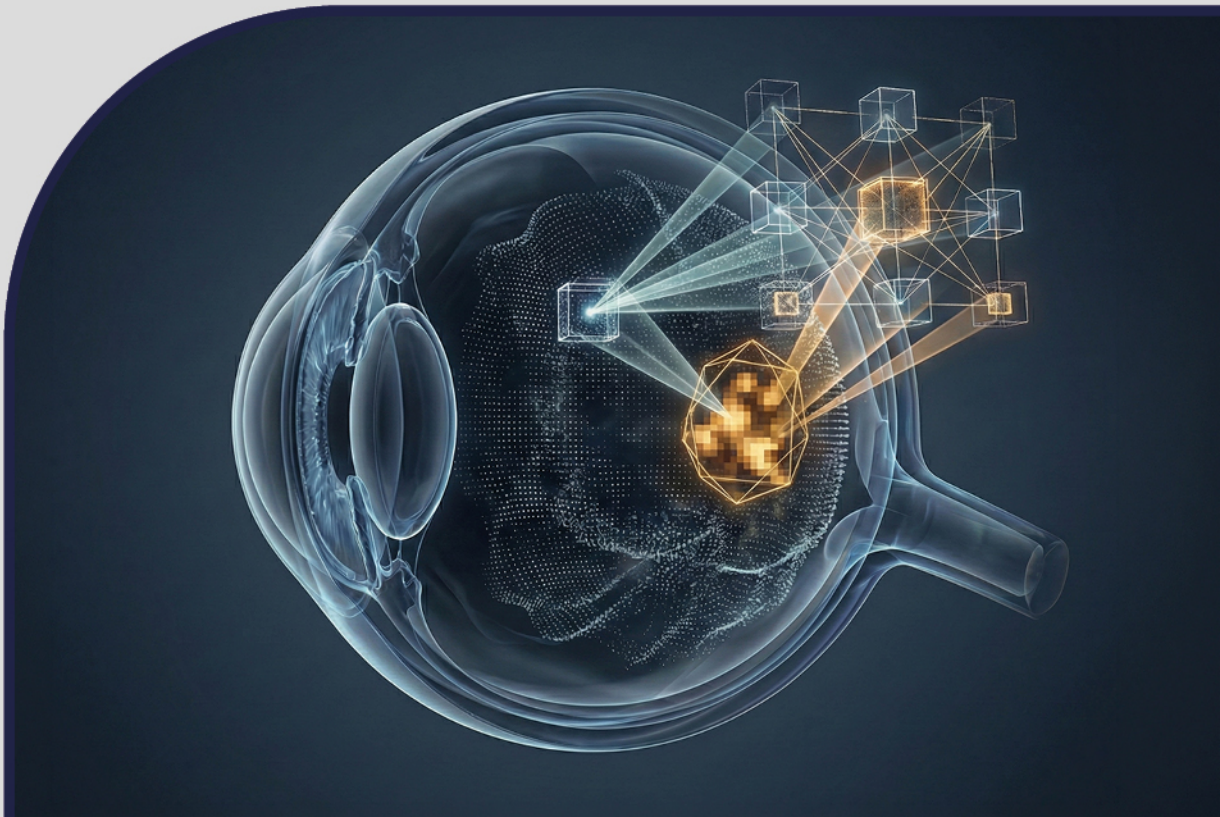
MONOGRAPHS

POLITECHNIKA
LUBELSKA
LUBLIN UNIVERSITY
OF TECHNOLOGY

Deep Learning Approaches for Retinal Diseases Recognition

From Data Augmentation Towards Vision Transformers

Paweł Powroźnik



Deep Learning Approaches for Retinal Diseases Recognition

From Data Augmentation Towards Vision Transformers

MONOGRAPHS
OF LUBLIN UNIVERSITY OF TECHNOLOGY

Deep Learning Approaches for Retinal Diseases Recognition

From Data Augmentation Towards Vision Transformers

Paweł Powroźnik

LUBLIN UNIVERSITY
OF TECHNOLOGY
PUBLISHING HOUSE

Lublin 2026

Reviewers:

Houda Harkat, Ph.D. (Universidade Lusofona, Lisbon)
prof. **Slawomir Nasuto,** Ph.D. (University of Reading, School of Biological Sciences)

Author:

dr **Paweł Powroźnik**, ORCID: 0000-0002-5705-4785
Politechnika Lubelska – ROR: https://ror.org/024zjzd49

Lead editor: **Magdalena Chołojczyk**
Linguistic proofreading: **Ronald E. Day**
Technical editor **Katarzyna Pełka-Smętek**
Series graphic design: **Łukasz Maj**

The illustrations included in the book are the author's own work.

Published with the approval of **Rector of Lublin University of Technology**

ISBN: 978-83-7947-666-4 (print version)
ISBN: 978-83-7947-667-1 (electronic version)
DOI: 10.35784/9788379476671
Publisher: Wydawnictwo Politechniki Lubelskiej
www.wpl.pollub.pl
ul. Nadbystrzycka 36C, 20-618 Lublin
tel. (81) 538-46-59

LUBLIN UNIVERSITY
OF TECHNOLOGY
PUBLISHING HOUSE

Print run: 50 copies

Contents

Metody głębokiego uczenia w rozpoznawaniu chorób siatkówki. Od augmentacji danych do transformerów wizyjnych

Streszczenie

Monografia poświęcona jest problematyce automatycznego rozpoznawania rzadkich chorób siatkówki z użyciem zaawansowanych metod głębokiego uczenia. Praca stanowi spójne, wieloaspektowe studium naukowe łączące zagadnienia inżynierii danych medycznych, projektowania architektur neuronowych, metod zwiększania efektywności uczenia przy ograniczonych zbiorach danych oraz interpretowalności modeli sztucznej inteligencji w kontekście klinicznym. Monografia wypełnia istotną lukę badawczą dotyczącą braku skutecznych, wiarygodnych i wyjaśnialnych narzędzi wspomagających diagnostykę rzadkich chorób siatkówki, takich jak retinopatia barwnikowa czy zwyrodnienie plamki żółtej. Głównym celem monografii jest opracowanie i kompleksowa ocena nowoczesnych metod głębokiego uczenia zdolnych do precyzyjnej identyfikacji rzadkich chorób siatkówki na podstawie obrazów okulistycznych. Autor dąży do zwiększenia skuteczności diagnostycznej w warunkach ograniczonej liczby danych, dużej zmienności fenotypowej oraz subtelnych zmian patologicznych. Cele szczegółowe pracy obejmują: (i) opracowanie zaawansowanych strategii augmentacji danych, w tym z użyciem generatywnych sieci przeciwstawnych, (ii) zaprojektowanie autorskich architektur konwolucyjnych sieci neuronowych oraz transformerów wizyjnych, (iii) redukcję złożoności obliczeniowej przy zachowaniu wysokiej skuteczności klasyfikacji, (iv) integrację metod zespołowych oraz (v) zapewnienie interpretowalności predykcji modeli w kontekście klinicznym.

Monografia prezentuje szerokie spektrum metod obliczeniowych i eksperymentalnych. W pierwszej kolejności autor szczegółowo analizuje charakterystykę wybranych rzadkich chorób siatkówki oraz opisuje proces tworzenia i przygotowania dedykowanych zbiorów danych. Kluczowym elementem metodologicznym jest rozbudowany moduł augmentacji danych. Oprócz klasycznych technik przekształceń geometrycznych i fotometrycznych, autor proponuje zastosowanie Deep Convolutional Generative Adversarial Networks (DCGAN) do generowania syntetycznych obrazów siatkówki, co pozwala istotnie zredukować problem niedoboru danych i niezbalansowanych klas. W zakresie architektur identyfikacyjnych monografia obejmuje zarówno analizę modeli referencyjnych (ResNet, DenseNet, Inception), jak i autorskie rozwiązania. Szczególną uwagę poświęcono modelowi Deep CNN-GRU, który łączy ekstrakcję cech przestrzennych z modelowaniem zależności kontekstowych, konwolucyjnej sieci GRU U-Net do jednoczesnej segmentacji i klasyfikacji oraz dedykowanej Residual Attention Network, umożliwiającej selektywne skupienie uwagi na kluczowych obszarach patologicznych. Istotnym wkładem metodologicznym jest wprowadzenie konwolucji Kroneckera jako zamiennika klasycznych filtrów splotowych, co pozwala na ograniczenie liczby parametrów i zwiększenie efektywności uczenia przy małych zbiorach danych. Kolejna część pracy poświęcona jest transformerom wizyjnym, w tym autorskiej architekturze Deep Residual Vision Transformer, wzbogaconej o mechanizm resztkowej samo-uwagi. Autor analizuje proces tokenizacji obrazu, mechanizmy uwagi jedno- i wielogłowicowej oraz wpływ pozycjonowania tokenów na jakość klasyfikacji. Metody zespołowe (bagging, boosting, stacking, soft voting) zostały użyte do dalszej poprawy stabilności i dokładności klasyfikacji. Całość uzupełniają techniki wyjaśnialnych narzędzi, w szczególności Grad-CAM oraz SHAP, umożliwiające wizualną i ilościową interpretację decyzji modeli. Przeprowadzone eksperymenty wykazały, że zastosowanie zaawansowanej augmentacji danych z użyciem DCGAN prowadzi do istotnego wzrostu jakości klasyfikacji w porównaniu z klasycznymi metodami augmentacji. Autorskie architektury CNN-GRU, RAN oraz Deep Residual ViT osiągnęły bardzo wysokie wartości dokładności, czułości i miary F1-score, przekraczające 95%, przy jednoczesnym zachowaniu dobrej generalizacji. Konwolucja Kroneckera pozwoliła ograniczyć złożoność obliczeniową modeli bez utraty skuteczności, a w wielu przypadkach prowadziła do jej poprawy. Zastosowanie uczenia zespołowego dodatkowo zwiększyło stabilność klasyfikacji, szczególnie w warunkach niezbalansowanych klas. Analiza statystyczna z użyciem testu McNemary'ego potwierdziła istotność uzyskanych różnic względem modeli bazowych. Metody wyjaśnialności wykazały, że modele skupiają uwagę na obszarach siatkówki zgodnych z kliniczną wiedzą okulistyczną, co zwiększa zaufanie do proponowanych rozwiązań i ich potencjał aplikacyjny w praktyce klinicznej.

Monografia dowodzi, że połączenie zaawansowanych technik augmentacji danych, nowoczesnych architektur głębokiego uczenia, mechanizmów uwagi, redukcji złożoności obliczeniowej oraz metod interpretowalnych stanowi skuteczną strategię w rozpoznawaniu rzadkich chorób siatkówki. Zaproponowane rozwiązania znacząco przewyższają klasyczne podejścia, oferując jednocześnie transparentność i potencjał klinicznego wdrożenia. Praca wnosi istotny wkład do rozwoju medycznej sztucznej inteligencji, wskazując kierunki dalszych badań, obejmujące m.in. adaptacyjne modele generatywne, hierarchiczne transformery wizyjne oraz integrację z wielomodalnymi danymi klinicznymi.

Słowa kluczowe: kasyfikacja obrazów siatkówki, diagnostyka medyczna oparta na wizji komputerowej, augmentacja obrazów medycznych, wyjaśnialna sztuczna inteligencja, głębokie uczenie w okulistyce

Deep Learning Approaches for Retinal Diseases Recognition. From Data Augmentation Towards Vision Transformers

Abstract

The monograph covers the problem of automatic recognition of rare retinal diseases using advanced deep learning methods. The work constitutes a coherent, multi-faceted scientific study that integrates issues of medical data engineering, the design of neural network architectures, methods for improving learning efficiency under limited data availability, and the interpretability of artificial intelligence models in clinical context. The monograph fills a significant research gap related to the lack of effective, reliable and explainable tools supporting the diagnosis of rare ophthalmic diseases, such as retinitis pigmentosa and acquired vitelliform lesion.

The primary objective of the monograph is to develop and comprehensively evaluate modern deep learning methods capable of accurate identification of rare retinal diseases based on ophthalmic images. The author seeks to enhance diagnostic performance under conditions of limited data availability, high phenotypic variability, and subtle pathological changes. The specific objectives include: (i) the development of advanced data augmentation strategies, including those based on generative adversarial networks; (ii) the design of original convolutional neural network architectures and vision transformers; (iii) the reduction of computational complexity while maintaining high classification performance; (iv) the integration of ensemble learning methods; and (v) ensuring the interpretability of model classification in clinical context.

The monograph is based on a broad spectrum of computational and experimental methods. First, the author provides a detailed analysis of the characteristics of selected rare retinal diseases and describes the process of constructing and preparing dedicated datasets. A key methodological component is an extensive data augmentation module. In addition to classical geometric and photometric transformation techniques, the author proposes the use of Deep Convolutional Generative Adversarial Network (DCGAN) to generate synthetic retinal images, which significantly alleviates the problem of data scarcity and class imbalance.

With regard to identification architectures, the monograph covers both an analysis of reference models (ResNet, DenseNet, Inception) and original solutions. Particular attention is paid to the Deep CNN–GRU model, which combines spatial feature extraction with contextual dependency modelling, the convolutional GRU U-Net for simultaneous segmentation and classification, and a dedicated Residual Attention Network that enables selective focus on key pathological regions. An important methodological contribution is the introduction of Kronecker convolution as a substitute for classical convolutional filters, allowing a reduction in the number of parameters and improved learning efficiency while dealing with limited datasets.

The subsequent part of the study covers vision transformers, including the author's Deep Residual Vision Transformer architecture, enhanced with a residual self-attention mechanism. The author analyses the image tokenisation process, single-head and multi-head attention mechanisms, and the impact of token positioning on classification quality. Ensemble methods (bagging, boosting, stacking, soft voting) are employed to further improve identification stability and accuracy. The study is complemented by explainable AI techniques, in particular Grad-CAM and SHAP, which enable visual and quantitative interpretation of model decisions.

The conducted experiments demonstrate that the application of advanced data augmentation using DCGAN leads to a significant improvement in classification performance compared to classical augmentation methods. The proposed CNN–GRU, RAN and Deep Residual ViT architectures achieved high accuracy, sensitivity and F1-score, exceeding 95%, while maintaining good generalisation capability. Kronecker convolution reduced the computational complexity of the models without compromising performance and, in many cases, resulted in further improvements. The use of ensemble learning additionally increased classification stability, particularly under class imbalance conditions. Statistical analysis using McNemar's test confirmed the significance of the observed improvements relative to baseline models.

Explainability analyses showed that the models focus their attention on retinal regions consistent with established clinical knowledge, thereby increasing trust in the proposed solutions and their potential applicability in clinical practice.

The monograph demonstrates that the combination of advanced data augmentation techniques, modern deep learning architectures, attention mechanisms, computational complexity reduction, and explainable methods constitutes an effective strategy for the recognition of rare retinal diseases. The proposed solutions significantly outperform classical approaches while offering transparency and clear potential for clinical deployment. The study makes an important contribution to the development of medical artificial intelligence and outlines directions for future research, including adaptive generative models, hierarchical vision transformers, and integration with multimodal clinical data.

Keywords: retinal image classification, vision-based medical diagnosis, medical image augmentation, explainable artificial intelligence, deep learning in ophthalmology

List of symbols

AM	– Attention Mechanisms
AMD	– Age-related Macular Degeneration
AVL	– Acquired Vitelliform Lesions
BN	– Batch Normalization
CBAM	– Convolutional Block Attention Module
CNN	– Convolutional Neural Network
CORD	– Cone-rod Dystrophy
DCGAN	– Deep Convolutional Generative Adversarial Networks
DenseNet	– Densely Connected Convolutional Networks
DKCN	– Deep Kronecker Convolutional Network
FAF	– Wide-field Fundus Autofluorescence
GAN	– Generative Adversarial Networks
Grad-CAM	– Gradient-weighted Class Activation Mapping
GRU	– Gated Recurrent Unit
KCNN	– Kronecker Convolutional Neural Network
LSTM	– Long Short-Term Memory
MLP	– Multilayer Perceptron
OCT	– Optical Coherence Tomography
ResNet	– Residual Networks
RNN	– Recurrent Neural Network
RP	– Retinitis Pigmentosa
SHAP	– Shapley Additive Explanations
UWFAF	– Ultra-widefield Autofluorescence
UWFP	– Ultra-widefield Photography
UWPC	– Ultra-widefield Pseudocolour
VGG	– Visual Geometry Group
ViT	– Vision Transformer
XAI	– Explainable Artificial Intelligence
XGBoost	– Extreme Gradient Boosting

Preface

This monograph addresses the crucial topic of employing advanced deep learning techniques for the automated diagnosis of rare retinal diseases, such as retinitis pigmentosa, acquired vitelliform lesions and drusen-type lesions. The author emphasizes the importance of developing precise retinal image classification methods, which significantly enhance in the early stage of diagnosis.

Chapter One introduces the motivation for these studies and clearly identifies the research gap regarding insufficient methods for classifying rare eye diseases. This gap arises mainly from limited datasets, subtle pathological changes, and high phenotypic variability. The necessity of utilizing innovative deep learning techniques, including advanced data augmentation and attention mechanisms is highlighted, as crucial to achieving a high level of diagnostic performance.

Chapter Two provides an extensive literature review, pointing out current approaches to ocular disease diagnostics. The shortcomings in existing methods are noted, as well as, specifically, the need for dedicated solutions targeting rare retinal conditions specifically.

Chapter Three details the characteristics of rare retinal diseases, such as: retinitis pigmentosa, acquired vitelliform lesions and drusen-type lesions. This chapter also describes the datasets prepared as a basis for subsequent experiments. The procedures for data collection and selection, ensuring their high quality and readiness for the deep learning processes are also thoroughly discussed.

In Chapter Four, the author introduces an innovative data augmentation technique using Deep Convolutional Generative Adversarial Networks. The implementation is described in detail and experimental results confirm the significant advantage of this approach over traditional augmentation methods. The use of generative models led to a remarkable increase in classification accuracy, demonstrating the effectiveness of the proposed method, especially with limited datasets.

Chapter Five elaborates on modules important for ocular disease classification, such as: convolutional neural networks, attention modules and gated recurrent units. The architecture of these modules and their impact on enhancing the accuracy of diagnostic models are thoroughly explained, especially concerning subtle and challenging pathological changes in eye diagnostic images.

Chapter Six reviews typical algorithms for classifying medical imaging data. Various architectures such as residual networks, densely connected convolutional neural networks and Inception networks are compared. Their strengths and limitations in the context of ophthalmic datasets are highlighted.

Chapter Seven presents the author's custom solutions, including the adapted bidirectional connection of gated recurrent units and U-Net architectures. Special attention is paid to the dedicated residual attention module implemented in both the residual attention network and the author's deep residual vision transformer. Experiments confirm the high effectiveness of these models in detecting even the smallest pathological changes.

Chapter Eight introduces another innovation. Standard convolutional operations are replaced with Kronecker convolution. This approach significantly reduces computational complexity and improves the models' capability to extract essential diagnostic features from limited datasets, resulting in increased classification accuracy.

Chapters Nine and Ten are devoted to vision transformers. The token-level attention mechanisms, through which vision transformer networks achieve high accuracy and provide interpretability of results, are comprehensively described. In addition, the author's deep residual vision transformer architecture is introduced, improved with a dedicated residual attention module. This approach further improves classification performance, enabling a precise model focus on subtle pathological retinal changes.

In Chapter Eleven, the ensemble learning techniques such as boosting, stacking and voting are discussed. Implementing these methods improves

diagnostic accuracy and reduces the risk of overfitting, particularly important for small and imbalanced datasets.

Chapter Twelve concentrates on explainable artificial intelligence techniques, such as Grad-CAM and SHAP values, used to identify critical image elements influencing classifier decisions. Employing these methods increases model transparency and facilitates collaboration with clinicians.

Chapter Thirteen examines the statistical significance of the obtained results. McNemar's test is used as a comparative tool.

Chapter Fourteenth discusses the presented achievements and compares them with other well-known methods. The entire monograph is summarized in the last chapter.

1. Introduction

Computer vision is the part of artificial intelligence that empowers a computer to understand and analyse sight data from the real world. This has radically changed almost all industries, especially healthcare. In medical diagnostics, computer vision methods help to automate the reading of medical images so that it is possible to improve how diseases are spotted and tracked [1], [2], [3]. Related to this, areas like ophthalmology have greatly profited from recent improvements in deep learning. For example disease detection can now be done automatically through different types of retinal images, such as Optical Coherence Tomography (OCT) [4], [5].

Uncommon retinal disorders, though due to the low prevalence of conditions like retinitis pigmentosa (RP) and acquired vitelliform lesion (AVL), along with their variable presentations, the pathology is understood in its importance when it comes to diagnosis. Timely and accurate diagnosis can lead to better outcomes for patients. However, existing deep learning models fail to provide clinically acceptable accuracy owing to imbalanced datasets, insufficient data, high variability in phenotype expression, and subtle changes in disease status [3], [6], [7].

This monograph tries to fill such gaps by suggesting and assessing advanced deep learning techniques, specifically convolutional neural networks (CNNs), generative adversarial networks (GAN) and vision transformers (ViTs), for the recognition of rare retinal diseases. Significant consideration has been

given to cutting-edge data augmentation approaches to deal with dataset constraints, stressing Deep Convolutional Generative Adversarial Networks as well as traditional methods. This study further investigates several architectural upgrades that include attention mechanisms, Kronecker convolutions, and gated recurrent units for capturing the subtle pathological signatures of retinal diseases.

Reviews of the literature and current methods show major shortcomings in conventional models, especially their inability to generalize well from small, often heterogeneous datasets. As a result, ensemble learning and explainable artificial intelligence (XAI) approaches, specifically Gradient-weighted Class Activation Mapping (Grad-CAM) and SHapley Additive exPlanations (SHAP), will be explored to enhance interpretability and clinical reliability. The main aim of this monograph is to build strong, precise, and understandable deep learning frameworks that support confident and quick diagnoses of unusual retinal conditions.

1.1. Study Motivation

Uncommon ocular disorders like retinitis pigmentosa and acquired vitelliform lesions are stealthy culprits in the field of eye care, mainly due to their progressive traits coupled with a real chance of permanent loss of vision. AVL is an eye condition that deals with the macula, which is the central portion of the retina and enables sharp central sight [8]. The key to these maladies lies in early, precise identification so that action can be taken not only in time to delay progression but also to uplift the quality of life for those afflicted.

There is a considerable gap in the literature about effective classification methods for these rare diseases, even after deep learning and medical imaging advancements. Current CNNs and ViTs have shown great success in the detection of common ocular conditions such as diabetic retinopathy and glaucoma [9], [10].

However, these models often fail when applying them to rare eye diseases like RP and AVL:

1. **Data Scarcity**: Limited data is available to train robust models due to the rarity of the diseases. A high-quality annotated image dataset is required for CNNs and ViTs to learn discriminative features relevant to conditions such as RP and AVL [11], [12].

2. **Variability in Presentation**: Rare ocular diseases can show great differences in how they appear, making it hard for models learned from small groups of patients to work well across various individuals [13]. For example, AVL may come with different size lesions, stages and patterns; this makes automatic finding much harder [14].
3. **Minor Pathological Changes**: At the onset of conditions like RP and AVL, small alterations in the retina may be undetectable by typical models [15]. This requires the evolution of approaches that can discern minute details [16].

1.2. Defining Study Gap

A review of the current literature indicates a significant gap in research focused on the automated classification of these rare eye diseases. Many studies prioritize conditions with larger datasets and higher prevalence rates, leaving rare diseases underrepresented [1]. The existing models for rare diseases often do not achieve sufficient levels of accuracy, sensitivity, or specificity required for clinical applications [4].

To address this gap, there is an essential need for studies of advanced classification methods tailored for rare eye diseases. Potential areas of exploration include:

1. Data Augmentation: Utilizing generative models like Generative Adversarial Networks to augment limited datasets can enhance model training.
2. Transfer Learning: Applying knowledge from models trained on related tasks or diseases to improve performance on rare disease classification.
3. Attention Mechanisms: Incorporating attention modules within CNNs and ViTs to focus on critical regions of retinal images may improve detection of subtle features associated with rare diseases.
4. Kronecker Convolution: Replacing standard convolution with the Kronecker product in CNNs to improve rare eye disease classification by reducing model complexity and overfitting, enhancing feature extraction from limited data, and increasing computational efficiency.
5. Ensemble Learning: Utilizing ensemble learning techniques can be particularly beneficial due to their ability to enhance classification

accuracy, robustness to overfitting, diversity of models and handling variability.

6. Explainable Artificial Intelligence: Providing explanations for model predictions to increase clinicians' trust as well as help to identify when and why a model might be making incorrect predictions, which is crucial for refining models and improving accuracy.

The current limitations of CNNs and ViTs in detecting rare eye diseases like retinitis pigmentosa and macular degeneration highlight a significant gap in ophthalmic artificial intelligence research. By undertaking focused studies on classification methods for these conditions, a tool that provides early and accurate diagnosis can be developed. By focusing on dystrophia maculae vitelliformis and retinitis pigmentosa, this summary highlights the need for specialized research into classification methods for these rare eye diseases. Addressing the challenges associated with limited data and subtle pathological features is crucial for developing effective diagnostic tools [3].

Ensemble learning and explainable AI play critical roles in advancing the detection of rare eye diseases like retinitis pigmentosa and dystrophia maculae vitelliformis. Ensembles improve predictive performance and robustness, while explainable AI ensures that models are transparent and their predictions are interpretable by clinicians. Together, they contribute to more accurate, trustworthy, and clinically applicable diagnostic tools.

2. Related Works

In the last few years, the rapid development of automatic ways for sorting medical pictures has helped greatly in improving diagnostic abilities in many areas of medicine. OCT pictures are a good example because their high spatial resolution helps to identify eye diseases accurately including odd diseases that are hard to diagnose and are a challenge for clinicians.

Rare diseases have few clinical cases available, which translates into little training data being accessed. This problem is very difficult for methods of artificial intelligence where large and varied data sets are needed to reach a high efficiency in diagnosis.

Respecting these limits, several ways have been created that allow generating synthetic or changed OCT images thus helping better quality training for machine learning models. Deep learning models, particularly CNNs and ViTs, have been developed recently with much efficiency in classifying medical images.

While CNNs are already well-established in ophthalmological diagnostics, transformer models, originally used for natural language analysis, introduce new possibilities for interpreting image data and can bring additional benefits for complex classification tasks related to rare diseases.

One major prerequisite for evaluating AI models within the field of medicine is their interpretability (explainable AI), which means one can understand what mechanism leads to decisions by algorithms. In this respect,

methodologies like Grad-CAM or SHAP gain principal relevance as they allow identifying key regions in OCT images that play a determining role in disease classification.

2.1. Medical Images Augmentation

Increased interest in medical image augmentation, especially for OCT images, has led to the rapid development of data augmentation techniques. These methods are critically important because they provide a solution to very important problems that occur when there is not enough good-quality training data needed for creating robust machine learning models. Recent papers show some advancement in medical image augmentation, emphasizing the role that generated images play in improving diagnostics for several ocular conditions.

Efficacy studies have reported that GANs help improve OCT images by facilitating high-resolution synthetic images pertaining to specific eye diseases such as glaucoma and diabetic retinopathy. According to Kumar et al., GANs can successfully synthesize circumpapillary OCT images thereby greatly improving classification performance when glaucomatous changes inpatients are considered [17]. This was further confirmed by Zheng et al., who showed that GANs could synthesize OCT images. This allowed one for increasing accuracy in eye condition detection [18]. GAN-based approaches go beyond conventional data augmentation techniques by learning the underlying data distribution and generating entirely new, realistic samples, rather than applying predefined geometric or intensity transformations to existing images. As demonstrated by Zafar et al., this capability helps alleviate class imbalance commonly observed in medical datasets [19].

In addition, recent studies further highlight the innovative use of advanced GAN architectures, such as the Pix2Pix framework, which has been successfully employed to translate OCT images into other imaging modalities, including fluorescein angiography. This is however one of the few studies that has verified this particular application in terms of medical image translation [20]. The use of adapted GANs has been promising in creating different datasets based on specific tasks for learning that deal with issues related to over-and-underfitting in the analysis of medical images [21], [22].

Many papers also stress that the gamut of retinal pathologies must be addressed using GAN-generated synthetic images. Remtulla et al. narrate

how images synthetically created by GANs can supplement existing imaging modalities and finally take diagnostic accuracy for vitreoretinal pathologies to the next level [23]. Further, works by Danesh et al. show that labelled OCT images can be automatically generated, hence validating further the efficacy of such methods in increasing diversity in databases [24], [25].

Other studies focusing on diseases of the retina have thereby demonstrated the role of GANs in several diagnostic frameworks. For example, a conditional GAN model may be applied to create synthetic OCT images for various ocular conditions, which model is later validated through research conducted by Han et al., underlining that applying GANs helps to represent diverse eye conditions through synthesized imagery [26]. This model demonstrates the ability of GANs to create disease-specific images. This has great implications for the challenges facing the medical imaging community, especially in the case of rare diseases.

As per Moon et al.'s finding, predictive models based on synthetic OCT images have improved future treatment outcomes for neovascular age-related macular degeneration, denoting true clinical utility for synthetic data in real-world applications [27]. Thus, combining GAN-generated images with classifications from traditional datasets marks a significant step toward integrating artificial intelligence into ocular diagnostics.

Adjustability and strength toward multi-modal imaging transformations are further underscored here; for example, standard OCT imagery can be converted into characteristic synthetic versions. Also, GANs have been used to improve the quality of OCT images by increasing their resolution through advanced modelling techniques. Techniques like intensity inhomogeneity correction have augmented the accuracy of OCT data interpretation, facilitating more reliable segmentation of retinal features [28], [29]. This aspect is particularly crucial because, segmentation serves as a precursor to various diagnostic workflows, impacting subsequent treatment decisions for many ocular conditions. Moreover, segmentation is the basis for classification with U-Nets [30], [31].

Also, GANs have been used to improve the quality of OCT images by increasing their resolution through advanced modelling techniques. Liang showed that GANs can recover realistic details in micro-OCT images and may help research as well as clinical practices [32]. Such novelties do justice to the merger of artificial intelligence with medical imaging because here not only are the resolution and details captured in images increased, but even automated diagnostics get a boost.

As GANs and their versions keep on advancing, new methods that come up, which incorporate multi-layered approaches for segmentation and

classification, like deep learning frameworks, show an enormous potential. Their use ranges everywhere in these real-time clinical assessments, as seen in studies for coronary plaque characterization that highlight how adaptable synthetic data can be utilized across different imaging modalities [19], [33].

However, issues about the credibility of synthetic OCT images still remain. A "few-shot" learning approach can be seen as a first step toward enhancing diagnostic accuracy for rare retinal conditions, with the caveat that most of the generated images may not be clinically appropriate [34]. This warning has to be taken into consideration, stressing further the importance of continuous assessment of quality and applicability once synthetic images tend to find usage in any clinical setting.

To sum up, the existing literature on OCT image augmentation sets the stage for a technological innovation balanced against clinical need. Now that GAN-based methodologies are being thrust ahead, there arises an accompanying need also to ensure that generated images meet the stringent quality standards demanded in medical practice. All in all, collaboration possibilities for enlarging datasets via synthetic augmentation are bound to change the diagnostic landscape by instituting strong frameworks well-equipped to manage ophthalmological diseases' vast inherent variability.

2.2. Convolutional Neural Networks in OCT Images Classifications

The integration of CNNs into medical image classification has shown remarkable promise, particularly in the domain of OCT imaging. The last decade has seen rapid advancements in both network architectures and application contexts, significantly enhancing diagnostic capabilities for a range of ocular conditions. This literature review consolidates recent findings, focusing on studies that utilize various typical CNN architectures, including VGG, EfficientNet, Inception, DenseNet, ResNet, as well as more complicated models.

In previous works found in the literature, the use of CNNs was particularly useful for analysing OCT images regarding age-related macular degeneration (AMD), diabetic retinopathy, and glaucoma. An equally critical application area beyond traditional retinal imaging for using CNNs was described by Liu et al., through detecting cochlear endolymphatic hydrops using an architecture called ELHnet architecture, highlighting an important application of CNNs

beyond traditional retinal imaging [35]. A segmentation-based classification method for retina was developed by Wang et al., which can be used for very accurate early detection of glaucoma [36]. This strategy highlights the flexibility of CNNs in handling both segmentation and classification aspects when it comes to OCT images.

In their work regarding the competency of CNNs to separate AMD from healthy individuals, Lee et al. shared metrics that reflected a high level of sensitivity and specificity, which is crucial for early detection and action [16]. In another significant investigation, Motozawa et al., with great success, classified different types of AMD, mainly stressing their identification between non-exudative and exudative forms, which they classified with much precision [37]. These results prove that CNNs have enormous potential to improve diagnoses via optical imaging, thus providing clinicians advanced tools in making critical decisions.

Classification capabilities have been further fine-tuned with the use of more intricate architectures like Inception and EfficientNet. A Deep Convolutional Neural Network (DCNN) model was presented by Mohan et al., which shows better performance than usual architectures by attaining substantial classification accuracy regarding diseases related to the retina [38]. Similarly, Mishra et al. utilized a multi-level dual-attention CNN, improving the accuracy of OCT image classification related to macular conditions. Their design shows how attention layers can focus on important features enhancing model power [39].

The application of CNNs to differentiate between various pathologies, has rapidly gained traction. Yoon et al. developed a CNN that successfully classified central serous chorioretinopathy (CSC). This study reinforced the strength of deep learning models in analysing OCT images, providing insights into the chronicity of CSC [40]. Similarly, Jee et al. evaluated multiple OCT imaging modalities for predicting persistent CSC, achieving a high classification accuracy with their ResNet-50 model [41]. Studies focusing on the classification of other retinal conditions, such as pachychoroid, have also shown promise [42].

Adding to the paradigm of deep learning applications, Tsuji et al. explored capsule networks for OCT image classification. In this study, a comparative analysis with ophthalmologists was presented. The results showed that clinicians achieved a mean diagnostic accuracy at the same level as neural models. It underscores the potential of machine learning models for matching human expert performance [43]. As CNN architectures continue to evolve, innovative strategies are being developed to improve robustness against variations in optical imaging. The incorporation of multi-modal data has also been considered, as seen in the study by Barbosa et al., which combined OCT with infrared imaging

to distinguish between edema types [44]. The work of Powroznik et al. utilized a hybrid CNN-GRU architecture that achieved a maximum classification accuracy of 98.9% on the OCT Retina dataset, with mean accuracy exceeding 95%, demonstrating strong potential for clinical application in macular disease detection [45]. Moreover, the combination of segmentation models with classification models was presented in the work of Skublewska-Paszkowska et al., where the U-Net network enriched with Bidirectional GRU elements was used [30]. Also, Xiao et al. showed the combination of U-Net architecture with convolutional networks for the same purposes [46].

Also, well-known models such as VGG, Inception, ResNet and DenseNet are used in the classification of OCT images. A study by Jaimes et al. employed a VGG16 architecture integrated with transfer learning to enhance the detection of various retinal diseases. Their work proved that transfer learning can use pre-trained models in a very efficient way to accelerate and make training better with those smaller datasets which are quite typical in medical imaging scenarios. The depth of the VGG16 architecture allows for the extraction of detailed hierarchical features, leading to improved discrimination among different pathologies [47], [48].

Another important contribution has been made by Kirankumar, who worked on automated classification of subtypes of coloboma using an InceptionV3 algorithm applied to OCT images with the purpose of developing a diagnostic tool through which a clinician can easily diagnose six distinct types of coloboma. Using a pre-trained InceptionV3 architecture known for image data handling through parallel convolutional paths, the model showed good performance in identifying different subtypes of coloboma. This work demonstrates that automated differentiation of multiple coloboma subtypes from OCT images is feasible, paving the road towards stable and efficient classification tools reducing diagnostic complexity and supporting timely clinical decision-making [49].

Yusufoğlu et al. used a new CNN architecture that integrates Inception modules, Squeeze-and-Excitation blocks, and a ConvMixer to further classify AMD. The challenge of computational efficiency is always secondary when it comes to the accuracy of classification, especially in a disease like AMD where misclassification can have varying implications due to its varied presentations. Therefore, the combination of these state-of-the-art techniques led to a model of superior performance metrics which indeed could prove useful in practical scenarios for ophthalmologists [50]. Furthermore, Bahr et al. conducted a systematic review of existing deep learning and machine learning algorithms

for retinal image analysis in neurodegenerative diseases, detailing diverse datasets and models utilized across studies. Their findings suggest that, while considerable progress has been made, ongoing efforts are needed to standardize datasets and improve model transparency for broader clinical applicability [51].

Incorporating novel network architectures, the study by Hamid et al. introduced an LSTM approach for the classification of age-related macular degeneration using OCT images. By integrating LSTM with CNN features, the model aims to harness temporal relations between images alongside spatial features, enabling a comprehensive analysis of the progression of AMD over time. This innovative approach represents a multipronged methodology in exploring deep learning frameworks, highlighting the multitude of ways CNNs can be extended to accommodate the complex nature of neurodegenerative diseases. The exploration of such models emphasizes the need for continuous evolution in classification methodologies to improve the accuracy and reliability of OCT image interpretation in clinical settings, ultimately providing timely interventions for patients [52].

2.3. Vision Transformers in OCT Images Classifications

Vision transformers have recently brought significant novelty to the challenge of Optical Coherence Tomography image classification. A study by Cai et al. employed a Vision Transformer model for the classification of diabetic maculopathy stages using OCT images, achieving superior performance compared to standard CNN-based approaches in terms of accuracy and related evaluation metrics. These results demonstrate that Vision Transformers are capable of effectively capturing discriminative pathological patterns in OCT scans, enabling accurate and timely diagnosis of diabetic retinopathy [53].

Wu et al. also reported a study on the application of Vision Transformer architectures to identify grades of diabetic retinopathy directly from OCT images. The authors pointed out that, based on their results, ViTs can parse very efficiently through complex visual data intrinsic to retinal imaging, hence achieving classification metrics close to perfection. This fact underlines that Vision Transformers can be very useful when challenged with the varying presentations of retinal diseases [54].

Another approach was taken by Akça et al., who stated in their report that the choroidal neovascularization, diabetic macular edema, and drusen from OCT images would be classified well by using Token-to-Token Vision Transformer and Mobile Vision Transformer among other advanced Vision Transformer models. Their experiments reported classification accuracies exceeding 95% across different model configurations, with consistently high precision (91–100%) and F1-score (92–99%) for all retinal conditions [55].

Also, the progress of ViTs in ophthalmology is evidenced in a study by Jiang et al., where a Vision Transformer helped to make a computer-aided diagnosis system for classifying OCT images linked to age-related macular degeneration and normal conditions. This study revealed impressive classification accuracy, showing the great potential of Vision Transformers in making diagnostic accuracy better and giving quick assessments that could be very important in clinical settings [56].

Similarly, Cai et al. investigated the effectiveness of Vision Transformer models in improving retinal disease image classification. Their study pointed out that domain-specific optimization is needed when changing models to enable effective medical image analysis, recommending practices that take advantage of the spatial and spectral features found uniquely in OCT data [57].

Huang et al. introduced DAT-SegNet, a transformer-based multi-organ segmentation network designed for medical image analysis, which was successfully applied to ocular imaging tasks, demonstrating the effectiveness of transformer architectures in medical image segmentation. The results of their study show that Vision Transformers can reach at least performance of the conventional segmentation methods. Thus, Huang et al. prove models' adaptability in complex image analyses consistent with the OCT images [58].

Further investigations conducted by Ma et al. and Zhou et al. confirm conclusions of [58] by classifying pathologies seen in OCT scans, such as glaucoma and AMD, using transformer-based architectures. Taken together, these results provide strong impetus for further research into the deployment of Vision Transformers in actual clinical settings and thus push the need for even more development and fine-tuning of these models as they might play a crucial role in automating and enhancing diagnostic accuracy in ophthalmology [59], [60].

For comparison, 34 works were selected that deal with the subject of OCT image classification. The neural network models used and their quality measures were compared. Additionally, information on the data sets used was collected. The compared data are presented in Fig. 2.1. and Fig. 2.2.

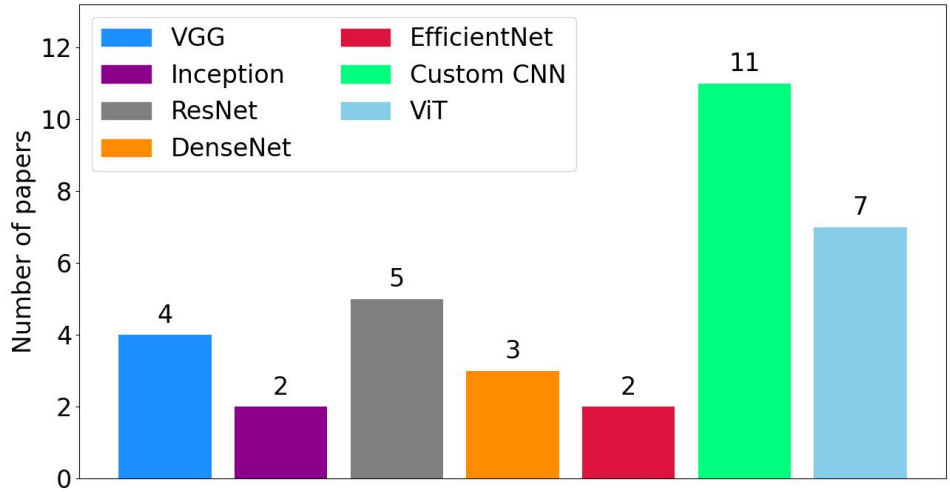


Fig. 2.1. A comparison of the number of publications employing different neural network models in OCT image classification tasks.

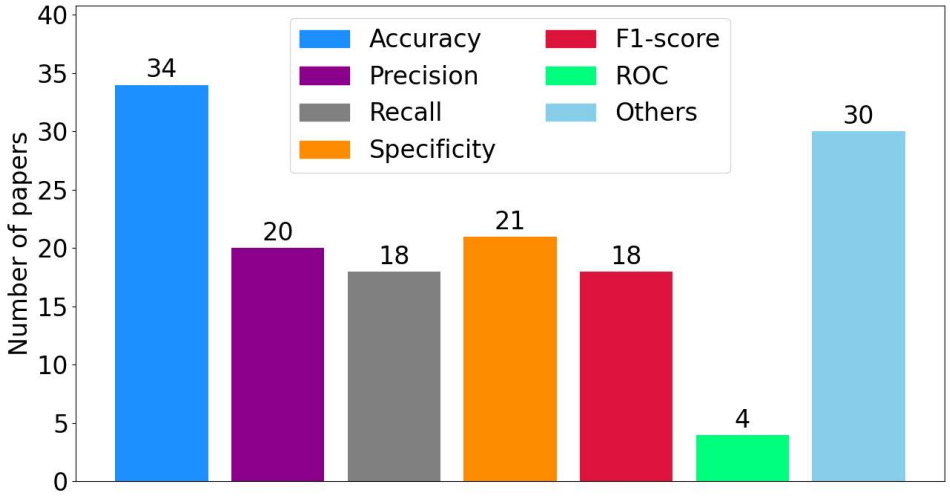


Fig. 2.2. A comparison of the most commonly reported classification metrics used in OCT image classification studies.

In summary, the reviewed studies demonstrate that a wide range of neural network architectures have been explored for OCT image classification, including both convolutional and transformer-based models. Fig 2.1 and 2.2 present an overview of the neural network families most frequently employed in the literature and the classification metrics most commonly reported in OCT-related studies. These figures are intended to summarize methodological

choices and reporting practices. The observed diversity of models and evaluation protocols, together with the limited availability of large, balanced ophthalmic datasets, motivates further research toward data-efficient and clinically interpretable solutions, which is addressed in the subsequent chapters.

Recent studies show that hybrid CNN-Transformer architectures are effective for retinal image analysis. These models capture both local structural features and global contextual information in OCT and fundus images. Transformer-based and attention-driven approaches have achieved strong results in OCT-based retinal disease classification as well as in multi-label fundus image analysis, demonstrating their applicability to complex retinal disease recognition across different imaging modalities [61], [62], [63], [64].

3. Selected Rare Eye Diseases

3.1. Retinitis Pigmentosa

Patients diagnosed with retinitis pigmentosa (RP) commonly experience night blindness and a gradual decline in peripheral vision in both eyes, typically during their first few decades of life. This leads to the loss of central vision and progress to total blindness. The standard form of RP is a chronic disorder that usually unfolds over several decades.

However, some cases exhibit rapid progression within about 20 years, while others progress so slowly that complete blindness never occurs. This lifelong disease is categorized into three stages: early, middle, and advanced.

In its early stage, night blindness is the primary symptom. As the disease advances to the middle stage, visual field assessments reveal small peripheral scotomas, which gradually expand toward the far periphery and macular region of the eye. In the final stage, significant peripheral vision loss results in "tunnel vision", severely restricting independent movement of the patient. Patients at this stage retain only a small degree of central vision, making visual field testing the key diagnostic tool for assessing functional vision. The clinical manifestations of RP can vary significantly among patients, even among those with identical genetic mutations, individuals from the same family or even between the two eyes of a single patient [45].

A characteristic triad of fundoscopic findings defines RP: attenuation of retinal blood vessels, a waxy pallor of the optic disc, and intraretinal pigment deposits forming a bone-spicule pattern in the retinal periphery. These pigmentary abnormalities arise when degenerating retinal pigment epithelial cells release pigment into the inner retinal layers in response to photoreceptor cell loss. Initially, this manifests as a fine dusting of pigment in the mid-and-far periphery, which later evolves into the bone-spicule formations surrounding retinal blood vessels. In advanced RP, choriocapillaris atrophy exposes the underlying larger choroidal vessels [65].

Cone-rod dystrophy (CORD), another related retinal disorder, is characterized by progressive visual acuity loss, photophobia, and impaired colour perception (dyschromatopsia). In these cases, electroretinography typically reveals abnormal cone photopic responses, with little or no rod involvement in the early stages. Pigmentary changes are also observed in the posterior pole of the retina [66].

Diagnosing and monitoring retinitis pigmentosa requires a thorough evaluation, combining patient history with various imaging and functional tests. These include fundus photography, widefield fundus autofluorescence (FAF), visual field assessments and full-field ERG. Such diagnostic tools are critical for detecting early signs in at-risk individuals and tracking disease progression over time [67].

Recent advancements aim to prevent photoreceptor degeneration or to restore rod cell function in retinal dystrophies. Additionally, there is a growing need for standardized devices to quantify structural and functional retinal changes in RP. With the development of molecular therapies for inherited retinal diseases, sensitive imaging techniques are essential for detecting structural disease progression and establishing robust analytical methods for disease monitoring [68].

Ultra-widefield photography (UWFP) and ultra-widefield autofluorescence (UWFAF) imaging, utilizing scanning laser ophthalmoscopy, have emerged as valuable tools for noninvasive, high-resolution retinal imaging with a 200° field of view. FAF, in particular, provides insights into retinal disease progression by visualizing lipofuscin accumulation, a marker of RPE function. In the early stages of RP, FAF images typically reveal hyperfluorescent patterns, while later stages exhibit hypofluorescence due to the presence of retinal lesions. A distinct autofluorescent ring, seen as a hyperfluorescent border in FAF images, often demarcates the transition from a functional to a dysfunctional retina in RP patients [68], [69], [70].

3.2. Acquired Vitelliform Lesion

Acquired Vitelliform Lesion is a retinal condition characterized by the accumulation of yellowish deposits in the subretinal space, specifically above the retinal pigment epithelium (RPE). These deposits consist mainly of lipofuscin, melanolipofuscin, melanosomes, and outer segment debris from photoreceptors. AVL most commonly occurs in the macula, but lesions can also develop outside the central retina. The condition is thought to arise due to impaired degradation of photoreceptor outer segments and their gradual accumulation in the subretinal space [8].

Initially, AVL may be asymptomatic and discovered incidentally during routine eye examinations. As deposits increase in size, some patients may experience mild visual disturbances, such as slight blurring or a subtle decline in visual acuity. Unlike other retinal conditions, AVL typically does not cause sudden vision loss, and distortion of vision is less common unless the lesion affects the foveal contour. Ophthalmologists usually detect these changes as yellowish subretinal deposits during a fundus examination, prompting further imaging tests such as Optical Coherence Tomography. OCT provides detailed cross-sectional images of the retina, confirming the presence of subretinal hyperreflective material above the RPE [71], [72].

On OCT, AVL appears as a well-defined hyperreflective lesion in the subretinal space, often with a stratified or heterogeneous internal structure. The accumulation of material may cause shadowing effects, obscuring deeper retinal layers. An example of OCT image is depicted in Fig. 3.1.

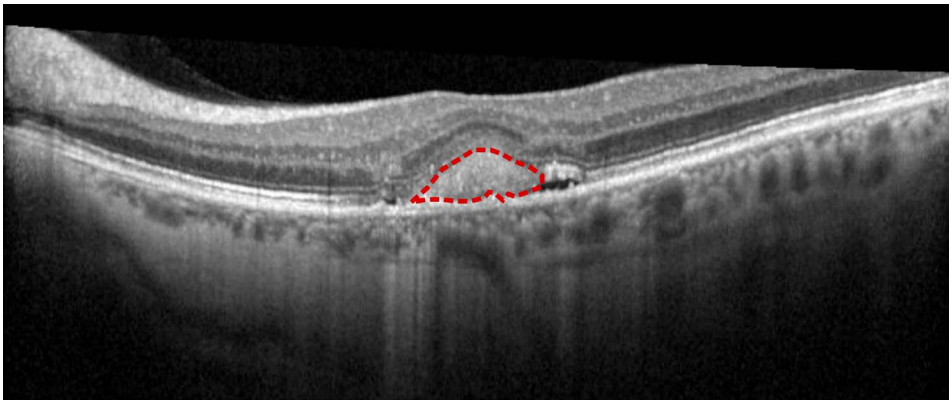


Fig. 3.1. OCT image of an eye affected by an acquired vitelliform lesion. The figure presents a representative OCT image illustrating an acquired vitelliform lesion. Pathological changes identified by ophthalmologists are highlighted in red.

The etiology of AVL is not fully understood, but it is most commonly associated with aging and possibly genetic predisposition. Unlike other macular degenerative conditions, AVL does not result from systemic metabolic disorders such as diabetes or lipid dysregulation. Regular monitoring with OCT is crucial for assessing the disease's progression and identifying potential complications [73].

3.3. Drusen

Drusen deposits are deposits made up of lipids, proteins and other metabolic products that are deposited between the retinal pigment epithelium and Bruch's membrane. These changes take the form of yellowish or whitish dots visible during a fundus examination and may occur singly or in larger clusters. They are usually located in the central area of the retina, although they can sometimes be observed outside the macula. In the early stages of development, they are usually small and do not cause noticeable clinical symptoms, which is why they are often detected accidentally during routine ophthalmological examinations [74].

As the number and size of deposits increase, symptoms such as decreased visual acuity, difficulty recognizing details and increased sensitivity to bright light may occur. In some cases, patients also describe distortions of the lines and contours of objects. Diagnosis is primarily based on a fundus examination, which shows characteristic changes in colour or transparency in the retinal area. In order to obtain a detailed image and assess the size of deposits, an OCT test is performed, which shows a cross-section of the retinal layers [73].

In the OCT image (Fig. 3.2), drusen are revealed primarily as small, dome-shaped elevations of the retinal pigment epithelium layer. In cross-section, a clear deformation of the continuity of the RPE line, under which deposits accumulate, is visible. Sometimes, differences in the reflectivity of these changes are also observed. Some of them may appear as areas with higher or lower signal reflectivity, depending on the composition of the deposit and the degree of its calcification or the presence of lipids [30].

Drusen deposits may be small, point-like and quite difficult to distinguish from the background, but they also sometimes take the form of more extensive elevations. In such cases, a dome-shaped protrusion of the RPE above the Bruch's membrane is clearly visible. In some people, a certain fluid space or a disturbed arrangement of the adjacent retinal layers can be

visible around drusen, which indicates accompanying changes in the retinal microenvironment. The formation of Drusen deposits is promoted by genetic factors, aging of the body and various systemic disorders, including changes in lipid metabolism [73], [74].

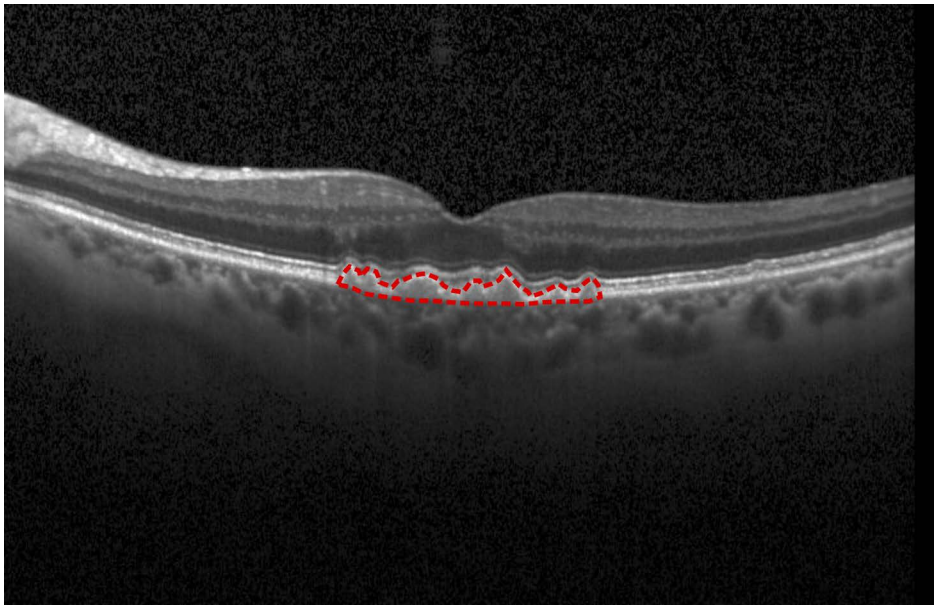


Fig. 3.2. OCT image illustrating drusen-related retinal changes. This figure shows an example OCT cross-sectional scan with drusen deposits visible as dome-shaped elevations of the retinal pigment epithelium. Pathological changes identified by ophthalmologists are highlighted in red.

3.4. Datasets Description and Preparation

For the purposes of the research conducted within this monograph, two data sets were constructed. The first one contains data related to retinitis pigmentosa disease. The second set contains OCT images of acquired vitelliform lesions. All studies were carried out in accordance with the relevant guidelines and regulations, as well as, the Helsinki Declaration. Their scope and method of the study were approved by the Ethics Committee of the Medical University of Lublin (no. KE-0254/260/12/2022). In addition, informed consent was obtained from all study participants or their legal guardians. All data available as part of the conducted studies were anonymized to prevent obtaining personal patient data [75].

3.4.1. Retinitis Pigmentosa Dataset

Medical imaging of Ultra-widefield Photography (UWPC) and Ultra-widefield Autofluorescence (UWFAF) have been obtained using an Optos 200TX device (Optos PLC) at the Chair and Department of General and Pediatric Ophthalmology of the Medical University of Lublin, Poland. The study involves 348 patients suffering from retinal dystrophy, such as: RP and CORD. The obtained data were reviewed by two independent experts. For this study, 1226 image data were collected involving rare eye diseases: 686 (RP) and 166 (CORD). The creation of the data is depicted in Fig. 3.3.

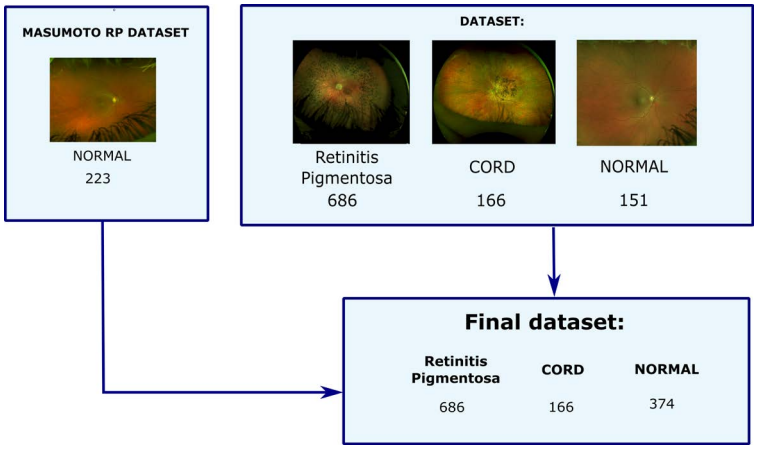


Fig. 3.3. Structure and class-wise composition of the RP dataset. The diagram illustrates the overall organization of the RP dataset, including images from retinitis pigmentosa (RP), cone-rod dystrophy (CORD), and healthy control subjects. The figure also presents the distribution of samples across training, validation, and test subsets.

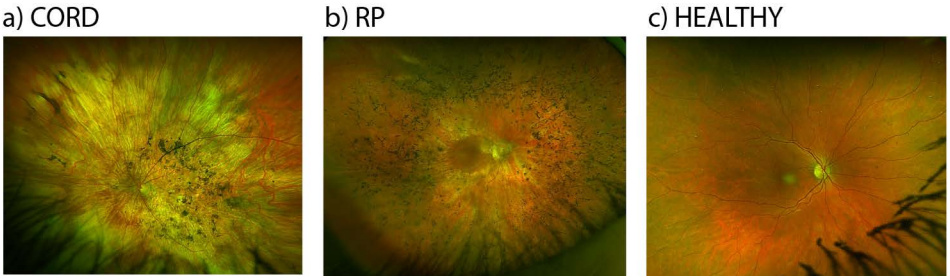


Fig. 3.4. Representative ultra-widefield retinal images from the RP dataset. The figure shows example images corresponding to the three classes included in the RP dataset: cone-rod dystrophy (CORD), retinitis pigmentosa (RP), and healthy controls. Both pathological and normal retinal appearances are illustrated.

The control group consisting of 374 healthy optos data were gathered both from the masumoto dataset (223 images) [76], as well as from the Medical University of Lublin – 151 images. An example of all types of images is presented in Fig. 3.4.

3.4.2. AVL Dataset

The structure of the OCT dataset, curated specifically for this study, is illustrated in Fig. 3.5. The AVL OCT subset from the same retinal imaging modality (AngioVue OCT-A system – Optovue Inc., Fremont, CA, USA) were gathered from two ophthalmic research centers: the Eye Clinic, Department of Public Health, Federico II University in Italy (ORC1) and the Chair and Department of General and Pediatric Ophthalmology at the Medical University of Lublin in Poland (ORC2). The collected AVL subset has been thoroughly verified by a group of experienced ophthalmologists. The images that were blurred or illegible have been excluded from the study. In total, 125 images qualified for this study.

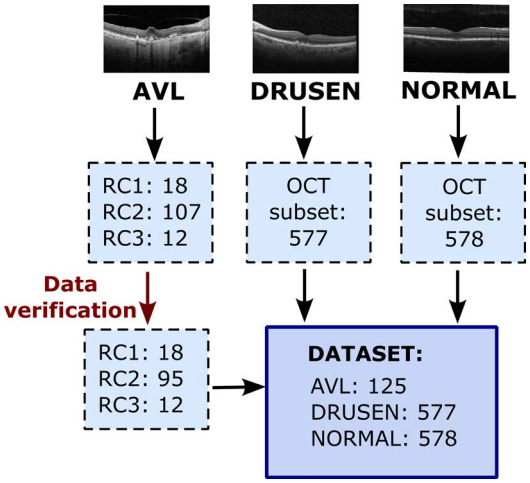


Fig. 3.5. The structure of AVL dataset. This figure depicts the composition of the OCT dataset used in the AVL-related experiments, including the number of images acquired from different ophthalmic research centers and their allocation into training, validation, and test subsets.

All OCT images were initially examined for motion artifacts and overall quality to remove scans with excessive blur or poor contrast. Each retained image was then processed with a Gaussian and median filter to reduce speckle noise while preserving key structural boundaries.

First, the images were denoised. Next, for AVL images, a rectangular region around the macula was manually selected to exclude irrelevant peripheral areas. The images were then resized to a fixed resolution of *224×224* pixels to match the classification networks' input requirements. Finally, each image's pixel intensities were scaled to a [0, 1] range based on its individual minimum and maximum values, minimizing variability across different imaging sessions.

The Drusen and normal cases subsets were collected from the Labelled Optical Coherence Tomography [77]. Thus, ORC1/ORC2 are specialized ophthalmology research centres focusing primarily on advanced AMD (or a particular variant such as AVL), and they have a limited collection of either normal eyes or mild-to-moderate AMD (Drusen) images. For this reason, it was decided to include two other classes from public data sets in the conducted studies.

By using the publicly available UCSD dataset, we ensure that both the Drusen and Normal classes have sufficient representation. Including a well-known dataset also increases the study's credibility and reproducibility. The final dataset contains 125 AVL images, 577 DRUSEN images, and 578 images from healthy cases. The entire dataset was split into training, validation, and testing subsets, following a 70%, 15%, and 15% split, respectively, while maintaining balanced proportions across the classes.

3.4.3. Data augmentation

Data augmentation is a commonly used technique in image classification, aimed at improving the performance and generalization ability of machine learning models [22]. It consists of increasing the dataset by generating new training samples using different transformations applied to the original images. This process increases the diversity of available samples, which at the same time reduces the risk of overfitting the model.

Image modifications take into account typical variability such as differences in lighting, scale, orientation, and other features, which allows the model to adapt better to real-world conditions during training. Moreover, data augmentation enables the enlargement of the dataset without the need for obtaining additional samples qualified by a given criterion. It also improves the model's ability to generalize new data, previously unused in the training process. Thus, data augmentation plays a pivotal role in DL models training by adapting it to the diverse real word images [78].

The following data augmentation method was utilized [78], [79]:

- Rotation: Images are rotated by a chosen angle to provide various perspectives and orientations.
- Scaling: Images are resized while maintaining their aspect ratio, allowing the model to learn from different distances or sizes.
- Crop and Resize: Randomly cropping an image and then resizing it helps the model to learn features from various focal points and scales within the image.
- Colour Jittering: Modifying brightness, saturation, contrast, and hue simulates different lighting conditions and colour variations.
- Gaussian Noise: Small distortions are introduced using random noise to reduce the model's sensitivity.
- Random Erasing: Selected parts of the image are replaced with a solid colour or noise, helping the model focus on relevant features.
- Cutout: Randomly masking out square regions in the image improves robustness, encouraging the model to focus on various parts of the image during training.
- Horizontal and Vertical Flipping: These operations simulate various image orientations and perspectives, improving the training process, especially for symmetrical features.

Due to the small number of datasets, the above-mentioned methods were applied to enhance the model's performance. An example of image transformation can be seen in Fig. 4.2 in Chapter 4. Moreover, the data was generated applying the dedicated Deep Convolutional Generative Adversarial Network (DCGAN) method described in detail in Chapter 4. The final structures of datasets are shown in Table 3.1.

Table 3.1. The final structure of data used in described studies. The amount of data in the training set after the data augmentation process.

Dataset	Class	Train	Validation	Test
RP dataset	RP	500	103	103
	CORD	475	25	25
	HEALTHY	485	56	56
AVL dataset	AVL	400	19	19
	DRUSEN	450	87	87
	HEALTHY	450	86	87

3.5. Studies Environment

All studies in this monograph were conducted using a computer equipped with a 24-core AMD Ryzen Threadripper PRO 7965WX processor clocked at 4.20 GHz. The computer uses 512 GB of RAM. The system also benefits from the support of a NVIDIA GeForce RTX 4090 graphics card. The equipment runs in a 64-bit operating system environment. This configuration enables both rapid prototyping of models and training of extensive neural network architectures. The high volume of RAM reduces the risk of problems related to a lack of resources, enabling larger batch processing, which can ultimately accelerate the model training process. The RTX 4090 GPU significantly accelerates matrix calculations, which are crucial for training deep learning models.

The medical data were divided into training, validation, and test sets (or into training and validation sets, depending on the experimental scenario) while maintaining strict patient separation. Images from the same patient did not appear simultaneously in different subsets. This eliminated the risk of data leakage and ensured a reliable assessment of the model's generalization ability. Additionally, separation was introduced at the level of medical centers. Data from the same center were not simultaneously present in the training and test/validation sets. This limited the impact of specific acquisition protocols and hardware differences. In the case of division into three subsets (training, validation, test), the process involved five iterations. Each iteration included data division, model training, and testing. The final results were reported as the average of five replicates. In the case of division into two subsets, 5-fold cross-validation was applied. The data were divided into five distinct parts. In each iteration, four parts were used for training and one for validation. The process was repeated five times, changing the validation set. The final result was the average of the five runs. Synthetic and augmented data were added exclusively to the training sets.

4. Medical Data Augmentation

This section overviews the dataset, classical augmentation methods, DCGAN, XGBoost, as well as evaluation metrics and measures. It aims to enhance retinitis pigmentosa image outcomes by integrating DCGAN with transfer learning. For classification purposes, it applies the architecture of the VGG16 model and outlines the entire proposed methodology in Fig. 4.1.

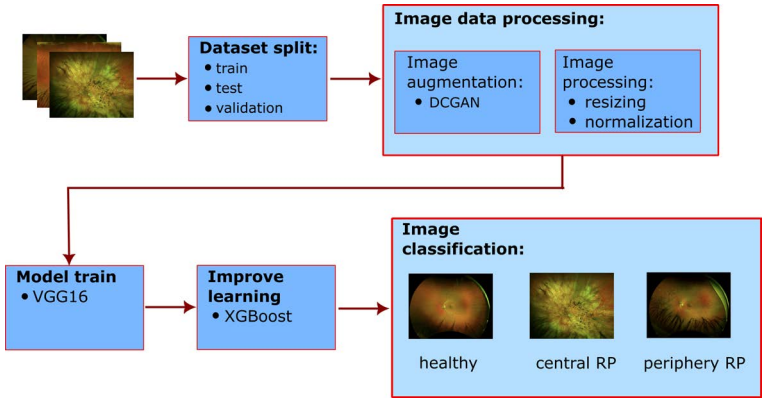


Fig. 4.1. Proposed methodology for DCGAN retinitis pigmentosa images augmentations and VGG16+XGBoost Classification. For other used datasets the workflow is analogous. The figure presents the complete workflow of the proposed data augmentation and classification framework. Synthetic retinal images are generated using a Deep Convolutional Generative Adversarial Network (DCGAN) and subsequently combined with real images to form an extended training set. Feature extraction is performed using a VGG16 backbone, while final classification is carried out using the XGBoost algorithm. The same methodological pipeline is applied analogously to all datasets considered in this study.

4.1. Classical Image Augmentation Methods

Classical image augmentation encompasses a variety of traditional methods that manipulate images in ways that simulate real-world scenarios, enhance the dataset, and encourage the model to learn invariant representations. Some common augmentation techniques include rotation, flipping, scaling, translation, brightness and contrast adjustments, noise addition, cropping and padding, shearing, elastic deformations, and gamma correction.

These image augmentation techniques help increase the diversity of the training dataset, improve the model's ability to generalize to unseen data and enhance its performance on various computer vision tasks. By incorporating these traditional augmentation methods, it is possible to build more reliable and effective machine learning models for image analysis and classification tasks.

Rotating medical images involves changing the angle of the image around its centre. By rotating images at different angles, the model can learn to recognize patterns from various perspectives. This augmentation technique helps increase the variability in the dataset and thus can enhance the model's ability to correctly classify unseen data. Rotation augmentation is particularly useful in medical imaging tasks where the orientation of structures may vary across different scans or modalities.

Flipping medical images horizontally or vertically involves mirroring the image along a specified axis. This augmentation technique can provide additional variations to the dataset and help the model learn invariant features. Flipping images is a simple yet effective way to increase the diversity of the training data without the need for collecting new samples. In medical imaging, flipping can be useful for tasks where the orientation of structures does not affect the diagnosis, such as in certain segmentation tasks.

Scaling medical images involves resizing the images to different dimensions. This augmentation technique can simulate images taken from varying distances or with different resolutions. Scaling helps the neural networks model learn to recognize objects at different sizes and can improve its robustness to variations in image resolution. In medical imaging, scaling augmentation is beneficial for tasks where the size of anatomical structures may vary across patients or imaging modalities.

Translating medical images involves shifting the image horizontally or vertically. This augmentation technique can simulate changes in position or perspective and help the model learn to detect objects at different locations within the image. Translation augmentation increases the variability in the

dataset and enhances the model's ability to localize structures accurately. In medical imaging tasks such as object detection or localization, translation augmentation can improve the model's performance by exposing it to diverse spatial transformations.

Adjusting the brightness and contrast of medical images involves changing the intensity levels of pixel values. This augmentation technique can help the model learn to be unchanging to variations in lighting conditions. By augmenting images with different brightness and contrast levels, the model becomes more robust to changes in illumination and can better process unseen data. In medical imaging, brightness and contrast adjustments are important for tasks such as image classification or segmentation, where variations in image quality can affect the model's performance. The examples of modified images are shown in Fig. 4.2.

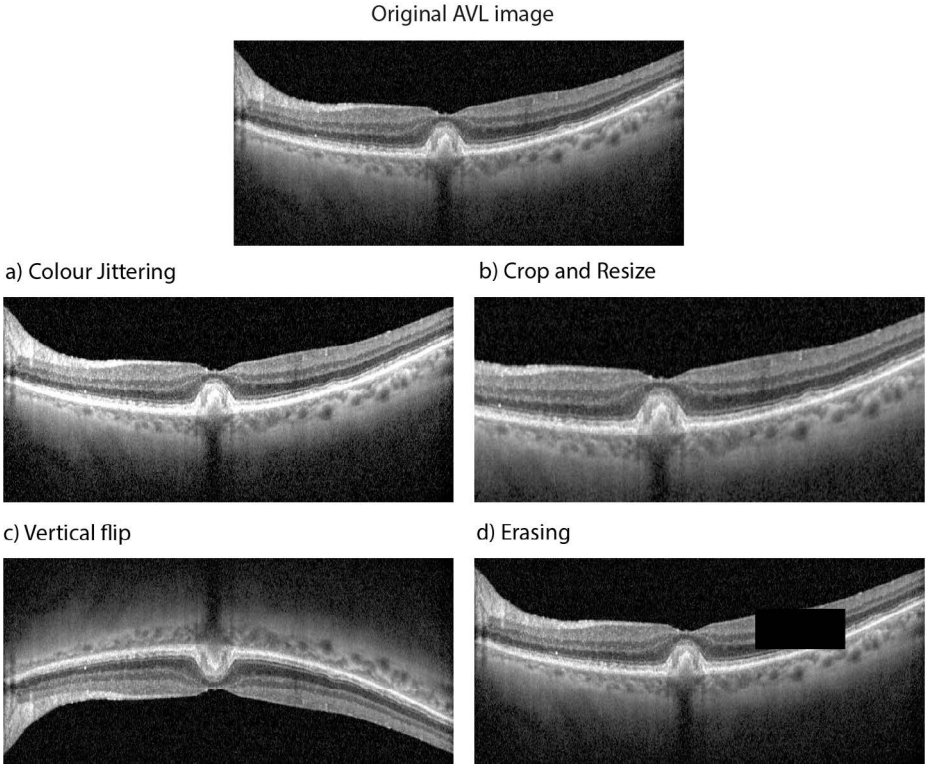


Fig. 4.2.a–d. Examples of classical image augmentation techniques applied to retinal images. Figure illustrates representative examples of classical image augmentation methods used to increase dataset diversity, including rotation, horizontal and vertical flipping, brightness adjustment, cropping and resizing, Gaussian noise injection, random erasing, and combined transformations. These operations simulate realistic variability in retinal image acquisition conditions and improve model robustness by exposing classifiers to a wider range of appearance variations.

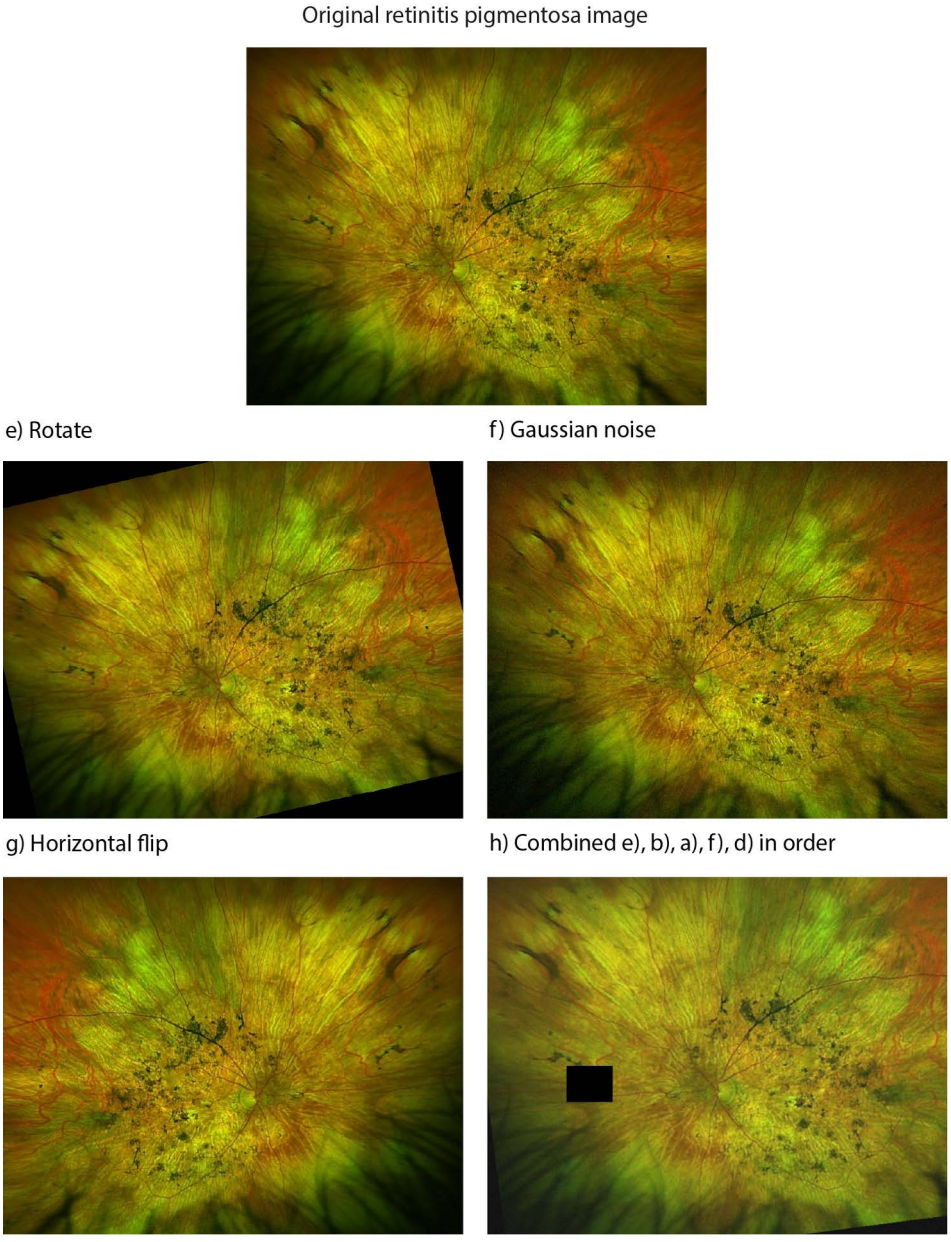


Fig. 4.2.e–h. Examples of classical image augmentation techniques applied to retinal images. Figure illustrates representative examples of classical image augmentation methods used to increase dataset diversity, including rotation, horizontal and vertical flipping, brightness adjustment, cropping and resizing, Gaussian noise injection, random erasing, and combined transformations. These operations simulate realistic variability in retinal image acquisition conditions and improve model robustness by exposing classifiers to a wider range of appearance variations.

The small size of the datasets requires limited image modification. The applied transformations included:

- horizontal flip, performed with a certain probability (selected by user);
- brightness adjustment, applied with probability p, where the brightness scaling factor was sampled uniformly from the interval [0.85, 1.15];
- resizing to 85–115% of the original dimensions, shifting by 7% to +7% along each axis, and rotating between 7 and +7 degrees, all with the same probability.

More significant changes to each image were due to a higher probability value. This probability, denoted by p, represented the intensity level of augmentation. Pixels erased during these transformations were filled with a zero value, representing the background intensity. This procedure is detailed in Algorithm 4.1. Probability values of 0.25 and 0.5 were tested.

Algorithm 4.1. Traditional Augmentation Technique.

Require: Input: An eye image $I(x,y)$ and probability p

Step 1: $I_{new1}(x,y) = Horizontal\_flip(I)$ with probability p

Step 2: $I_{new2}(x,y) = AdjustBrightness(I_{new1},(0.85,1.15))$ with probability p

Step 3:

$I_{new3}(x,y) = Scale(I_{new2},(0.85,1.15))$ with probability p

$I_{new4}(x,y) = Translate(I_{new3},(-7,+7))$ with probability p

$I_{new5}(x,y)$ $(I_{new4},(-7,+7))$ with probability p

return: $I_{new5}(x,y)$

4.2. Generative Adversarial Networks

GANs represent one of the most intriguing advancements in the field of artificial intelligence over the past decade. This cutting-edge category of machine learning frameworks has not just sparked the interest of AI researchers and practitioners but has also demonstrated a wide array of applications, ranging from generating photorealistic images to creating medical datasets. The essence of GANs lies in their unique architecture that pits two neural networks against each other in a game-theory contest.

The basics of GANs were presented in [80]. The innovation was to have two convolutional networks engage in a dynamic contest: a Generator that makes information and a Discriminator that assesses it. The Generator's objective

is to deliver indistinguishable information from a real dataset, whereas the Discriminator's goal is to discern real data from fake data. This results in a continuous learning process for both networks, leading to the generation of increasingly convincing synthetic data.

The elegance of GANs lies in their simplicity and power. They do not require labelled data to learn, making them a form of unsupervised learning. This is a significant advantage in the era of big datasets where numerous unlabelled data, but labelled information, can be rare or expensive to obtain.

However, preparing GANs is notoriously troublesome. It requires a fragile adjustment as both systems progress. If one network outpaces another altogether, the training can fail. Although there are numerous techniques and variations to stabilize the training process and improve the quality of the generated data, this is an extremely time-consuming operation.

The process of training the Discriminator and Generator and their mutual competition is based on minimizing the loss function. However, the loss function is defined separately for each of them. In the rest of the chapter, the meanings will be as follows:

x – real data;
z – vector in the Latent space;
$G(z)$ – data from the Generator, fake data;
$D(x)$ – Discriminator's assessment of real data;
$D(G(z))$ – Discriminator's assessment of fake data;
$p_{data}(x)$ – Original data distribution;
$p_g(x)$ – Generated distribution;
$Error(a,b)$ – Error between a and b.

The Discriminator's objective is to accurately identify generated images as fake and real data points as genuine. Consequently, the loss function L_D for the Discriminator can be defined as:

$$L_{\mathrm{D}} = Error(D(x), 1) + Error(D(G(z)), 0\ ). \tag{4.1}$$

Here, the generic notation for Error is used to denote some function that measures the distance or discrepancy between the two functional parameters. Generally, concepts like cross-entropy or Kullback-Leibler divergence are commonly applied in these situations. Similarly, a loss function for the Generator L_G can be defined. The Generator's goal is to deceive the Discriminator to the extent that the latter incorrectly classifies generated images as real.

$$L_{\mathrm{G}} = Error(D(G(z)), 1\ ). \tag{4.2}$$

It is important to remember that a loss function is something that needs to be minimized. For the Generator, this means minimizing the difference between 1 (the label for real data) and the Discriminator's assessment of the generated fake data.

Loss functions

In information theory, cross-entropy between two probability distributions p and q, which cover the same set of events, quantifies the average number of bits required to specify an event from the set when the coding scheme is optimized based on the estimated distribution q instead of the actual distribution p [81].

A commonly used loss function in binary classification problems is binary cross-entropy. Mathematically cross-entropy function of the distribution q relative to a distribution p over a given set can be defined as Eq. (4.3) [81]:

$$H(p, q) = E_{x\sim p_x}[-logq(x)], \tag{4.3}$$

where: $\mathrm{E}_{x\tilde{}px}$ denotes the expected value operator with respect to the distribution p_x.

In classification tasks, the random variable is discrete, thus the expectation can be represented as a summation [81]:

$$H(p, q) = -\sum_{x\in X} p(x)logq(x). \tag{4.4}$$

In case of binary cross-entropy, the expression (4.4) can further be simplified since there are only two labels: one and zero [81]:

$$H(y, \hat{y}) = -\sum \left[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})\right]. \tag{4.5}$$

This is defined as the Error function. Binary cross-entropy serves to measure the difference between two distributions in the context of binary classification, determining whether an input data point is true or false. Applying this to loss functions in (4.1) [81]:

$$L_D = -\sum_{x\in X, z\in \zeta} [\log(D(x)) + \log(1 - D(G(z)))]. \tag{4.6}$$

The same can be done for Eq. (4.2) [81]:

$$L_G = -\sum_{z \in \zeta} \log(D(G(z))). \tag{4.7}$$

Loss functions to train both the Discriminator and the Generator are defined by Eq. (4.6) and (4.7), respectively. It has to be noted that for the Generator's loss function, the loss is minimized if *D(G(z))* is close to 1, while log(1) = *0*. The effectiveness of Eq. (4.6) can be easily seen with a similar approach.

The original paper [80] demonstrated a slightly different version of the two loss functions discussed above:

$$\max_D \{\log(D(x)) + \log(1 - D(G(z)))\}. \tag{4.8}$$

Essentially, the Distinction between Eq. (4.6) and (4.8) lies in the sign and the aim to minimize or maximize a particular quantity. In Eq. (4.6), the function was formulated as a loss function to be minimized, whereas the original formulation presented it as a maximization problem, with the sign flipped accordingly.

In [80] frames, Eq. (4.8) is described as a min-max game, where the objective of the Discriminator is to maximize the given quantity while the Generator aims to do the opposite:

$$\min_G \max_D \{\log(D(x)) + \log(1 - D(G(z)))\ \}. \tag{4.9}$$

The min-max formulation directly illustrates the adversarial character of the competition between the Discriminator and the Generator. However, in practice, separate loss functions for the Generator and the Discriminator are defined as Eq. (4.9). This is because the gradient of the function $y = \log(x)$ is steeper near $x = 0$ compared to the function $y = \log(1\text{-}x)$. Thus, maximizing $\log(D(G(z)))$, or equivalently, minimizing $-\log(D(G(z)))$, leads to quicker and more substantial improvements in the Generator's performance than minimizing $\log(1\text{-}D(G(z)))$.

The training of the Generator and Discriminator models constitutes an optimization problem for the loss functions. Their parameters must be chosen to minimize the respective loss functions to the greatest extent possible.

Discriminator Training

When a GAN is being trained, one model is typically trained at a time. In other words, when the Discriminator is being trained, the Generator is assumed to be fixed. The min-max game can now be revisited. The quantity of interest can be defined as a function of *G* and *D*, referred to as the value function [82]:

$$V(G,D) = \mathrm{E}_{x\sim P_{\mathrm{data}}}[\log(D(x))] + \mathrm{E}_{z\sim P_z}[\log(1-D(G(z)))]. \tag{4.10}$$

In practical applications, more interest lies in the distribution modelled by the Generator rather than by P_z. Therefore, a new variable, $y = G(z)$, can be defined, and this substitution can be used to rewrite the value function [82]:

$$\begin{aligned} &V(G,D) = \mathrm{E}_{x\sim p_{\mathrm{data}}}[\log(D(x))] + \mathrm{E}_{y\sim p_g}[\log(1-D(y))] \\ &= \int_{x\in X}(p_{\mathrm{data}}(x)\log(D(x)) + p_g(x)\log(1-D(x)))\mathrm{d}x. \end{aligned} \tag{4.11}$$

The goal of the Discriminator is to maximize this value function. By taking the partial derivative of $V(G,D)$ with respect to $D(x)$, it can be observed that the optimal Discriminator, denoted as $D^*(x)$, occurs when [82]:

$$\frac{p_{\mathrm{data}}(x)}{D(x)} - \frac{p_g(x)}{1-D(x)} = 0. \tag{4.12}$$

By rearranging Eq. (4.12), the following can be obtained:

$$D^*(x) = \frac{p_{\mathrm{data}}(x)}{p_{\mathrm{data}}(x) + p_g(x)}. \tag{4.13}$$

This constitutes the condition for the optimal Discriminator. It is noteworthy that the formula is intuitively logical: if a given sample x is highly genuine, $p_{data}(x)$ is expected to be close to one, while $p_g(x)$ should converge to zero. In such a case, the optimal Discriminator would assign a value of 1 to that sample. Conversely, for a generated sample $x = G(z)$, the optimal Discriminator would be expected to assign a label of zero, as $p_{data}(G(z))$ and is anticipated to be close to zero.

Generator Training

To train the Generator, the Discriminator is assumed to be fixed and analysis of the value function proceeds. First, the result found above, namely Eq. (4.13), is plugged into the value function to observe the outcome [83]:

$$\begin{aligned} V(G, D^*) &= \mathrm{E}_{x \sim p_{\text{data}}} \left[\log(D^*(x))\right] + \mathrm{E}_{x \sim p_g} \left[\log(1 - D^*(x))\right] \\ &= \mathrm{E}_{x \sim p_{\text{data}}} \left[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}\right] \\ &+ \mathrm{E}_{x \sim p_g} \left[\log \frac{p_g(x)}{p_{\text{data}}(x) + p_g(x)}\right]. \end{aligned} \tag{4.14}$$

Furthermore, taking inspiration from [83], Eq. (4.14) can be transformed into:

$$\begin{aligned} V(G, D^*) &= \mathrm{E}_{x \sim p_{\text{data}}}[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}] \\ &+ \mathrm{E}_{x \sim p_g}[\log \frac{p_g(x)}{p_{\text{data}}(x) + p_g(x)}] \\ &= -\log 4 \\ &+ \mathrm{E}_{x \sim p_{\text{data}}}[\log p_{\text{data}}(x) - \log \frac{p_{\text{data}}(x) + p_g(x)}{2}] \\ &+ \mathrm{E}_{x \sim p_g}[\log p_g(x) - \log \frac{p_{\text{data}}(x) + p_g(x)}{2}]. \end{aligned} \tag{4.15}$$

Essentially, the properties of logarithms are being exploited to extract a –log4 that previously did not exist. By extracting this number, changes are applied to the terms in the expectation, specifically by dividing the denominator by 2. The necessity of this step lies in the ability now to interpret the expectations as Kullback-Leibler divergence (D_{KL}) [83]:

$$\begin{aligned} V(G, D^*) &= -\log 4 + D_{KL}\left(p_{\text{data}} \| \frac{p_{\text{data}} + p_g}{2}\right) \\ &+ D_{KL}\left(p_g \| \frac{p_{\text{data}} + p_g}{2}\right). \end{aligned} \tag{4.16}$$

At the end there is a need to recalculate the Jensen-Shannon divergence (*DJS*) and is reencountered, which is defined as [83]:

$$J(P, Q) = \frac{1}{2}(D(P \| R)) + D(Q \| R). \tag{4.17}$$

where: $R = ½ (P+Q)$, for any probability density functions P and Q.

This implies that the Eq. (4.16) can be represented as a Jensen-Shannon divergence [83]:

$$V(G, D^*) = -\log 4 + 2 \cdot D_{JS}(p_{\text{data}} \| p_g). \tag{4.18}$$

The conclusion of this analysis is straightforward: the goal of training the Generator, which is to minimize the value function *V(G,D)*, aligns with minimizing the Jensen-Shannon divergence between data distribution and generated examples' distribution. The Generator is desired to learn the underlying distribution of the data from sampled training examples. In other words, p_g and p_{data} should be as close to each other as possible. The optimal Generator is thus the one that can mimic p_{data} to model a compelling distribution p_g.

4.3. Deep Convolutional Generative Adversarial Networks

Deep Convolutional Generative Adversarial Networks (DCGANs) are a specialized type of GANs that utilize deep convolutional layers to enhance performance, particularly in image generation tasks. However, the original GAN architecture did not fully leverage the capabilities of convolutional layers, which are highly effective for image processing. In 2015, the architecture of DCGANs was proposed in [84]. This new architecture incorporated few enhancements over the original GANs to improve stability and performance during training. Several key differences between DCGANs and GANs can be highlighted.

In DCGANs, deep convolutional layers are extensively used in both the Generator and the Discriminator. This contrasts with the original GANs, which relied on fully connected layers. Convolutional layers allow for better spatial structure capture, which is crucial for generating high-quality images.

Moreover, the Generator in DCGANs employs transposed convolutions (also known as deconvolutions) to upsample the input noise vector into an image. This method helps in producing images with finer details. In contrast, the original GANs typically used fully connected layers followed by reshaping and upsampling operations, which could lead to less detailed images.

Furthermore, the Discriminator in DCGANs uses strided convolutions instead of pooling layers to downsample images. This approach preserves

more spatial information, enhancing the Discriminator's ability to distinguish between real and fake images. Original GANs often used max-pooling or average-pooling layers for downsampling, which might lose important details.

Another difference is Batch normalization, which is applied in both the Generator and the Discriminator in DCGANs. This technique stabilizes the training process by normalizing the inputs of each layer, thus preventing issues like mode collapse.

Specific activation functions are used in DCGANs to improve performance. For instance, *Leaky ReLU* is used in the Discriminator, and *ReLU* followed by hyperbolic tangent is applied in the Generator's output layer.

By incorporating the above-mentioned enhancements, DCGANs have improved the quality and stability of the image generation process compared to the original GANs. The differences in architectural designs and training techniques allow DCGANs to generate more realistic and higher-resolution images, making them a significant advancement. The idea of the DCGAN workflow for retinitis pigmentosa data augmentation is presented in Fig. 4.3.

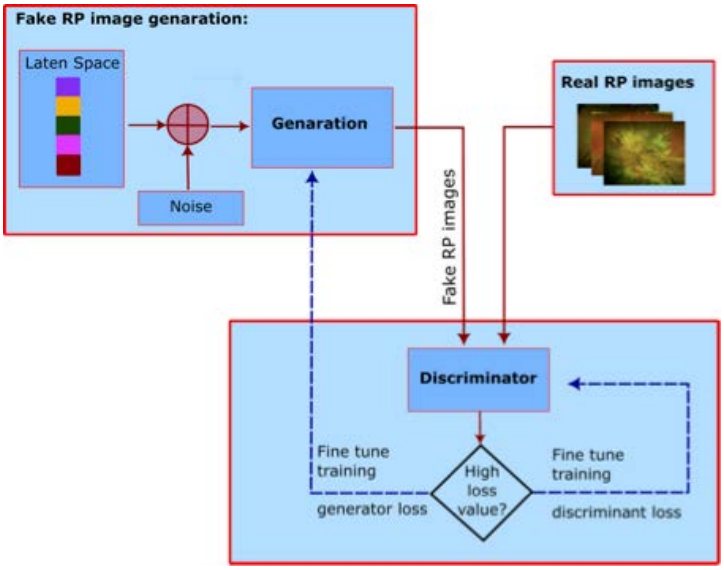


Fig. 4.3. The DCGAN workflow for retinitis pigmentosa data augmentation. Random noise vectors are sampled from the latent space and transformed by the Generator into synthetic retinal images. While the Discriminator iteratively learns to distinguish between real and generated samples. The adversarial training process continues until the generated images become indistinguishable from real data. The same workflow is employed for other datasets used in this monograph.

Generator in DCGAN

In a GAN, the Generator is one of two neural networks engaged in a zero-sum game, competing against the Discriminator. The Generator's task is to learn how to create data that is indistinguishable from the real data provided during training. It takes a random noise vector (latent space vector) as input and generates data, aiming to produce outputs that are indistinguishable from genuine data.

The Generator network is built with an 11-layer architecture, containing over 15 million adjustable parameters, as shown in Fig. 4.4. It starts by accepting a 128-element vector of random numbers uniformly distributed between 0 and 1 (exclusive). These values first pass through a Fully Connected (FC) layer with 15,360 neurons. The output is then reshaped into a three-dimensional structure, similar to a *6×5* pixel image but with a depth of 512 channels.

Next, the network alternates between standard 2D convolutions (Conv) and 2D transposed convolutions (also known as Conv trans up or "deconvolutions"). These layers use a *5×5* kernel size with 'same' padding to maintain dimensions, while the transposed convolutions use a stride of 2, doubling the spatial dimensions of their input. Except for the final layer, each layer uses the *Leaky ReLU* function for activation, introducing non-linearity. The final layer uses a hyperbolic tangent (tanh) activation function, chosen because its output range matches the desired bound of [–1, 1] for image generation and offers a cantered zero output, beneficial for training.

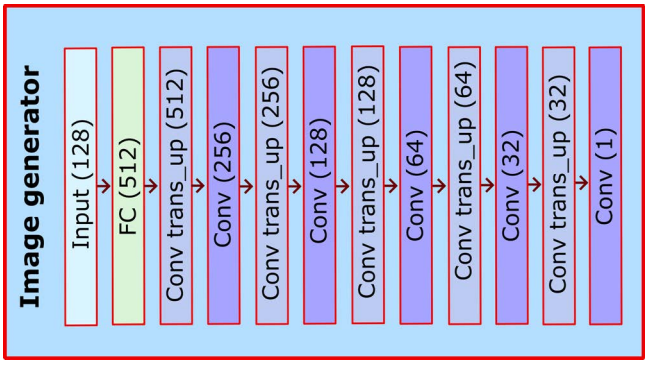


Fig. 4.4. The schema of Deep Convolutional image Generator. The network transforms a latent noise vector into a synthetic retinal image through a sequence of fully connected layers followed by transposed convolutional layers.

After five cycles of alternating convolution and transposed convolution layers, with each transposed layer expanding the input size, the architecture produces an image with a resolution of *192×160* pixels and a single-colour channel [22].

Discriminator in DCGAN

The Discriminator used in this study is based on a standard CNN design optimized for tasks involving binary classification. It consists of a total of 11 layers with around 9.5 million trainable parameters, as shown in Fig. 4.5.

The Discriminator work starts with a grayscale image, with dimensions of *192×160*, used as input. It goes through five convolutional layers, with strides of 1 and 2 alternating between each pass. A stride of 2 aids in downsampling, as there are no pooling layers included in the structure. The Discriminator is completed by two fully connected (FC) layers. Across the entire network, all layers utilize a *Leaky ReLU* activation function, except for the final layer which does not have an activation function [22].

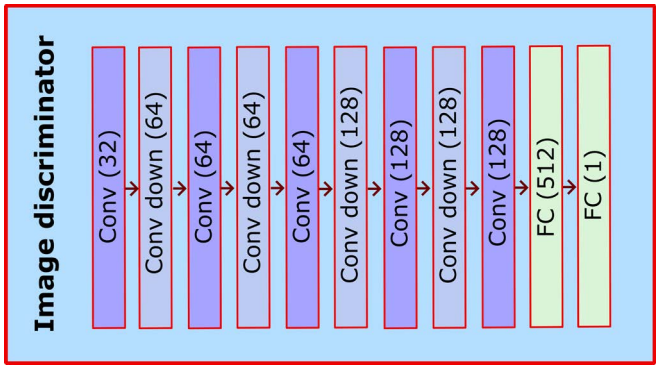


Fig. 4.5. The schema of Deep Convolutional image Discriminator. The network receives either real or synthetic retinal images as input and processes them through multiple convolutional layers with strided downsampling, followed by fully connected layers for binary classification.

Hybrid Approach Using Xtreme Gradient Boost

In this study the traditional XGBoost algorithm originally described in [85] underwent minor adjustments. XGBoost was designed for datasets containing n examples and m features expressed as $D\{x_i, y_i\}(|D| = n, x_i \in R^m, y_i \in R)$. While applying the additive function S as detailed in Eq. (4.19), it facilitates the prediction of outcomes:

$$y_{ii} = \sum_{s=1}^{S} f_s(x_i), f_s \in \mathscr{F}. \tag{4.19}$$

In this context, $\mathscr{F} = \left\{f(x) = \vartheta_{q(x)}\right\}(q : R^m \to T, \vartheta \in R^T)$ represents the spatial regression of trees, q denotes the tree structures and T is the count of leaves on the tree. The overall score is determined by $\mathfrak{9}$, which consolidates the respective leaves. The regularization goal of obtaining a set of functions is outlined in Eq. (4.20) and Eq. (4.21) [85].

$$J() = \sum_i l(y_{ii} - y_i) + \sum_j \phi(f_j), \tag{4.20}$$

$$\phi(f) = \gamma T + \frac{1}{2}\lambda \| \vartheta \|^2. \tag{4.21}$$

The loss function l quantifies the discrepancy between the predicted value as denoted by y_{ii} and the actual target value y_i. The parameter Φ signifies the model's complexity. The additional regularization term λ aids in smoothing the final learned weights, thereby reducing the risk of overfitting. Essentially, the regularized objective tends to favour models that employ straightforward and predictive functions. Consequently, [85] utilized an additive strategy, as illustrated in Eq. (4.22). Here, $y_i(t)$ represents the prediction for the i-th instance at the t-th iteration.

$$J^{(t)} = \sum_{i=1}^{n} l\left(y_i, y_{ii}^{(t-1)} + f_f(x_i)\right) + \vartheta(f_t). \tag{4.22}$$

Eq. (4.22) can be enhanced by incorporating a second-order approximation, as illustrated in Eq. (4.23) [85].

$$J^{(t)} \simeq \sum_{i=1}^{n} \left[l\left(y_i, y_{ii}^{(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \vartheta(f_t). \tag{4.23}$$

At step t, $g_i = \partial_{y_{ii}^{(t-1)}} l\left(y_i, y_{ii}^{(t-1)}\right)$ and $h_i = \partial^2_{y_{ii}^{(t-1)}} l\left(y_i, y_{ii}^{(t-1)}\right)$ are used as loss functions for the first and second gradient statistics, respectively. Eq. (4.23) can be simplified by removing the constant term, leading to Eq. (4.24) [85].

$$\widetilde{J^{(t)}} = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \vartheta(f_t). \tag{4.24}$$

Furthermore, if $S_j = i|q(x_i) = j$ will be defined as a leaf j, the Eq. (4.24) can be expressed in a form of Eq. (4.25) by expanding $\mathfrak{9}$ [86]:

$$\begin{aligned}\widetilde{J^{(t)}} &= \sum_{i=1}^{n}\left[g_i f_t(x_i) + \tfrac{1}{2}h_i f_t^2(x_i)\right] + \gamma T + \tfrac{1}{2}\lambda\sum_{j=1}^{T}\omega_j^2 \\ &= \sum_{j=1}^{T}\left[\sum_{i\in S_j}(g_i)\omega_j + \tfrac{1}{2}\left(\sum_{i\in S_j}(h_i+\lambda)\omega_j^2\right)\right] + \gamma T.\end{aligned} \tag{4.25}$$

The optimal value ω of leaf j for the static function q is expressed by Eq. (4.26) and Eq. (4.27) [86]:

$$\omega_j^*(q) = -\frac{\sum_{i\in s_j} g_i}{\sum_{i\in s_j} h_i + \lambda}, \tag{4.26}$$

$$\widetilde{J^{(t)}}(q) = -\frac{1}{2}\sum_{j=1}^{T}\frac{\left(\sum_{i\in s_j} g_i\right)^2}{\sum_{i\in s_j} h_i + \lambda}. \tag{4.27}$$

Algorithm 4.2 outlines the complete data augmentation process using the XGBoost module. DCGANs have been shown to be very effective at generating high-quality images. In the case of this monography all used datasets were augmented by DCGANs.

Algorithm 4.2. Data augmentation with VGG16 and XGBoost classification.

Require: Input: An eye image *I(x,y)*

Step 1: Converting image into gray scale and applying normalization

$I : \{X \subset \mathbb{R}^n\} \to \{Min, ..., Max\}$ change into $I_N : \{X \subset \mathbb{R}^n\} \to \{newMin, ..., newMax\}$

$$I_N = (newMax - newMin)\frac{1}{1+e^{-\frac{I-\beta}{\alpha}}} + newMin$$

Step 2: Basic image augmentation (rotation, scaling, flipping)

Rotation: I_{new} = (x_r,y_r)

$$x_r = (x - center_x)\cdot\cos(angle) - (y - center_y)\cdot\sin(angle) + center_x$$

$$y_r = (x - center_x)\cdot\sin(angle) + (y - center_y)\cdot\cos(angle) + center_y$$

Scale: I_{new} = (x_{ratio},y_{ratio})

$$x_{\text{ratio}} = \frac{old\_image_x}{new\_image_x}, y_{\text{ratio}} = \frac{old\_image_y}{new\_image_y}$$

$$I_{\text{new}}(\lfloor x \cdot x_{\text{ratio}} \rfloor, \lfloor y \cdot y_{\text{ratio}} \rfloor)$$

Horizontal flip: I_{new} = (width - x - 1,y)

```
Step 3: Image augmentation using DCGAN
Step 4: Feature extraction by VGG16 network
Step 5: Feature mapping with XGBoost
    Adding output to the XGBoost tree by predicting the weight
    of leaf j (Eq. 4.24)
return: I_new = (x_r, y_r)
```

4.4. Evaluation Metrics

Classification evaluation methods are essential in assessing the performance of machine learning models in categorizing data into different classes. Various metrics, including accuracy, precision, recall, F1-score, and specificity are commonly used to evaluate the effectiveness of classification models. These metrics provide insights into the model's predictive power, its ability to correctly classify instances and its overall performance. Each metric plays a crucial role in understanding different aspects of the model's performance. In Eq. (4.28)–(4.31) *TP*, *FP*, *TN* and *FN* are true-positive, false-positive, true-negative and false-negative values, respectively [87].

True Positive is the number of instances that were correctly predicted as positive by the model. True Negative denotes a number of instances that were correctly predicted as negative by the model. False Positive indicates the number of instances that were incorrectly predicted as positive by the model (actually negative). False Negative is the number of instances that were incorrectly predicted as negative by the model (actually positive).

Accuracy is a fundamental metric in classification evaluation that measures the proportion of correctly predicted instances out of the total instances in the dataset. It is calculated as the ratio of the number of correct predictions to the total number of predictions made by the model. Accuracy is expressed as:

$$Accuracy\ (Acc) = \frac{TP + TN}{TP + TN + FP + FN} * 100\%. \tag{4.28}$$

In classification problems, precision is a metric that measures the accuracy of the positive predictions made by a model. Specifically, precision is defined as the ratio of true positive (*TP*) predictions to the total number of positive predictions (i.e., the sum of true positives and false positives). Mathematically, it is expressed as:

$$Precision\ (Prec) = \frac{TP}{TP + FP} * 100\%. \tag{4.29}$$

Precision is crucial in scenarios where the cost of false positives is high. For example, in medical diagnostics, predicting that a patient has a disease (when they do not) could lead to unnecessary stress and treatment. High precision ensures that when the model predicts a positive outcome, it is likely to be correct. Precision helps in understanding the quality of the model's positive predictions. High precision means that most of the positive predictions are relevant and correct, indicating that the model is not easily misled by noise in the data. Precision is particularly useful when dealing with imbalanced datasets where the positive class is rare compared to the negative class. In such cases, precision helps to evaluate the model's performance specifically on the positive class, providing insights into its ability to correctly identify instances of interest.

Recall, also known as sensitivity or true positive rate, measures the proportion of correctly predicted positive instances (TP) out of all actual correctly predicted instances in the dataset. Recall assesses the model's ability to capture all positive instances without missing any. Recall is calculated as:

$$Recall = \frac{TP}{TP + FN} * 100\%. \tag{4.30}$$

Recall is used in scenarios where it is important to identify all positive cases, even if it means including some false positives. For instance, in medical screening for a serious disease, it is vital to catch every potential case to ensure that no patient goes untreated.

F1-score is a metric that combines precision and recall to provide a single measure of a model's performance. It is the harmonic mean of precision and recall, giving equal weight to both metrics. F1-score is particularly useful when there is a need to balance between precision (the accuracy of positive predictions) and recall (the ability to find all positive instances). Mathematically, it is expressed as:

$$F1\text{-}score\ (F1) = 2 * \frac{Precision * Recall}{Precision + Recall} * 100\%. \tag{4.31}$$

F1-score provides a single, comprehensive metric that reflects both precision and recall, making it easier to compare the performance of different models or to tune model parameters. In cases where the classes are imbalanced, the F1-score is more informative than accuracy. Accuracy can be misleading

in such scenarios because it can be high even if the model is not effectively identifying the minority class. F1-score gives a better indication of how well the model is performing on the minority class.

Specificity, also known as True Negative Rate (TNR), measures the percentage of correctly detected negative cases out of all true negatives. In other words, it tells how well the model does not confuse negative and positive examples. Mathematically, it is expressed as:

$$Specificity = \frac{TN}{TN + FP} * 100\%. \tag{4.32}$$

Accuracy quantifies the proportion of correct predictions across all classes. However, when dealing with imbalanced datasets, for instance, cases of RP/CORD/AVL/DRUSEN being relatively rare compared to HEALTHY, relying on Accuracy alone, may be misleading. The model frequently defaults to the majority (HEALTHY) class.

Precision measures the fraction of predicted positives (i.e., "sick" patients) that are truly positive. High Precision therefore minimizes the number of False Positives, preventing healthy individuals from being misclassified as sick.

Recall indicates how many of the truly sick individuals are correctly identified. High Recall reduces the number of False Negatives, ensuring that few sick patients are missed.

Specificity gauges the model's ability to correctly identify healthy individuals, thereby avoiding false alarms. A high Specificity value minimizes the number of False Positives, indicating that healthy patients are rarely mislabelled as sick.

Finally, the F1-score is the harmonic mean of Precision and Recall, and is especially beneficial when data are imbalanced. A high F1-score indicates that the model simultaneously misses few truly sick patients (high Recall) and does not over-diagnose (high Precision). A confusion matrix is a table that is used to evaluate the performance of a classification model by summarizing the predictions made by the model on a test dataset. It is a matrix with rows representing actual classes and columns representing predicted classes. Each cell in the matrix provides information about the number of instances that fall into a particular category based on the model's predictions.

The confusion matrix provides a detailed breakdown of the model's predictions, allowing one to evaluate its performance across different classes. By analysing the *TP*, *TN*, *FP* and *FN*, the model's strengths and weaknesses can be observed. In scenarios where classes are imbalanced, the confusion matrix

helps to assess how well the model is performing for each class. It allows one to identify if the model is biased towards the majority class and if specific classes are being misclassified more frequently.

4.5. Augmentation Results

This section's study aims to show how the model, when trained with data augmented using DCGANs, outperforms the model trained without such augmentation. Furthermore, it compares the effectiveness of DCGANs for data augmentation to traditional augmentation techniques and analyses the impact of both methods. It uses various metrics – including accuracy, precision, recall and F1-score – to evaluate the performance of the study.

For all tests, to verify the accuracy of the developed model, Leave-One-Patient-Out Cross-Validation (LOPOCV) is conducted. It involved training the model without data from one patient, which was then used for testing. This algorithm was repeated for each patient. Although this procedure is computationally intensive, it provides precise and unbiased insights into the model's performance. Through LOPOCV, the root mean squared error (RMSE) is calculated for n tests [87]:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2. \tag{4.33}$$

where n – indicates the number of tests, y_i is a true value, $\hat{y}_i$ denotes predicted value.

$$RMSE = \sqrt{MSE}. \tag{4.34}$$

DCGAN Augmentation

The proposed DCGAN models employ a batch size of 32 with 375 steps per epoch, utilize the Adam optimizer, apply the Wasserstein loss function and include a gradient penalty of 10. The Wasserstein distance is used to address instability during training. Training ends once the Discriminator's loss reaches 0, indicating its inability to differentiate between real and generated images. This criterion is achieved after approximately 180 epochs, as shown in Fig. 4.6 and Fig. 4.7, which

illustrates the training losses and density distribution of the DCGAN trained on a subset of retinitis pigmentosa peripheral changes subjects (referred to as *RP_PC*). Similar training trends are observed in other subsets and datasets.

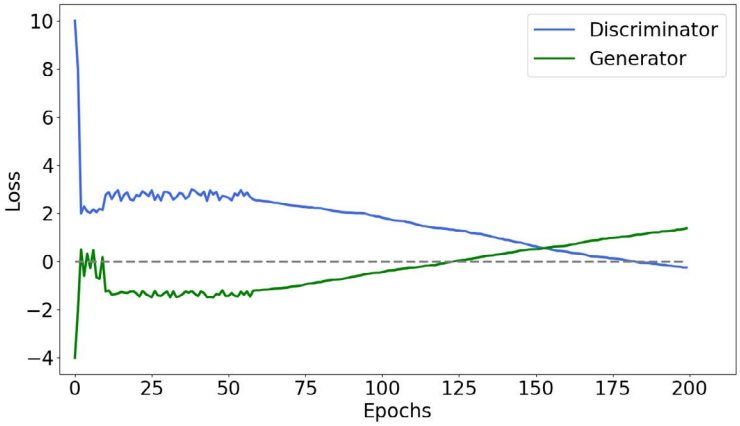


Fig. 4.6. Training loss curves of the DCGAN model on the RP_PC subset. The convergence of losses indicates stable adversarial training and suggests that the Discriminator gradually loses its ability to differentiate between real and synthetic images, reflecting improved realism of generated samples.

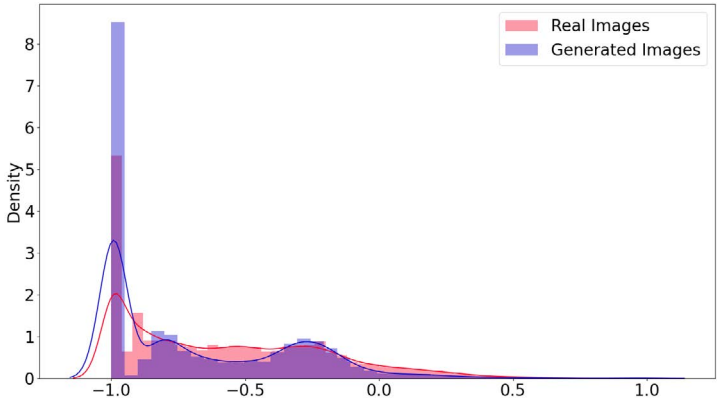


Fig. 4.7. Kernel density estimation of standardized pixel intensity values (dimensionless) for real and generated retinal images for the RP_PC subset. The distribution demonstrates the ability of the Generator to approximate the underlying data distribution of real retinal images, confirming that the synthetic samples capture key statistical and structural properties required for effective data augmentation.

The process begins with the generation of fake images using the DCGAN. Initially, the image resembles a blank canvas. By the 20th iteration, faint shadows start to emerge on the retinitis pigmentosa images. By the 50th iteration,

the eye region becomes more defined. Upon reaching the 100th iteration, the generated image displays a clear eye image. Subsequently, details gradually emerge throughout subsequent iterations. Ultimately, beyond iteration 180, the generated image exhibits a compelling outcome, as depicted in Fig. 4.8.

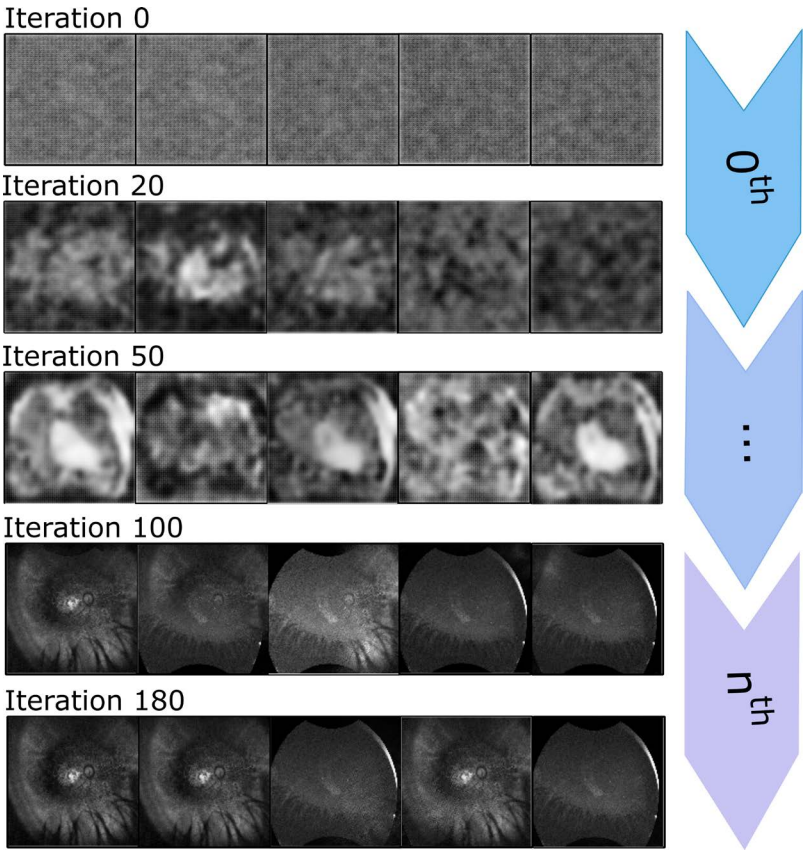


Fig. 4.8. Evolution of DCGAN-generated retinal images across training iterations. This figure illustrates the progressive improvement in the visual quality of retinal images generated by the DCGAN model at different stages of training, shown after 0, 20, 50, 100, and 180 training iterations. Early iterations produce highly noisy and unstructured patterns, while subsequent stages reveal increasingly coherent retinal anatomy and disease-related texture.

Once fully trained, DCGANs generate synthetic images for each class, amounting to 200% of the number of images relative to each subset of the dataset. These generated images are then merged with the images from the original dataset to create eight distinct composite datasets, each characterized by varying ratios of synthetic to authentic images (i.e., 25%, 50%, 75%, 100%, 125%, 150%, 175% and 200%).

Classification Results

The objective of this experiment is to establish a basic classification accuracy for future studies. Three dropout probabilities: 0%, 25% and 50% are examined, aligning with the probabilities detailed in the traditional augmentation section. The outcomes are illustrated in Fig. 4.9. The top-performing model, without dropout, achieves an accuracy of 72.20%, serving as the "baseline" for future iterations. Introducing dropout degrades the model's performance by approximately 4%.

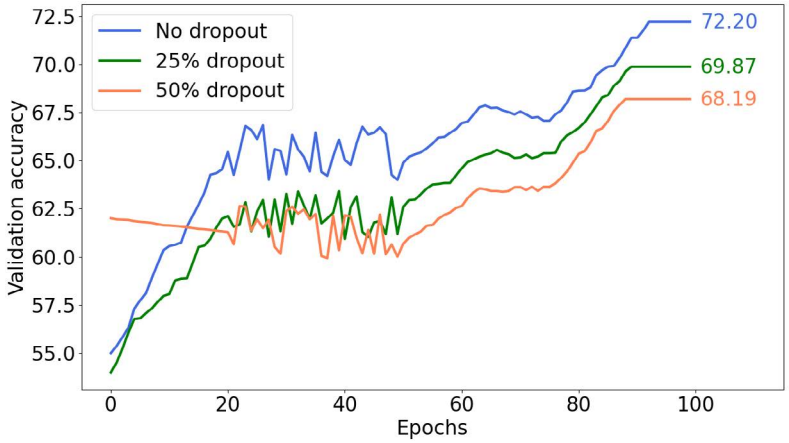


Fig. 4.9. VGG16+XGBoost trained without augmentation methods on RP_PC subset. The obtained results will be used as a reference point for further studies.

After setting the initial performance benchmark, the next phase involves examining the impact of integrating conventional augmentation methods on the efficacy of the models. These methods are described in the section related to classical augmentation techniques and Algorithm 4.1. Fig. 4.10 illustrates the comparative performance of two distinct models: one incorporating a 25% dropout rate and the other devoid of dropout. Both are analysed using the enhanced dataset. The non-dropout model demonstrates higher performance compared to the model with other parameter settings. However, its application may be limited due to the potential advantages of regularization in scenarios involving imperfect data.

Contrary to the expectation that augmentation often increases model capabilities, the current study observes that only when a small amount (25%) of the augmentation data is used and no dropout is applied, the baseline is

exceeded. This can be attributed to the rigorous nature of medical image formatting, where even minimal magnification interventions can have a detrimental effect on the classification process. The efficiency of the best model in this part of the study reaches 72.89%.

Moreover, the augmentation process worsens the model's ability to converge consistently, as reflected by the pronounced fluctuations in the performance curves depicted in Fig. 4.10.

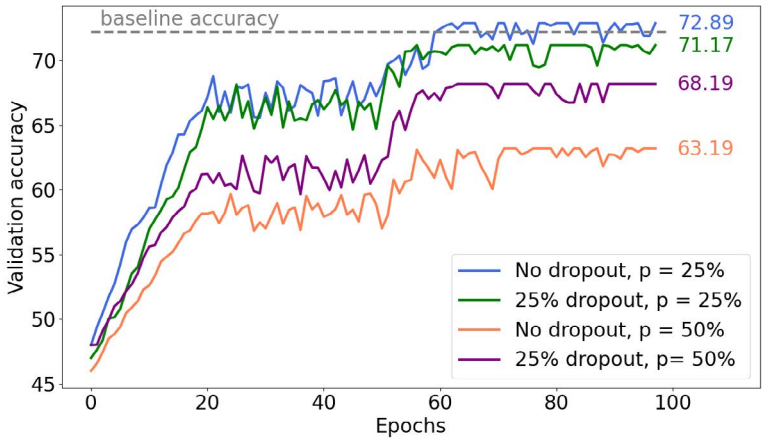


Fig. 4.10. VGG16 + XGBoost trained with traditional augmentation techniques on RP_PC subset. Two distinct network architectures were analysed across two varying values of *p*. The dash, grey line in the graph signifies the baseline (72.20%) established in the prior experiment. Although the validation accuracy without dropout temporarily exceeds the baseline accuracy, this observation should be interpreted as an indication of training dynamics rather than as a statistically significant improvement.

The primary objective of this study is to assess the performance of a VGG16+XGBoost classifier trained on a dataset enhanced with GAN-generated images. To identify the optimal quantity of artificial images to include in the original dataset, eight experiments are conducted, as detailed in the section about DCGAN augmentation. The results of these experiments are illustrated in Fig. 4.11. Validation accuracy was evaluated exclusively on real images, while GAN-generated samples were used only for training data augmentation.

Every run outperforms the baseline, demonstrating that DCGANs can serve as effective data augmentors, even in scenarios where traditional augmentation techniques fall short. The optimal ratio of 75% achieves a score almost 19% higher than the baseline.

Using DCGAN-generated images for data augmentation does not significantly impact model convergence, unlike traditional augmentation techniques. Despite the increased dataset size, nearly all models converge by epoch 80. In contrast, most traditional data augmentation runs (Fig. 4.10) fail to converge even after 100 epochs. The ideal ratio appears to be between 75% and 125%, with the highest score of 91.17% obtained by the model trained with a 75% fake/real ratio.

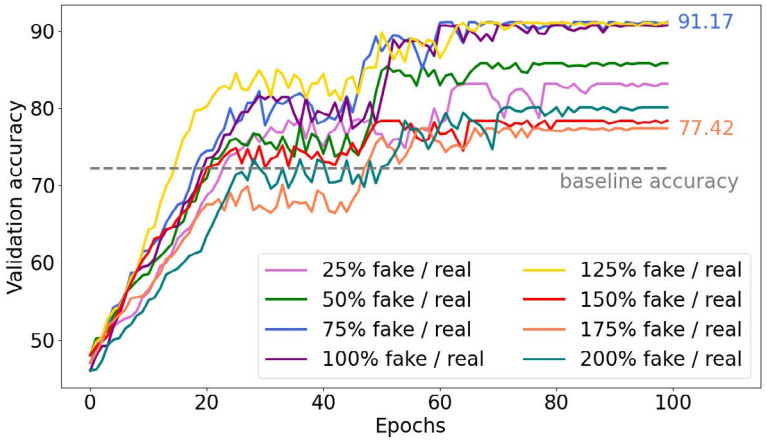


Fig. 4.11. VGG16 + XGBoost trained with DCGAN augmentation techniques on RP_PC subset. Eight distinct datasets were analysed with different levels of fake images and no dropout function. The dash, grey line in the graph signifies the baseline (72.20%).

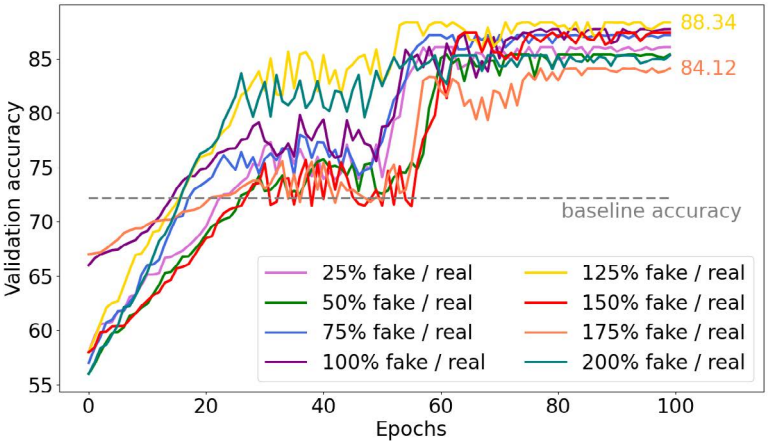


Fig. 4.12. VGG16 + XGBoost trained with DCGAN and traditional augmentation techniques on RP_PC subset. Eight distinct datasets were analysed with different levels of fake images and 25% dropout function and $p = 0.25$. The dash, grey line in the graph signifies the baseline (72.20%).

Finally, researchers evaluate the combination of traditional and DCGAN-based augmentation methods. The optimal model in this instance features a 25% dropout rate and a 125% fake-to-real ratio, achieving a score of 88.34%. Unlike previous experiments, dropout proves beneficial in this scenario. Fig. 4.12 illustrates the models trained with a 25% dropout rate across eight different fake-to-real ratios. As shown in Fig. 4.13, the models incorporating a 25% dropout rate perform marginally better than those without dropout.

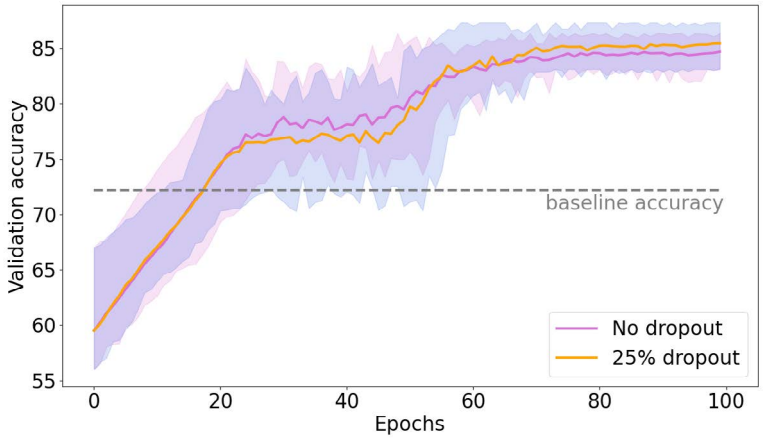


Fig. 4.13. VGG16 + XGBoost trained with DCGAN and traditional augmentation techniques on RP_PC subset. Two tests for no dropout and 25% dropout with $p = 0.25$ were conducted for 8 datasets represent different fake to real image ratios. The thick line represents the mean value, while the shaded area was made by the minimum and maximum values achieved in each epoch. The dash, grey line in the graph signifies the baseline (72.20%).

The use of Generative Adversarial Networks in medical data augmentation and classification addresses critical challenges posed by small and unbalanced datasets. By generating realistic and diverse synthetic data, GANs enhance the training process, improve model accuracy and contribute to more reliable and equitable healthcare outcomes. Their ability to balance datasets and preserve patient privacy further underscores their value in advancing medical imaging and diagnostic applications.

4.6. Chapter Summary

This chapter shows that Generative Adversarial Networks have emerged as powerful tools in medical imaging and data analysis, offering significant benefits in data augmentation and classification, especially when dealing with small and unbalanced datasets. The most important achievements presented in this chapter include: enhanced data augmentation for small datasets, balancing unbalanced datasets, preservation of data privacy, reduction of overfitting, improved model robustness, cross-modality data generation.

1. **Enhanced Data Augmentation for Small Datasets**

- Synthetic Data Generation: GANs generate realistic synthetic medical images that closely resemble real patient data. This augmentation increases the size of the training dataset without the need for additional data collection, which is often expensive or impractical in medical settings.
- Diversity and Variability: By introducing variations in the generated images, GANs help models learn a wider range of features, improving their ability to generalize new, unseen data.

2. **Balancing Unbalanced Datasets**

- Minority Class Amplification: In datasets where certain classes (e.g., rare diseases) are underrepresented, GANs produce additional synthetic examples of these classes, helping to balance the dataset.
- Improved Classification Performance: Balancing the dataset reduces bias towards majority classes and enhances the model's ability to correctly classify instances of minority classes, which is critical in medical diagnosis.

3. **Preservation of Data Privacy**

- Anonymization: GANs generate data that do not correspond to real patients, mitigating privacy concerns associated with sharing sensitive medical information.
- Regulatory Compliance: Using synthetic data help comply with data protection regulations.

4. **Reduction of Overfitting**

- Robust Model Training: With more data available through augmentation, models are less likely to overfit to the training data, leading to better performance on validation and test sets.

5. **Improved Model Robustness**

- Noise and Artifact Introduction: GANs simulate various imaging conditions, including noise and artifacts, enabling models to become more robust to real-world imperfections in medical images.

6. **Cross-Modality Data Generation**

- Multi-Modal Learning: GANs generate data across different imaging modalities, assisting in training models that can interpret multiple types of medical images.

5. Eye Diseases Classification Based on Artificial Intelligence Models

The classification of eye diseases has seen significant advancements in recent years, driven largely by the integration of cutting-edge imaging technologies and sophisticated machine learning algorithms. One of the pivotal imaging modalities that has revolutionized ophthalmology is OCT, which provides high-resolution, cross-sectional images of the retina, which are essential for diagnosing and monitoring various eye conditions, such as age-related macular degeneration, diabetic retinopathy and glaucoma.

OCT images allow clinicians to visualize the intricate layers of the retina, facilitating early detection and precise characterization of pathological changes. The complexity and richness of these images also make them ideal candidates for automated analysis using deep learning techniques, particularly CNNs, as well as their variants.

Convolution operations lie at the heart of CNNs, enabling the automatic extraction of features from images. Traditional convolutions have been remarkably successful in image processing tasks. However, recent innovations such as Kronecker convolutions have expanded the horizon of what is possible. Kronecker convolution, in particular, offers a unique approach by decomposing the convolution operation into a series of smaller, more manageable matrix multiplications. This not only reduces computational complexity but also enhances the network's ability to learn and generalize from the data.

5.1. Convolutional Neural Networks

Convolutional Neural Networks have revolutionized the field of medical image analysis, offering powerful tools for the automated classification of eye diseases. By leveraging hierarchical feature extraction, CNNs can detect subtle pathological changes in ocular images, aiding in early diagnosis and treatment [88]. This chapter describes the critical components of CNNs: convolution layers, channels, strides, padding, pooling layers, fully connected layers and activation functions, as well as, explores their specific impacts on the classification of eye diseases.

Convolution Layer

The convolution layer is the fundamental building block of CNNs, responsible for extracting local features from input images through learnable filters. In the context of eye disease classification, convolution layers play a pivotal role in detecting various pathological features such as microaneurysms, hemorrhages, drusen and neovascularizations. These features are indicative of conditions like retinitis pigmentosa, AVL and drusen [89], [90].

Mathematically, the convolution operation between an input image I, of size $H \times W$, and a kernel K, of size $M \times N$, is defined as Eq. (5.1):

$$O(i,j) = (I^{*}K)(i,j) = \sum_{m=1}^{M}\sum_{n=1}^{N} K(m,n) \cdot I(i+m-1, j+n-1), \qquad (5.1)$$

where O is the output feature map, and (i, j) are spatial coordinates in the output feature map.

In this setting, convolutional kernels act as feature extractors by computing linear combinations of local input regions. During training, the network minimizes the overall loss function by iteratively updating the kernel weights through backpropagation. For classifying eye diseases, different patterns of these kernels can be understood to represent such disease patterns. Namely, some of these kernels strive to detect the circular patterns typical of microaneurysms and the distorted shapes characteristic of vitelliform lesions.

The role of the depth and number of convolutional layers is significant to the ability of the network to capture elaborate characteristics, whereas shallower networks will capture simple edge and texture related features emerging in the image. However, more complex features, such as those of the disc edge

or the layers in the retina, can be attained in deeper networks [91]. Increasing network depth and incorporating additional information may lead to optimization challenges, including the vanishing gradient problem. Residual connections help alleviate this issue by enabling more stable and efficient gradient propagation.

Furthermore, convolutional layers promote parameter sharing and sparsity. This means that every neuron in a convolution layer is connected to a small region of the input called the receptive field – i.e., it is characterized by possibly non-zero output only over some range of the input. On the one hand, this local connectivity renders the network more computationally efficient and conserves the number of trainable parameters as opposed to simple densely connected layers. On the other hand, the parameter sharing strategy also implies that the same set of weights is utilized to filter all portions of the picture at different locations, allowing the network to locate features irrespective of where they occur within the image – a key consideration in medical images since the pathology can manifest anywhere within the retina.

In the scenario of eye diseases classification, topics where convolutional layers play an important role include:

- Detecting Lesions: Identifying small pathological features such as microaneurysms requires high-resolution feature extraction. Convolution layers with small kernels can capture fine details.
- Modelling Anatomical Structures: Larger kernels or successive convolution layers can model broader structures like the optic nerve head or macular region.
- Learning Hierarchical Features: Early layers capture low-level features (edges, textures), while deeper layers capture high-level features (lesions, anatomical abnormalities), improving diagnostic accuracy.

Channels

Channels in CNNs represent different dimensions or aspects of the input data. In ocular imaging, channels can correspond to colour components in RGB images or to different imaging modalities. The effective utilization of channels is crucial for capturing comprehensive information necessary for accurate eye disease classification [92].

For standard fundus photographs, the input image has three channels corresponding to the red, green and blue colour components Eq. (5.2):

$$I = \{I_{red}, I_{green}, I_{blue}\}. \tag{5.2}$$

Each channel is able to provides unique information:

- Red Channel: Sensitive to features like hemorrhages and blood vessels.
- Green Channel: Provides high contrast for microaneurysms and exudates.
- Blue Channel: Less commonly used due to higher noise but can highlight certain drusen lesions or vitelliform deposits.

In convolution layers, kernels operate across all input channels to produce output feature maps. If the input has C_{in} channels and the convolution uses C_{out} filters, the operation for each output channel k is defined as Eq. (5.3).

$$O_k(i,j) = \sum_{c=1}^{C_{in}} \sum_{m=1}^{M} \sum_{n=1}^{N} K_{k,c}(m,n) \cdot I_c(i+m-1, j+n-1), \tag{5.3}$$

where $K_{k,c}$ is the kernel for the k-th output channel and c-th input channel and I_c indicates the c-th input channel.

In the classification of eye diseases, the channels may have the following significance:

- Colour Information: Utilizing all colour channels enhances the network's ability to detect colour-dependent features, such as the reddish hue of hemorrhages or the yellowish appearance of drusen.
- Multi-Modal Data: Additional imaging modalities like Optical Coherence Tomography can be integrated as separate channels or inputs, providing depth information of retinal layers critical for diagnosing diseases [93].
- Channel Attention Mechanisms: Advanced models may incorporate attention mechanisms that weigh the importance of different channels, allowing the network to focus on the most relevant features [94].

Strides

Strides in CNNs define the step size with which the convolutional kernel moves across the input image. Strides affect the spatial resolution of the output feature maps and play a significant role in computational efficiency and feature detection granularity.

The output size considering strides is calculated as Eq. (5.4) [94]:

$$O_{size} = \left\lfloor \frac{I_{size} - K_{size} + 2P}{S} \right\rfloor + 1. \tag{5.4}$$

where I_{size} is the input size, K_{size} denotes the kernel size, P indicates padding, S denotes a stride.

In eye diseases classification using a stride of 1 ensures that the convolutional kernel examines every pixel of the input image. This is essential for detecting small lesions such as microaneurysms or early-stage drusen, which might be missed with larger strides. Increasing the stride reduces the spatial dimensions of the output feature map, leading to faster computations and lower memory usage. However, this may result in a loss of spatial detail, potentially overlooking critical pathological features. Selecting the appropriate stride involves balancing computational efficiency with the need for detailed feature extraction. For high-resolution retinal images where fine details are crucial, smaller strides are preferred despite the increased computational cost [95].

Some advanced architectures employ variable strides or dilated convolutions to expand the receptive field without losing resolution [96]. Dilated convolutions introduce a dilation rate d, modifying the convolution operation to Eq. (5.5):

$$O(i,j) = \sum_{m=1}^{M} \sum_{n=1}^{N} K(m,n) \cdot I(i + d \cdot (m-1), j + d \cdot (n-1)). \tag{5.5}$$

Padding

Padding involves adding extra pixels around the input image borders, typically with zeros (zero-padding). Padding is crucial for controlling the spatial dimensions of the output feature maps and ensuring that edge features are not neglected during convolution operations.

Without padding, the spatial dimensions of the output feature map decrease with each convolution. This reduction can be problematic in deep networks, leading to a significant loss of spatial information.

In the case of eye disease classification using appropriate padding, the output dimensions can be preserved by Eq. (5.6):

$$P = \frac{(K_{size} - 1)}{2}, \tag{5.6}$$

For odd-sized kernels (e.g., $K_{size} = 3$), padding of $P = 1$ ensures the output size remains the same as the input size when $S = 1$.

Important features like the optic disc, retinal blood vessels or peripheral lesions often lie near the image borders. Padding ensures that convolutional kernels can process these edge areas, capturing critical pathological features [97].

Three types of padding are typically used: zero padding, reflective padding and replication padding. The first one is the most common form, where zeros are added. It is simple but may introduce artificial edges. In reflecting one mirrors the edge values of the input image, reducing artifacts introduced by zero padding. Replication padding extends the edge pixels of the input image outward.

Pooling Layer

Pooling layers reduce the spatial dimensions of the feature maps, which decreases computational load, controls overfitting and provides translational invariance. Max pooling and average pooling are the most common types of pooling in CNNs. The first one retains the maximum value within a pooling window Eq. (5.7) [88]:

$$O(i,j) = \max_{(m,n)\in W} I(S \cdot i + m, S \cdot j + n), \tag{5.7}$$

where W is the set of indices in the pooling window and S is the stride. Average pooling computes the average of the values within a pooling window Eq. (5.8):

$$O(i,j) = \frac{1}{|W|} \sum_{(m,n)\in W} I(S \cdot i + m, S \cdot j + n). \tag{5.8}$$

Pooling layers contribute to dimensionality reduction by decreasing the spatial dimensions and also reduce the number of parameters and computational complexity. Moreover, they prevent overfitting by reducing the number of activations, which helps to prevent the network from learning spurious patterns specific to the training data. Another advantage is translational invariance. Pooling helps the network recognize features regardless of their exact position in the image, which is beneficial when lesions appear at varying locations.

Fully Connected Layer

Fully connected layers are traditional neural network layers where each neuron is connected to every neuron in the preceding layer. FC layers are typically used after convolutional and pooling layers to integrate the extracted features and perform classification.

The operation of an FC layer is given by Eq. (5.9):

$$y = \sigma(Wx + b), \tag{5.9}$$

where is the flattened input vector from the previous layer, W denotes the weight matrix, b, σ indicate the bias vector and the activation function, respectively, and is the output vector representing class scores or probabilities.

In eye disease classification, the fully connected layer combines high-level features learned by convolutional layers and provides the representational capacity needed to model complex feature interactions. Dropout is employed to mitigate overfitting and enhance generalization. Importantly, the FC layer can be replaced with almost any classifier such as Support Vector Machine, k-Nearest Neighbours or Bayesian one.

Activation Functions in CNNs

Activation functions introduce non-linearity into neural networks, enabling them to model complex relationships between inputs and outputs. The choice of activation function affects the network's convergence behaviour, learning dynamics and overall performance. In eye disease classification, activation functions influence how the network responds to variations in pixel intensities, contrast and textures: key factors in identifying pathological features. The choice of activation function can affect the following aspects:

- Convergence Speed: Some activation functions lead to faster training, which is beneficial when computational resources are limited.
- Model Capacity: Activation functions that allow for negative outputs can increase the network's ability to model complex patterns.
- Robustness: Appropriate activation functions can make the network more resistant to variations in imaging conditions, such as lighting and noise.

Commonly used activation functions in medical applications include: Rectified Linear Unit (*ReLU*), *Leaky ReLU*, Exponential Linear Unit (ELU), and Softmax Activation [88], [98], [99], [100].

One of the most commonly used functions is the Rectified Linear Unit, defined as Eq. (5.10) [98]:

$$ReLU(x) = \max(0, x). \tag{5.10}$$

The *ReLU* function is computationally efficient and simple to implement. It helps mitigate the vanishing gradient problem, allowing for the training of deeper networks. It also promotes sparse activations, which can improve the generalization of the model. In the context of eye disease classification, *ReLU* enables the network to focus on important features in ophthalmic images by suppressing negative activations. This accelerates convergence during training, which is beneficial when working with large medical datasets.

An alternative to *ReLU* is the *Leaky ReLU* function, which addresses the issue of "dead" neurons. It is defined as Eq. (5.11) [98]:

$$Leaky\ ReLU(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha x, & \text{otherwise,} \end{cases} \tag{5.11}$$

where α is a small positive constant (e.g., 0.01). *Leaky ReLU* allows a small gradient when the unit is not active, improving the learning process. In eye disease classification, it enhances the network's ability to learn from a wider range of features, including those represented by negative activations. This can lead to better performance in detecting subtle pathological features.

Another function is the Exponential Linear Unit, defined as Eq. (5.12) [98]:

$$\mathrm{ELU(x)} = \begin{cases} \mathrm{x}, \text{ if } \mathrm{x} > 0 \\ \alpha(\mathrm{e^x} - 1), \text{ otherwise,} \end{cases} \tag{5.12}$$

where α is a positive constant. ELU brings the mean activation closer to zero, which accelerates the learning speed. It also provides negative output values, which can help the network model complex patterns. In the case of eye disease classification, ELU may improve the model's ability to capture variations in intensity and contrast in retinal images, leading to faster convergence and increased accuracy.

The Softmax function is often used in the output layer for multi-class classification tasks and is defined by the formula Eq. (5.13) [98]:

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}, \tag{5.13}$$

where z_i is the input to the i-th neuron and N is the number of classes. This function converts the output scores into probabilities, facilitating the interpretation of the model's predictions. It is particularly important in tasks involving multiple eye diseases or severity grading.

The choice of an appropriate activation function has a significant impact on the neural network's ability to learn effectively and accurately classify medical images. By suppressing or promoting specific activations, these functions influence which features are utilized by the model, which is crucial in the precise diagnosis of eye diseases.

5.2. Attention Modules

Attention mechanisms have become pivotal in the development of deep learning models, especially in the field of computer vision. Integrating attention mechanisms with convolutional neural networks has significantly enhanced the models' ability to focus on the most relevant parts of an image. This focus leads to improved performance in tasks such as image classification, object detection or segmentation. Key advancements in attention mechanisms within CNNs include soft attention, hard attention, residual attention and modules like the Convolutional Block Attention Module (CBAM) [101].

Traditional CNNs, while effective in extracting hierarchical features, struggle to capture long-range dependencies due to their local receptive fields. Attention mechanisms alleviate this limitation by allowing the network to assign varying levels of importance to different spatial regions or channels in feature maps. Spatial attention determines where important features are located in an image. By generating spatial attention maps, models can emphasize significant regions and suppress less relevant ones. Channel attention, on the other hand, identifies what features are important across different channels. It assigns weights to each channel, enabling the model to focus on the most informative features.

Soft attention computes a weighted sum over all possible locations in the input, with weights that sum to one. These weights are typically

generated using a Softmax function, making the process differentiable and suitable for gradient-based optimization. Since the attention weights are continuous and differentiable, soft attention allows for end-to-end training using backpropagation. It considers all parts of the input simultaneously, assigning different levels of importance to each, thereby capturing global context. Although it requires computing attention weights over the entire input, optimizations and hardware advancements have made this feasible for large-scale models.

Hard attention involves selecting specific parts of the input to focus on, effectively disregarding the rest. It selects specific regions or features, often modelled as a sampling process. Due to the discrete nature of selection, hard attention introduces non-differentiable operations, making traditional backpropagation inapplicable. Training hard attention models often requires reinforcement learning techniques or approximations like the reinforce algorithm to handle the non-differentiable components [102].

Residual attention combines the power of residual learning with attention mechanisms to build very deep networks with enhanced performance [103]. In its architecture, attention mechanisms are integrated within residual blocks, creating an attention module that is stacked throughout the network. Each attention module consists of multiple branches performing mask operations to generate attention-aware features at different scales. This integration allows the network to focus on relevant features more effectively, improving feature representation. Moreover, the use of residual connections allows for training deeper networks without degradation problems, enhancing scalability.

CBAM is a simple yet effective attention module for feedforward convolutional neural networks, combining both spatial and channel attention mechanisms [101]. In its architecture, the channel attention module utilizes both max-pooling and average-pooling operations across spatial dimensions to generate channel-wise attention maps, helping the model focus on what is important. The spatial attention module applies pooling operations along the channel axis and uses a convolutional layer to generate spatial attention maps, highlighting where important features are located. CBAM applies channel attention first, followed by spatial attention, progressively refining the feature maps. The benefits of CBAM include a performance boost, as it enhances feature representations by focusing on important channels and spatial locations, leading to better performance in tasks like classification and detection. Additionally, it offers ease of integration, as it can be easily incorporated into any CNN architecture and trained end-to-end without requiring additional

supervision. Studies have shown that integrating CBAM with architectures such as ResNet, DenseNet and MobileNet improves accuracy [101].

Self-attention mechanisms allow a model to consider the relationships between all pairs of positions in the feature maps, capturing global dependencies. Non-local Neural Networks introduce self-attention mechanisms into CNNs, enabling the modelling of long-range dependencies beyond local receptive fields [103]. By relating distant spatial positions, the model gains a better understanding of the overall image structure, leading to improved performance in tasks requiring global context, such as video classification and image segmentation.

Although less common due to training challenges, hard attention mechanisms have been explored in convolutional networks. Recurrent Attention Models use hard attention to sequentially focus on different parts of an image, making discrete decisions about where to focus [102]. Training methods employ reinforcement learning to optimize the non-differentiable hard attention policies.

Soft attention is differentiable, captures global context and allows for end-to-end training, but is computationally intensive due to the need to process all parts of the input. Hard attention is computationally efficient during inference and can sharply focus on relevant regions but is difficult to train due to non-differentiability, requiring complex optimization techniques.

Attention mechanisms have various applications in computer vision. In image classification, they improve feature discrimination, leading to higher accuracy. In object detection and segmentation, by focusing on relevant regions and features, attention-enhanced models perform better in detecting and segmenting objects [94]. In fine-grained recognition, attention helps models capture subtle differences between similar classes by focusing on critical features.

Attention mechanisms, including soft attention, hard attention, residual attention and modules like CBAM, have significantly advanced the field of computer vision. They enable models to focus on the most relevant features and spatial regions, improving performance across various tasks.

5.2.1. Soft Attention Modules

Attention mechanisms have become integral in enhancing convolutional neural networks for medical image classification tasks. Medical images often contain complex patterns and subtle features that are critical for accurate diagnosis. Soft attention mechanisms enable models to focus on the most relevant regions of an image, improving classification performance by highlighting important features and suppressing irrelevant information [104].

In medical image classification, soft attention modules help identify regions of interest, such as tumours, lesions or anatomical structures, by assigning higher weights to these areas. This selective focus allows the network to learn more discriminative features, which is crucial given the high variability and noise present in medical imaging data.

Mathematically, soft attention mechanisms in CNNs involve computing attention weights over spatial locations or feature channels. Consider a feature map $F \in R^{C \times H \times W}$, where C is the number of channels, and H and W are the height and width of the feature map, respectively. The attention mechanism aims to generate an attention map $\mathbf{A}$ that highlights important regions or features.

Spatial attention identifies important spatial regions by assigning higher weights to locations that are more informative. This is particularly useful in medical images, where abnormalities may appear in specific regions. The attention map $M_s \in R^{H \times W}$ can be computed using average pooling Eq. (5.14) and max pooling Eq. (5.15) across the channel dimension [101].

$$f_{avg}(i,j) = \frac{1}{C}\sum_{c=1}^{C} F_{c,i,j}, \tag{5.14}$$

$$f_{max}(i,j) = \max_{c} F_{c,i,j}. \tag{5.15}$$

These pooled features are concatenated (channel-wise concatenation, resulting in a two-channel spatial tensor) and passed through a convolutional layer followed by a sigmoid activation to generate the attention map Eq. (5.16).

$$M_s = \sigma\left(f^{conv}\left([f_{avg}; f_{max}]\right)\right), \tag{5.16}$$

where denotes the sigmoid function, f^{conv} is the convolution operation, and [;] represents channel-wise concatenation. The feature map is then refined by Eq. (5.17).

$$\widetilde{F}_{c,i,j} = M_s(i,j) \cdot F_{c,i,j}. \tag{5.17}$$

Channel attention determines the importance of each feature channel by assigning weights based on the relevance of the extracted information. In medical images, certain features may be more indicative of a particular

pathology. The channel attention map $M_c \in R^C$ can be computed using global average pooling Eq. (5.18) and global max pooling Eq. (5.19) across spatial dimensions [101]:

$$z_{avg}(c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} F_{c,i,j}, \tag{5.18}$$

$$z_{max}(c) = \max_{i,j} F_{c,i,j}. \tag{5.19}$$

These descriptors are passed through a shared network (typically consisting of fully connected layers) and combined using the sigmoid function Eq. (5.20)

$$M_c = \sigma(\text{MLP}([z_{avg}, z_{max}])), \tag{5.20}$$

where MLP denotes a multilayer perceptron. The refined feature map is obtained by Eq. (5.21).

$$\widetilde{F}_{c,i,j} = M_c(\text{c}) \cdot F_{c,i,j}. \tag{5.21}$$

Combining both spatial and channel attention mechanisms can further enhance the model's ability to focus on relevant features. In medical image classification, this approach allows the network to simultaneously consider which features are important and where they are located. Sequential application to refine the feature map for channel and spatial attention is defined by Eq. (5.22) and Eq. (5.23), respectively [104].

$$\widehat{F}_{c,i,j} = M_c(\text{c}) \cdot F_{c,i,j}, \tag{5.22}$$

$$\widetilde{F}_{c,i,j} = M_c(\text{c}) \cdot \widehat{F}_{c,i,j}. \tag{5.23}$$

This method is utilized in the CBAM, which sequentially applies channel and spatial attention to refine feature maps adaptively. Integrating CBAM into CNN architectures for medical image classification has been shown to improve performance by focusing on the most informative features and regions [105].

Moreover, U-Net architecture can be extended by incorporating attention gates, resulting in the Attention U-Net. Although originally designed for segmentation tasks, the attention mechanisms within the network enhance

feature extraction, which can be beneficial for classification tasks as well [106]. The attention coefficients are computed as Eq. (5.24) [107]:

$$\alpha_i = \sigma\left(W_\alpha^\intercal \left[F_i^l; G^g\right] + b_\alpha\right), \tag{5.24}$$

where F_i^l is the feature vector from the encoder at location *i,* G^g is the gating signal from the decoder, W_α and b_α are learnable parameters, σ denotes the sigmoid function, [;] represents concatenation. The attended feature is then expressed in a form of Eq. (5.25) [107].

$$\widetilde{F_l} = \alpha_i \cdot F_i^l. \tag{5.25}$$

This mechanism allows the network to focus on relevant regions, improving the accuracy of segmentation and potentially benefiting classification tasks when adapted appropriately [107].

In classification tasks, the final prediction is often based on a global feature vector. Using soft attention mechanisms, the global feature vector $v \in R$ can be computed as Eq. (5.26) [107]:

$$v = \sum_{i=1}^{H}\sum_{j=1}^{W} \alpha_{i,j} \cdot F_{:,i,j}, \tag{5.26}$$

where $\alpha_{i,j}$ represents the attention weight at spatial location *(i, j),* $F_{:,i,j} \in R^c$ is the feature vector at that location *(i, j)* across all channels, *i, j* are spatial location indices and *H, W* indicate height and width of the feature map. The attention weights $\alpha_{i,j}$ are computed using a Softmax function over the attention scores $e_{i,j}$ Eq. (5.27) [107]:

$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{k=1}^{H}\sum_{l=1}^{W}\exp\left(e_{k,l}\right)}, \tag{5.27}$$

where *k* and *l* are indices for summation over all spatial locations.

5.2.2. Hard Attention Modules

Hard attention mechanisms in convolutional neural networks play a crucial role in the classification of rare eye diseases such as retinitis pigmentosa and dystrophia maculae vitelliformis. These mechanisms enable models to focus on the most informative regions of retinal images, enhancing the detection of subtle and varied pathological features characteristic of these conditions. Unlike soft attention, which assigns continuous weights across the entire input and remains differentiable, hard attention involves making discrete, non-differentiable decisions about where to concentrate computational resources. This discrete selection presents unique challenges for training, as traditional backpropagation cannot be directly applied. To overcome this, specialized techniques like reinforcement learning and differentiable approximations are utilized [108], [109].

In the context of CNNs for rare eye disease classification, hard attention allows the model to selectively process specific patches or regions of a retinal image that are most indicative of disease presence. For instance, in diagnosing RP and macular vitelliform dystrophy (DMV), the model may focus on areas showing characteristic signs such as pigmentary changes or vitelliform deposits, respectively. This selective focus not only reduces computational overhead by limiting the number of regions processed but also enhances the model's ability to detect nuanced features that might be overlooked when analysing the entire image uniformly.

Mathematically, hard attention can be formalized using a stochastic policy that governs the selection of attention locations based on probability distributions. Consider an input retinal image and a CNN model that sequentially selects attention locations where l_t represents the chosen location at time step t. The objective of the model is to maximize the expected reward $E_p(l_{1:T}|I)[R]$ where R is a reward function associated with task performance such as classification accuracy [110].

The policy $p(l_t|s_{t-1}; \theta)$ where θ is the set of learnable parameters, and which depends on the current state s_{t-1}, which encapsulates information gathered up to the previous time step. This state is typically modelled using a recurrent neural network (RNN) such as a LSTM network or GRU [30], [52], which maintains a memory of past attention decisions and observations.

Training the model involves optimizing the expected reward $J(\theta) = E_p(l_{1:T}|I; \theta)[R]$ with respect to the parameters θ. The gradient of this expected reward is estimated using reinforcement learning algorithms [110],

which provides a way to compute gradients despite the non-differentiable nature of hard attention decisions. The gradient is expressed as Eq. (5.28):

$$\nabla_\theta J(\theta) = E_{p(l_1|I;\theta)}\left[R\sum_{t=1}^{T}\nabla_\theta \log p(l_t|s_{t-1};\theta)\right], \tag{5.28}$$

where $J(\theta)$ is the expected reward as a function of parameters θ, $\nabla_\theta J(\theta)$ indicates the gradient of the expected reward with respect to θ, R is the reward signal, $p(l_t|s_{t-1};\theta)$ denotes the policy probability of selecting location l_t given state s at time step $t-1$.

Within CNNs, hard attention models such as the Recurrent Attention Model (RAM) process input images by iteratively selecting and analysing specific regions or glimpses. At each time step t, the model extracts an observation x_t from the image at the selected location l_t: $x_t = f_{extract}(I,l_t)$, where $f_{extract}$ is a function that extracts a localized glimpse centred at l_t. The internal state st is then updated using an RNN: $s_t = f_{RNN}(s_{t-1},x_t;\theta_s)$. The subsequent attention location l_{t+1} is sampled from the policy distribution: $l_{t+1} \sim p(l_{t+1}|s_t;\theta_s)$ $s_t = f_{RNN}(s_{t-1},x_t;\theta_s)$. After a predefined number of time steps T, the model generates a final classification prediction y based on the accumulated state: $y = \mathrm{f}_{\mathrm{output}}(\mathrm{s}_\mathrm{T};\theta_y)$. Here, foutput is the function that maps the final state s_T to the output prediction y, and $\theta = \{\theta_s, \theta_l, \theta_y\}$ represents the set of all model parameters [111].

To facilitate differentiable training despite the discrete nature of hard attention, differentiable approximations such as the Gumbel-Softmax trick are employed [108], [112]. The Gumbel-Softmax distribution provides a continuous relaxation of categorical distributions, allowing gradients to flow through the sampling process. Given unnormalized log probabilities Eq. (5.29)

$$\tilde{z}_i = \frac{\exp((\log \pi_i + g_i)/\tau)}{\sum_{j=1}^{K}\exp\Big(\Big(\log \pi_j + g_j\Big)/\tau\Big)}x, \tag{5.29}$$

where g_i are independent samples from the *Gumbel(0,1)* distribution, $\tau > 0$ is the parameter controlling the smoothness of the approximation and K is the number of categories. As τ approaches zero, the distribution becomes closer to a categorical distribution, effectively mimicking hard attention while retaining differentiability.

5.2.3. Convolution Block Attention Module

The Convolutional Block Attention Module is an attention mechanism designed to enhance the representational power of CNNs by focusing on meaningful features along both the channel and spatial dimensions [101]. CBAM is lightweight and can be seamlessly integrated into existing CNN architectures, making it particularly useful in medical imaging.

Given an intermediate feature map $F \in R^{C\times H\times W}$ where C denotes the number of channels, and H and W represent the spatial dimensions, CBAM sequentially infers a 1D channel attention map $M_c \in R^{C\times 1\times 1}$ and 2D spatial attention map $M_s \in R^{1\times H\times W}$. The refined feature map F' is obtained by Eq. (5.30) [101]:

$$F' = M_s \odot M_c \odot F, \tag{5.30}$$

where denotes element-wise multiplication.

The channel attention mechanism emphasizes features which are important by focusing on their inter-channel relationships. To achieve this, global average pooling and global max pooling are applied to the input feature map F along the spatial dimensions, generating two context descriptors f_{avg} Eq. (5.18) and f_{max} Eq. (5.19), both in R^C [101]:

These descriptors are then passed through a shared multilayer perceptron to generate the channel attention map. The MLP consists of two fully connected layers with a reduction ratio r to reduce computational overhead. It is defined as Eq. (5.31) [113]:

$$\mathrm{MLP} = W_1\delta(W_0 x), \tag{5.31}$$

where $W_0 \in R^{\frac{C}{r}\times C}, W_1 \in R^{C\times\frac{C}{r}}$ are weight matrices of the first and second FC layer, respectively, r is the dimensionality reduction factor, and δ represents the *ReLU* activation function. The channel attention map M_c is computed by applying the sigmoid activation function (σ) to the sum of the outputs from the MLP applied to both descriptors Eq. (5.32) [113]:

$$M_{\mathrm{c}} = \sigma((\mathrm{MLP})(f_{\mathrm{avg}})) + \mathrm{MLP}(f_{\mathrm{max}}). \tag{5.32}$$

This attention map is then applied to the input feature map F via element-wise multiplication Eq. (5.33) [113]:

$$F_c = M_c \odot F, \tag{5.33}$$

where M_c is broadcasted along the spatial dimensions.

The spatial attention mechanism focuses on the location of important features by emphasizing informative spatial locations. After applying the channel attention, the feature map M_c is used as input for the spatial attention module. Similarly, to the channel attention, average pooling and max pooling are performed along the channel axis, resulting in two 2D spatial context descriptors $f^s_{(avg)}$ Eq. (5.34), $f^s_{(max)}$ Eq. (5.34), both in $R^{H\times W}$ [114]:

$$f^s_{\text{avg}} = \text{AvgPool}_{\text{chan}}(F_c) = \frac{1}{C}\sum_{c=1}^{C} F_{c,i,j}, \tag{5.34}$$

$$f^s_{\text{max}} = \text{MaxPool}_{\text{chan}}(F_c) = \max_{i,j} F_{c,i,j}. \tag{5.35}$$

These two descriptors are concatenated along the channel dimension and passed through a convolutional layer with a kernel size of *7×7* to produce the spatial attention map Eq. (5.36) [114]:

$$M_s = \sigma\left(f^{7\times7}_{\text{conv}}\left(\left[f^s_{\text{avg}}; f^s_{\text{max}}\right]\right)\right), \tag{5.36}$$

where [. ; .] denotes channel-wise concatenation, and $f^{7\times7}_{conv}$ is the convolution operation with kernel size *7×7*. The sigmoid activation function σ ensures that the attention map values are between 0 and 1. The use of a *7×7* convolutional kernel in the spatial attention module follows the original design of the CBAM [101]. This kernel size provides a sufficiently large receptive field to capture broader spatial context and long-range dependencies across the feature map, which is essential for effective spatial attention modelling. Smaller kernels (e.g., *3×3*) have been shown to be less effective in aggregating global spatial information, whereas larger kernels do not yield additional performance gains and introduce unnecessary computational overhead [101].

Finally, the spatial attention map M_s is applied to the feature map F_c to obtain the final refined feature map *F'* Eq. (5.37) [114]:

$$F' = M_s \odot F_c. \tag{5.37}$$

where M_s is broadcasted along the channel dimension.

In medical imaging tasks, such as classifying rare eye diseases like retinitis pigmentosa and Acquired Vitelliform Lesion, the CBAM enhances the CNN's ability to focus on subtle pathological features. For instance, in retinitis pigmentosa, the degeneration of photoreceptor cells leads to pigmentation changes that are often diffuse and difficult to detect. The channel attention mechanism helps to emphasize features related to these pigmentation changes by modelling inter-channel dependencies. Similarly, the spatial attention mechanism highlights the specific regions where these changes occur, enabling the network to concentrate on relevant spatial locations. In the case of Acquired Vitelliform Lesion, which presents as yellowish subretinal deposits, the CBAM aids in detecting these lesions by assigning higher attention weights to spatial regions where the lesions are present and emphasizing the channels corresponding to features of the lesions. By refining the feature maps through sequential application of channel and spatial attention, CBAM-enhanced CNNs can improve their classification performance on these rare eye diseases, even when dealing with limited data and subtle features.

5.2.4. Residual Attention Modules

Residual attention modules in convolutional neural networks have emerged as a powerful approach for enhancing image classification tasks. Residual attention networks integrate attention mechanisms with residual learning to improve feature representation and model performance [115]. The attention mechanism enables the network to focus on the most informative regions of the input image, while residual connections facilitate the training of deeper networks by mitigating issues like vanishing gradients.

In a residual attention module, the attention mechanism is embedded within the residual block of a CNN. Given an input feature map $F \in R^{C\times H\times W}$ where C denotes the number of channels, and H and W represent the spatial dimensions, the module computes an attention map $M \in R^{C\times H\times W}$ to refine the feature map. The attention map M is typically generated using a combination of spatial and channel attention mechanisms. The spatial attention focuses on where the important features are located, while the channel attention emphasizes which features are significant [116].

The spatial attention map M_s is computed by applying average pooling and max pooling operations along the channel dimension, followed by a convolution layer and a sigmoid activation function: $M_s = \sigma(f_{conv}([AvgPool(F); MaxPool(F)]))$, and channel attention map M_c are calculated by performing average pooling and

max pooling along the spatial dimensions, followed by shared fully connected layers: $M_c = \sigma(f_{fc}([AvgPool(F^{spatial}), MaxPool(F^{spatial})]))$. The combined attention map M is obtained by element-wise multiplication of the spatial and channel attention maps: $M = M_s \odot M_c$, where $\odot$ denotes element-wise multiplication [116].

The refined feature map $\widetilde{F}$ is produced by applying the combined attention map to the input feature map: $\widetilde{F} = M \odot F$ Finally, the output of the residual attention module is computed using a residual connection: $F_{output} = F + \widetilde{F}$ [116].

This connection allows the network to learn both the original features and the ones highlighted by the attention mechanism, enhancing the overall representation.

Incorporation of residual attention modules in convolutional neural networks offers a promising avenue for improving the classification of rare eye diseases such as retinitis pigmentosa and acquired vitelliform lesion. By enabling the model to focus on critical regions within retinal images and enhancing feature representations, these modules contribute to higher accuracy and provide valuable support in decision-making.

5.3. Gated Recurrent Unit

Gated Recurrent Unit is a recurrent neural network architecture designed to address the challenges of modelling long-term dependencies in sequential data, particularly the vanishing gradient problem inherent in traditional RNNs. Although introduced over eight years ago, GRU remains a fundamental component in modern neural network applications due to its computational efficiency and effectiveness in capturing temporal patterns.

At each time step t, GRU updates its hidden state by incorporating the current input and the previous hidden state h_{t-1}. This update is regulated by two gating mechanisms: the update gate and the reset gate r_t.

The update gate z_t determines the extent to which the previous hidden state is preserved versus updated with new information. It is computed as Eq. (5.38) [45]:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z), \tag{5.38}$$

where σ denotes the sigmoid activation function, W_z and U_z are weight matrices for the input and hidden state, respectively, and b_z is the bias vector.

The reset gate r_t decides how much of the previous hidden state to forget Eq. (5.39):

$$r_t = \sigma(W_r x_r + U_r h_{t-1} + b_r). \tag{5.39}$$

Using the reset gate, the candidate hidden state is computed as Eq. (5.40):

$$\widetilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h), \tag{5.40}$$

where tanh is the hyperbolic tangent function and $\odot$ denotes element-wise multiplication.

The final hidden state h_t is then updated by interpolating between the previous hidden state and the candidate hidden state, weighted by the update gate Eq. (5.41):

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h}_t. \tag{5.41}$$

These mechanisms enable GRU to adaptively capture dependencies over different time, frame and convolution scales without the computational overhead of more complex architectures. This advancement showcases the adaptability of GRUs in integrating with other neural network components to enhance performance [30], [45].

5.4. Chapter Summary

Convolutional Neural Networks and Gated Recurrent Units play a pivotal role in the classification of rare eye diseases such as retinitis pigmentosa. CNNs are highly effective in processing medical images due to their ability to learn hierarchical feature representations from visual data. In the context of retinitis pigmentosa, a genetic disorder that leads to progressive retinal degeneration, CNNs can automatically extract and identify characteristic features from retinal images, such as bone spicule pigmentation, attenuation of retinal vessels and optic disc pallor. These features are essential for accurate diagnosis and differentiation from other retinal conditions.

Combining CNNs and GRUs allows for the integration of spatial and temporal information. While CNNs focus on extracting spatial features from

retinal images, GRUs process the sequence of these features over time. This synergy enhances the model's ability to classify rare eye diseases accurately, facilitating early detection and intervention.

Moreover, appropriate use of soft attention mechanisms in the context of detecting rare eye diseases has allowed for the following benefits:

1. **Improved Focus on Pathological Regions**: Soft attention allows models to concentrate on areas more likely to contain abnormalities, enhancing diagnostic accuracy.
2. **Reduced Influence of Irrelevant Regions**: By down-weighting less informative areas, the model becomes more resistant to noise and irrelevant background information common in medical images.
3. **Enhanced Interpretability**: Attention maps can provide visual explanations of where the model is focusing, aiding clinicians in understanding the decision-making process.

Implementing hard attention mechanisms in CNNs offers several advantages for the classification of rare eye diseases:

1. **Focused Analysis**: By directing attention to specific regions of retinal images, models can more effectively identify subtle pathological changes associated with diseases like RP and DMV, which might be missed when processing the entire image uniformly.
2. **Computational Efficiency**: Limiting the number of regions analysed reduces the computational burden, enabling the processing of high-resolution images without excessive resource consumption.
3. **Enhanced Interpretability**: Hard attention decisions can provide insights into which regions of the image the model deems most relevant for diagnosis, aiding clinicians in understanding and trusting the model's predictions.
4. **Improved Performance on Limited Data**: Rare diseases often suffer from limited and imbalanced datasets. Hard attention mechanisms, combined with techniques like data augmentation and transfer learning, can enhance model performance even with scarce data.

In medical imaging, particularly in classifying rare eye diseases such as retinitis pigmentosa and acquired vitelliform lesion, soft attention modules into convolutional neural networks significantly enhance their ability to focus on subtle pathological features. Retinitis pigmentosa, for example, involves the degeneration of photoreceptor cells, leading to diffuse pigmentation changes that are challenging to detect. The channel attention mechanism emphasizes features related to these pigmentation alterations by modelling dependencies

between channels. Meanwhile, the spatial attention mechanism highlights the specific regions where these changes occur, allowing the network to concentrate on relevant spatial locations.

For acquired vitelliform lesion, characterized by yellowish subretinal deposits, CBAM aids in detection by assigning greater attention weights to the spatial regions containing the lesions and emphasizing the channels corresponding to lesion-specific features. By refining feature maps through the sequential application of channel, spatial and residual attention enhanced convolutional neural networks improve their classification performance on these rare eye diseases, even when dealing with limited data and subtle manifestations.

6. Typical Medical Data Classification Algorithms

Rare eye diseases, such as retinitis pigmentosa and AVL, present significant challenges in ophthalmology due to their low prevalence and complex manifestations. Early and accurate diagnosis is crucial for managing these conditions and preventing irreversible vision loss. Advances in imaging technologies like OCT have enabled detailed visualization of retinal structures, providing valuable data for analysis. CNNs, a subset of deep learning models specialized in processing visual information, have become instrumental in interpreting these complex images.

This chapter delves into the most common CNN architectures utilized for classifying rare eye diseases using OCT images. By automatically learning hierarchical features from raw imaging data, CNNs can detect subtle pathological changes that may be overlooked by traditional analysis methods. Selected models will be explored, such as InceptionV3, Residual Networks (ResNet), U-Net and Dense Convolutional Networks (DenseNet). Their architectures, applications and impact on the diagnosis of rare retinal disorders will be discussed. The aim of this study is to highlight how these deep learning models contribute to improved diagnostic accuracy and support clinicians in delivering timely interventions for patients affected by these challenging conditions.

6.1. Residual Blocks

Residual learning constitutes a commonly adopted architectural mechanism in deep neural networks, particularly in convolutional architectures. In the following, the structure and operating principle of residual blocks are briefly described [117].

Let us take into consideration a small section of a neural network, as depicted in Fig. 6.1, where x represents the input. The objective is to learn a function $f(x)$ that captures the desired transformation of the input data. In traditional networks, this function is learned directly. However, in residual networks, we instead focus on learning the residual function $g(x) = f(x) - x$. Learning the residual function simplifies the training process, especially when $f(x)$ is close to the identity function x. In such cases, the residual $g(x)$ becomes zero or near zero, making it easier for the network to learn. This is achieved by adjusting the weights and biases in the upper layers (such as fully connected or convolutional layers) so that they contribute minimally to the output [118].

The residual block is illustrated on the right side of the figure, where a shortcut connection bypasses certain layers by directly adding the input x to the output of those layers. This connection is often referred to as a residual or shortcut connection. By allowing the input to bypass intermediate layers, residual blocks enable faster forward propagation and help mitigate issues like vanishing gradients during training [119].

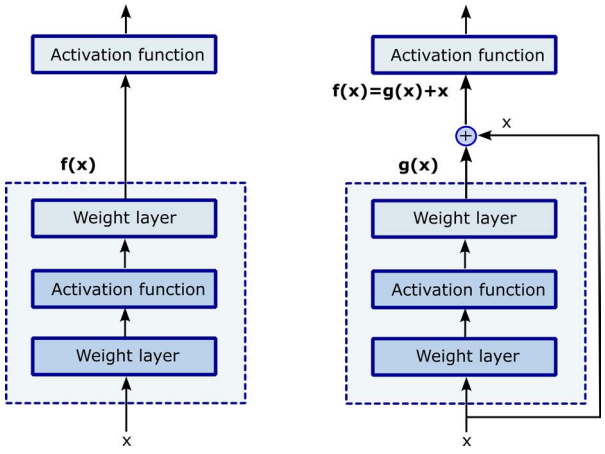


Fig. 6.1. Structure of a typical and residual block used in convolutional neural networks. In a regular block (left), the portion within the dotted-line box must directly learn the mapping $f(x)$. In a residual block (right), the portion within the dotted-line box needs to learn the residual mapping $g(x) = f(x)-x$, making the identity mapping $f(x) = x$ easier to learn.

Furthermore, a residual block can be seen as a simplified version of multi-branch architectures like the Inception block. It consists of two parallel paths: one that processes the input through a series of layers and another that passes the input directly into the output without any modifications.

A residual block typically consists of two convolutional layers with *3×3* filters, both producing outputs with the same number of channels. Each convolutional layer is followed by batch normalization and a *ReLU* activation function. Instead of processing the input solely through these convolutional layers, the design includes a shortcut that bypasses them, adding the original input directly before the final *ReLU* activation. This approach requires that the output from the convolutional layers matches the shape of the input so they can be combined through addition. If there's a need to change the number of channels, an additional *1×1* convolutional layer is used to adjust the input dimensions appropriately for the addition operation (see Fig. 6.2) [120].

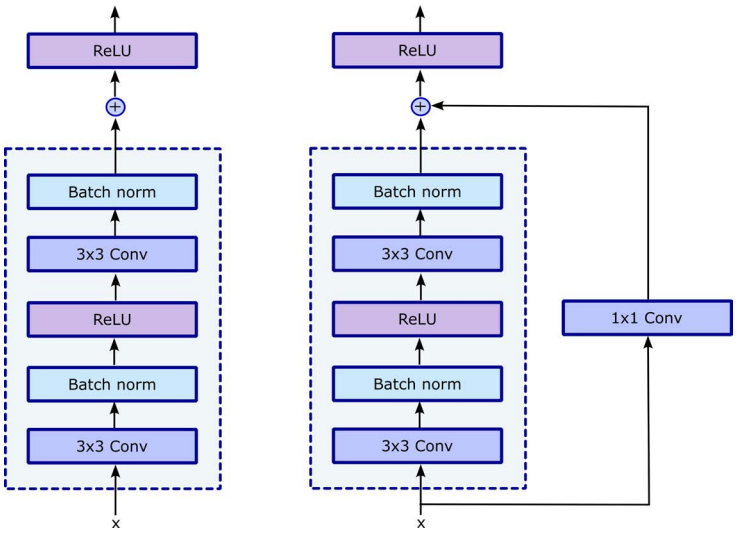


Fig. 6.2. Residual block with and without convolution, which transforms the input into the desired shape for the addition operation.

6.2. Residual Network Model

The backbone of ResNet was introduced by He et al. and was designed to address the vanishing gradient problem that impedes the training of deep neural networks. This phenomenon occurs when gradients used to update the network's weights diminish exponentially through layers, making it challenging

to train networks beyond a certain depth. ResNets circumvent this issue by incorporating residual learning through shortcut connections, allowing the network to learn identity mappings that facilitate the flow of gradients during backpropagation [91]. The ResNet-18 model, one variant of this architecture, consists of 18 layers, including convolutional layers and residual blocks. In the context of retinal disease diagnosis, ResNet-18 can effectively learn intricate features from high-resolution retinal images, aiding in the detection of subtle pathological changes associated with retinitis pigmentosa or AVL.

The architecture of ResNet-18 begins with a *7×7* convolutional layer with 64 output channels and a stride of 2, capturing low-level features from the input images. This layer is followed by a *3×3* max-pooling layer with a stride of 2, reducing the spatial dimensions and emphasizing the most salient features. Each convolutional layer within the network is typically followed by a Batch Normalization layer [121] and a *ReLU* activation function, which together enhance the network's training efficiency and nonlinear representation capabilities [122].

ResNet-18 comprises four modules (Fig. 6.3), each containing residual blocks with the same number of output channels. In the first module, the number of channels remains consistent with the input channels, and the spatial dimensions are preserved due to the preceding max-pooling layer. For each subsequent module, the first residual block doubles the number of channels while halving the height and width of the feature maps. This progression is achieved by using a convolutional layer with a stride of 2 within the residual block. The transformation can be formalized as Eq. (6.1) [123]:

$$C_{\text{out}} = 2C_{\text{in}}, \quad H_{\text{out}} = \frac{H_{\text{in}}}{2}, \quad W_{\text{out}} = \frac{W_{\text{in}}}{2}, \tag{6.1}$$

where C_{in} and C_{out} are the input and output channels, respectively, and H_{in}, W_{in} and H_{out}, W_{out} are the spatial dimensions before and after the transformation, respectively.

Following the residual modules, a global average pooling (GAP) layer is applied to aggregate spatial information and reduce the dimensionality of the feature representations while preserving their semantic content. By replacing fully connected layers operating on spatial feature maps, GAP reduces the number of learnable parameters and mitigates overfitting [61]. This layer computes the average of each feature map, summarizing the spatial information and reducing each feature map to a single scalable value. The operation is defined as Eq. (6.2) [91], [123]:

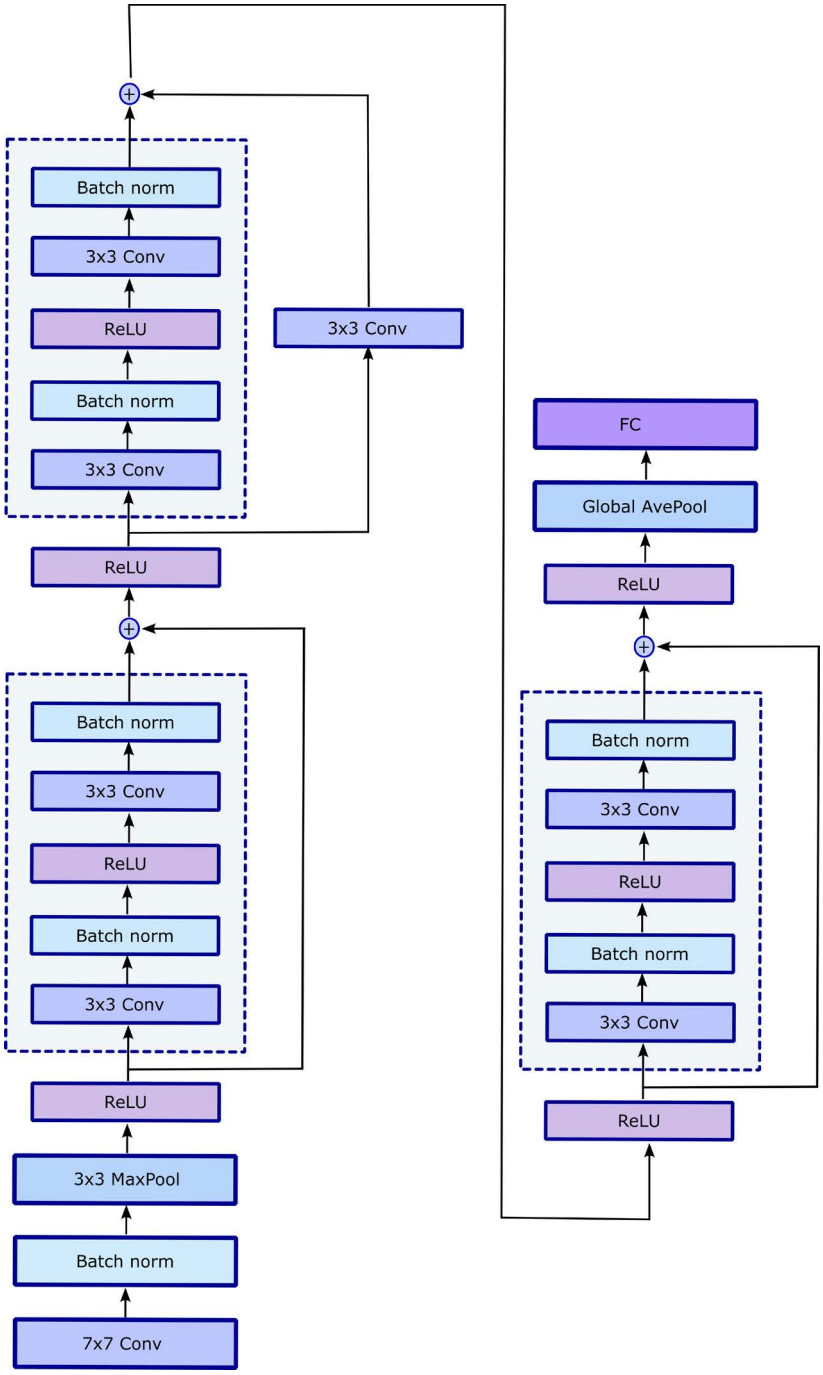


Fig. 6.3. The architecture of RestNet-18. Initial layers extract low-level retinal textures, while deeper layers focus on high-level disease-related patterns. The skip connections preserve diagnostically relevant information across depths, enabling accurate classification.

$$\widehat{y_k} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} y_{k,i,j}, \tag{6.2}$$

where $\widehat{y_k}$ is the pooled output for the k-th feature map and $y_{k,i,j}$ represents the activation at spatial position *(i, j)*. This pooling layer mitigates the risk of overfitting and reduces the computational complexity of the subsequent fully connected layer, which produces the final classification output. This architecture strikes a balance between depth and computational efficiency, making it suitable for medical imaging tasks where large datasets may be limited.

Recent studies have underscored the effectiveness of ResNet architectures in ophthalmology. Li et al. developed a deep learning system based on ResNet to detect glaucomatous optic neuropathy from fundus photographs, achieving high sensitivity and specificity [124]. Similarly, Milea et al. utilized deep CNNs for the automated detection of papilledema and other optic disc abnormalities [125]. These applications highlight the potential of ResNet-18 in facilitating early diagnosis and monitoring of retinal diseases, including retinitis pigmentosa and AVL.

Understanding the input shapes and feature representations within ResNet-18 is crucial for optimizing its application to retinal imaging. As the network processes the input images, the resolution decreases due to pooling operations and strided convolutions, while the feature dimensionality increases. This progression aligns with the clinical need first to capture fine-grained local features, such as the characteristic pigmentation patterns in retinitis pigmentosa or the subretinal deposits in AVL, and second to integrate them into higher-level abstractions pertinent to disease classification.

ResNet-18 offers a robust and efficient framework for the analysis of retinal images in the diagnosis of retinitis pigmentosa and acquired vitelliform lesions. Its architectural design addresses fundamental challenges in training deep networks, such as the vanishing gradient problem, by leveraging residual learning to enhance feature extraction. Ongoing advancements in deep learning methodologies, including the incorporation of attention mechanisms [58] and optimized training procedures [126], hold promise for further improving diagnostic accuracy and clinical utility in ophthalmology.

6.3. Densely Connected Convolutional Networks

Densely Connected Convolutional Networks (DenseNets) have revolutionized deep learning architectures by introducing direct connections between all layers, facilitating feature reuse and efficient gradient flow [127]. Unlike traditional CNNs, where each layer connects only to its immediate predecessor, DenseNets connect each layer to every other layer in a feed-forward fashion. The differences between the individual types of connections between layers are shown in Fig. 6.5.

The core idea of DenseNets lies in the dense connectivity between layers. In a DenseNet with L layers, $\frac{L(L+1)}{2}$ there are direct connections. Each layer receives the feature maps of all preceding layers as input and passes on its own feature maps to all subsequent layers (Fig. 6.4).

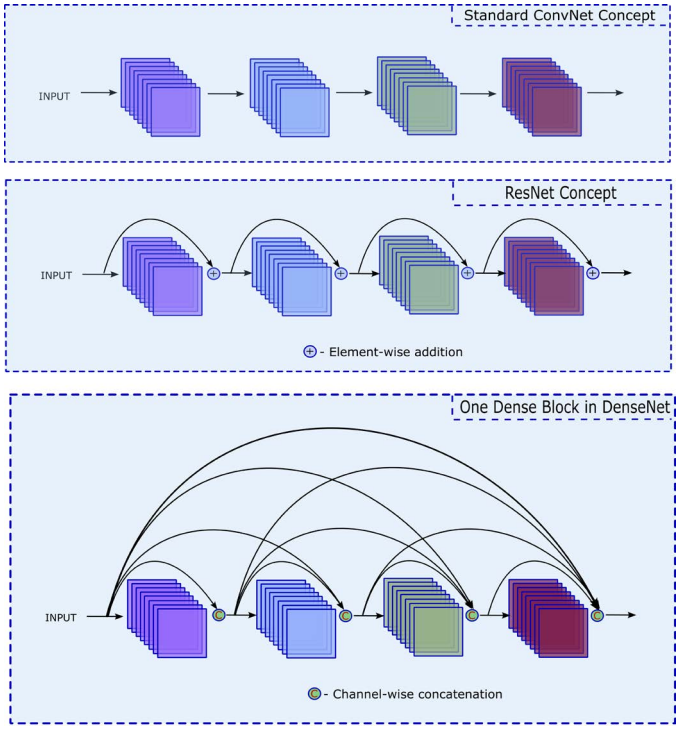


Fig. 6.4. In standard CNN, each input image goes through multiple convolutions and obtains high-level features. In ResNet, identity mapping is proposed to promote the gradient propagation. Element-wise addition is used. It can be viewed as algorithms with a state passed from one ResNet module to another one. In DenseNet, each layer obtains additional inputs from all preceding layers and passes on its own feature-maps to all subsequent layers. Concatenation is used. Each layer receives a "collective knowledge" from all preceding layers.

DenseNets are structured into dense blocks and transition layers. Within a dense block, layers are densely connected, but between blocks, transition layers are used to control the dimensions of the network. Dense Blocks are composed of multiple convolutional layers, each receiving inputs from all preceding layers within the same block. The *l*-th layer in a dense block receives the feature maps from all previous layers Eq. (6.3):

$$x_l = H_l([x_0, x_1, ..., x_{l-1}]), \tag{6.3}$$

where x_l is the output of the *l*-th layer, denotes a composite function of operations (e.g., Batch Normalization, *ReLU*, Convolution), and [x_0, x_1, ..., x_{l-1}] represents the concatenation of feature maps from layers 0 to *l-1*.

Transition Layers are placed between dense blocks to perform downsampling, they typically consist of a Batch Normalization layer, a *1×1* convolutional layer, and an *2×2* average pooling layer. Transition layers help in reducing the spatial dimensions and controlling the complexity of the model.

The growth rate *k* is a crucial hyperparameter in DenseNets, defining the number of feature maps each layer contributes. A smaller growth rate leads to a more compact model, while a larger growth rate increases the model's capacity. The growth rate determines how many new feature maps each layer adds. The total number of input feature maps for layer *l* is $k_0+k\times(l-1)$ where k_0 is the number of channels in the input layer.

To improve computational efficiency, DenseNets employ bottleneck layers using *1×1* convolutions before the *3×3* convolutions. This reduces the number of input feature maps, decreasing computational cost without sacrificing performance.The applied model is built upon the DenseNet121 architecture [127] that accepts an RGB image of *224×224* pixels. The parameters were previously pre-trained, adjusted by individual datasets and used to obtain adequate features to refine the weights of the fully connected layer as output. The convolutional layers are also known as the feature extraction phase and the last layer as the classification phase. The model output uses a Softmax activation to assign probabilities to each class. Used architecture is depicted in Fig. 6.5.

When the convolutional layers handle the image, the features obtained are passed to the classification stage. Two fully connected layers are part of the classifier. Its first layer contains 1024 units with a *ReLU* activation, followed by a dropout layer with a probability of 25% and another layer with 1024 units and *ReLU* activation. The output layer has 3 (number of classes in each dataset) units with a Softmax activation.

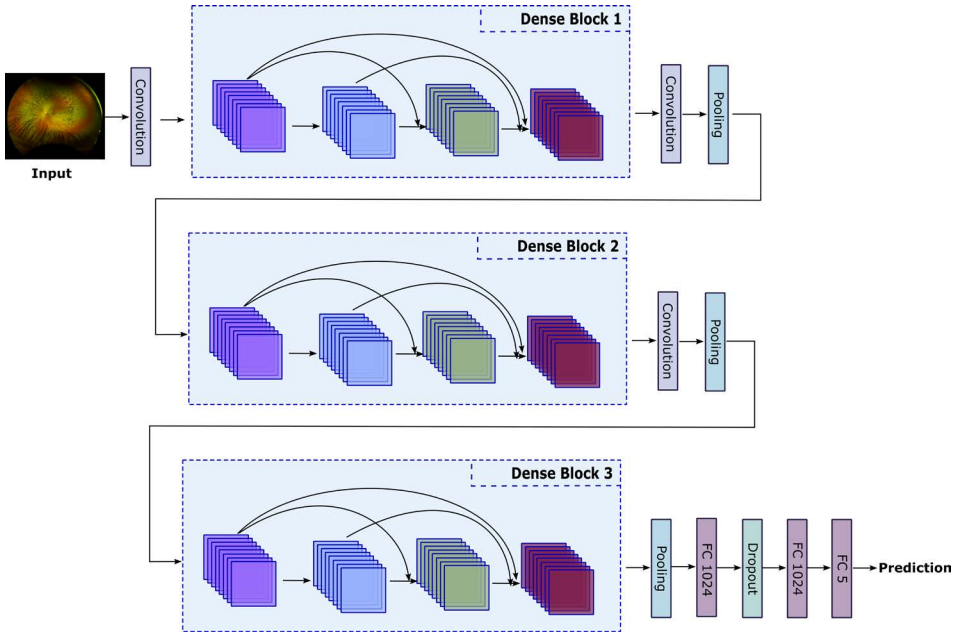


Fig. 6.5. DenseNet121 architecture with three dense blocks. Layers between two adjacent blocks are transition layers and change feature-map sizes via convolution and pooling. Adapted from [127].

6.4. Inception Networks

The Inception module is a fundamental building block introduced in the GoogLeNet architecture by Szegedy et al. to enhance neural network performance by efficiently capturing spatial correlations at multiple scales while keeping computational costs at a reasonable level. The module achieves this by combining convolutional operations of different sizes and incorporating dimension reduction techniques to limit the number of parameters [128].

At its core, the Inception module (Fig. 6.6) connects convolutional layers with filters of sizes 1×1, 3×3 and 5×5, connected with a max-pooling operation. This concatenation allows the network to process information at multiple receptive field sizes simultaneously, effectively capturing both fine and coarse features.

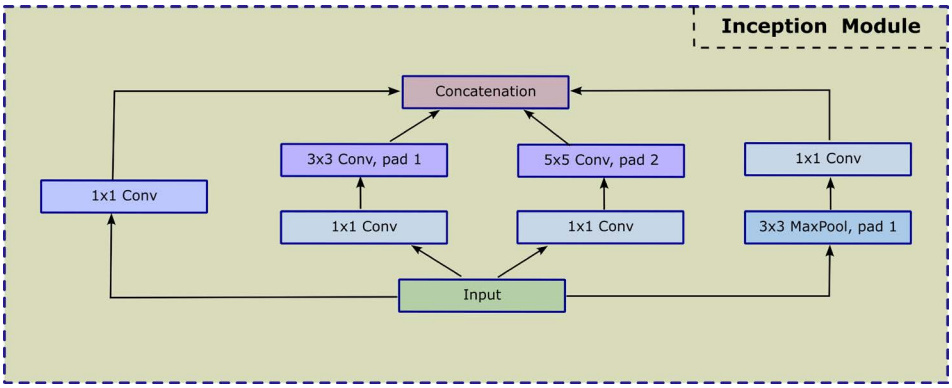


Fig. 6.6. Single inception module with dimension reduction. The module processes the input feature map through parallel branches to extract multi-scale spatial information. *1×1* convolutions reduce feature dimensionality before *3×3* and *5×5* convolutions, significantly lowering computational cost while preserving discriminative information. The max-pooling branch enhances robustness to local variations, and all branch outputs are concatenated to form a unified multi-scale representation.

Dimension reduction within the Inception module is primarily achieved through *1×1* convolutions, which serve as bottleneck layers. These layers reduce the number of input channels before the computationally expensive *3×3* and *5×5* convolutions are applied. Mathematically, a *1×1* convolution can be expressed as Eq. (6.4) [129]:

$$Y_{i,j,k} = \sum_{c=1}^{C} X_{i,j,k} * W_{c,k}, \tag{6.4}$$

where $X_{i,j,k}$ is the input feature map at spatial location *(i, j)* and channel *c*, $W_{c,k}$ is the weight connecting input channel to output channel *k*, and $Y_{i,j,k}$ is the output feature map.

Adding *1×1* convolutions before the larger convolutions, the module reduces the depth (number of channels) of the feature maps, thereby decreasing the computational load. For instance, if the input has *C*, channels and it can be reduced to *E* channels using a *1×1* convolutions, the computational cost of the subsequent *n×n* convolution decreases from $O(n^2*F*C)$ to $O(n^2*F*E)$, where *F* is the number of filters. The final output of the Inception module is obtained by concatenating the feature maps from all parallel paths along the channel dimension. If the outputs are denoted from the *1×1, 3×3* and *5×5* convolutions, and the pooling layer as Y^1, Y^3, Y^5 and Y^{pool} respectively, the concatenated output *Y* Eq. (6.5) [129] is:

$$Y = \text{Concat}(Y^1, Y^3, Y^5, Y^{\text{pool}}). \tag{6.5}$$

This architecture allows the network to learn both sparse and dense patterns efficiently. The use of dimension reduction not only lowers computational complexity but also acts as a form of regularization, mitigating overfitting by restricting the model's capacity.

To understand why this network is highly effective, consider how it combines filters. They analyse the image using a range of sizes, enabling details at different scales to be efficiently recognized by filters of corresponding sizes. At the same time, different parameters to each filter can be allocated.

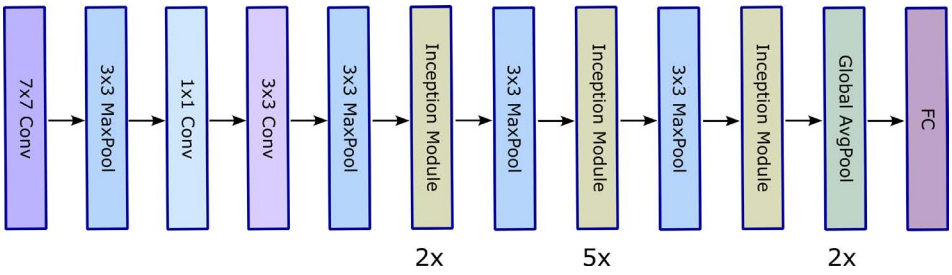


Fig. 6.7. Example Inception network architecture used for study purposes. Initial convolutional and pooling layers capture low-level patterns, while repeated Inception blocks refine increasingly abstract representations. Max-pooling layers control spatial resolution, and the global average pooling aggregates spatial information before the final fully connected layer performs classification.

As depicted in Fig. 6.7, one version of the Inception model utilizes a sequence of nine initial blocks, arranged into three groups with max-pooling layers inserted between them. It employs global average pooling at its head to produce estimates. The max-pooling between the initial blocks serves to reduce dimensionality.

The initial module employs a *7×7* convolutional layer with 64 output channels. The second module comprises two convolutional layers: first, a *1×1* convolutional layer with 64 channels, followed by a *3×3* convolutional layer that triples the number of channels. This configuration corresponds to the second branch in the Inception block and completes the body's design. At this stage, the network has 192 channels [128].

The third module connects two full Inception blocks in sequence. The first Inception block produces 256 output channels, distributed among the four branches in a ratio of 2:4:1:1. In the second Inception block, the number of output channels increases to 480, with a branch ratio of 4:6:3:2.

The fourth module is more complex, connecting five Inception blocks in series with output channels of 512, 512, 512, 528, and 832, respectively. The channel allocation among these branches is similar to those in the third module: the second branch with the *3×3* convolutional layer outputs the most channels, followed by the first branch with only a *1×1* convolutional layer, then the third branch with the *5×5* convolutional layer, and finally the fourth branch with the *3×3* max-pooling layer. The second and third branches initially reduce the number of channels according to specific ratios, which vary slightly between different Inception blocks [129].

6.5. Models' Evaluation

The above-mentioned neural network models were tested for the classification of images characteristic of retinitis pigmentosa and AVL. In Fig. 6.8 learning performance for RP dataset is depicted, while in Fig. 6.9 for AVL dataset. The obtained results are presented in Tables 6.1–6.4 and Fig. 6.10–6.13. The obtained results will serve as a base value for additional, custom classifiers. To ensure the reliability of the results, a series of experiments was performed with a random split of the data into training (80%) and test (20%) sets. To enhance the reproducibility and stability of the results, the 5-fold cross validation is applied.

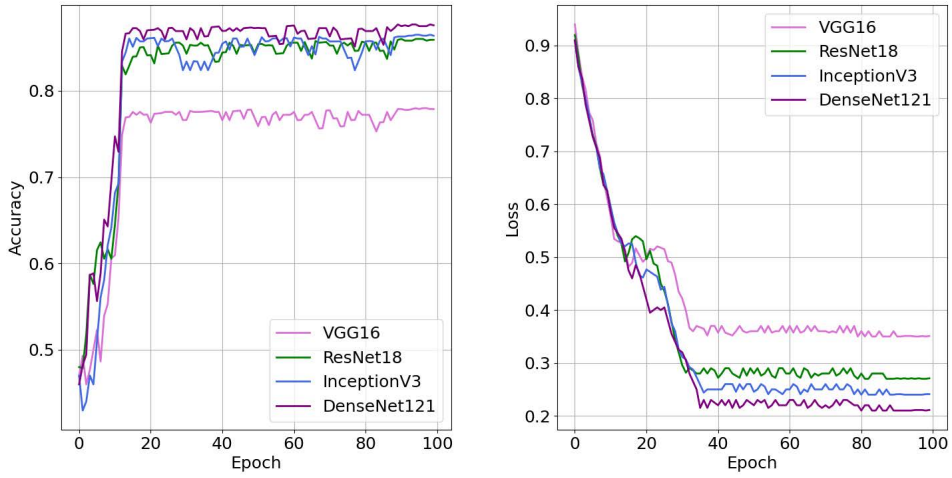


Fig. 6.8. Validation accuracies (left) and loss (right) obtained for RP dataset. The curves illustrate convergence speed and stability of each model, revealing differences in learning dynamics and final performance when trained on retinitis pigmentosa data.

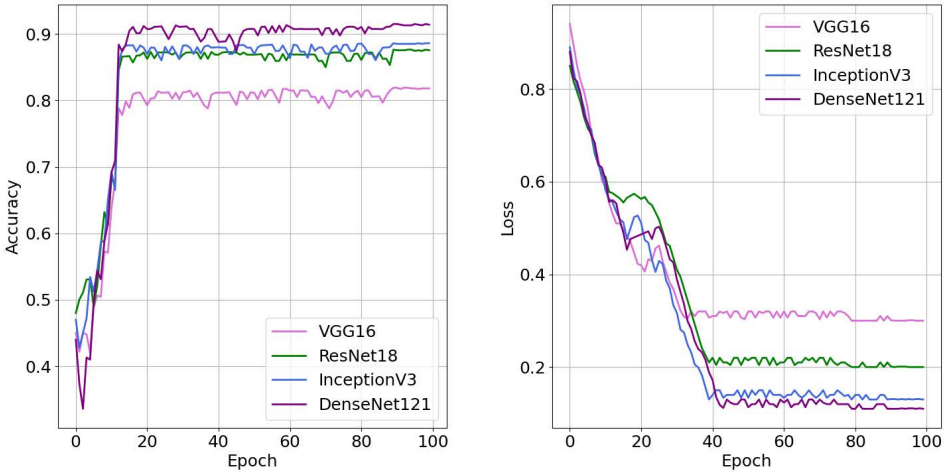


Fig. 6.9. Validation accuracies (left) and loss (right) obtained for AVL dataset. Faster loss convergence and lower final loss values indicate more effective feature learning and better optimization behaviour, particularly for deeper architectures.

Table 6.1. The results obtained for RP and AVL datasets classification by VGG16. All values in %.

Dataset	Class	Precision	Recall	F1-score	Specificity
RP dataset	RP	88.00	73.00	80.00	87.65
	CORD	50.00	68.00	58.00	89.31
	HEALTHY	66.00	77.00	71.00	82.81
AVL dataset	AVL	48.00	84.00	62.00	90.23
	DRUSEN	81.00	76.00	79.00	85.85
	HEALTHY	84.00	76.00	80.00	87.74

Table 6.2. The obtained results for RP and AVL datasets classification by ResNet-18. All values in %.

Dataset	Class	Precision	Recall	F1-score	Specificity
RP dataset	RP	89.00	83.00	86.00	87.65
	CORD	59.00	80.00	68.00	91.19
	HEALTHY	78.00	77.00	77.00	90.63
AVL dataset	AVL	48.00	84.00	62.00	90.23
	DRUSEN	87.00	82.00	84.00	89.94
	HEALTHY	92.00	86.00	87.00	94.34

Table 6.3. The obtained results for RP and AVL datasets classification by InceptionV3. All values in %.

Dataset	Class	Precision	Recall	F1-score	Specificity
RP dataset	RP	92.00	83.00	88.00	91.36
	CORD	52.00	68.00	59.00	89.94
	HEALTHY	79.00	82.00	81.00	90.63
AVL dataset	AVL	52.00	79.00	62.00	91.95
	DRUSEN	90.00	83.00	86.00	92.45
	HEALTHY	98.00	86.00	88.00	91.51

Table 6.4. The obtained results for RP and AVL datasets classification by DenseNet121. All values in %.

Dataset	Class	Precision	Recall	F1-score	Specificity
RP dataset	RP	89.00	83.00	86.00	87.65
	CORD	70.00	84.00	76.00	94.34
	HEALTHY	78.00	82.00	80.00	89.84
AVL dataset	AVL	54.00	74.00	62.00	93.10
	DRUSEN	95.00	83.00	88.00	96.23
	HEALTHY	88.00	92.00	90.00	89.61

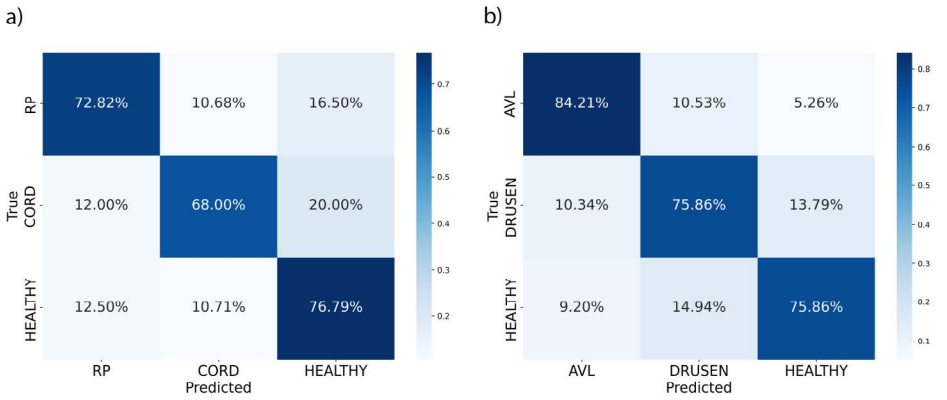


Fig. 6.10. Confusion matrices obtained for VGG16 model for a) RP dataset b) AVL dataset. The confusion matrices reveal that the most frequent misclassifications occur between retinitis pigmentosa and cone–rod dystrophy, reflecting their similar peripheral degeneration patterns. Additional confusion is observed between diseased classes and healthy controls.

The obtained learning curves (both accuracy and loss function) for the RP dataset, show that all four architectures, VGG16, ResNet-18, InceptionV3 and DenseNet121, stabilized in a similar range of epochs. VGG16 had more fluctuations in the initial stages of training, which also resulted in slightly

worse final accuracy. ResNet-18 and DenseNet121 reached the point where the validation curves stopped improving significantly more quickly, while DenseNet121 showed less tendency to overfitting. InceptionV3 was placed between ResNet-18 and DenseNet121 in this respect, which was also reflected in the results on the test set.

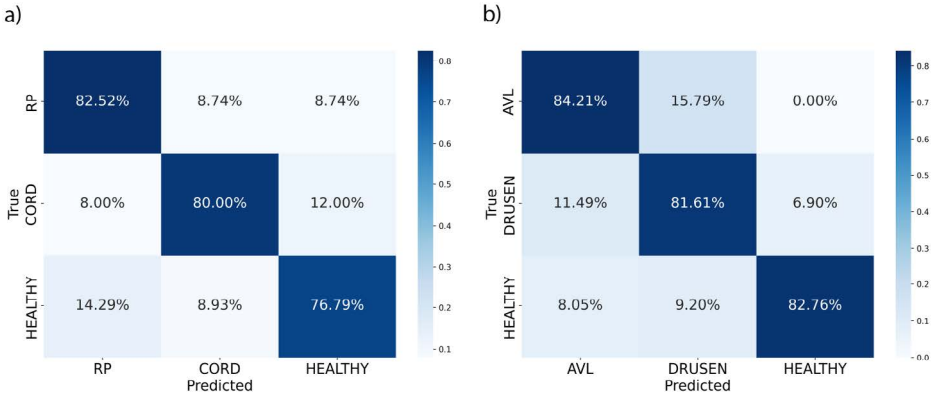


Fig. 6.11. Confusion matrices obtained for ResNet-18 model for a) RP dataset b) AVL dataset. The figure shows that the dominant source of misclassification lies between AVL and DRUSEN, which share similar subretinal deposit characteristics in OCT images. Healthy samples are occasionally confused with mild pathological cases.

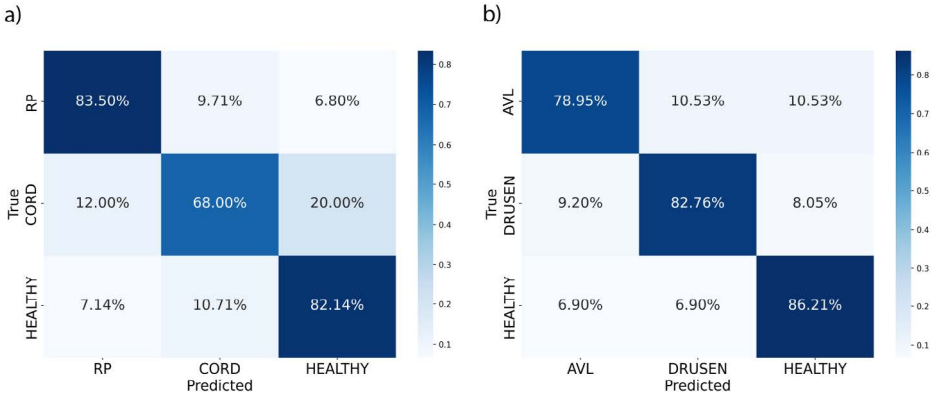


Fig. 6.12. Confusion matrices obtained for InceptionV3 model for a) RP dataset b) AVL dataset. The confusion matrices indicate reduced misclassification between RP and healthy subjects in the optimized model, while CORD remains the most challenging class, being confused with both RP and healthy cases. This pattern highlights the difficulty of separating intermediate disease phenotypes and motivates the use of more expressive architectures.

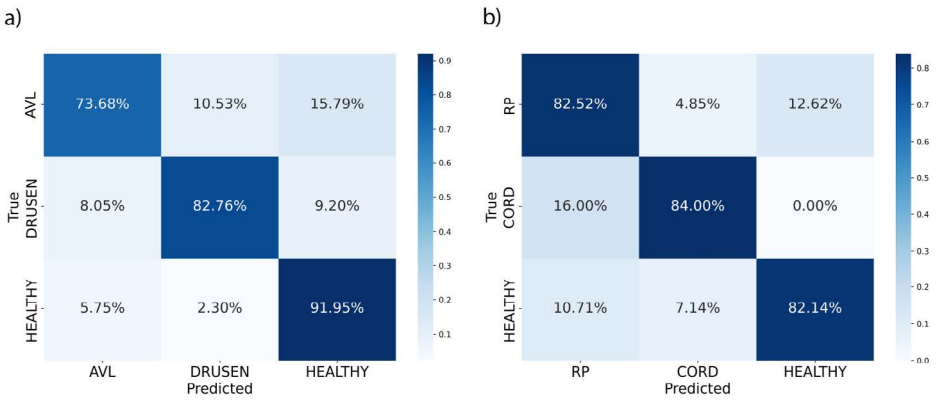


Fig. 6.13. Confusion matrices obtained for DenseNet121 model for a) RP dataset, b) AVL dataset. The figure demonstrates that misclassifications primarily occur between AVL and DRUSEN, whereas healthy samples are classified with higher reliability.

Analysis of classification metrics confirmed that DenseNet121 achieved the highest overall performance, as seen in the accuracy values (around 83%) and in the balanced precision, recall and F1-score metrics. This model was most effective in recognizing the CORD class, which was the least numerous in the RP set and at the same time quite problematic for the other solutions. InceptionV3, with an overall accuracy comparable to DenseNet, achieved very high specificity in recognizing RP, but distinguished CORD classes less well, which resulted in a lower F1-score in this category. ResNet-18 achieved good results in each of the three classes and dealt quite confidently with the classification of RP and HEALTHY, but with the CORD class it presented average accuracy and slightly lower stability than DenseNet. VGG16 performed worst in terms of overall accuracy (around 73%) and had the lowest specificity in recognizing HEALTHY. Analysis of the learning curves suggests that VGG16 showed a stronger tendency to overfitting; the difference between the training and validation curves remained at a higher level than in the case of DenseNet121 or ResNet-18.

6.6. Chapter Summary

This chapter introduces the fundamental challenges and state-of-the-art approaches for classifying rare retinal diseases, specifically retinitis pigmentosa and acquired vitelliform lesions, using convolutional neural networks and advanced deep learning techniques.

It begins by highlighting the clinical importance of early and accurate detection of such diseases, given their potential to cause progressive vision loss. OCT imaging and CNN-based methods are presented as crucial tools in this domain, owing to their ability to capture subtle retinal abnormalities that might otherwise go unnoticed.

A central focus is placed on the architectures most commonly employed in medical image analysis: Inception-based networks, Residual Networks and Densely Connected Convolutional Networks. The discussion of residual blocks illuminates how shortcut connections in ResNet mitigate training issues like vanishing gradients, permitting deeper networks to learn identity mappings when needed. This concept is linked to the broader problem of function classes, where adding layers does not always guarantee better performance unless the new function space is nested. This chapter also examined the DenseNet paradigm, emphasizing its layer-to-layer dense connectivity that reuses features and improves gradient flow. The role of Inception modules was similarly addressed, showing how multi-scale convolutional filters can process features of varying sizes in parallel while dimension-reduction techniques (*1×1* convolutions) keep computational loads feasible.

Empirical evaluations demonstrated the efficacy of these CNN architectures in classifying OCT images, particularly for RP and AVL. Each network showed unique strengths and limitations, with DenseNet121 emerging as the most balanced and effective for the least-represented classes, and InceptionV3 excelling in specificity for the RP class.

CNN architectures have proven instrumental for early diagnosis and classification of rare eye conditions like retinitis pigmentosa and AVL. By leveraging deep networks such as ResNet, DenseNet, and InceptionV3, clinicians and researchers can detect subtle pathologies in OCT images with greater reliability than by using traditional methods. Residual learning alleviates vanishing gradient issues, dense connectivity promotes efficient feature reuse, and inception modules capture multi-scale information: all of which enhance classification performance. The reported experimental results reinforce the notion that no single architecture is universally optimal. Nonetheless, the methods explored in this chapter collectively underscore the broad applicability and clinical value of deep CNNs for improving diagnostic accuracy and facilitating timely interventions for rare retinal disorders.

7. New Convolutional-Based Architecture for Eye Diseases Identification

This chapter presents three novel approaches to OCT image classification, based on advanced convolutional neural network architectures. The first is the Deep CNN-GRU Network [45], which combines the features of a deep convolutional network with GRU layers, enabling efficient processing of feature sequences extracted from images. The second approach is the Convolutional Gated Recurrent Units U-Net [30], an adaptation of the classical U-Net architecture, enhanced with GRU cells to better model sequential data. The third proposed model is Residual Attention Networks [115], which incorporates an attention mechanism and residual connections between layers, allowing for more precise analysis of critical image regions and more effective learning of deep representations.

To evaluate the classification quality achieved by each of the proposed solutions, the metrics detailed in Section 4.2 are used. Additionally, for each classifier, a confusion matrix is presented to identify the most challenging classes to distinguish and to verify potential systematic errors. Learning parameter graphs illustrating the training process and cross validation results are also provided, offering objective insights into the generalization capabilities of each model.

7.1. Deep CNN-GRU Network

The structures described in the previous chapters constituted the foundation for the construction of dedicated classifiers for detecting changes caused by diseases such as retinitis pigmentosa and acquired vitelliform lesions. Particularly important is the use of convolutional networks in combination with recurrent GRU blocks and attention modules, which enable the effective extraction of both spatial and sequential features, important for the detection of subtle pathological changes. This hybrid architecture, combining the ability of CNN to capture the complex structure of the image and GRU to model correlations between successive retinal slices, leads to significant improvement in the efficiency of disease classification [45], [130], [131]. The use of Deep GRU CNN networks in OCT analysis is part of a broader trend of searching for deep architectures, capable of interpreting complex medical data and supporting doctors in the precise and early diagnosis of eye diseases.

In order to ensure high quality classification of changes in OCT images, the Deep CNN-GRU classifier has been proposed. It consisted of five consecutive blocks. Each of them contained two convolution and one max-pooling layers. The data from the last block was processed by GRU elements, then flattened, processed by a fully-connected layer and subjected to a classification process using the Softmax function. The structure of the proposed model was presented in Fig. 7.1 and Table 7.1 [45].

As can be observed from the structure of the network, after performing several convolutional and pooling operations, high-level features are flattened into a vector and then processed by a fully connected layer. The purpose of such a layer is to connect every neuron to neurons in the bi-directional layers, enabling the network to learn complex relationships between extracted features and the target output.

Moreover, these kinds of deep neural networks are characterized by a high number of parameters and trained on low-quality or noisy data, which may lead to issues related to overfitting. This occurs when the network achieves excellent performance on the training set while struggling to accurately classify new test data from the same domain. This situation was observed in our case due to the limited number of available examples and the necessity of generating them using the methods described in Chapter Two. To mitigate the problem of overfitting, a strategy called dropout is employed. This involves randomly deactivating neurons in fully connected layers during training with a certain

probability. The application of dropout can be described by the following Eq. (7.1) [132]:

$$y_k = \sum_{k \in K^*} \text{Probability}(x) y_k^k, \tag{7.1}$$

where y_k denotes predicted unit k, K^* the set of all narrowed networks and y^K represents the output from unit K.

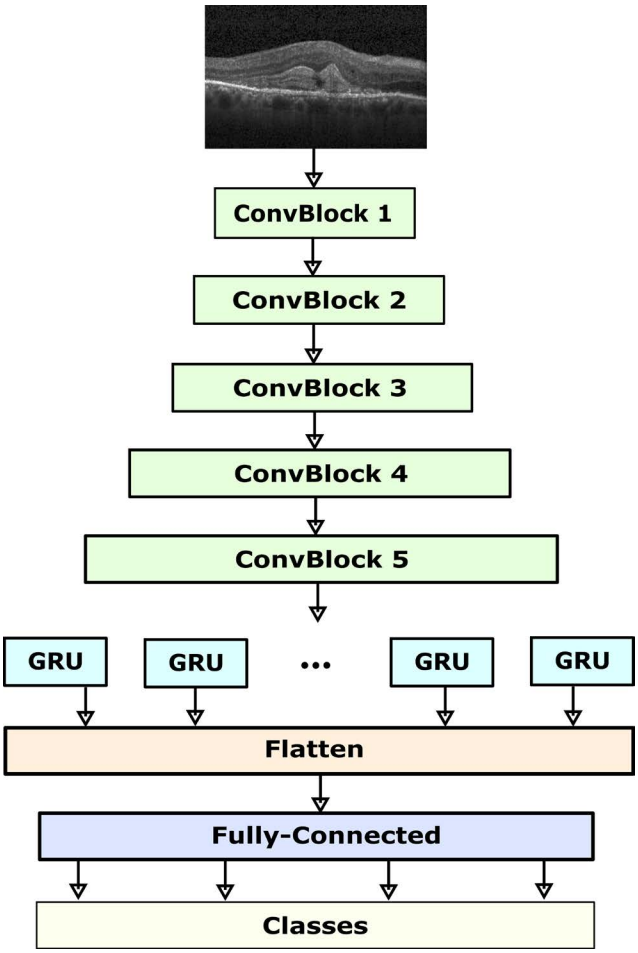


Fig. 7.1. Architecture of the proposed Deep CNN–GRU network for retinal image classification. The network first extracts hierarchical spatial features from OCT images using a sequence of convolutional blocks with increasing representational depth. The resulting feature maps are then processed by parallel GRU units, which model contextual dependencies within the extracted representations. The GRU outputs are flattened and passed to fully connected layers, where spatial and contextual information is jointly integrated.

Table 7.1. The structure of Deep CNN-GRU network.

Layer	Conv Block No	Type	Kernel size	Features No	Input size
1.	1	Conv2D	*3x3*	64	*3x224x224*
2.		Conv2D	*3x3*	64	*64x224x224*
3.		Pooling	*2x2*	-	*64x112x112*
4.	2	Conv2D	*3x3*	128	*64x112x112*
5.		Conv2D	*3x3*	128	*128x112x112*
6.		Pooling	*2x2*	-	*128x56x56*
7.	3	Conv2D	*3x3*	256	*128x56x56*
8.		Conv2D	*3x3*	256	*256x56x56*
9.		Pooling	*2x2*	-	*256x28x28*
10.	4	Conv2D	*3x3*	256	*256x28x28*
11.		Conv2D	*3x3*	512	*512x28x28*
12.		Pooling	*2x2*	-	*512x14x14*
13.	5	Conv2D	*3x3*	512	*512x14x14*
14.		Conv2D	*3x3*	512	*512x14x14*
15.		Pooling	*2x2*	-	*512x7x7*
16.	-	GRU	-	-	*512x49*
17.	-	FC	-	64	25088
18.	-	Output	-	4	64

The process of feature extraction from an image involves several steps, starting with convolutional operations and pooling layers, followed by the application of a GRU layer, and ending with the flattening and classification stage. In the first phase, each OCT image with dimensions of *224×224* pixels and three channels (R, G, B) is processed through the initial convolutional block. As a result of this stage, feature maps with a resolution of *112×112* are obtained, which are then dimensionally reduced using a max-pooling layer, leading to a final size of *64×112×112*. Next, the second convolutional block receives input data with dimensions of *64×112×112* and generates new feature maps of size *128×112×112*. Another max-pooling operation reduces these dimensions to *128×56×56*. Following the same approach, the third, fourth, and fifth convolutional blocks further transform and reduce the data, ultimately producing a set of feature maps with dimensions of *512×7×7*. These final representations are then passed to the GRU layer, which is used for the final classification.

7.1.1. Deep CNN-GRU Evaluation

Evaluation of the proposed model was conducted based on the metrics described in Subchapter 4.2. To ensure the reliability of the results, a series of experiments were performed with a random split of the data into training (80%) and validation (20%) sets. To enhance the reproducibility and stability of the results, the 5-fold cross validation is applied. In the case of the AVL dataset, the abbreviations TP, FP, TN and FN correspond to the following scenarios: TP represents correctly identified DRUSEN cases; FP refers to cases incorrectly classified as DRUSEN (although they actually belong to the HEALTHY or AVL categories); TN denotes correctly classified HEALTHY or AVL cases; and FN represents DRUSEN cases incorrectly classified as HEALTHY or AVL. A similar scheme was applied for the evaluation of all investigated classes, as well as for the RP dataset.

The key learning statistics for each dataset are depicted in Fig. 7.2 and 7.3. The results achieved by the Deep CNN-GRU model, presented in Table 7.2 and Fig. 7.4, demonstrate its ability to distinguish healthy eyes from those affected by disease, thereby confirming the effectiveness of the deep learning approach. An average accuracy exceeding 84%.

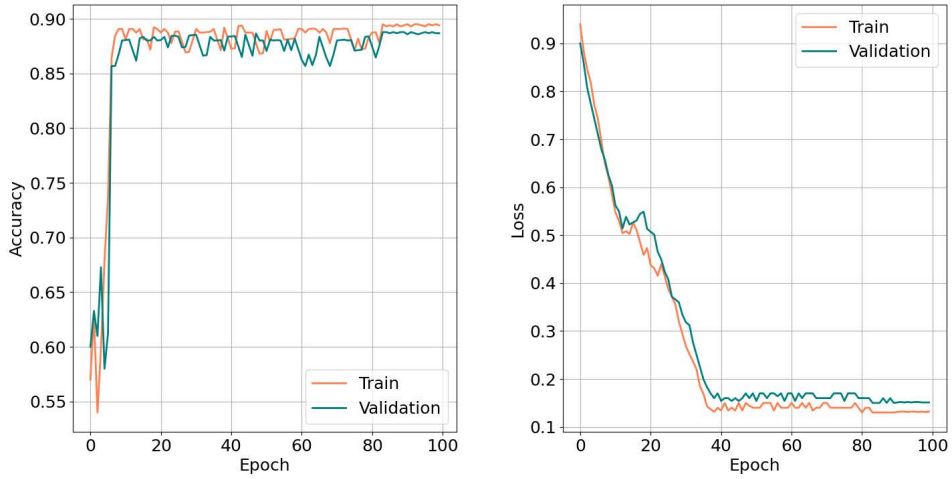


Fig. 7.2. Training and validation accuracies (left) and loss (right) obtained for RP dataset. Rapid convergence and a small gap between training and validation curves indicate stable optimization and good generalization. The smooth loss decay and consistent validation accuracy confirm the robustness of the architecture with respect to training settings and hyperparameter variations.

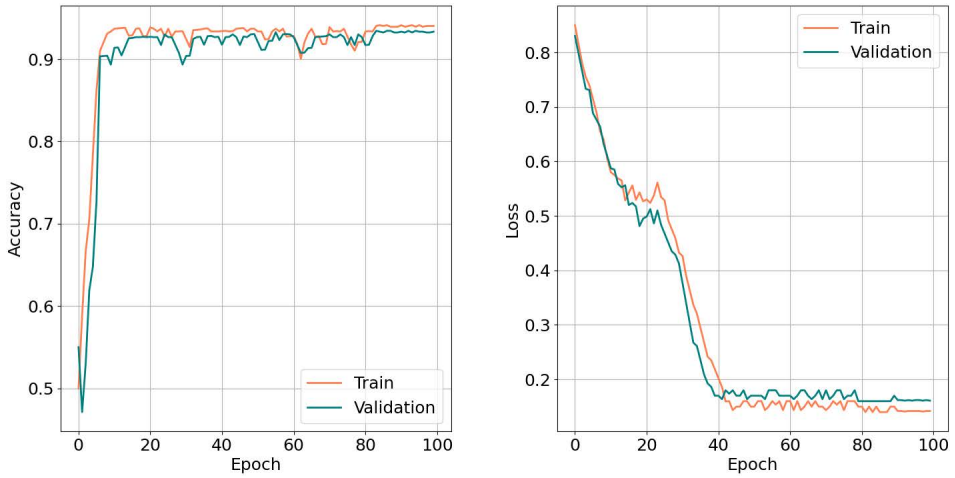


Fig. 7.3. Training and validation accuracies (left) and loss (right) obtained for AVL dataset. Rapid convergence in the early epochs indicates efficient feature learning, while the close alignment of training and validation curves demonstrates good generalization and limited overfitting. The stable loss plateau at later epochs confirms effective optimization of the proposed Deep CNN–GRU architecture.

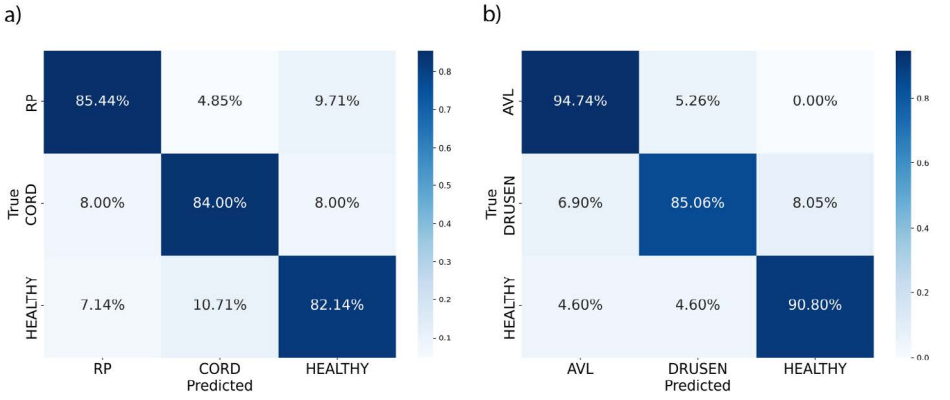


Fig. 7.4. Testing accuracies obtained for CNN-GRU model for a) RP dataset b) AVL dataset. For the RP dataset, the most frequent misclassifications are observed between healthy and CORD cases (10.71%) as well as between RP and healthy subjects (9.71%), indicating partial overlap between early pathological changes and normal retinal structures. Confusion between RP and CORD remains limited, with misclassification rates below 8%. For the AVL dataset, misclassifications mainly occur between DRUSEN and healthy samples (8.05%) and between DRUSEN and AVL cases (6.90%), while AVL is classified with very high accuracy (94.74%) and shows no confusion with the healthy class.

Table 7.2. The results obtained for RP and AVL datasets classification by CNN-GRU. All values in %.

Dataset	Class	Precision	Recall	F1-score	Specificity
RP dataset	RP	93.62	85.44	89.31	92.59
	CORD	65.63	84.00	73.70	93.08
	HEALTHY	79.31	82.14	80.69	90.63
AVL dataset	AVL	64.29	97.74	76.61	94.25
	DRUSEN	93.67	85.06	89.08	95.28
	HEALTHY	91.86	90.80	91.32	93.40

Fig. 7.2 presents the learning curves for the proposed CNN network with the GRU module, trained on the RP dataset. A rapid increase in accuracy is visible in the initial epochs. The model quickly learns key patterns characterizing the RP, CORD, and HEALTHY classes. In the subsequent stages of training, both the training and validation graphs stabilize at around 85–90%. The difference between the training and validation lines is not large. The loss function curves are analogous: initially, the loss values drop rapidly, to stabilize in the range of 0.1–0.2 after a dozen or so epochs. This behaviour suggests that the network effectively learns distinctive features for individual RP, CORD, and HEALTHY categories, without overfitting the training data.

For the AVL dataset (Fig. 7.3), the learning dynamics are similar, although the model here reaches an even higher level of accuracy, above 90%, after about 20 epochs. The difference between the training and validation curves remains low. In the context of the difficult, sparse AVL class, high validation accuracy suggests that the network effectively recognizes even relatively rare disease patterns. At the same time, the loss function decreases smoothly for both training and validation, stabilizing within the range of 0.1–0.2.

Analysis of the confusion matrix (Fig. 7.4.a) for the RP dataset shows that the model recognizes the CORD class quite reliably (84% correct classifications), which is important in the context of the smallest number of samples belonging to this category. The RP class achieves about 85% accuracy, and HEALTHY 82%. Incorrect assignments occur, but the scale of errors between RP and HEALTHY is relatively small. This indicates that the use of the GRU module allows the model to effectively capture sequential or contextual dependencies in OCT data and to improve the quality of classification. Fig. 7.4.b presents the confusion matrix in the AVL dataset. Particularly important is the high accuracy in the AVL class (over 94%), which means that the network rarely misses this rare but clinically important disease entity. DRUSEN and HEALTHY are also recognized with high efficiency, although a certain part of DRUSEN samples is sometimes confused with

HEALTHY and vice versa. A small number of errors at such a high level of accuracy confirms that the CNN architecture enriched with the GRU layer can capture even subtle differences in the structure of the OCT image, crucial for correctly distinguishing DRUSEN from HEALTHY, as well as AVL from other classes.

The total cost of the proposed model is in the order of ~22.38 GFLOPs per forward pass, with the number of parameters ~11.12M. In practice, the model performs a single inference in ~6ms on a mid-range GPU (Table. 7.3, Table. 7.4), which confirms clinical applicability in near-real-time settings.

Table 7.3. Comparison of inference complexity and speed for the baseline CNN and Deep CNN-GRU variants. Timing measurements for RTX 4090 and single-core Ryzen 7950X.

Model	Params [M]	GFLOPs	RTX 4090 [ms]	Ryzen 7950X [ms]
Baseline CNN	11.01	22.37	61.3	479
Deep CNN-GRU	11.12	22.38	63.7	582

Table 7.4. FLOPs for a single image *224×224*. Values in billions of floating-point operations (GFLOPs).

Layer no.	Layer	Configuration/ size	FLOPs [G]	Part
Block 1 (3→64, *224×224*)				
1.	*Conv 3×3*	(3→64, *224×224*)	0.1734	0.8%
2.	*Conv 3×3*	(64→64, *224×224*)	3.36994	16.5%
Block 2 (64→128, *112×112*)				
3.	*Conv 3×3*	(64→128, *112×112*)	1.8497	8.3%
4.	*Conv 3×3*	(128→128, *112×112*)	3.36994	16.5%
Block 3 (128→256, *56×56*)				
5.	*Conv 3×3*	(128→256, *56×56*)	1.8497	8.3%
6.	*Conv 3×3*	(256→256, *56×56*)	3.36994	16.5%
Block 4 (256→512, *28×28*)				
7.	*Conv 3×3*	(256→512, 28×*28*)	1.8497	8.3%
8.	*Conv 3×3*	(512→512, 28×*28*)	3.36994	16.5%
Block 5 (512→512, *14×14*)				
9.	*Conv 3×3*	(512→512, *14×14*)	0.9248	4.1%
10.	*Conv 3×3*	(512→512, *14×14*)	0.9248	4.1%
Sequential section and classification				
11.	GRU	*In* = 512, *h* = 64, seq. 49	0.0108	<<0.01%
12.	FC	(25088→64)	0.0032	<<0.01%
13.	FC	(64→3)	<0.0001	<<0.01%
Total			**22.384**	**100.0%**

Table 7.5. Ablation study for Deep CNN – GRU model (Mean values in %).

Dataset	Retinitis Pigmentosa			AVL		
Model	Precision	Recall	F1-score	Precision	Recall	F1-score
Deep CNN-GRU	79.52	83.86	81.23	83.27	91.20	85.67
– GRU	77.16	81.07	78.93	81.40	88.50	84.30
– Batch Normalization	78.80	82.56	80.49	82.17	89.64	85.10
– Dropout in FC layers	79.50	82.55	80.73	82.46	90.11	85.39

The ablations studies (Table 7.5) confirm the design choices in CNN–GRU. Removing the GRU influents recall the most (and thus F1-score), showing the recurrent part is key for capturing spatial context.

To sum up, the presented results indicate that designed own CNN network with the GRU module included effectively copes with the classification of both the RP and AVL sets. Stable learning curves indicate an effective training process, while high values of correct classifications in the confusion matrices suggest that this architecture can well capture key morphological features in OCT images. This is particularly important in the context of rare classes.

7.2. Convolutional Gated Recurrent Units U-Net

U-Net is a convolutional neural network architecture initially designed for biomedical image segmentation [133]. It has become a cornerstone in the field of deep learning for its ability to capture intricate features in visual data. Characterized by its unique U-shaped structure with symmetric encoding and decoding paths, U-Net excels at preserving spatial information while extracting contextual features. While its proficiency in segmentation tasks is well-documented, recent advancements have extended its application to classification problems [30], [112], [134]. This chapter describes how U-Net architectures can be adapted for classification tasks, exploring the modifications required, the benefits over traditional models, and real-world applications that showcase their versatility and effectiveness in classification scenarios.

The inspiration for this network is a combination of [127], [133], [135], [136]. The network utilizes the strengths of Bidirectional GRU networks (Fig. 7.7) as well as densely connected convolutions. The individual components are described in detail in the following sections. The whole structure is presented in Fig. 7.5.

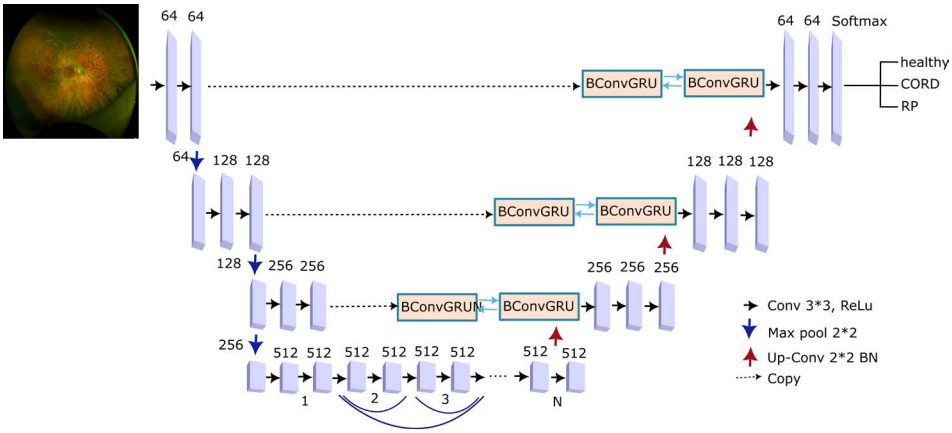


Fig. 7.5. Architecture of the proposed convolutional U-Net with bidirectional GRU blocks for retinal image classification. The network follows an encoder–decoder U-Net structure, where convolutional blocks progressively extract multi-scale spatial features and max-pooling layers reduce spatial resolution to capture global context. Skip connections transfer high-resolution features from the encoder to the decoder, preserving fine structural details. Bidirectional GRU blocks are integrated to model contextual dependencies within feature maps. The decoder reconstructs enriched feature representations, which are finally aggregated and classified using fully connected layers with a softmax output.

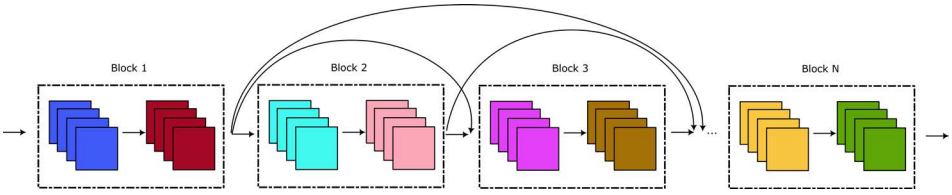


Fig. 7.6. The architecture of U-Net dense layer. By aggregating low – and high-level features throughout the network, the dense U-Net structure enhances sensitivity to subtle retinal abnormalities while improving training stability.

7.2.1. Encoding

The modified U-Net architecture consists of an encoding path composed of four stages. Each stage includes two *3×3* convolutional filters, followed by a *2×2* max-pooling operation and activated using the *ReLU*. With each progression, the number of feature maps is doubled. This progressive transformation allows the network to capture increasingly abstract and high-level image features. As

a result, the final layer of the encoding path produces a high-dimensional feature representation containing rich semantic information [30].

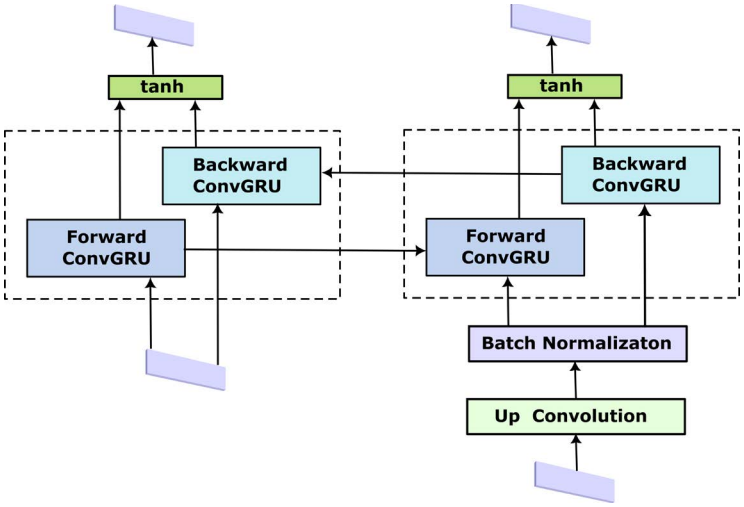


Fig. 7.7. The architecture of Bi-directional ConvGRU. This bidirectional scheme enables the model to capture contextual dependencies from both preceding and succeeding spatial locations. Batch normalization stabilizes the recurrent activations, while the tanh nonlinearity fuses forward and backward representations into a unified feature map.

In the traditional U-Net, the last step of the encoding path consists of a series of convolutional layers, enabling the network to learn various types of features. However, this can lead to the learning of redundant features due to successive convolutions. To address this issue, densely connected convolutions were introduced [127]. This enhancement improves network performance by leveraging the concept of "collective knowledge". It involves concatenating the feature maps from all preceding convolutional layers with the feature map from the current layer. These combined feature maps are then passed as input to the next convolutional layer. This strategy promotes the reuse of valuable feature maps throughout the network, reducing the likelihood of learning redundant features.

Densely connected convolutions offer several advantages over regular convolutions [127]. Firstly, they help the network learn a diverse set of feature maps rather than redundant ones. Additionally, this concept enhances the network's representational power by facilitating information flow and feature reuse throughout the network. Moreover, dense connections can utilize all previously generated features, helping the network avoid issues like exploding or vanishing gradients. Furthermore, gradients are propagated more efficiently

to their respective locations in the network during backpropagation. The proposed network incorporates densely connected convolutions by introducing two consecutive convolutions as one block. There is a sequence of N such blocks in the last convolutional layer of the encoding path, as shown in Fig. 7.5 and Fig. 7.6. These blocks are densely connected.

7.2.2. Decoding

The decoding path starts by applying an up-sampling function to the output of the previous layer. In the conventional U-Net architecture, corresponding feature maps from the contracting path are cropped and copied over to the decoding path. These feature maps are then concatenated with the result of the up-sampling function. However, in this method, a bidirectional convolutional GRU is used to process these two types of feature maps in a more sophisticated manner. The feature maps copied from the encoding part are initially passed through an up-convolutional layer. This layer performs an up-sampling function followed by a 2×2 convolution, which doubles the spatial dimensions of each feature map while halving the number of feature channels. Consequently, the feature maps progressively increase in size throughout the expanding path, layer by layer, until they return to the original input image size at the final layer.

7.2.3. Bidirectional Convolutional GRU

After the up-sampling step, the feature map is processed using Batch Normalization (BN). During training, intermediate layers often struggle with varying activation distributions, which can significantly slow down the training process because each layer must adapt to new distributions at every training step. To mitigate this issue, BN [121] is employed to enhance the neural network's stability. BN standardizes the inputs to a layer by subtracting the batch mean and dividing by the batch standard deviation. This normalization not only speeds up the training of the neural network but also provides a regularization effect in some cases, further improving the model's performance.

The output from the BN step is then fed into a bidirectional convolutional GRU layer. A major drawback of the standard GRU is that it does not consider spatial correlations because it uses fully connected layers for input-to-state and state-to-state transitions. To address this limitation, a bidirectional convolutional GRU was proposed that leverages convolution operations. This concept, based on LSTM, was introduced in [137]. The layer consists of an input vector x_t, an

output vector u_t, a reset gate r_t, an update gate a_g and a candidate activation vector $\widetilde{u}_t$. The update rule for the input vector and the previous output is defined by the following equations (for clarity, subscripts and superscripts on the parameters have been omitted) [138]:

$$r = \sigma(W_{r*n}[u_{t-1}; x_t] + b_r), \tag{7.2}$$

$$a = \sigma(W_{u*n}[u_{t-1}; x_t] + b_u), \tag{7.3}$$

$$c = \rho(W_{c*n}[x_t; r \odot u_{t-1}] + b_c), \tag{7.4}$$

$$u_t = a \odot u_{t-1} + (1 - a) \odot c, \tag{7.5}$$

where σ, ρ are sigmoidal and *ReLU* function respectively. $*n$ denotes a convolution kernel of size $n \times n$ and $\odot$ indicates a Hadamard product operation. Brackets are presented to indicate a feature concatenation. r, a, c, u_t denotes the typical elements (reset, update, current memory content, and a new state) of a GRU unit.

7.2.3. Convolutional Gated Recurrent Units U-Net Evaluation

The proposed model was evaluated using the metrics outlined in Subchapter 4.2. To ensure result reliability, experiments were conducted with a random split of the data into training (80%) and validation (20%) sets. Additionally, 5-fold cross validation was employed to enhance reproducibility and stability. The key learning statistics for each dataset are illustrated in Fig. 7.8 and 7.9. The results obtained from the Deep CNN-GRU model, as shown in Table 7.6 and Fig. 7.10, highlight its ability to differentiate between healthy eyes and those affected by disease, validating the effectiveness of the deep learning approach.

Table 7.6. The obtained results for RP and AVL datasets classification by GRU U-Net. All values in %.

Dataset	Class	Precision	Recall	F1-score	Specificity
RP dataset	RP	90.00	87.38	88.66	87.65
	CORD	62.50	80.00	70.18	92.45
	HEALTHY	88.46	82.14	85.23	95.32
AVL dataset	AVL	60.71	89.47	72.43	93.68
	DRUSEN	92.94	90.80	91.83	94.34
	HEALTHY	96.25	88.51	92.36	97.17

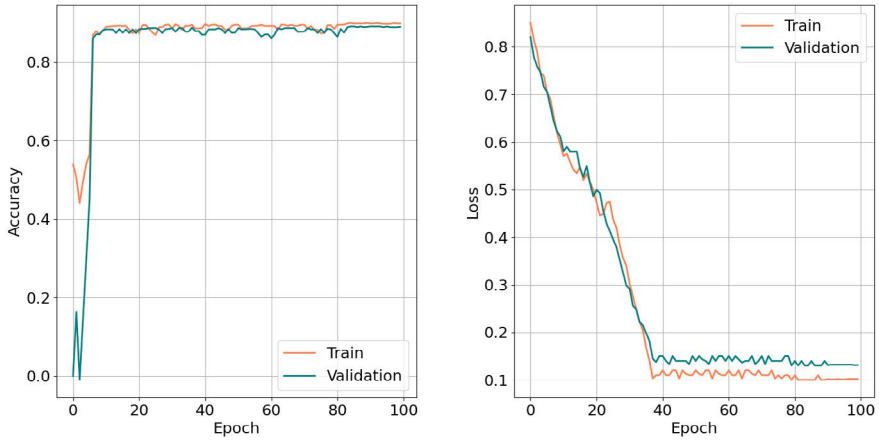


Fig. 7.8. Training and validation accuracies (left) and loss (right) obtained for RP dataset. Rapid growth of accuracy in the initial epochs indicates efficient feature extraction, while the close alignment of training and validation curves demonstrates good generalization and limited overfitting. The smooth decrease and stabilization of the loss function confirm effective optimization and stable convergence during training.

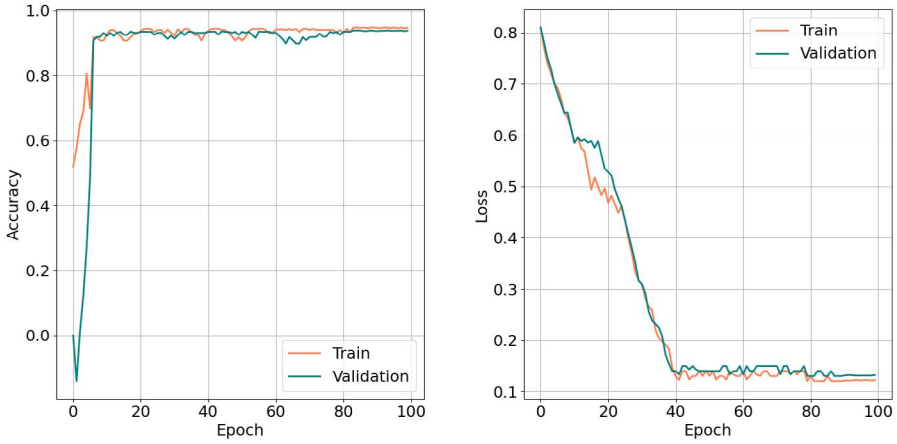


Fig. 7.9. Training and validation accuracies (left) and loss (right) obtained for AVL dataset. Rapid convergence in the initial epochs indicates efficient learning of disease-relevant features, while the close agreement between training and validation curves demonstrates good generalization.

In the presented results for the RP dataset (Fig. 7.8), a rapid increase in accuracy is noticeable in the first dozen or so epochs, which indicates the effective learning of the GRU U-Net network of key features differentiating the RP, CORD and HEALTHY classes. Then, both the training and validation graphs stabilize

around high accuracy of about 90%, with a moderately low loss function, which suggests that the network successfully captures specific patterns in OCT images.

Analysis of the confusion matrix in the same set reveals that the RP and HEALTHY classes are distinguished with high precision and sensitivity, while the specificity also remains at a level exceeding 85–90%. This means that the model relatively rarely classifies RP images as HEALTHY (and vice versa), minimizing false alarms. A slightly bigger problem appears in the case of the CORD class, which is understandable due to the smaller number of samples in this category. Despite this, the obtained recall and specificity values remain at a satisfactory level, indicating effective detection of true CORD cases and a small number of incorrect assignments.

For the AVL dataset (Fig. 7.9), a similar course of learning curves is observed, although the network reaches the target accuracy even faster. This is influenced by the better recognition of the DRUSEN and HEALTHY classes, which dominate the set numerically. The effectiveness in detecting the AVL class (significantly less numerous) is still good, as evidenced by relatively high sensitivity and precision. The results of the confusion matrix show, however, that although there are occasional mistakes between AVL and DRUSEN, the F1-score and specificity values indicate a moderate number of false classifications. Similar to the RP dataset, the difference between the training and validation curves is small.

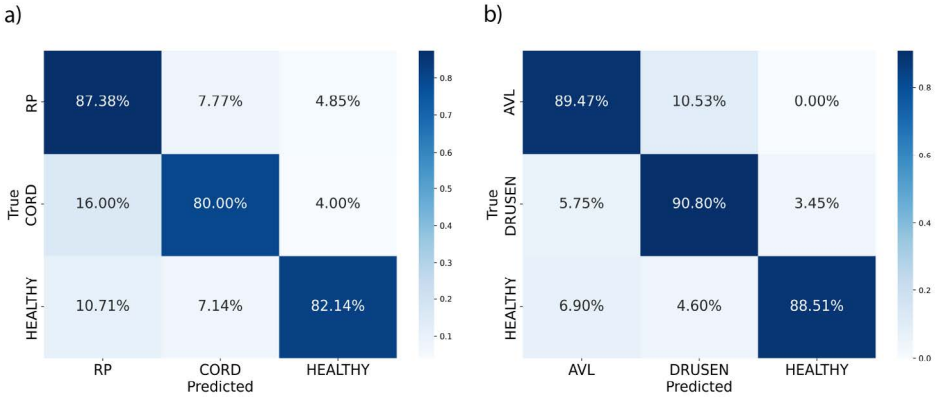


Fig. 7.10. Testing accuracies obtained for GRU U-Net model for a) RP dataset b) AVL dataset. The model achieves quite high recognition rates for all classes, with the most frequent misclassifications occurring between CORD and RP (16.00%), reflecting overlapping degeneration patterns between these inherited retinal diseases. Additional confusion is observed between healthy and RP cases (10.71%). In case of the AVL dataset, the dominant misclassifications occur between AVL and DRUSEN (10.53%), while confusion with healthy samples remains limited. The high diagonal values across both datasets demonstrate effective class separability.

Table 7.7. Comparison of inference complexity and speed for the baseline U-Net and Deep GRU U-Net variants. Timing measurements for RTX 4090 and single-core Ryzen 7950X.

Model	Params [M]	GFLOPs	RTX 4090 [ms]	Ryzen 7950X [ms]
Baseline U-Net	7.9	36.5	22.6	279
GRU U-Net	9.2	39.8	24.9	382

Table 7.8. FLOPs for a single image *256×256*. Values in billions of floating-point operations (GFLOPs).

Layer no.	Layer	Configuration/ size	FLOPs [G]	Part
Encoder				
1.	Block E1	(1→32→32, *256×256*)	6.4	16.1%
2.	Block E2	(32→64→64, *128×128*)	8.10	20.3%
3.	Block E3	(64→128→128, *64×64*)	5,9	14.8%
4.	Block E4	(128→256→256, *32×32*)	4.1	10.3%
Dense				
5.	Dense	(4×*Conv 3×3 with concatenation, C ≈ 512, 16×16)*	8.9	22.4%
Decoder				
6.	UpConv D4	(*up×2+ Conv 512→256, 32×32)*	1.50	3.8%
7.	Bi-ConvGRU D4	*C* = 256, kernel *32×32*	0.16	0.4%
8.	Block D4	(2×*Conv 3×3 256→256, 32×32)*	1.35	3.4%
9.	UpConv D3	(*up×2+ Conv 256→*128*, 64×64)*	1.25	3.1%
10.	Bi-ConvGRU D3	*C* = 128, kernel *32×32*	0.14	0.4%
11.	Block D3	(2×*Conv 3×3 128→64, 128×128)*	1.10	2.8%
12.	UpConv D2	(*up×2+ Conv 128→64, 128×128)*	0.90	2.3%
13.	Bi-ConvGRU D2	*C* = 64, kernel *32×32*	0.08	0.2%
14.	Block D2	(2×*Conv 3×3 64→64, 128×128)*	0.80	2.0%
15.	UpConv D1	(*up×2+ Conv 64→32, 256×256)*	0.60	1.5%
16.	Block D1	(2×*Conv 3×3 32→32, 256×256)*	0.75	1.9%
Sequential section and classification				
17.	FC	(*32*→3)	<0.02	<<0.01%
Total			**39.8**	**100.0%**

Compared to the baseline U-Net, the proposed Convolutional GRU U-Net increases the number of parameters by about 18% and GFLOPs by ~9%. This translates into a slight increase in response time: ~2.3 ms on an RTX 4090 and ~+103 ms on a single CPU core (Table 7.7). The largest influence of the computational overhead is attributed to the encoder (about 61%) and the "dense" layers (about 22%). The decoder section, despite the presence of Bi-ConvGRU, is lightweight:

a single module accounts for ~0.2–0.4% of the total. This highlights that the cost is due to early high-resolution convolutions, while the sequential mechanisms add functionality with marginal overhead (Table. 7.8). Table 7.9 presents the assessment of the impact of individual model elements on the classification quality.

Table 7.9. Ablation study for GRU U-Net model (Mean values in %).

Dataset	Retinitis Pigmentosa			AVL		
Model	Precision	Recall	F1-score	Precision	Recall	F1-score
GRU U-Net	80.32	83.17	81.36	83.30	89.59	85.54
– Dense	78.60	81.70	80.00	81.60	88.10	84.00
– Bi-ConvGRU	77.90	81.00	79.10	80.90	87.40	83.30
– Batch Normalization	79.30	82.40	80.50	82.10	88.80	84.60

Combining the GRU module with the U-Net architecture clearly improves the discrimination of key features for the classification of rare eye diseases, which translates into high quality measures in both sets. The results suggest that the network not only handles large classes well (e.g., DRUSEN and HEALTHY), but also maintains a decent level of performance on smaller classes (CORD, AVL). This aspect is extremely important for practical clinical applications, where the rare disease class is crucial for early diagnosis. At the same time, there is space for further improvement in the recognition of the most problematic classes.

7.3. Residual Attention Network

Attention mechanisms (AM) play a critical role in enhancing the ability of CNNs to focus on relevant regions of input while eliminating unnecessary background fragments. In applications related to object detection, these relevant regions can be associated with objects of interest or specific areas of selected classes in the image, requiring both classification and precise localization. The AM architecture includes key components such as a basic 2D convolutional layer, an MLP supporting channel-wise attention and a sigmoid function generating a mask based on the input feature map [101]. An example of the basic structure of an attention module is illustrated in Fig. 7.11.

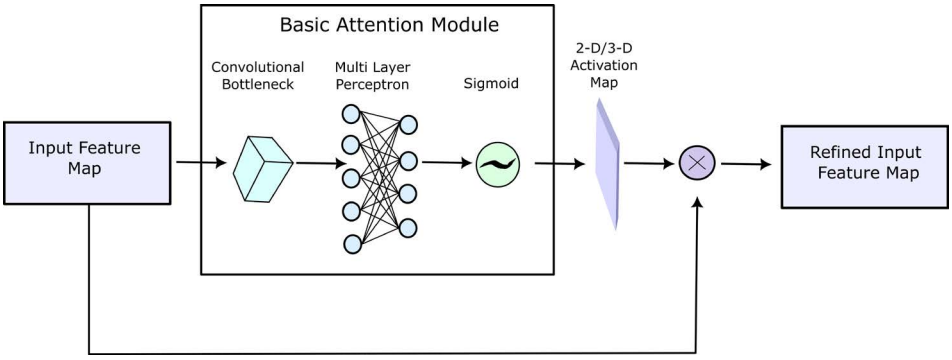


Fig. 7.11. The structure of Basic Attention Module [91] consisting of Convolutional Bottleneck, Multi Layer Perceptron together with Sigmoid function. It can be seen that a refined input feature map is generated based on the input feature map and the feature map calculated utilizing AM.

The attention module receives an input feature map with dimensions $C \times H \times W$, producing an output attention map with dimensions $1 \times H \times W$ or $C \times H \times W$ (in the case of three-dimensional attention). The resulting attention map is then combined with the original feature map through element-wise multiplication, enabling a clearer and more focused outcome. Broadly, attention mechanisms can be applied in both spatial and channel dimensions. Typically, two attention mechanisms are executed sequentially, as shown in Fig. 7.12, or in parallel [139].

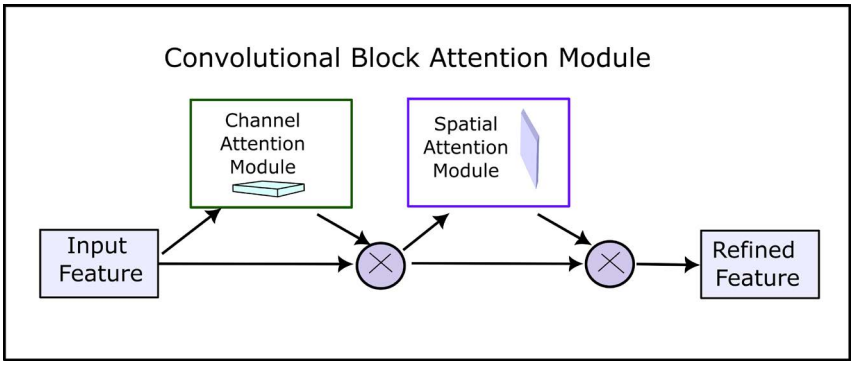


Fig. 7.12. The location of Channel and Spatial Attention Module in Convolutional Block AM [79].

The first implementations of these attention mechanisms were observed in ResNet architectures. Fig. 7.13 presents the complex structure of an attention layer, including all key components and processing stages.

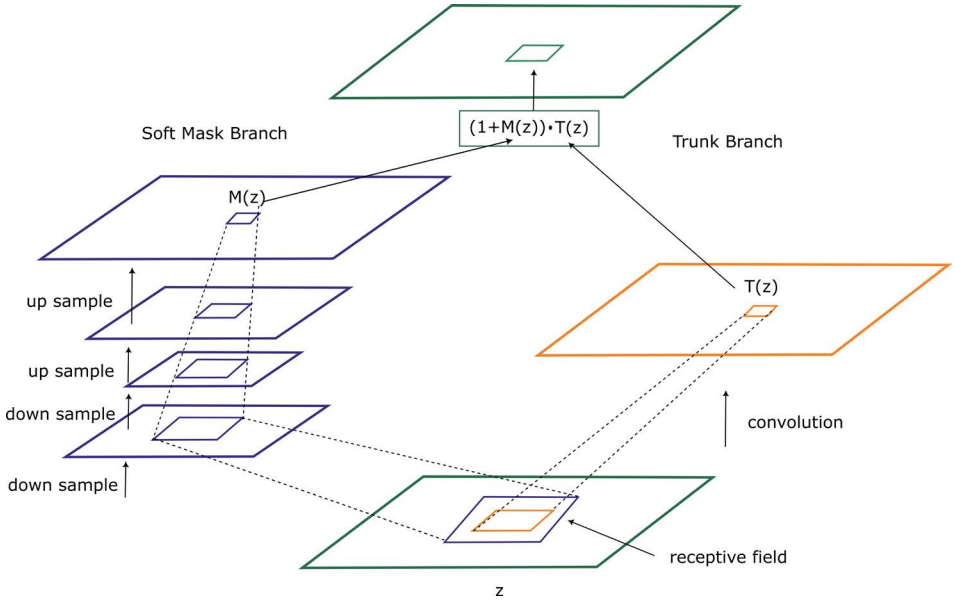


Fig. 7.13. Attention learning process [103] utilizing mask branch and trunk branch. The architecture consists of two cooperating components. The trunk branch extracts task-relevant feature representations T(z) using standard convolutional operations, while the soft mask branch learns an attention map M(z) through successive down-sampling and up-sampling stages to model spatial importance at multiple scales. The final feature response is computed as (1+M(z))•T(z), which adaptively amplifies informative regions and suppresses less relevant background areas.

Taking inspiration from the residual attention mechanism presented in the previous sections and described in [97], this work uses a mask branch that includes both top-down feedback steps and fast forward processing. These operations allow for fast collection of global information from the whole image, while the second pooling integrates two values: the global data and the initial feature maps. In the context of convolutional networks, this approach can be understood as a full convolutional structure that includes bottom-up and top-down processes.

The process starts with input data, over which max-pooling operations are repeatedly performed, allowing for a fast increase of the receptive field using a small number of residual units. Once the lowest resolution is reached, a symmetric top-down architecture is applied to extend the global information and target the input features at each spatial location in the image. Linear interpolation is used to upsample the output over the set of residual units, with the number of interpolation operations corresponding to the number of max-pooling operations, ensuring that the output size matches the input

features. Then, a sigmoid layer normalizes the output range to [0,1], followed by two more *1×1* convolutional layers. Additionally, to efficiently capture information at different scales, skip connections are introduced between the bottom-up and top-down segments. The end-to-end structure of the module is illustrated in Fig. 7.14 [115].

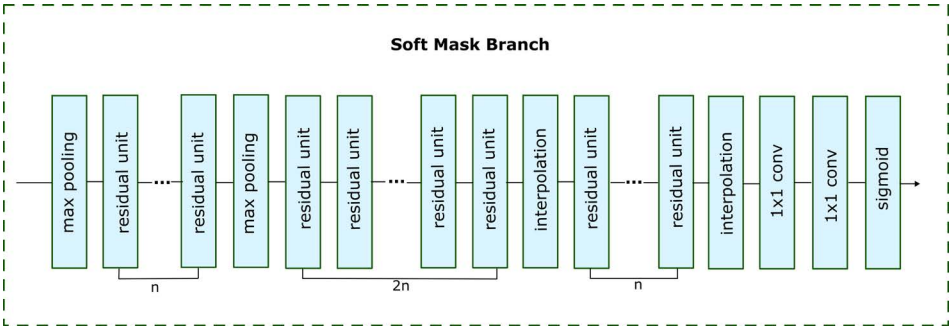


Fig. 7.14. A Soft Mask Branch structure [115] where n denotes the number of Residual Units between adjacent pooling layer in the mask branch.

The Residual Attention Network (RAN) is built by aggregating multiple Attention Modules. Each of these modules is composed of two parts: the mask branch and the trunk branch. The mask branch is responsible for processing features and can be seamlessly integrated into modern network architectures. In our study, the pre-activation residual unit was used [115].

For the output *T(z)* of the trunk branch with input *z*, the mask branch employs a bottom-up top-down architecture [140], [141], [142] to generate a mask *M(z)* of the same dimensions. This mask subtly modulates the output features *T(z)*. The fundamental structure of the bottom-up top-down approach mimics the rapid development and feedback loops characteristic of attention processes. The resulting mask functions as a control mechanism. The final output of the AM module, denoted as *H*, is defined as follows Eq. (7.6) [103]:

$$H_{\mathrm{i,c}}(z) = M_{\mathrm{i,c}}(z) * T_{\mathrm{i,c}}(z), \tag{7.6}$$

where *i* encompasses all spatial locations and $c \in \{1, 2, ..., C\}$ denotes the channel index. This setup is optimized for end-to-end training, functioning both as a feature selector during forward inference and as a gradient filter during backpropagation. In the case of the soft mask branch, the gradient with respect to the mask for the input features is defined as Eq. (7.7):

$$\frac{\partial M(z,\vartheta)T(z,\tau)}{\partial \tau} = M(z,\vartheta)\frac{\partial T(z,\tau)}{\partial \tau}, \tag{7.7}$$

where ϑ denotes the parameters of the mask branch, and represents the parameters of the trunk branch. The full structure of the employed model is illustrated in Fig. 7.15.

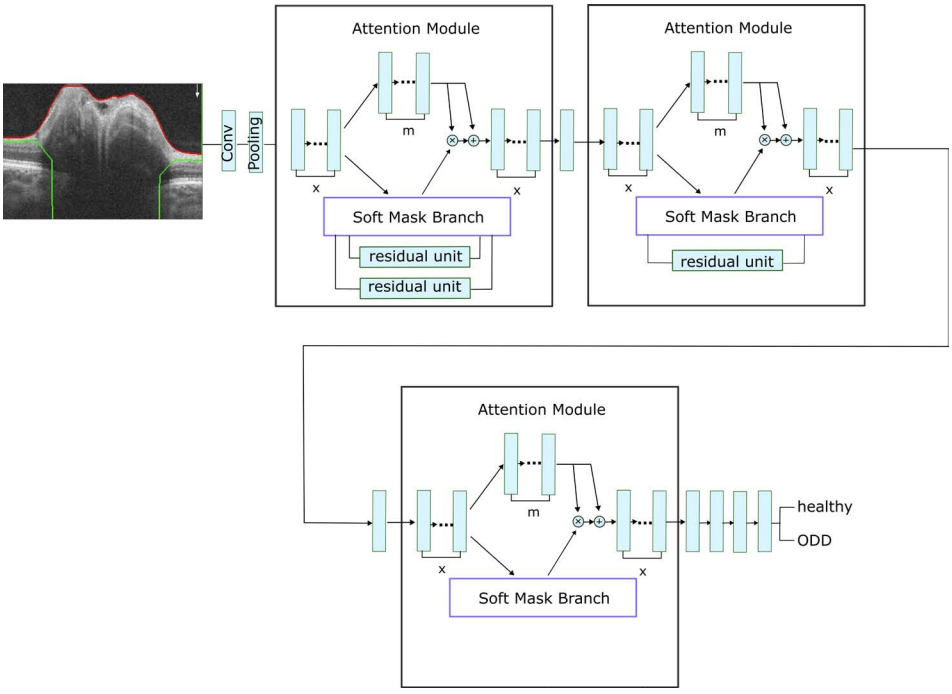


Fig. 7.15. The structure of the proposed classification model consists of Convolutional Layer, Pooling Layer, three following AM and Convolutional Layers, Fully Connected Layer activated by Softmax function.

Three hyperparameters are used to shape the AM module: x, n and m. The first represents the number of residual units that preprocess the input before splitting into the trunk and mask branches. m denotes the number of residual units within the trunk branch, n refers to the number of residual units placed between the mask branch and adjacent pooling layers. In these empirical studies, the following parameter configuration is used: $x = 1$, $n = 1$, and $m = 2$. Notably, the number of channels in the soft mask residual unit matches that of the corresponding trunk branches.

7.3.1. Attention Learning

In each AM, every trunk branch is paired with its corresponding mask branch, each tasked with developing attention patterns specific to its unique features. In the images being analysed, a mask is utilized to diminish the influence of irrelevant background elements, while another mask enhances the significant components. Additionally, the progressively accumulative design of the stacked network structure allows for a gradual refinement of attention mechanisms tailored to complex images.

However, adopting a straightforward stacking method for Attention Modules leads to a noticeable decline in performance. This deterioration can be attributed to several factors. First, the repeated dot product operations involving masks that range from zero to one cause a degradation of feature values in the deeper layers. Second, the use of soft masks can disrupt desirable characteristics of the trunk branch, such as the consistent mapping provided by Residual Units. To address this issue, a mask selection method based on Eq. (7.8) has been proposed [143]:

$$H_{\mathrm{i,c}}(z) = (1 + M_{\mathrm{i,c}}(z)) * F_{\mathrm{i,c}}(z). \tag{7.8}$$

The key distinction between the proposed mask selection method and those found in other studies [143] is that $F_{i,c}(z)$ represents the features generated by deep convolutional networks, rather than an approximation of the residual function. In both branches, the masks act as selectors that enhance valuable features while suppressing noise from the trunk function. Consequently, $M(z)$ varies within the range of [0,1] and, as $M(z)$ approaches *0*, $H(z)$ approximates the original features $F(z)$. This entire process is illustrated in Fig. 7.14.

Furthermore, the stacking of Attention Modules reinforces the learning of attention mechanisms through its progressive structure. This approach preserves the beneficial properties of the initial features while enabling them to bypass the soft mask branch and directly propagate to higher layers, thereby reducing the influence of the mask branch on feature selection. This cumulative arrangement of stacked attention modules facilitates a step-by-step enhancement of feature maps [144].

7.3.2. Residual Attention Network Evaluation

During the first experiment, the test set was randomly selected from the entire dataset and constituted 20% of the available images, while the remaining 80% were used for training. All images have been resized to *352×352*. Normalization of the RGB space in the range [0,1] is used. The network is trained using stochastic gradient descent with a momentum value equals to 0.9. An initial learning rate is set to 0.1. This value is divided by 10 every 10 iterations. The iteration limit is defined at 100. The entire structure of the used network is presented in Table 7.10 [115], while obtained results in Table 7.11, 7.12 and Fig. 7.18.

Table 7.10. Detailed structures of used RAN model.

Layer	Output size
Convolution 1	*176×176*
Max pooling	*88×88*
Residual Unit	*88×88*
Attention Model	*88×88*
Residual Unit	*44×44*
Attention Model	*44×44*
Residual Unit	*22×22*
Attention Model	*22×22*
Residual Unit	*11×11*
Average pooling	*1×1*
Fully Connected	*2*

Table 7.11. The obtained results for RP and AVL datasets classification by custom RAN. All values in %.

Dataset	Class	Precision	Recall	F1-score	Specificity
RP dataset	RP	95.80	88.35	92.00	95.06
	CORD	64.71	88.00	74.58	92.45
	HEALTHY	89.09	87.50	88.29	95.31
AVL dataset	AVL	64.29	94.74	76.54	94.25
	DRUSEN	96.47	94.25	95.38	97.17
	HEALTHY	95.00	87.36	91.00	96.23

Table 7.12. Ablation study for RAN model (Mean values in %).

Dataset	Retinitis Pigmentosa			AVL		
Model	Precision	Recall	F1-score	Precision	Recall	F1-score
RAN	83.20	87.95	84.96	85.25	92.12	87.64
– AM	80.60	84.10	81.10	82.00	88.83	84.21
– Mask	81.90	86.15	82.90	83.73	90.72	85.81
– Trunk	79.30	82.40	80.50	82.10	88.80	84.60
Shallower mask	82.74	86,80	83,66	84,61	91,34	86,62

Compared to the baseline ResNet-18, the proposed RAN increases the number of parameters by about 26.6% and the computational overhead by ~7.8%. This translates into a moderate increase in response time: +3 ms on the RTX 4090 and +50 ms on a single CPU core (Table 7.13). Analysis of the cost distribution (Table 7.14) shows that attention modules (AMs) dominate the computational overhead: *trunk* and *mask* accounts for about 81% GFLOPs in total, with the *44×44* and *22×22* stages accounting for ~63% of the total cost. The contribution of early processing and simple residual units is relatively small (~5% and ~14%, respectively). In practice, this means that the main cost for quality improvement comes from *mask/trunk* mechanisms operating at medium resolutions, where attention brings the greatest representational gain while still maintaining an acceptable computational overhead.

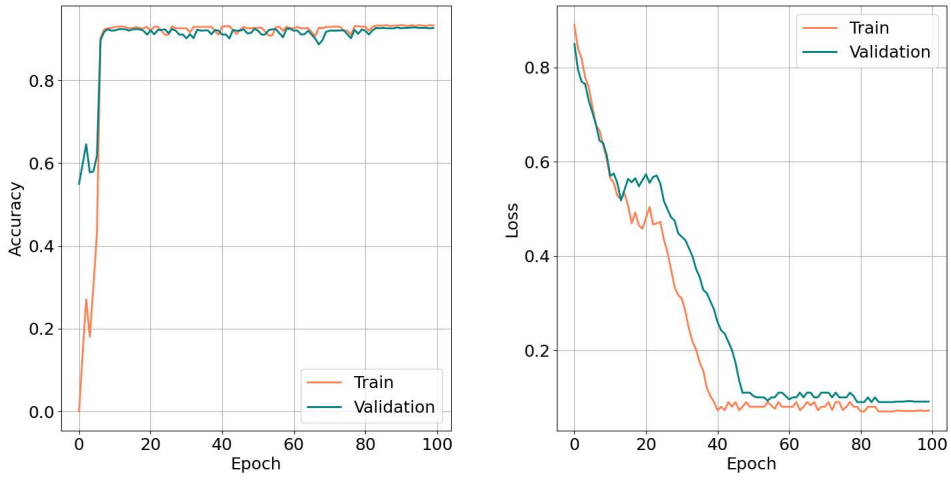


Fig. 7.16. Training and validation accuracies (left) and loss (right) obtained for RP dataset. Rapid convergence in the initial epochs indicates effective extraction of disease-relevant features, while the close alignment between training and validation accuracy curves demonstrates good generalization performance.

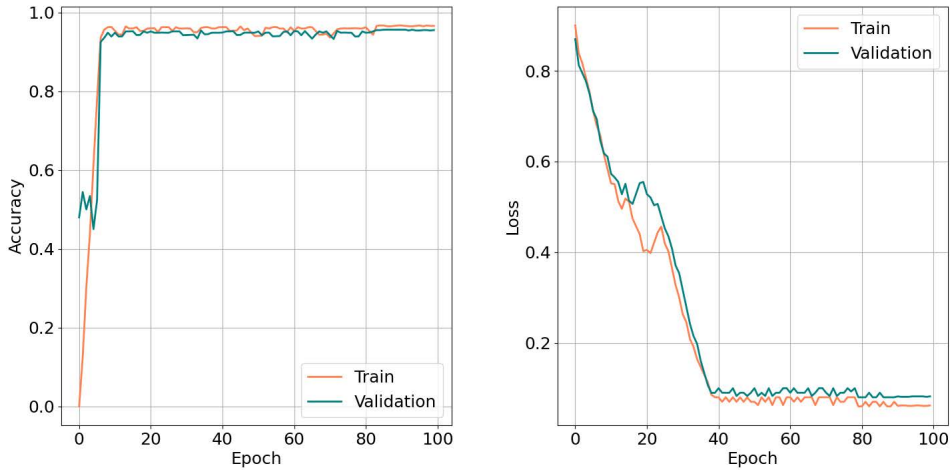


Fig. 7.17. Training and validation accuracies (left) and loss (right) obtained for AVL dataset. Also, in case of the AVL set, the model achieved fast convergence and a low error rate, and therefore good generalization quality.

Table 7.13. Comparison of inference complexity and speed for the baseline ResNet-18 and RAN variants. Timing measurements for RTX 4090 and single-core Ryzen 7950X.

Model	Params [M]	GFLOPs	RTX 4090 [ms]	Ryzen 7950X [ms]
Baseline ResNet-18	11.7	24.4	16	175
RAN	14.8	26.3	19	225

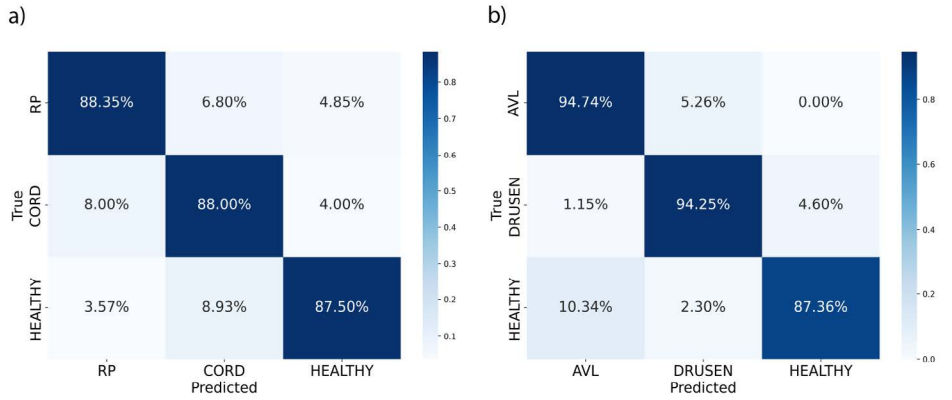


Fig. 7.18. Confusion matrices obtained for RAN model for a) RP dataset b) AVL dataset. For the RP dataset, the model achieves high classification accuracy across all classes, with the most frequent confusion observed between RP and CORD cases (6.80% and 8.00%), reflecting overlapping structural characteristics of inherited retinal degenerations. For the AVL dataset, the model demonstrates very strong class separability, particularly for AVL and DRUSEN classes, with limited confusion between DRUSEN and healthy samples (4.60%).

Table 7.14. FLOPs for a single image *352×352*. Values in billions of floating-point operations (GFLOPs) AM – attention module, RU – residual elements, BU/TD – Bottom-Up / Top-Down: bidirectional path in the mask branch – first spatial compression (BU, down), then reconstruction (TD, up) with intermediate level merges.

Layer no.	Layer	Configuration/ size	FLOPs [G]	Part
Preprocessing				
1.	*Conv 7×7*	(3→64→32, *176×176*)	1.36	5.2%
Stage *88×88*				
2.	RU (pre)	(64→64)	0.78	2.9%
3.	AM (trunk)	(t = 2 RU, 64→64)	2.72	10.3%
4.	AM (mask)	BU/TD *n = 2*	1.94	7.4%
Stage *44×44*				
5.	RU (down)	(64→128, stride 2)	0.97	3.7%
6.	AM (trunk)	(t = 2 RU, 128→128)	4.65	17.7%
7.	AM (mask)	BU/TD *n = 2*	3.49	13.3%
Stage *22×22*				
8.	RU (down)	(128→256, stride 2)	1.09	4.1%
9.	AM (trunk)	(t = 2 RU, 256→256)	5.04	19.2%
10.	AM (mask)	()	3.49	13.3%
Stage *11×11*				
8.	RU (down)	(128→512, stride 2)	0.78	2.9%
9.	Global average pooling	*512×11×11* →512)	-	-
Sequential section and classification				
10.	FC	(*512* →3)	<0.01	<<0.01%
Total			**26.3**	**100.0%**

The learning curves for the RP dataset (Fig. 7.16) indicate a rapid increase in classification accuracy during the initial training epochs, followed by a gradual levelling off at a high level for both the training and validation sets. The parallel decrease in the loss function aligns well with these improvements, suggesting that the model is effectively capturing the retinal features relevant to distinguishing RP, CORD, and HEALTHY classes. The relatively small gap between the training and validation curves points to limited overfitting, which is further supported by the balanced nature of the precision, recall and F1-score values for all three classes. RP and HEALTHY exhibit particularly high specificity, indicating the model rarely confuses these classes with others. CORD, although less numerous, is still recognized with strong recall, demonstrating the model's ability to detect subtle patterns even in sparsely represented conditions.

For the AVL dataset (Fig. 7.17), a similar trend is observed in the learning curves, though the model appears to stabilize at an even higher overall accuracy after the first few epochs. The consistently low loss function values for both training and validation highlight the robustness of the learned features in separating AVL, DRUSEN and HEALTHY. AVL, despite being the rarest class, is detected at a satisfactory recall level, and the model maintains high precision and specificity for DRUSEN and HEALTHY. The small differences between the training and validation metrics indicate that the network generalizes well and is not overly tailored to the training data. Overall, these results confirm that the proposed approach effectively identifies diverse retinal pathologies in OCT images, maintaining strong performance even in classes with comparatively fewer samples.

7.4. Chapter Summary

This chapter introduces three novel solutions for classifying OCT images, each designed to improve diagnostic accuracy in distinguishing complex retinal conditions such as retinitis pigmentosa and acquired vitelliform lesions. The first proposed model, Deep CNN-GRU, combines convolutions and gated recurrent units to capture both spatial and sequential features of OCT scans. As demonstrated by tests on the RP dataset, it achieves an average accuracy of around 85–90% and specificities above 90% for two out of three classes. On the AVL dataset, its accuracy consistently surpasses 90%, with recall for the sparse AVL class reaching nearly 98% in some trials.

The second approach, Convolutional Gated Recurrent Units U-Net, adapts classic U-Net segmentation architecture into a classification framework by adding dense connections and bidirectional GRU modules. This hybrid network attains high performance on both datasets, exceeding 88% F1-score for classes with more samples (e.g., DRUSEN and HEALTHY in the AVL set) and retaining decent recall, above 80%, even for classes with limited data like CORD or AVL.

Finally, the Residual Attention Network integrates attention mechanisms with residual learning to focus more precisely on key regions of the retina. By deploying a trunk-and-mask strategy, it achieves specificity levels above 95% in several classes, indicating that it rarely misclassifies RP as HEALTHY or vice versa. Similarly, in the AVL dataset, the RAN demonstrates strong performance for DRUSEN detection, with precision of about 96% and recall exceeding 94%.

All three models show smooth and rapidly convergent learning curves, generally stabilizing their accuracy within the first 20 to 30 epochs. Confusion matrices confirm that most classification errors arise between relatively similar pathologies, yet even the rarest classes are classified with a recall typically above 80%. These findings highlight the promise of combining deep convolutional features, recurrent modules, and attention-based strategies for improving early detection and precise recognition of rare eye diseases from OCT scans.

Although all three proposed architectures (CNN–GRU, GRU U-Net, and the dedicated Residual Attention Network) achieve strong classification performance, they have been introduced with distinct and complementary motivations rather than as direct replacements for one another.

The CNN–GRU architecture serves as a baseline hybrid model that combines convolutional feature extraction with recurrent modelling to capture contextual dependencies within OCT images. While effective, this architecture operates primarily on global feature representations and does not explicitly model spatial localization of pathological regions.

The GRU U-Net architecture is introduced to address this limitation by incorporating an encoder–decoder structure with skip connections, enabling the preservation of fine-grained spatial information. This design improves sensitivity to localized retinal abnormalities and allows the model to better capture structural variations within retinal layers, which are particularly important for certain eye diseases.

The dedicated Residual Attention Network is motivated by the need for enhanced interpretability and adaptive feature weighting. By explicitly integrating attention mechanisms within a residual framework, the RAN selectively emphasizes diagnostically relevant regions while suppressing irrelevant background information. This architecture directly addresses the limitations of uniform feature processing present in the previous models and improves robustness, especially under conditions of class imbalance and subtle pathological differences.

8. Kronecker Convolution

At the beginning of this chapter, I would like to explain that a Deep Kronecker Convolutional Network (DKCN) essentially refers to any convolutional network that uses the Kronecker product instead of the standard convolution operation.

Kronecker convolutions represent a sophisticated approach to convolution operations within CNNs, aiming to optimize computational efficiency and enhance feature extraction capabilities. This method decomposes the conventional convolution operation into a series of smaller, more manageable matrix multiplications, significantly altering the way convolutional layers process input data.

In traditional convolution operations, filters or kernels move across the input image, performing element-wise multiplications and summing the results to generate a feature map. While effective, this process can become computationally intensive, especially as the input image size and the number of filters increase.

Kronecker convolutions tackle this challenge by leveraging the mathematical properties of Kronecker products. This product of two matrices results in a block matrix, where each element of the first matrix is multiplied by the entire second matrix. In the context of CNNs, this means that a single convolution operation can be decomposed into multiple smaller operations using these block matrices. Kronecker convolutions are characterized by several important elements from the point of view of eye image processing:

- Decomposition of kernels.
- Efficient computation.
- Parameter sharing.
- Flexibility and adaptability.

The convolution kernel is decomposed into smaller sub-kernels. This decomposition leverages the Kronecker product to represent the original kernel as a series of smaller matrices, reducing the overall number of multiplications required.

By breaking down the kernel into smaller sub-kernels, a Kronecker convolution reduces the computational load. Each sub-kernel performs a smaller convolution operation, and these results are combined to form the final output. This method significantly decreases the number of operations compared to traditional convolutions, making it more efficient, especially for larger kernels and high-dimensional data.

Kronecker convolutions inherently promote parameter sharing across different parts of the network. Since the same sub-kernels are used multiple times in generating the final convolution result, the network can learn more generalized features from the data, potentially improving its ability to generalize from the training data to unseen samples.

The Kronecker convolution approach is highly flexible, allowing it to be adapted to various network architectures and applications. It can be fine-tuned by adjusting the decomposition level of the kernels, offering a trade-off between computational efficiency and the richness of the learned features [145].

The Kronecker convolution of two polynomials *f(x), g(x)* can be represented by the Kronecker product of the matrices associated with these polynomials. If $f(x) = \sum_{i=0}^{n} f_i x^i$ and $f(x) = \sum_{i=0}^{n} f_i x^i$ then the resulting polynomial is by Eq. (8.1) as follows [146]:

$$f(x) \otimes g(x) = \sum_{i=0}^{n} \sum_{j=0}^{m} (f_i g_j)(x^{i+j}). \tag{8.1}$$

This corresponds to standard polynomial multiplication, which can be written in matrix form as the Kronecker product of coefficient vectors.

In a typical CNN, the output of a convolutional layer with an input tensor X and a weight tensor W is represented in Eq. (8.2) as [147]:

$$out = X * W + b, \tag{8.2}$$

where '*' indicates the standard convolution operation and b denotes the bias term.

In a Kronecker Convolutional Neural Network (KCNN), the convolution kernel W is approximated by the sum or Kronecker product of smaller tensors, which reduces the number of parameters and computational operations. The convolution operation with such a decomposed kernel looks as follows [147]:

$$out = X * (W_1 \otimes W_2) + b, \tag{8.3}$$

where is the standard convolution, and W_1, W_2 are smaller subkernels.

The Kronecker product of two matrices A and B as defined in Eq. (8.4), results in a matrix whose dimensions are the product of the dimensions of the original matrices [147].

$$A \otimes B = [a(i, j) * B]\, i, j. \tag{8.4}$$

In this expression, *a(i, j)* denotes the element in the *i*-th row and *j*-th column of matrix A.

In this approach, many combinations of small Kronecker products are able to replace large weight matrices, reducing computational complexity while maintaining model performance. The computational efficiency of Kronecker convolution allows for the utilization of larger feature maps, enhancing classification accuracy. When the number of external product parameters is equal to the number of Kronecker product parameters corresponding to image pixel values, the Kronecker product yields less reconstruction error compared to the outer product [148].

For weight matrices and tensors in convolutional networks, this method can generate approximate models that run faster and have fewer parameters without significant accuracy loss. The primary difference between Kronecker convolution and standard convolution is illustrated in Fig. 8.1. Kronecker convolution uses kernels decomposed by the Kronecker product of smaller subkernels, reducing computational complexity. Standard convolution, on the other hand, performs full element-wise multiplication for sliding filter windows over the entire input image. KCNN significantly reduces the number of operations due to this decomposition. In contrast, standard convolution multiplies each subset of pixels with the filter, generating new samples on the feature map as shown in Fig. 8.1 [148].

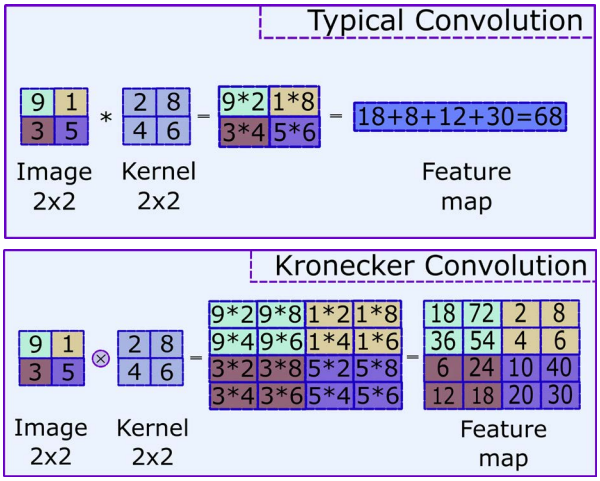


Fig. 8.1. A visual representation of the Kronecker and typical convolution operations on a *2×2* image. In the typical convolution (top), the output is computed by element-wise multiplication between the image patch and the kernel followed by summation, producing a single response value for the receptive field. In the Kronecker convolution (bottom), the convolution kernel is factorized into smaller matrices and expanded via the Kronecker product, yielding a structured filter that generates a feature map through organized element-wise interactions within the receptive field.

When the stride value is minimal, the resulting feature map size is large due to significant filter overlap, leading to more shared features in the outputs. Conversely, increasing the stride size reduces parameter sharing between filters and results in a smaller feature map. This increase in stride effectively downsamples the image, potentially obscuring lower-level details [148]. In the proposed technique, extracting deep features without filter overlap does not provide substantial advantages in classification.

The proposed models, in this chapter, integrate several layers, including an image input layer, a Kronecker convolutional layer (Conv2D), a *ReLU* activation layer, a max pooling layer, a flatten layer, a dropout layer and dense layers. CNNs have proven effective in image classification tasks [122], [131], [141]. However, they require large amounts of data for effective training and often involve millions of unknown parameters, making them computationally intensive and less interpretable.

Moreover, CNNs are considered black boxes, making them difficult to understand and limiting their efficiency in handling medical imaging data. Medical imaging data differs from standard images in two main aspects: (i) fewer images are typically available for training, and (ii) it is challenging for the designed model to interpret and classify the output effectively.

To address these issues, a new DKCN for the classification of rare eye diseases is proposed. DKCN can accumulate substantial knowledge from a small set of training images, offers good interpretability and provides better performance than conventional CNNs. The architecture of DKCN is primarily based on Kronecker products, which indirectly preserve the latent piecewise smooth properties of coefficients [149].

DKCN enables the identification of important regions within the image, enhancing model interpretability. In this study, several popular architectures within the DKCN framework, including ResNet, DenseNet and Inception are adapted. Although DKCN is based on a Kronecker structure, it can also be expressed in convolutional form and, unlike traditional CNNs, it has no overlaps between convolutions. This architecture provides improved model interpretability while reducing dimensionality as much as possible [149].

In standard deep learning architectures, overlapped convolutions are commonly used when the stride is equal to one. These overlapped convolutions lead to redundant information because the same regions of the image are processed multiple times, increasing computational complexity without adding new value. To eliminate this redundancy, the proposed network employs the Kronecker product to remove overlapping by multiplying the filter with individual pixels of the image, as illustrated in Fig. 8.1.

The *L*-layer Fully Convolutional Neural Network (FCNN) based on these Kronecker products is reconstructed. To demonstrate this operation, we assume A is an image of size 128×128 and the convolution filter is of size 3×3.

In this approach, the convolution operation is applied to every single element in a manner similar to a Kronecker product [150]. The resulting Kronecker-convoluted matrix image has the size $128 \times 3 \times 128 \times 3$.

After performing the Kronecker convolution in the first step, max-pooling is employed for down-sampling, which reduces the spatial dimensions of the input image by selecting the maximum value within a window of pixels. This operation enhances computational efficiency by reducing the size of the feature maps.

Subsequently, another Kronecker convolution operation is applied to the output of the max-pooling layer to extract more complex features from the input image. Following this, a dense layer is introduced, where every neuron is connected to each neuron in the preceding layer. This layer aims to map the features extracted from the Kronecker convolutional layers to a set of output classes.

A second max-pooling layer is then applied to the output of the dense layer, further reducing the dimensionality of the input. The final Kronecker

convolution operation is applied to the output of this second max-pooling layer, enabling the extraction of even more complex features.

An additional max-pooling layer is applied to the output of the third Kronecker convolution layer, further decreasing the input's dimensionality. The output of this third max-pooling layer is flattened into a vector to prepare it for the final classification layer.

To prevent overfitting, the model employs the regularization technique known as dropout. During training, random neurons are deactivated, forcing the remaining neurons to learn more robust features. The final dense layer maps the output from the previous layers to a set of output classes, using a Softmax activation function to produce class probabilities. The entire proposed methodology is presented in Algorithm 7.1 (pseudocode).

Algorithm 8.1. Deep Kronecker Network (DKN) Pipeline for Rare Eye Disease Classification.

1: Step 1: Collect and preprocess data
Collect a dataset of AVL and RP images and preprocess them by resizing them to a standard size and converting them to grayscale.
2: Step 2: Split the data
Split the dataset into training, validation and testing sets.
3: Step 3: Define the factors
Define the factor matrices for each layer of the DKN. These factor matrices will be used to define the weights for the horizontal and vertical convolutions in each layer.
4: Step 4: Define the model
Define the architecture of the DKN using the factor matrices from Step 3. Typically, the network consists of several Kronecker Conv2D layers, followed by a flattened layer and one or more dense layers.
5: Step 5: Design the model
Choose the appropriate loss function (e.g., categorical cross-entropy), optimizer, and metrics. This step is crucial for multi-class classification tasks such as classifying different types of eye diseases.
6: Step 6: Train the model
Train the DKN on the training set, adjusting its weights to minimize the chosen loss function.
7: Step 7: Evaluate the model
Assess the performance of the DKN on the validation set, computing metrics such as accuracy, precision, recall and F1-score.

```
8:  Step 8: Test the model
    Test the final DKN on the testing set and report the same metrics
    (accuracy, precision, recall, F1-score, specificity) for an unbi-
    ased performance estimate.
9:  Step 9: Fine-tune the model
    Refine the DKN by adjusting hyperparameters or modifying the archi-
    tecture, guided by the validation and testing results.
```

In the context of eye disease classification using OCT images, Kronecker convolutions enhance the ability of CNNs to detect subtle and complex patterns within high-resolution retinal scans. By efficiently handling the large amounts of data and intricate details present in OCT images, Kronecker convolutions contribute to more accurate and faster diagnostic models.

8.1. Kronecker Convolution Evaluation

This section examines the performance of Kronecker convolution when applied to a range of deep neural network architectures for rare eye disease classification, including those presented in Chapter 6: VGG16, ResNet-18, InceptionV3 and DenseNet121, as well as custom models introduced in Chapter 7: CNN-GRU, GRU U-Net and a Residual Attention Network. In each model, standard convolution is replaced by the Kronecker variant, allowing for more efficient parameter usage and improved capture of subtle retinal changes. In CNN architectures standard convolutional kernels are reformulated using Kronecker products, enabling parameter-efficient representations while preserving the original network topology.

8.1.1. Common CNN models evaluation

Tables 8.1–8.4 and Fig. 8.2–8.7 present detailed results for typical CNN models with Kronecker-based convolutions, focusing on key metrics such as precision, recall, F1-score, and specificity. Notable gains are observed for RP and AVL classes. In addition, the learning curves underscore faster convergence since the Kronecker models require fewer epochs to achieve similar or higher accuracy than conventional CNN counterparts. This heightened efficiency is particularly valuable given the limited number of specialized images typical in rare disease studies.

Table 8.1. The obtained results for RP and AVL datasets classification by VGG16 with Kronecker convolution. All values in %.

Dataset	Class	Precision	Recall	F1-score	Specificity
RP dataset	RP	96.34	76.60	85.28	96.30
	CORD	64.52	80.00	71.48	93.08
	HEALTHY	73.24	92.86	81.83	85.16
AVL dataset	AVL	53.33	84.21	65.28	91.95
	DRUSEN	89.41	87.36	88.38	91.51
	HEALTHY	88.46	79.31	83.64	91.51

Table 8.2. The obtained results for RP and AVL datasets classification by ResNet-18 with Kronecker convolution. All values in %.

Dataset	Class	Precision	Recall	F1-score	Specificity
RP dataset	RP	91.75	86.41	89.07	90.12
	CORD	77.78	84.00	80.67	96.23
	HEALTHY	85.00	91.07	87.94	92.97
AVL dataset	AVL	60.71	89.47	72.43	93.68
	DRUSEN	89.16	85.06	87.02	91.51
	HEALTHY	90.24	85.06	87.59	92.45

Table 8.3. The obtained results for RP and AVL datasets classification by InceptionV3 with Kronecker convolution. All values in %.

Dataset	Class	Precision	Recall	F1-score	Specificity
RP dataset	RP	96.00	88.00	94.59	95.06
	CORD	66.67	88.00	75.86	93.08
	HEALTHY	92.16	83.93	87.84	96.88
AVL dataset	AVL	60.00	94.74	73.44	93.23
	DRUSEN	95.00	87.36	91.02	96.23
	HEALTHY	91.57	87.36	89.54	93.40

Table 8.4. The obtained results for RP and AVL datasets classification by DenseNet121 with Kronecker convolution. All values in %.

Dataset	Class	Precision	Recall	F1-score	Specificity
RP dataset	RP	97.87	89.32	93.29	97.53
	CORD	77.33	88.00	80.05	94.97
	HEALTHY	86.67	92.86	89.66	93.75
AVL dataset	AVL	60.71	89.47	72.43	93.68
	DRUSEN	93.59	83.91	88.45	95.28
	HEALTHY	93.10	93.10	93.10	94.34

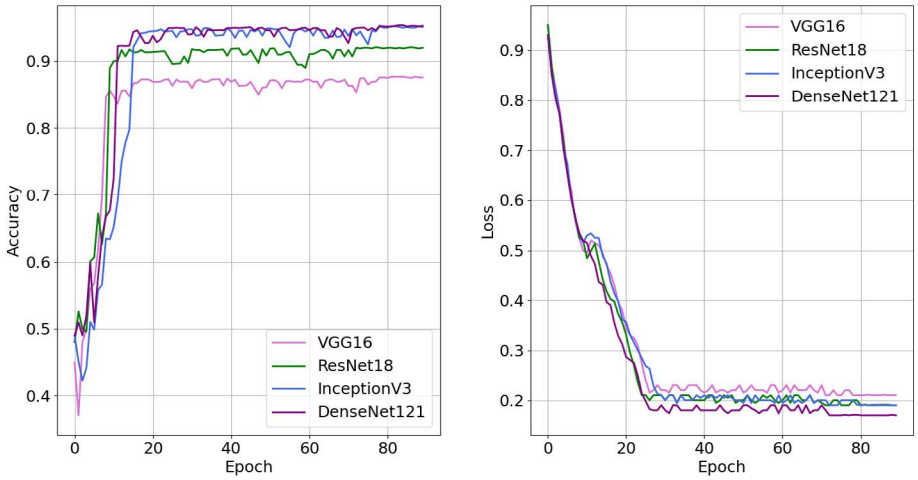


Fig. 8.2. Validation accuracies (left) and loss (right) obtained using RP dataset by VGG16, ResNe-t18, InceptionV3 and DenseNet121 with Kronecker convolution. Incorporation of Kronecker convolution leads to faster convergence, higher validation accuracy, and lower validation loss across all evaluated models. The most pronounced improvement is observed for deeper architectures, particularly DenseNet121 and InceptionV3 (see Fig. 6.9), indicating that the structured parameterization of Kronecker filters enhances feature extraction.

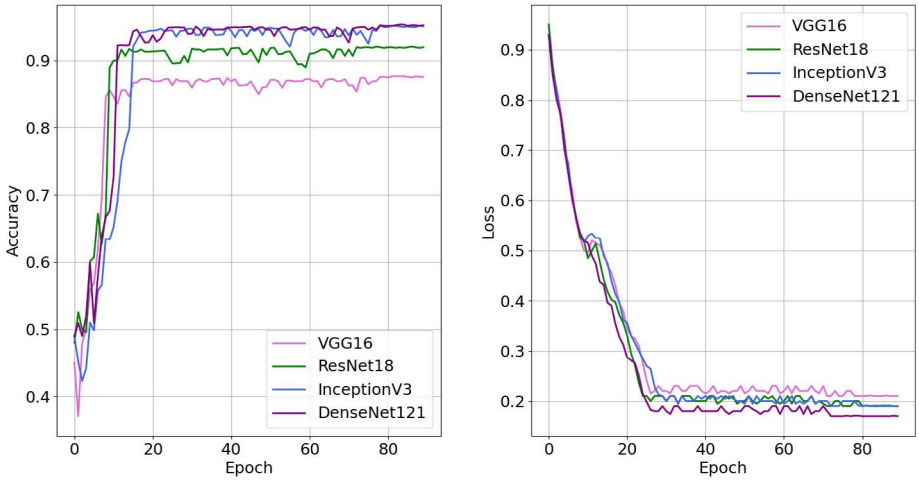


Fig. 8.3. Validation accuracies (left) and loss (right) obtained using AVL dataset by VGG16, ResNet-18, InceptionV3 and DenseNet121 with Kronecker convolution. Application of Kronecker convolution results in consistently higher validation accuracy and lower validation loss across all evaluated models (see Fig. 6.10).

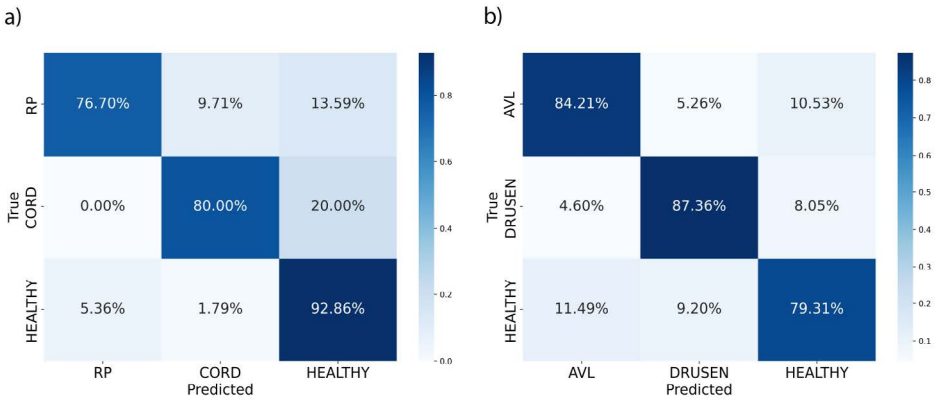


Fig. 8.4. Testing accuracies obtained for VGG16 model with Kronecker convolution for a) RP dataset b) AVL dataset. Compared with the corresponding baseline results shown in Fig. 6.11 the incorporation of Kronecker convolution leads to improved class-wise recognition performance and a more balanced confusion distribution across disease categories.

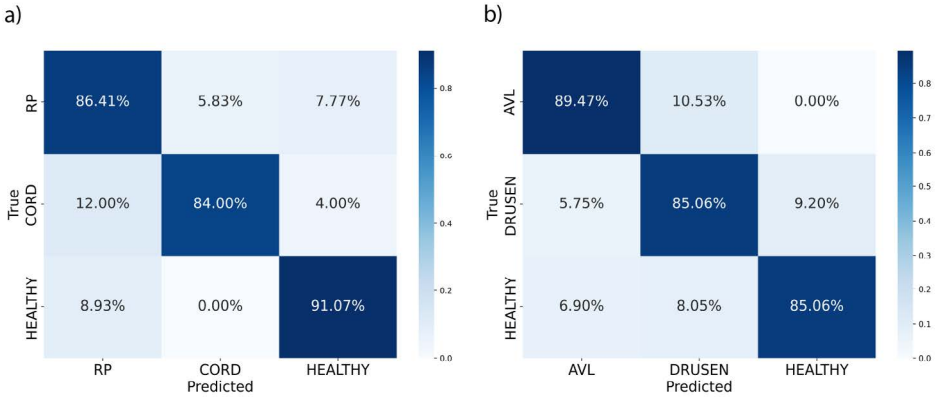


Fig. 8.5. Testing accuracies obtained for ResNet-18 model with Kronecker convolution for a) RP dataset b) AVL dataset. Compared to the corresponding baseline results (see Fig. 6.12), the use of Kronecker convolution leads to improved classification performance, in particular by increasing the accuracy of disease class recognition and reducing mutual confusion between classes.

The application of Kronecker convolution in deep networks has led to clear benefits in diagnosing rare eye diseases: retinitis pigmentosa and acquired vitelliform lesions. As it is evidenced in the results, sensitivity is markedly better than that based on the typical convolution (see summaries in Tables 6.1–6.4) for these two classes. For example, in the InceptionV3 network, recall RP was obtained at the level of 88% with a concomitant increase in precision, which greatly minimizes the risk of missing the right diagnosis. A similar impact is noted regarding AVL set results, where, among others, recall for AVL in DenseNet121

reached 89.47%. Diagnosing subtle changes in the macula makes this clinically important. It further decreases the number of patients left undiagnosed or wrongly adjudicated as healthy. The results suggest that Kronecker convolution allows the networks to identify delicate patterns that develop within the retina of RP and AVL patients, without losing classification performance regarding the other classes (CORD, DRUSEN, HEALTHY).

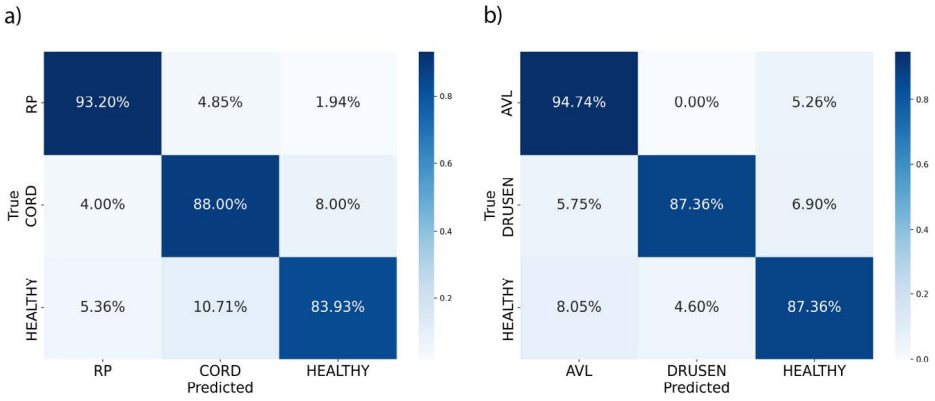


Fig. 8.6. Testing accuracies obtained for InceptionV3 model with Kronecker convolution for a) RP dataset b) AVL dataset. Application of Kronecker convolution leads to improved recognition of the RP and CORD classes in the RP set, as well as the AVL and DRUSEN classes in the AVL set, while increasing the accuracy of the HEALTHY class in both cases (see Fig. 6.13).

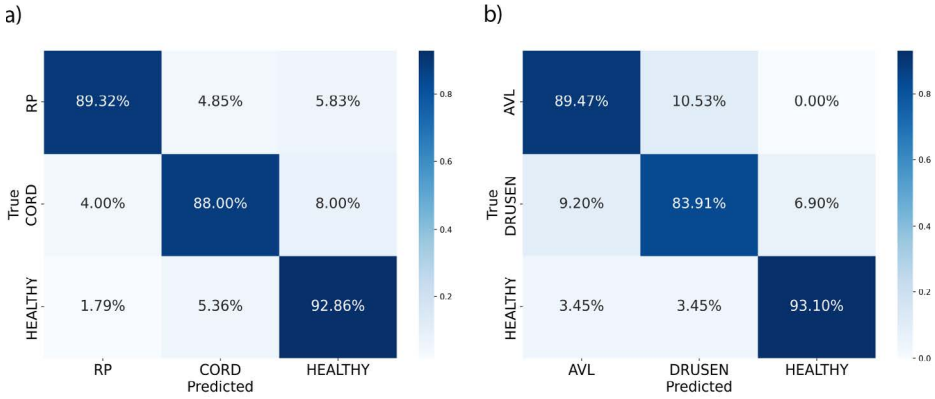


Fig. 8.7. Testing accuracies obtained for DenseNet121 model with Kronecker convolution for a) RP dataset b) AVL dataset. The introduction of Kronecker convolution improves classification performance mainly for the RP, AVL, and HEALTHY classes, as reflected by higher diagonal values and reduced inter-class confusion, while the recognition of the CORD/DRUSEN class remains comparable (see Fig. 6.14).

Learning curves analysed not only indicate that there was stability improvement in the final phases of training but also that epochs required to reach satisfactory convergence of the process reduced from 100, in the typical convolution, to 90 in networks with Kronecker convolution.

8.1.2. Custom CNN models evaluation

The impact of Kronecker convolution on the quality of learning and classification of rare diseases in the models discussed in Chapter 7 was also examined. Tables 8.5–8.7 and Figs. 8.8–8.12 provide detailed results for custom CNN models utilizing Kronecker-based convolutions, highlighting key metrics including precision, recall, F1-score and specificity.

Table 8.5. The obtained results for RP and AVL datasets classification by CNN-GRU model with Kronecker convolution. All values in %.

Dataset	Class	Precision	Recall	F1-score	Specificity
RP dataset	RP	98.98	94.17	96.54	98.77
	CORD	76.67	92.00	83.64	95.60
	HEALTHY	92.86	92.86	92.86	96.88
AVL dataset	AVL	90.00	94.74	92.33	98.85
	DRUSEN	95.29	93.10	94.19	96.23
	HEALTHY	93.18	94.25	93.69	94.34

Table 8.6. The obtained results for RP and AVL datasets classification by GRU U-Net model with Kronecker convolution. All values in %.

Dataset	Class	Precision	Recall	F1-score	Specificity
RP dataset	RP	98.99	95.15	96.60	98.77
	CORD	85.71	96.00	90.50	97.48
	HEALTHY	91.23	92.86	92.07	96.09
AVL dataset	AVL	78.26	94.74	85.62	97.13
	DRUSEN	96.34	90.80	93.59	97.17
	HEALTHY	95.45	96.55	95.94	96.23

Table 8.7. The obtained results for RP and AVL datasets classification by RAN model with Kronecker convolution. All values in %.

Dataset	Class	Precision	Recall	F1-score	Specificity
RP dataset	RP	97.03	95.15	96.06	96.29
	CORD	85.71	96.00	90.50	97.48
	HEALTHY	96.36	94.64	95.30	98.44
AVL dataset	AVL	81.82	94.74	87.70	97.70
	DRUSEN	97.62	94.25	95.90	98.11
	HEALTHY	96.55	96.55	96.55	97.17

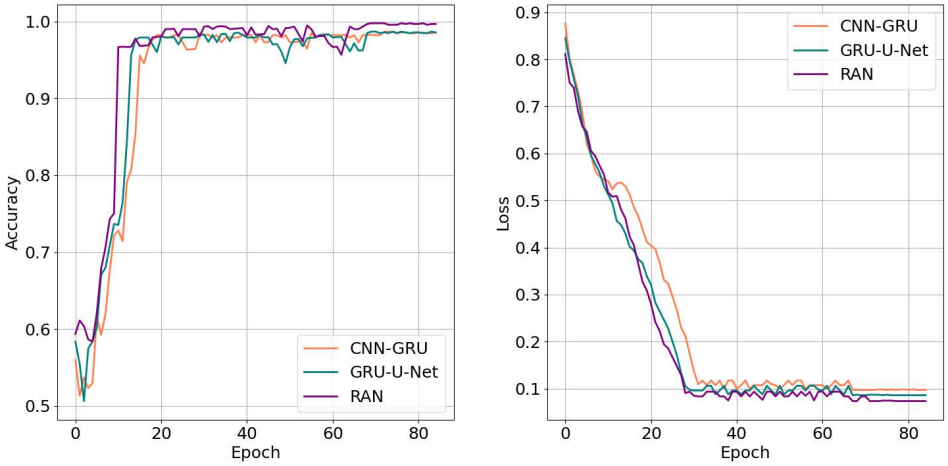


Fig. 8.8. Validation accuracies (left) and loss (right) obtained using RP dataset by CNN-GRU, GRU U-Net and RAN with Kronecker convolution. Compared to the corresponding baseline results (see Fig. 7.2, Fig. 7.8, and Fig. 7.16), models developed within this monograph exhibit faster convergence, reduced validation loss, and improved stability of validation accuracy, indicating a consistent performance gain across all considered architectures.

Precision reflects the proportion of positive predictions that truly belong to the targeted rare disease. High precision means fewer healthy patients are misclassified as sick. Recall indicates the model's ability to identify all actual cases of the disease. High recall means fewer sick patients go undetected. The F1-score combines both precision and recall to show the overall balance of these two metrics. Specificity measures the proportion of healthy cases correctly identified as healthy. In rare disease identification, recall is crucial for ensuring that patients with conditions like retinitis pigmentosa or acquired vitelliform lesions are not missed.

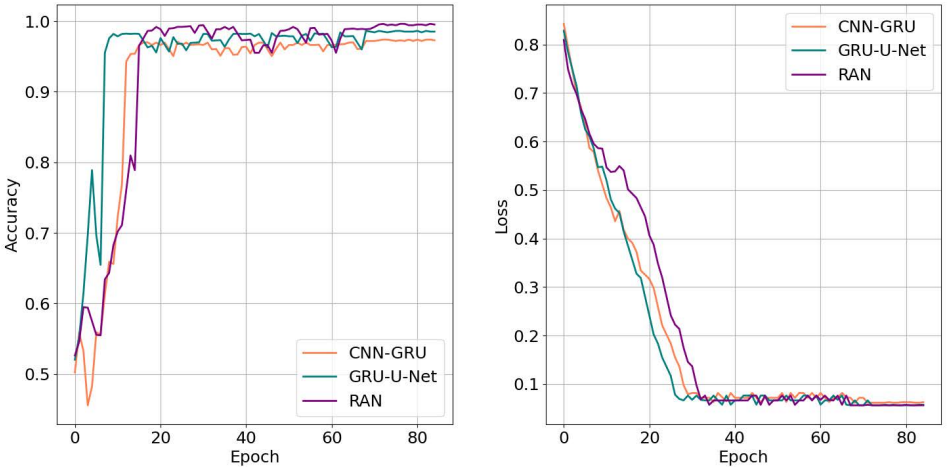


Fig. 8.9. Validation accuracies (left) and loss (right) obtained using AVL dataset by CNN-GRU, GRU U-Net and RAN with Kronecker convolution. All architectures reach high validation accuracy levels of approximately 94–96% after fewer than 15 epochs, while the validation loss rapidly decreases and stabilizes below 0.10. In comparison with the corresponding baseline models (see Fig. 7.3, Fig. 7.9, and Fig. 7.17), the Kronecker-based variants exhibit earlier convergence, reduced oscillations of validation accuracy, and consistently lower final loss values.

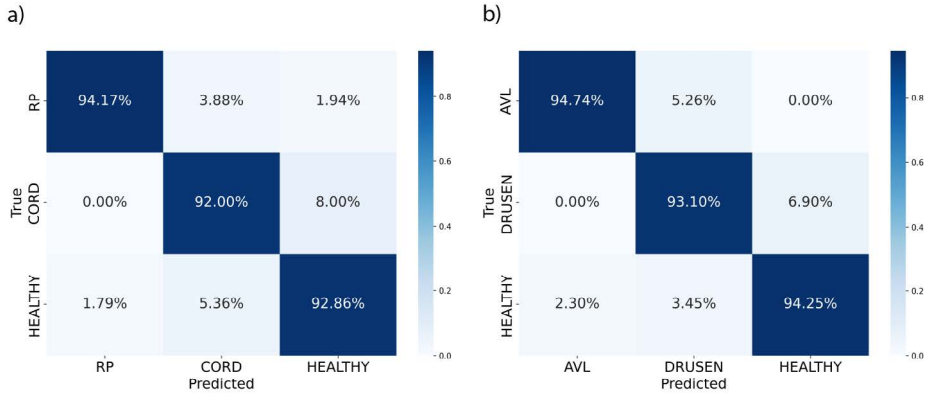


Fig. 8.10. Testing accuracies obtained for CNN-GRU model with Kronecker convolution for a) RP dataset b) AVL dataset. The application of Kronecker convolution leads to a clear improvement in class-wise recognition performance (see Fig. 7.4).

However, precision is also important for avoiding unnecessary false alarms. Tables 7.2, 7.3, and 7.5 show that CNN-GRU, GRU U-Net and RAN achieve solid recall values for these diseases, although precision sometimes fluctuates. The introduction of Kronecker convolution (Tables 8.5, 8.6 and 8.7) further improves these metrics. Models become more sensitive, ensuring they detect more rare

disease instances (higher recall), while maintaining or raising precision. As a result, the F1-scores increase, indicating a better trade-off between detecting sick patients and avoiding misdiagnoses. Specificity also remains high, limiting the risk of labelling healthy individuals incorrectly. These observations are consistent with the confusion matrices, which show fewer false negatives for rare classes. Using Kronecker convolution therefore supports reliable classification.

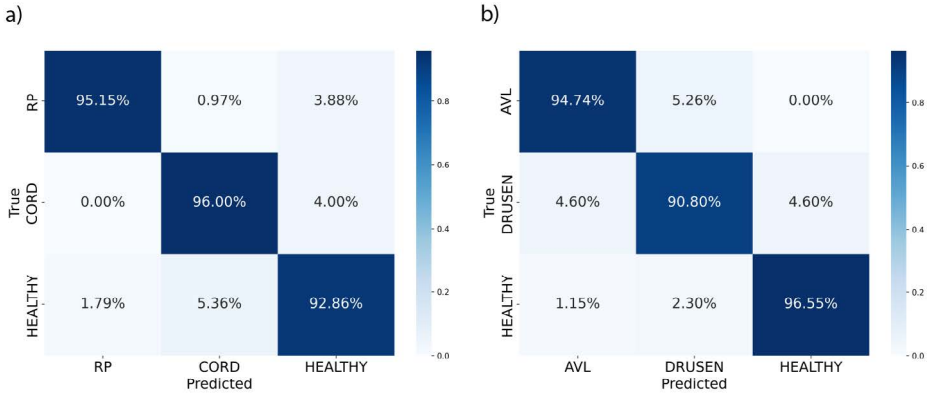


Fig. 8.11. Testing accuracies obtained for GRU U-Net model with Kronecker convolution for a) RP dataset b) AVL dataset. For the RP dataset, the Kronecker-based model increases correct recognition of all classes. For AVL dataset clear improvements are observed for DRUSEN and HEALTHY cases.

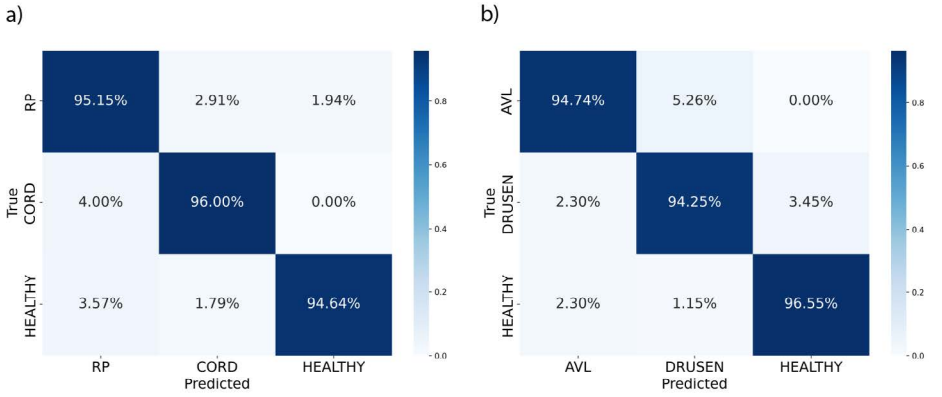


Fig. 8.12. Testing accuracies obtained for RAN model with Kronecker convolution for a) RP dataset b) AVL dataset. For the RP dataset, the accuracy for RP and CORD increases to 95.15% and 96.00%, respectively, with a simultaneous reduction of misclassification into the HEALTHY class. Similarly, for the AVL dataset, the correct recognition rates for DRUSEN and HEALTHY reach 94.25% and 96.55%.

8.2. Chapter Summary

Replacing the standard convolution operation in Convolutional Neural Networks with the Kronecker product offers several important benefits for classifying rare eye diseases such as Acquired Vitelliform Lesion and retinitis pigmentosa. Large convolutional kernels are decomposed into smaller, separable matrices, which significantly reduces the total number of parameters. This parameter efficiency lowers the risk of overfitting, especially in settings where data are limited and helps the model learn more generalizable features.

The Kronecker product also gives computational benefits by allowing faster training and inference. Smaller sub-kernels need less multiplication, so the network is running more efficiently. Lower memory requirements mean that input images can be bigger or deeper architectures without burdening the hardware resources that are available. These benefits are very important in ophthalmology because high-resolution retinal images need a lot of computational power. This method improves the process of feature extraction by representing complicated interactivity over all the space scales of the picture.

This approach enhances feature extraction by modelling complex interactions across the spatial dimensions of the image. It captures subtle and varied pathological structures in the retina, which is critical for detecting rare diseases. Compact representations help the network recognize small lesions or diffuse abnormalities that often appear in Acquired Vitelliform Lesion or retinitis pigmentosa.

The Kronecker-based method further demonstrates strong generalization when dealing with minimal datasets, a common situation in rare disease research. Its smaller parameter space demands fewer training samples and ensures resistance to variability in lesion size, shape or location. Such efficiency also facilitates transfer learning. Pre-trained models with fewer parameters can be adapted more easily to new or related tasks, reducing retraining time and effort.

In addition to practical improvements, the Kronecker product offers theoretical advantages for network interpretability and customization. Using the mathematical properties of the Kronecker product, it is possible to create architectures that reflect the specific structure of retinal images. The reduced redundancy that arises from eliminating overlapping convolutions helps focus the model's capacity on clinically relevant features.

By addressing overfitting and inefficiency in standard convolutions, the Kronecker approach supports faster convergence and higher accuracy. In many cases, the number of epochs required for training decreases from 100 to 90, shortening overall experimentation time while improving performance. This makes Kronecker convolution an appealing strategy for advancing deep learning methods in the classification of rare eye diseases.

The CNN-GRU, GRU U-Net and RAN models achieve high recall values, although precision is variable. The introduction of Kronecker convolution (Tables 8.5–8.7) improves these metrics, increasing model sensitivity and F1-score, while maintaining high specificity. Confusion matrices show fewer false negatives for rare classes, confirming the effectiveness of the approach. Additionally, for the custom CNN, the model converged even faster, after about 85 epochs, indicating its effectiveness in learning disease patterns.

9. Vision Transformers

In recent years, transformers, originally designed for processing text sequences [151], have been successfully extended to the field of computer vision. The Vision Transformers architecture [152] is based on a key assumption that images can be treated as sequences of fragments (so-called patches) and then processed with the attention mechanism previously used mainly in natural language processing. This approach has proven to be competitive with traditional convolutional neural networks, especially with large datasets. The method of introducing image element locations using position embeddings, methods aimed at more efficient image segmentation into tokens [153], as well as concepts enriched with residual attention mechanisms (Residual Attention Vision Transformers) remain intensively studied, striving for even more efficient and precise representation of visual information. This chapter presents the basic ideas and recent research directions on attention, positional embeddings and architectural variants, providing an overview of the key elements in this rapidly developing field. The overall architecture of the applied ViT network is presented in Fig. 9.1.

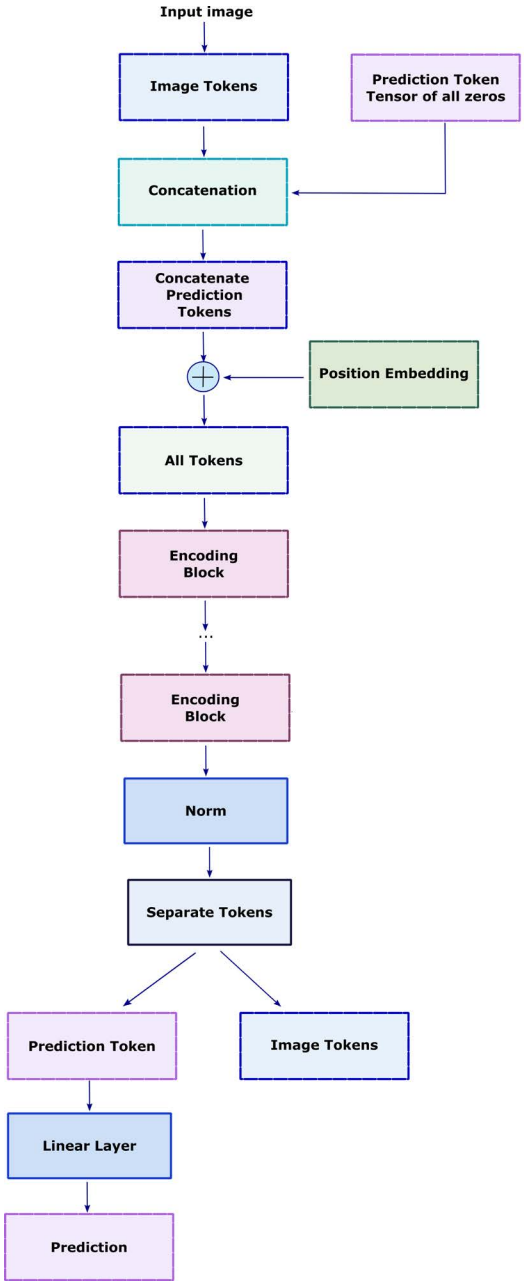


Fig. 9.1. Overview of the Vision Transformer architecture with an explicit prediction token. Image patches are first embedded into image tokens, which are concatenated with an initialized prediction token and enriched with positional embeddings. The resulting token sequence is processed by a stack of Transformer encoding blocks with self-attention and residual connections. After the final normalization, tokens are separated and only the prediction token is forwarded to a linear classification layer to produce the final decision.

9.1. Image Tokenization

The first step in Vision Transformers involves representing the input image as a sequence of tokens, allowing the use of the attention mechanism familiar from natural language processing [152]. While in Natural Language Processing (NLP) tokens typically correspond to words or subwords, in the case of images, the challenge lies in determining how to segment the input into fundamental visual elements.

In the ViT approach, the image is divided into small patches, each corresponding to a single token. In this way, an image with dimensions H (height), W (width), and C (channels) is transformed into N tokens Eq. (9.1), where each token represents a patch of size P [152]. Each token is of length P^2*C. This makes it possible to apply the transformer architecture directly to the image representation, minimizing the need for traditional convolutions.

$$N = \frac{HW}{P^2}. \tag{9.1}$$

Let us consider as an example of patch tokenization on the image of the retinitis pigmentosa disease. The original image has been cropped and converted into a single channel image. This means that each pixel has a value between zero and one. Single channel images are typically displayed in grayscale. However, it will display in an ocean colour scheme because it is easier to see (Fig. 9.2.a).

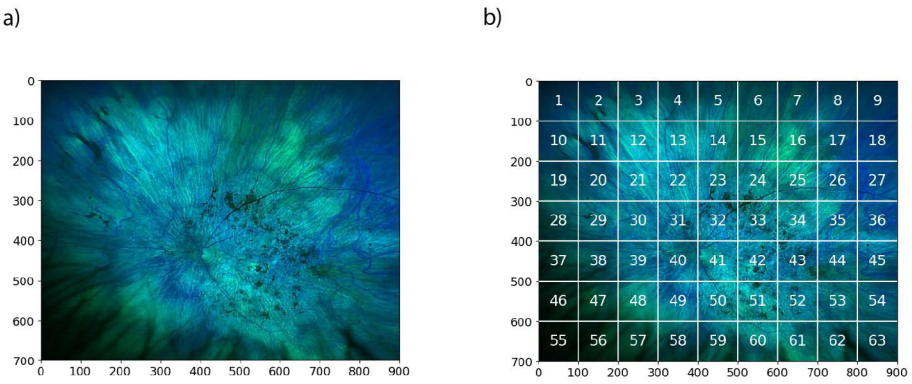


Fig. 9.2. a) Retinitis pigmentosa image in purple scale, b) Retinitis pigmentosa image divided into patches.

By flattening these patches, it becomes possible to observe the resulting tokens. Consider patch no. 32, for instance, as it presents four distinct shades within its composition (Fig. 9.3).

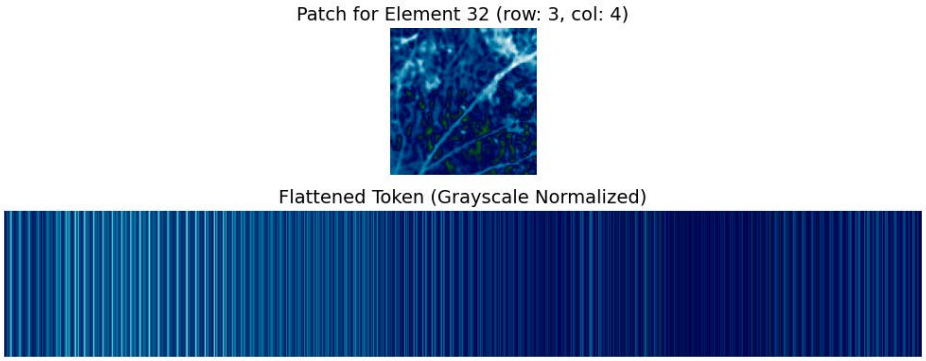


Fig. 9.3. A visual information about patch for element no. 32. Rows and columns numbered from 0.

Following the extraction of tokens from the image, a linear projection is typically applied to modify their dimensionality. This step is implemented through a trainable linear layer, and the resulting dimension is commonly referred to as the latent dimension, channel dimension or token length. Once this projection is completed, the tokens cease to maintain any direct visual correspondence to their original image patches.

9.2. Token processing

The subsequent two steps of the ViT architecture, conducted before the encoding blocks, are collectively referred to as "token processing". The token processing component of the ViT is illustrated in Fig. 9.4.

The first step involves prepending a blank token, designated as the Prediction Token, to the existing image tokens. Initially set to zero, this Prediction Token subsequently acquires information from the other image tokens, enabling its use at the encoding blocks' output to generate a prediction. At this stage, the model operates with 64 tokens, each with a length of 768, according to the base variant of ViT. The batch size employed is 13 [152]. Following the addition of the Prediction Token, a position embedding is applied

to all tokens. This positional information, incorporated by addition rather than concatenation, enables the transformer to interpret the ordering of the image tokens. With this step completed, the tokens are prepared for processing by the encoding blocks.

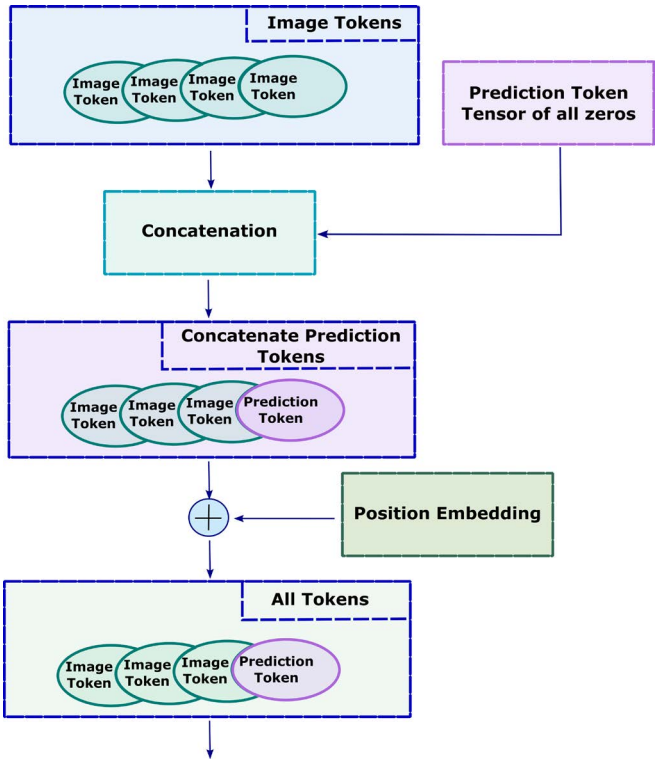


Fig. 9.4. Schematic overview of the ViT inference pipeline. Image tokens extracted from the input image are concatenated with an initialized prediction token and enriched with positional embeddings. The resulting token sequence is processed by a stack of encoding blocks based on self-attention and normalization. After the final normalization, the prediction token is separated and passed through a linear layer to produce the final classification output.

9.2.1. Position Embeddings

One of ViT's limitations is the absence of recurrence or convolution, meaning that transformers cannot inherently learn the order of token sequences. Without position embeddings, ViTs are insensitive to token order. In the context of images, this implies that rearranging image patches does not affect the predicted outcome (Fig. 9.5).

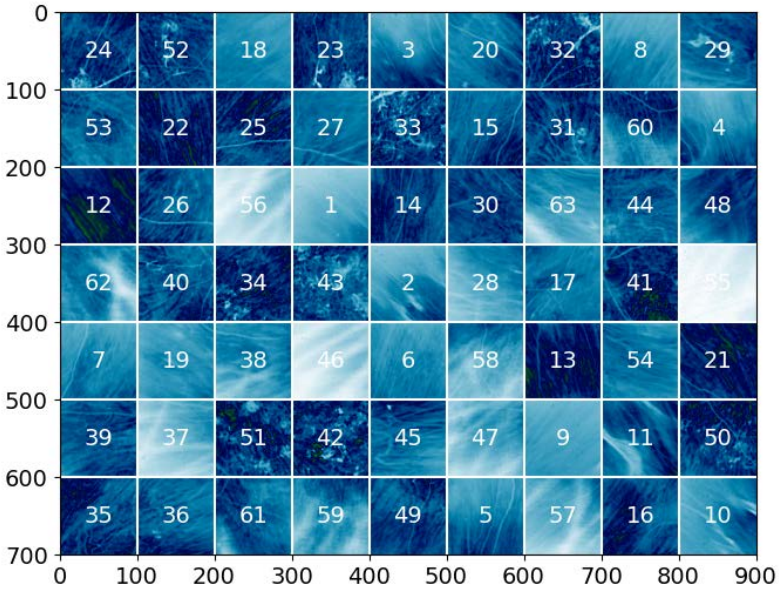


Fig. 9.5. Shuffled image patches.

To investigate the claim that ViTs remain invariant to token order, it is essential to consider which architectural component might induce this property. In particular, the attention module within the Transformer does not inherently encode positional relationships between tokens. A comprehensive explanation of this module's mechanisms can be found in Subchapter 9.3.

The attention process begins by deriving three matrices, Queries (Q), Keys (K) and Values (V), obtained through linear transformations of the input tokens. Once Q, K, and V are computed, the attention mechanism determines the relative importance of each token to every other token, without reliance on their original order. For demonstration purposes, four tokens, each possessing a projected length of nine (Fig. 9.6), are considered. The matrices are composed of integers. After generation, the positions of token 0 and token 2 are interchanged in all three matrices. Matrices reflecting these token swaps are denoted with a subscript 's'.

The first matrix multiplication in the attention formula, $A = Q*K^T$, produces a square matrix A, with dimensions corresponding to the number of tokens. When A_s is computed using Q_s and K_s, the resulting matrix A_s is obtained with both rows [0,2] and columns [0,2] swapped relative to A (Fig. 9.7).

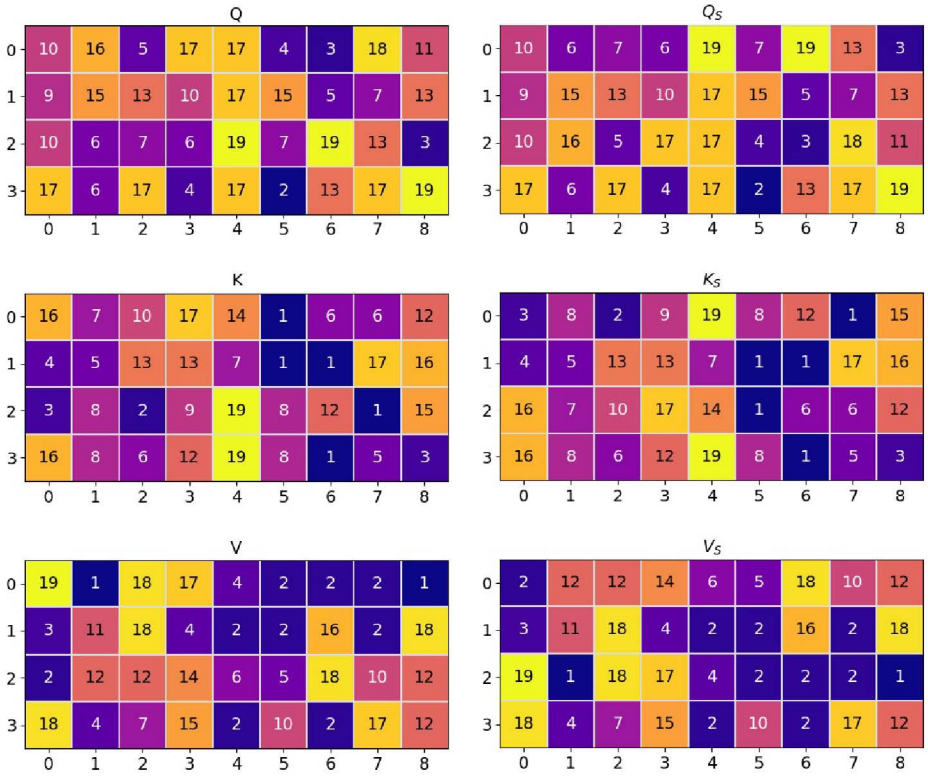


Fig. 9.6. Visualization of query (*Q*), key (*K*) and value (*V*) matrices generated for selected image patches, together with their modified counterparts (Q_s, K_s, V_s) obtained by substituting tokens 0 and 2. The comparison illustrates how local token substitution affects the internal attention representations, leading to noticeable changes in the corresponding *Q/K/V* activations while preserving the overall structural consistency of the token grid.

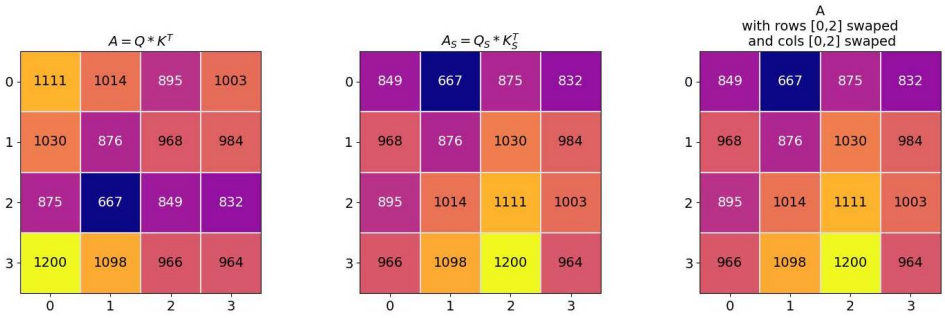


Fig. 9.7. Illustration of the attention score matrix $A = QK^T$ (left), the modified matrix. It can be seen that, regardless of the change in column order, the resulting matrix contains exactly the same values.

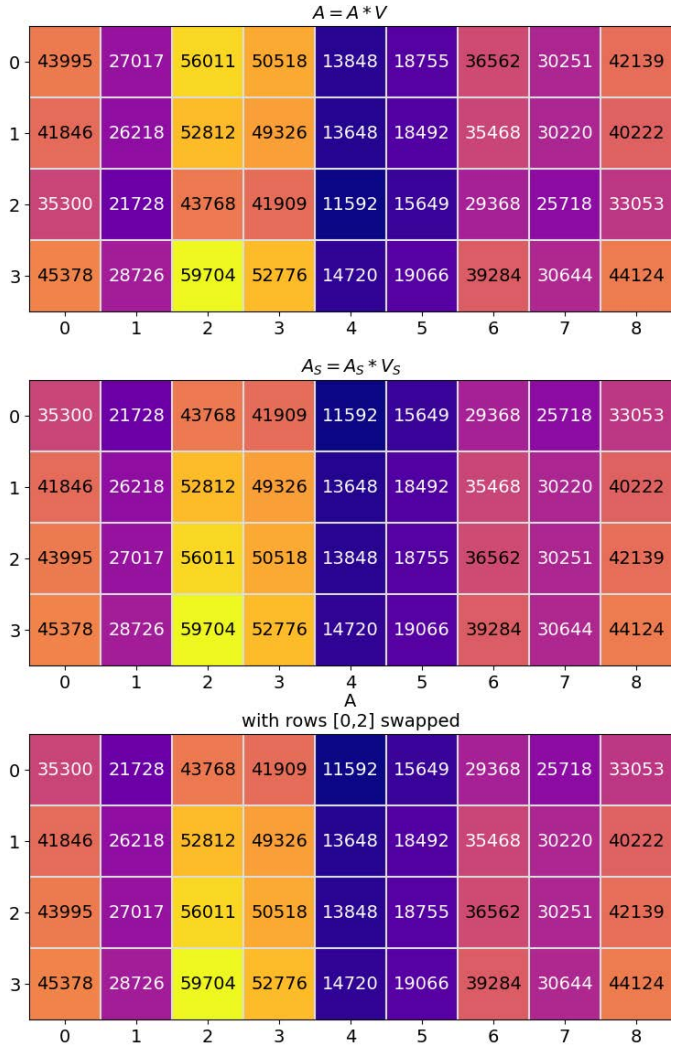


Fig. 9.8. Example showing that changing the order of tokens at the input to the attention layer results in an attention matrix at the output with the same changed token rows. This remains intuitive since attention is a computation of relationships between tokens. Without position information, changing the order of tokens does not change the way in which tokens are related.

The subsequent matrix multiplication involves A_s and the value matrix V_s, producing an output matrix that retains the same dimensionality as the original Q, K, and V matrices. When A_s is multiplied by V_s, the resulting representation preserves the row-wise index permutation [0,2] observed in A (Fig. 9.8).

In functional terms, the sinusoidal position embedding can be regarded as a matrix that shares the same shape as the tokens, as illustrated in Fig. 9.9.

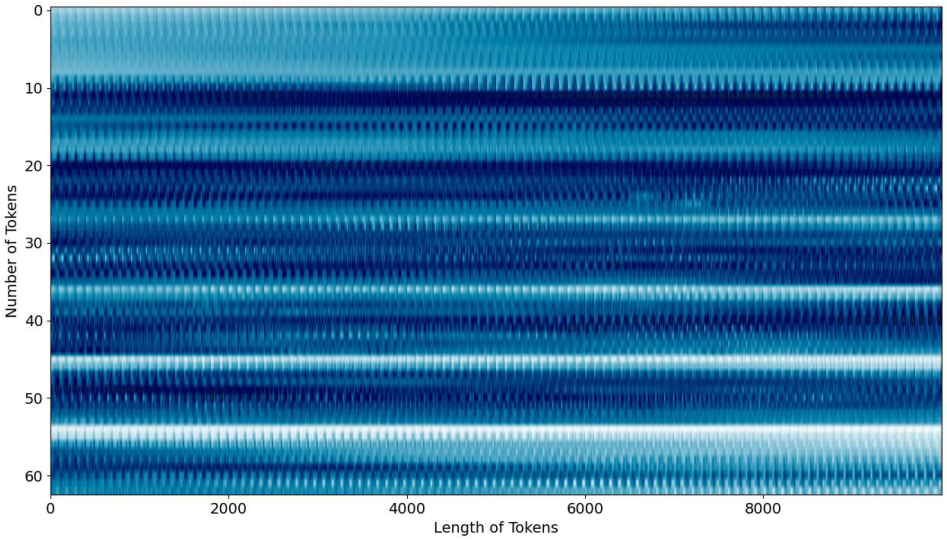


Fig. 9.9. Example of converting the patches into tokens using the same patch tokenization method as depicted in Fig. 9.2.b. While this process yields a fixed number of tokens (64), the dimensionality of each token depends on the embedding layer configuration (e.g., embedding dimension and projection strategy) and may therefore vary across different model settings.

The mathematical expressions (Eq. 9.2) for the sinusoidal position embedding ($\theta_{i,j}$) presented in [153] indicate that the position embedding matrix, denoted as *PE*, i is along the number of tokens, j is along the length of the tokens and d is the token length.

$$\begin{aligned} \theta_{i,j} &= \frac{i}{10000^{2jd^{-1}}} \\ PE_{i,2j} &= \sin(\theta_{i,j}) \\ PE_{i,2j+1} &= \cos(\theta_{i,j}) \end{aligned} \tag{9.2}$$

Once reshaping the patches into tokens, a sinusoidal position embedding of the appropriate dimensions is created and then added to the tokens (Fig. 9.10). As a result, ViTs can simultaneously observe the original token structure and positioning information, which will be transmitted to subsequent layers of the transformer (Fig. 9.11).

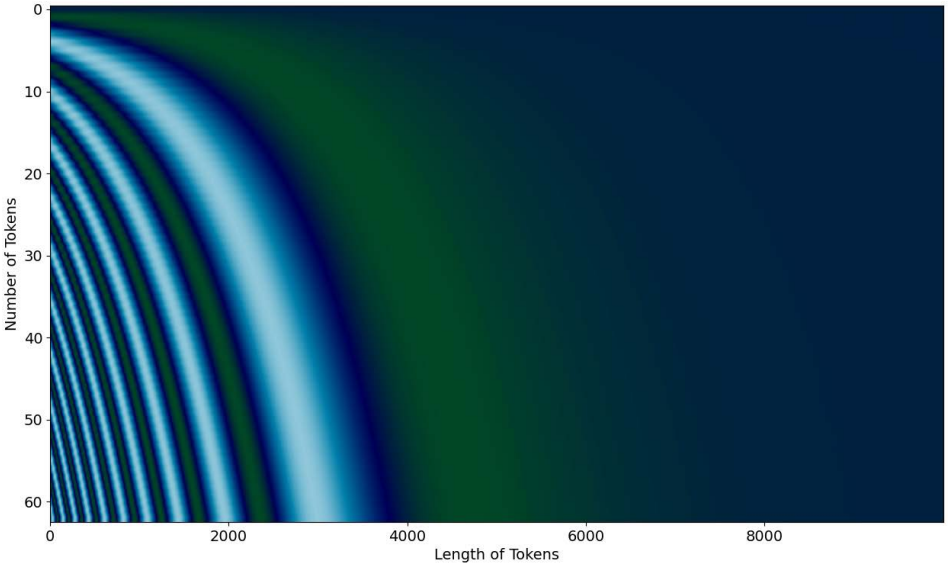


Fig. 9.10. Example of incorporating the sinusoidal position embedding into the tokens using the same patch tokenization method as depicted in Fig. 9.2.b. The light blue area will make the tokens darker, while a green one will make them lighter.

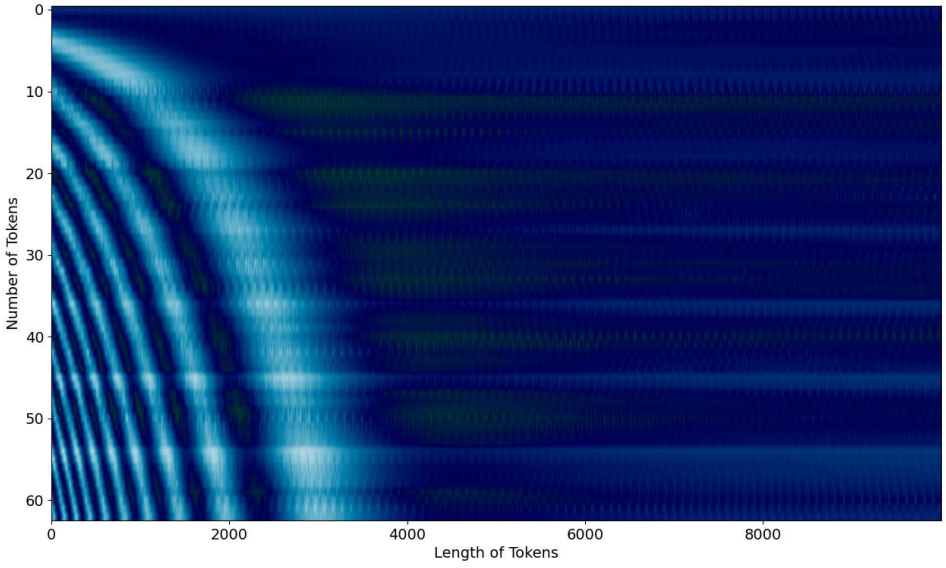


Fig. 9.11. Sample appearance of the structure from the original tokens as well as the structure in the position embedding.

9.2.2. Encoding Block

The encoding block (Fig. 9.12) in a Vision Transformer extends the core ideas of the Transformer architecture. Each block begins with a multi-head self-attention module that calculates pairwise relationships among all patch tokens, allowing the model to integrate both local and global context. These tokens, which have been projected from image patches via a linear embedding, are combined with learnable or sinusoidal positional embeddings to retain information about their spatial arrangement. The self-attention (described in Subchapter 9.3) operation is carried out by projecting tokens into multiple query, key and value spaces, then computing attention weights that emphasize relevant patch-token interactions. By aggregating attention outputs from parallel heads, the block captures a wider variety of contextual cues before concatenating and projecting them back into the token dimension.

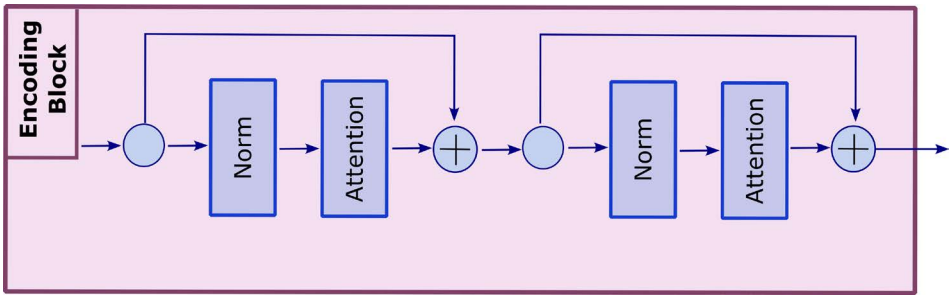


Fig. 9.12. The structure of the encoding block in ViTs. The block consists of two sequential self-attention sublayers, each preceded by layer normalization and followed by residual (skip) connections.

Following the attention module, a residual connection is added, and layer normalization is applied, resulting in enhanced training stability and improved gradient flow. The output then feeds into a feed-forward network comprising one or more linear transformations, interleaved with non-linear activations. Another skip connection and layer normalization step ensure that each token's representation is refined while preserving the flow of gradients. Through this iterative process, the model gradually captures higher-level features as more encoding blocks are stacked. These learned representations merge local patch-level information with the broader spatial layout of the image, enabling the Transformer to capture both fine-grained and global structures in a unified embedding space. The ability to learn contextual relationships in a position-aware manner is central to the encoding block, fostering a flexible mechanism for

image understanding without relying on explicit convolutional operations. By repeating the encoding block multiple times, Vision Transformers accumulate progressively richer descriptors, allowing them to represent tasks such as classification, detection and segmentation [153].

The Neural Network (NN) module is a sub-component of the encoding block. The NN module, in its standard form, has a fairly simple architecture, consisting of a fully-connected layer, an activation layer, and another fully-connected layer. The activation layer can be any layer that is passed as input to the module. It has also been noted that the NN module can be customized either to alter the input data's shape or to preserve its existing dimensions, depending on the design requirements.

A dedicated prediction token is explicitly separated from the remaining tokens and used as the sole input to the classification head. During the encoding process, this token accumulates global information about the input image through self-attention interactions with all other tokens. Consequently, only the prediction token is employed to generate the final output (Fig. 9.13).

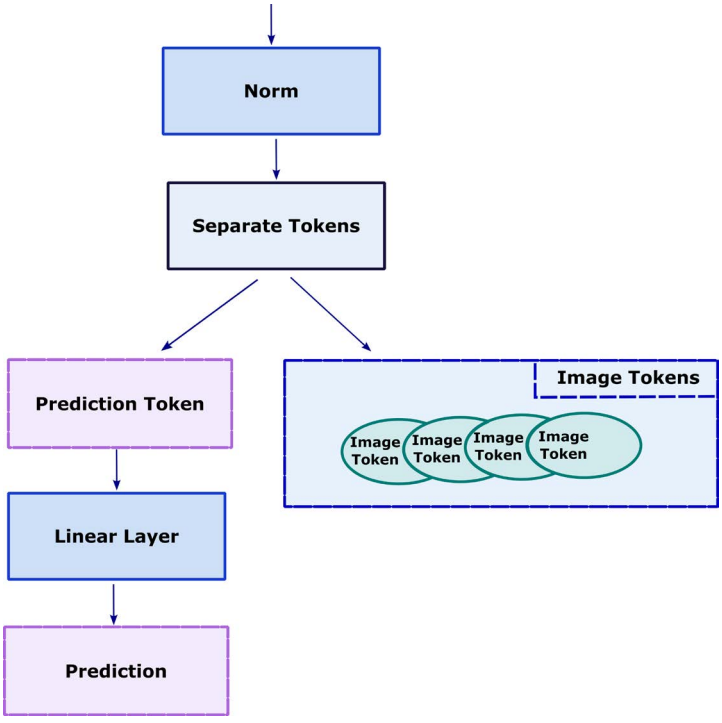


Fig. 9.13. The structure of prediction processing components of ViTs. After the final normalization, the token sequence is separated into the prediction token and image tokens. The prediction token is passed through a linear layer to produce the final model output.

Subsequently, the prediction token is fed into the classification head to generate the final output. The classification head is implemented as a neural network module, which may consist of a single linear layer, a multilayer perceptron, or a more complex structure, depending on the specific model configuration. The dimensionality of the head's output is defined by the learning task. In classification problems, it typically corresponds to a vector whose length equals the number of target classes.

9.3. Attention for Vision Transformers

In natural language processing, attention is frequently conceptualized as the relationship between words (tokens) within a sentence. When adapted to computer vision, attention instead characterizes the relationships among patches (tokens) in an image. There are various approaches for decomposing an image into a set of tokens. One potential method, illustrated in Subchapters 9.1 through 9.3, involves dividing the image into patches that are subsequently flattened into tokens. In the following, an attention layer is applied under the assumption that these tokens constitute its input.

At the initial stages of a transformer, these tokens directly represent patches in the input image. However, in deeper attention layers, attention operates on tokens that have already been modified by preceding layers, thereby obscuring the direct patch-to-token correspondence. The subsequent discussion examines dot-product (or multiplicative) attention [151] and explores both single – and multi-headed attention mechanisms.

9.3.1. Single-Head Attention

According to [151], attention is defined in terms of Query (Q), Key (K) and Value (V) matrices. The first step of computation entails deriving these matrices through a trainable linear transformation (Fig. 9.14). The linear transformations used to derive the Query, Key, and Value matrices play a crucial role in shaping the behaviour of the attention mechanism. Given an input token representation $X \in R^{N \times D}$, the matrices Q, K, and V are obtained by separate trainable projection matrices WQ, WK and WV, respectively, such that $Q = XW_Q$, $K = XW_K$, $V = XW_V$. These linear projections enable the model to embed the same input tokens into different representation subspaces. In this formulation, the query

vectors encode the information sought by each token, the key vectors determine how tokens are matched during similarity computation, and the value vectors carry the information to be aggregated. As a result, the projection matrices directly influence the attention patterns of the model [152].

Attention is then computed as in Eq. (9.3).

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{9.3}$$

where d_k is the dimension of the keys, equal to both the length of the key tokens and the channel dimension.

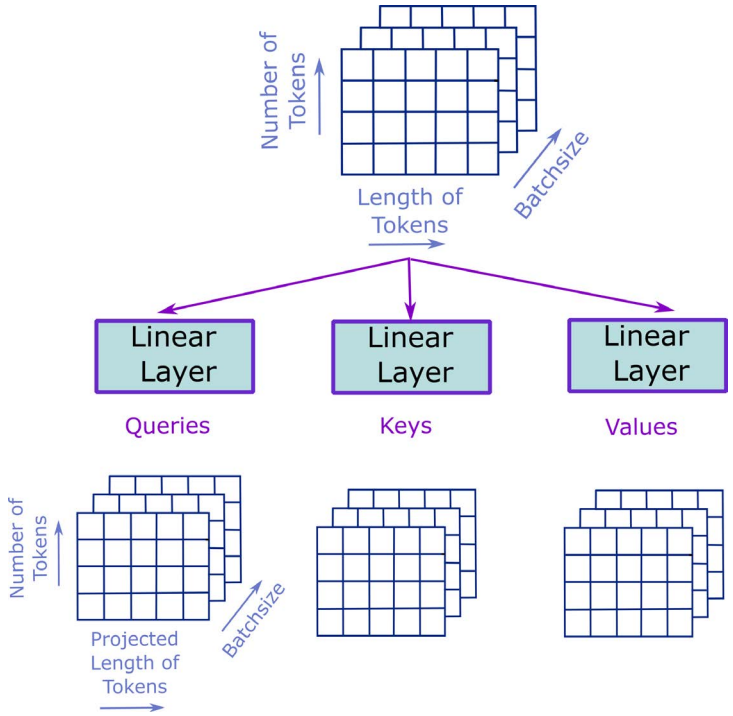


Fig. 9.14. Generation of Queries, Keys, and Values for single-head attention.

To better understand this pivotal process let's consider it in detailed steps. The first step is to compute intermediate attention matrices A Eq. (9.4).

$$A = \frac{QK^T}{\sqrt{d_k}}. \tag{9.4}$$

Graphically this process is presented in Fig. 9.15.

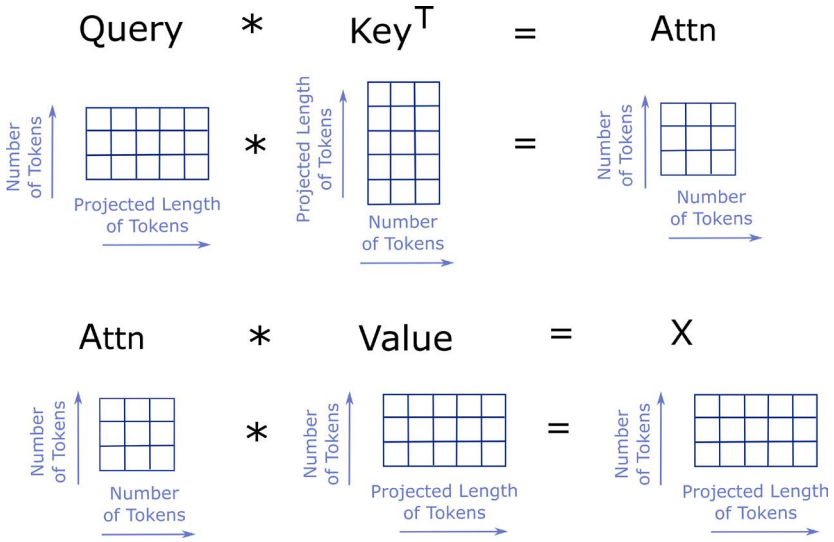


Fig. 9.15. Top: the matrix multiplication $Q \cdot K^T$ to obtain intermediate attention. Bottom: generating overall single-head attention.

Then there is usually a need to reshape output x to remove the attention head dimension. Lastly x is fed through a learnable linear layer that does not change its shape. It is worth noting that a skip connection is implemented. Since the current shape of x is different from the input shape, the V matrix is used for the skip connection and it is flattened in the attention head dimension. The whole process is depicted in Fig. 9.16.

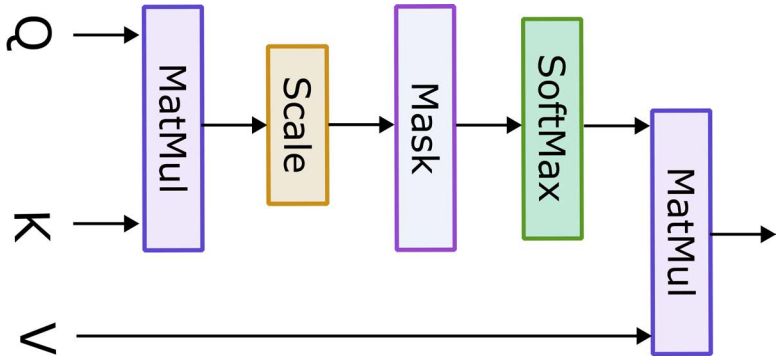


Fig. 9.16. Graphical illustration of single-head attention mechanism. As a scale is used $\sqrt{d_k}$. MatMul indicates matrix multiplication.

9.3.2. Multi-Head Attention

Despite the effectiveness of self-attention in replacing both recurrence and convolution, it introduces limited expressivity due to its averaging effect. Multi-head attention addresses this limitation by simultaneously modelling contextual dependencies in multiple representation subspaces. It achieves this by running several attention heads in parallel, concatenating their outputs, and then passing them through an affine transformation.

The method for computing *Queries*, *Keys* and *Values* is identical to that used in single-headed attention. However, the token length now becomes $\frac{\text{number of channels}}{\text{number of heads}}$.

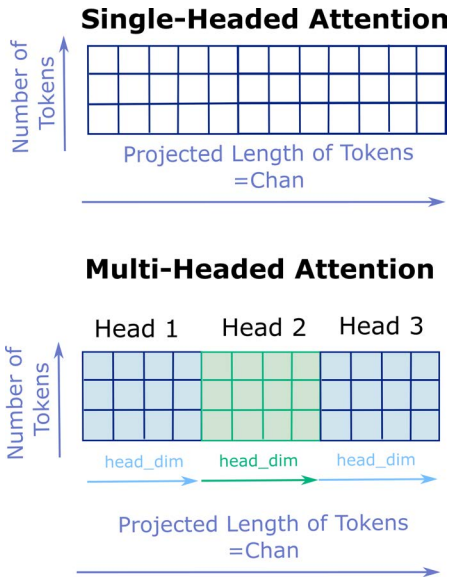


Fig. 9.17. Generation of heads for Multi-Headed Attention. In the single-headed attention mechanism (top), all tokens are projected into a single representation space of dimension Chan. In contrast, Multi-Head Attention (bottom) splits the projected token representation into multiple lower-dimensional subspaces (head_dim), which are processed independently by separate attention heads.

While the overall dimensions of the Q, K and V matrices remain unchanged, their contents are distributed across the newly introduced head dimension. In essence, the single-headed matrix is split into multiple segments, each assigned to a different head (Fig. 9.17). As in the case of single-head attention, a temporary attention matrix A_{Hi} should be determined Eq. (9.5) [152], [153].

$$A_{Hi} = \frac{Q_{Hi} K_{Hi}^T}{\sqrt{d_k}}. \tag{9.5}$$

However, in this case the scaling factor d_k is not equal to the number of channels but $\frac{\text{number of channels}}{\text{number of heads}}$.

All other operations are performed analogously to the single-head attention case. The entire process of data processing by multi-head attention is shown in Fig. 9.18, while a visualization of the first 16 attention maps in Fig. 9.19.

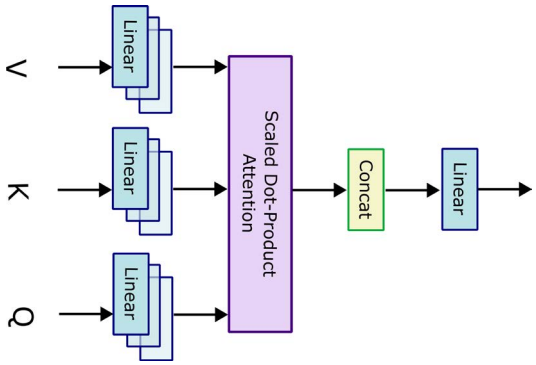


Fig. 9.18. Graphical illustration of multi-head attention mechanism.

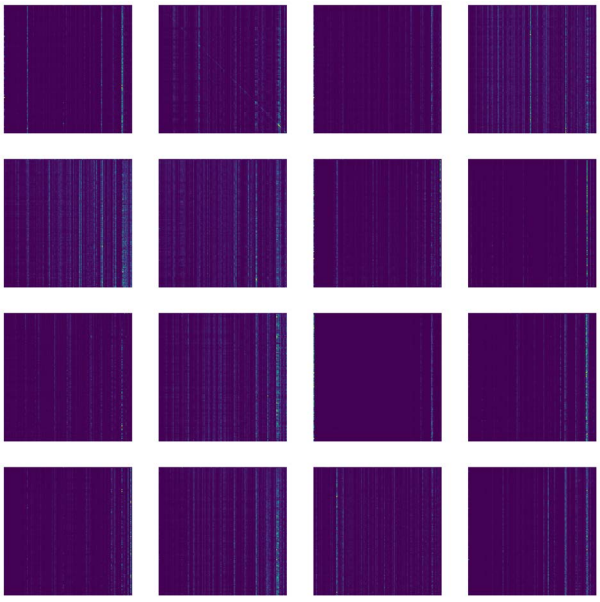


Fig. 9.19. Visualisation of 16 attention matrices from ViTs for Fig. 9.2a as an input image. Each subfigure represents the attention distribution learned by a single head, highlighting diverse token interaction patterns and complementary spatial focus across heads.

9.4. ViT Evaluation

This section presents the classification results obtained by the ViT model for two datasets. Particular attention is paid to four basic evaluation measures: precision, recall, F1-score and specificity (Table 9.1). These measures allow for assessing both the overall classification accuracy and the model's ability to distinguish diseased from healthy cases. In order to more fully illustrate the functioning of the ViT model, learning curves are also presented, showing the course of the training process and the convergence of the network in subsequent epochs, as well as confusion matrices (Fig. 9.22), which illustrate the distribution of correct and incorrect predictions between individual classes.

Table 9.1. The obtained results for RP and AVL datasets classification by ViT model. All values in %.

Dataset	Class	Precision	Recall	F1-score	Specificity
RP dataset	RP	98.97	93.20	96.00	98.77
	CORD	74.19	92.00	82.14	94.97
	HEALTHY	92.86	92.86	92.86	96.88
AVL dataset	AVL	85.00	89.47	87.17	98.28
	DRUSEN	96.55	96.55	96.55	97.17
	HEALTHY	95.35	94.25	94.80	96.23

The results presented in Table 9.1 confirm the high efficiency of the Vision Transformer model in recognizing rare retinal diseases on the RP and AVL sets. Particularly high results were obtained for the RP class: F1-score at the level of 96%, with very good values of both precision (98.97%) and sensitivity (93.20%). This means that the model not only rarely misses patients that are actually sick (high recall), but also does not make many mistakes by marking healthy people as RP (high precision). For the CORD class, the precision is slightly lower (74.19%), but the model still maintains high sensitivity (92.00%), which reduces the risk of missing the disease. The HEALTHY class maintains balanced results at around 93% for all measures, indicating the stability of the model in distinguishing people without pathological changes. A similar trend is observed in the AVL set, where the DRUSEN and HEALTHY classes demonstrate very high levels of precision and sensitivity, resulting in F1-score of 96.55% and 94.80%, respectively. For the AVL class, precision (85.00%) is slightly lower than for other categories, but high sensitivity (89.47%) protects against missing important disease cases. Confusion matrices (Fig. 9.22) confirm a small number of misclassifications, in particular limiting false negatives for RP and AVL.

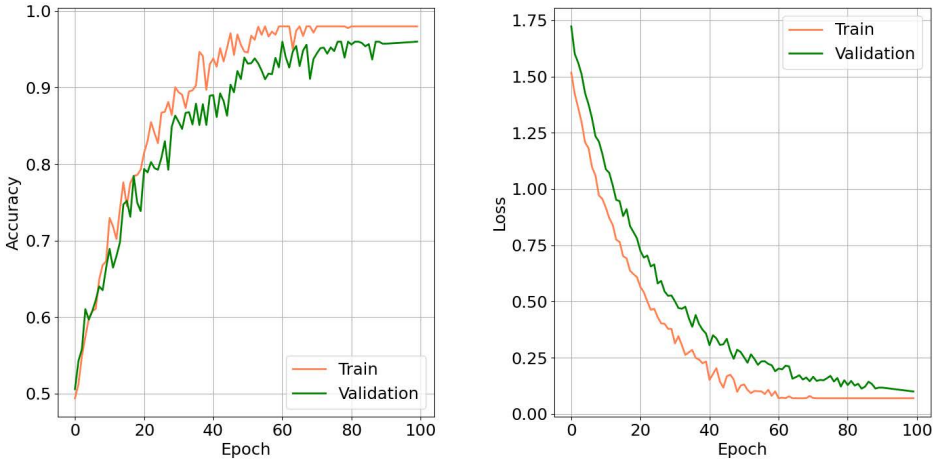


Fig. 9.20. ViT training and validation accuracies (left) and loss (right) obtained for RP dataset. The model demonstrates stable convergence, with validation accuracy exceeding 95% after approximately 60 epochs and a monotonic decrease of validation loss, indicating good generalization and no visible overfitting.

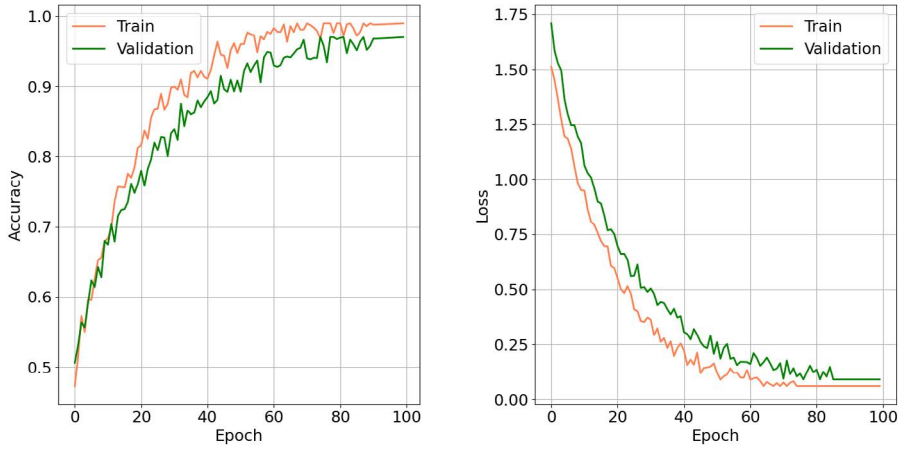


Fig. 9.21. ViT training and validation accuracies (left) and loss (right) obtained for AVL dataset. The model demonstrates stable convergence, a gradual reduction of validation loss, and close alignment between training and validation accuracies.

Learning curves (Fig. 9.20 and 9.21) suggest relatively fast convergence of the network and stability in the final epochs, which can be seen both in the increase in accuracy above 90% and in a clear reduction of the loss function. Additionally, a small discrepancy between training and validation results suggests that the model has not been overlearned, which is particularly important in tasks with limited data. Obtained results indicate that ViT can effectively extract subtle characteristics of various retinal changes while maintaining high stability and versatility.

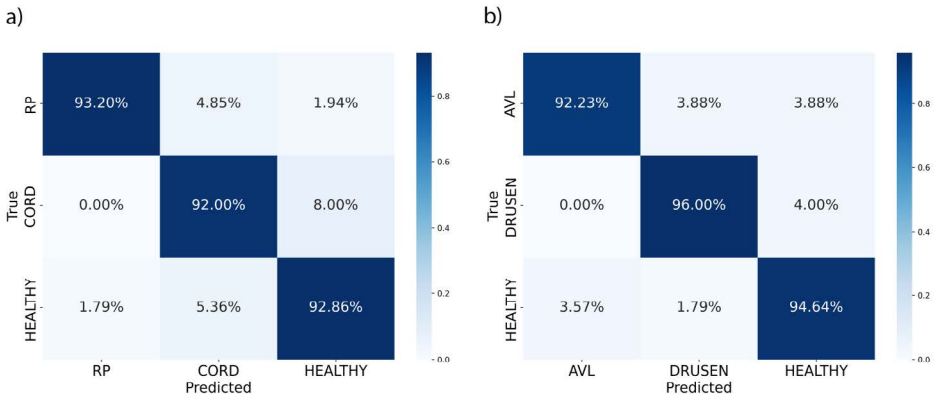


Fig. 9.22. Confusion matrices for ViT model for a) RP dataset b) AVL dataset.

9.5. Chapter Summary

The chapter on Vision Transformers shows how the approach based on image division into a sequence of small fragments and the mechanism of multi-headed attention can replace traditional convolutions in the analysis of retinal images. In the classification tasks of rare diseases, such as retinitis pigmentosa or acquired vitelliform lesions, ViT presents fully satisfactory results. According to Table 9.1, on the RP set, the RP class achieves F1-score as high as 96%, with precision of 98.97% and sensitivity of 93.20%, and for the CORD class with sensitivity of 92.00% there are no numerous false negatives (reducing the risk of missing the disease). Similarly stable results are noted for the HEALTHY class, for which all measures oscillate around 93%.

In the AVL set, the highest values are observed for the DRUSEN class, where precision, recall and F1-score are 96.55%. The HEALTHY class also remains at a similar level (e.g. F1-score 94.80%). Although the precision (85.00%) of the AVL class itself is lower, the high recall of 89.47% still effectively reduces the number of undetected cases. Analysis of the learning curves indicates a fast convergence rate and a relatively small difference between the training and validation sets, suggesting good generalizability even with a limited number of samples. The obtained results clearly confirm that ViT is able to extract subtle features that are key to recognizing changes in the retina, which makes it a promising tool for diagnosing rare eye diseases.

10. Deep Residual Vision Transformer

In recent years, several modifications to Vision Transformers have been proposed in the literature to overcome their inherent limitations, such as high computational complexity, feature collapsing, limited gradient propagation in deep architectures, and reduced effectiveness when trained on a small dataset. Examples include studies that combine the transformer concept with additional mechanisms introducing new signal pathways, which help preserve feature diversity. Particularly noteworthy is the concept of Residual Self-Attention, which introduces additional residual connections within self-attention layers. This increases the effective depth of the network, improves gradient propagation and reduces the risk of losing unique image features.

Although self-attention in ViTs provides the model with a global perspective on the image structure, it also has certain drawbacks. With a high number of patches (high image resolution), the computational complexity grows quadratically, which can hinder training and real-time application of the models. Additionally, there is the phenomenon of feature collapsing, where features represented by different patches become "flattened" or "merged". The self-attention mechanism generates an averaged version of various local features, potentially leading to a loss of diversity. The proposed residual connections allow partial feature vectors (from earlier layers) to flow into the outputs of later layers, aiding in maintaining the distinctiveness of these features.

10.1. Residual Self-Attention

The typical residual connection in ViT consists of adding the original input of the layer to the output of the self-attention layer (a procedure well known from ResNets). This allows one to preserve a certain portion of the original features in the output, unprocessed by the multi-head attention mechanism. However, in Residual Self-Attention, several things are observed.

First, the mechanism allows one for the incorporation of query and key matrices from previous layers. In a typical self-attention layer, the queries Q and keys K are calculated only for the current level. In Residual Self-Attention, information from previous layers can also be included in their calculation.

Second, aggregation of attention weights, from multiple levels, allows one to combine the self-attention results from a layer with the results from previous layers, which allows one to preserve the global and local context at the same time. Such a mechanism can be formalized as mixing the attention matrix using a coefficient $\boldsymbol{\xi}$, which decides which part of the current attention is to be taken into account in relation to the previous one.

Third, the risk of overfitting is reduced and diversity is preserved. Transferring queries and keys from previous layers helps to maintain the diversity of feature vectors. Since standard self-attention involves increasing averaging of representations in subsequent steps, introducing residual layers for the Q and K matrices make the model constantly remember low-level and mid-level relationships between patches.

The presence of skip connections around multi-head attention does not completely eliminate the feature collapsing phenomenon. Only additional paths transferring information between queries and keys in subsequent layers allow for greater feature diversification.

Medical images are often burdened with noise as well as non-standard artifacts, such a residual self-attention mechanism allowing better capturing of the anatomical structures of the eye, which translates into higher efficiency of disease classification in conditions of low data quality.

Formally, the proposed attention mechanism modifies the computation of the attention scores in the multi-head attention. Instead of computing the attention for each layer independently, the attention scores are aggregated with those from the previous multi-head attention layers. Specifically, for the i-th layer, the attention scores are computed as follows Eq. (10.1) [151].

$$SA_i \begin{cases} \frac{Q_0 K_0^T}{\sqrt{d_{k0}}} & \text{if } layer\ number\ i\ equals\ 0 \\ \xi\left(\frac{Q_i K_i^T}{\sqrt{d_i}}, SA_{i-a}\right) & otherwise, \end{cases} \tag{10.1}$$

where ξ indicates the permutation invariant aggregation function. For the first layer, the attention scores are computed in the standard manner as Eq. (10.2) [151]. For subsequent layers, the attention scores are computed as a weighted combination of the current layer's attention and the aggregated attention from the previous layer as Eq. (10.3).

$$A_0 = \frac{Q_0 K_0^T}{\sqrt{d_{k0}}}, \tag{10.2}$$

$$A_i = \xi \cdot \frac{Q_i K_i^{\mathsf{T}}}{\sqrt{d_k}} + \left(1 - \xi\right) \cdot A_{i-1}, \quad i > 0, \tag{10.3}$$

where ($\xi \in [0,1]$) controls the contribution of the current and previous attention layers.

The entire structure of the residual self-attention block is depicted in Fig. 10.2.

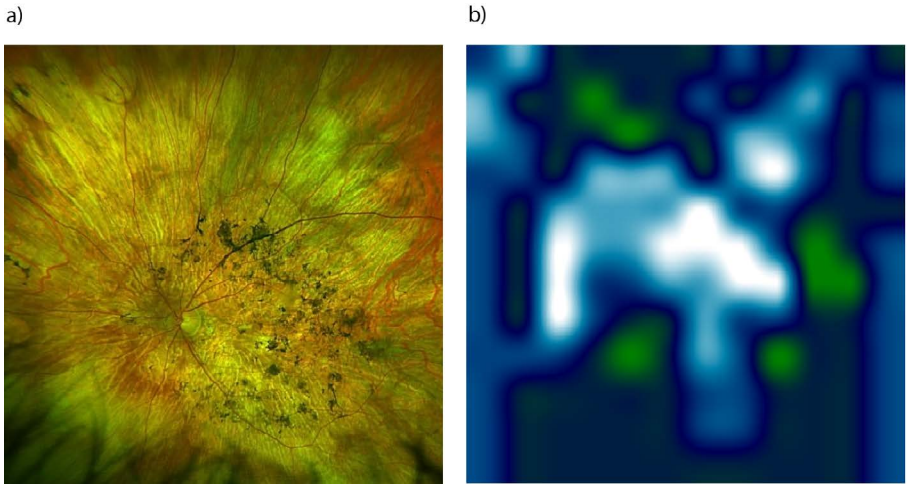


Fig. 10.1. An example of a global attention map (a) and an input image (b). The green areas represent the main regions of interest.

Multi-Head Self-Attention is a mechanism that empowers ViTs to capture complex relationships within a sequence by representing information in multiple subspaces. In essence, each attention head learns to focus on distinct aspects of the input. The process begins by projecting the input embedding matrix $X \in R^{N \times d}$ (with tokens and embedding dimension d) into three sets of learnable parameters Q, K, V. For a single attention head, this can be expressed as $Q = XW^Q$, $K = XW^K$, $V = XW^V$, where W^Q, W^K, W^V, are the trainable weight matrices that transform the input embeddings into Q, K, V, respectively. Instead of calculating, based on Q, K, V, a scaled dot-product attention in this study, a residual self-attention mechanism is employed (Fig. 10.2).

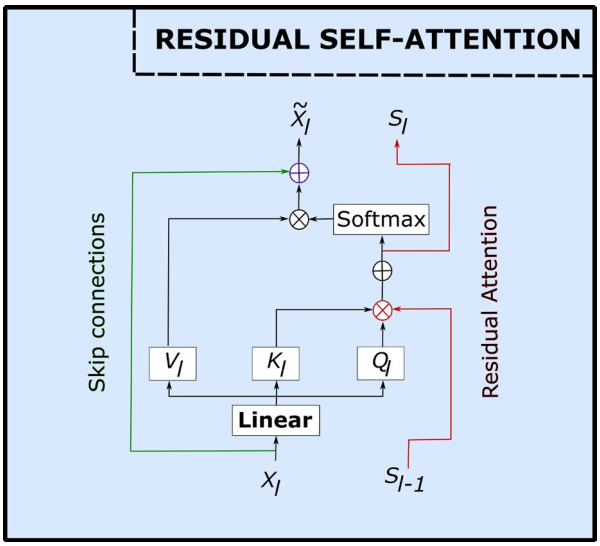


Fig. 10.2. The residual self-attention block. The input feature representation x_l is linearly projected to queries Q_l, keys K_l, and values V_l. Attention scores are computed using scaled dot-product attention followed by Softmax normalization, producing the attention output S_l. A residual connection propagates information from the previous layer S_{l-1}, while a skip connection adds the original input x_l, resulting in the refined representation.

In Multi-Head Self-Attention, multiple attention heads h are employed in parallel. Each head produces its own attention matrix, focusing on different relationships within the same sequence. Formally, if there are h heads, the queries, keys and values are split across h smaller subspaces $d_k = d/h$, where d_k is the dimensionality of the Q and K serve as a scaling factor in attention mechanism. The attention outputs from each head are concatenated and projected back to the original embedding dimension via a trainable matrix W^O Eq. (10.4) [152].

$$MHSA = Concat\left(head_1, \ldots, head_h\right) W^O$$
$$head_i = A_i(Q, K, V)\ , \tag{10.4}$$

where $A_i(Q, K, V)$ is defined by Eq. (10.3). By distributing the representation across multiple heads, the model gains an enriched perspective of the input data. Each head can detect unique dependencies and contextual cues, and the final aggregation incorporates these diverse insights into a comprehensive representation.

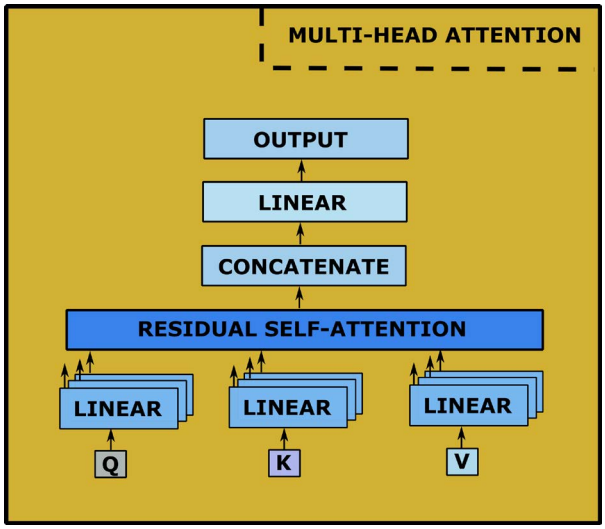


Fig. 10.3. The multi-head self-attention block with incorporated residual self-attention mechanism.

Attention globality describes the degree to which the attention mechanism in ViTs can incorporate information from the entire image, rather than focusing solely on local or adjacent elements. It represents the overall receptive field of the attention module. This feature becomes increasingly prominent in deeper layers of a ViT, where higher-level representations capture distant relationships among patches. A visual representation of the global attention map, which consists of a set of residual self-attention maps, is presented in Fig. 10.1. To quantitatively assess attention globality across heads and layers, one can use the non-locality metric expressed by D_i in Eq. (10.5). For each query patch m, this metric calculates the weighted sum of relative distances to all key patches j, where the weights are given by the attention scores $A_{m,j}^{i,h}$. Mathematically,

this metric is defined in terms of $||\delta_{m,j}||$, which denotes the positional distance between patch m (query) and patch j (key), while M is the total patch count, H is the number of heads, $D_{i,h}$ is the non-locality metric for layer i at head h and D_i is the average non-locality metric for that layer. Because $||\delta_{m,j}||$ remains constant (patch positions do not change throughout the network), the main determinant of $D_{i,h}$ is the set of attention scores $A_{m,j}^{i,h}$[154], [155].

$$\begin{aligned} D_i &= \tfrac{1}{H}\textstyle\sum_{h=1}^{H} D_{i,h} \\ D_{i,h} &= \tfrac{1}{M}\textstyle\sum_{m,j=1}^{M} A_{m,j}^{i,h}\,\|\delta_{m,j}\| \\ A_{m,j}^{i,h} &= \xi A_{m,j}^{i,h} + \left(1-\xi\right) A_{m,j}^{i-1,h} \end{aligned} \tag{10.5}$$

When residual attention is introduced, attention scores from the current layer *i* are combined with those from the previous layer *i-1*. This combination is governed by a parameter $\xi \in [0,1]$, which balances the contribution from the current and earlier layers. Consequently, the effective globality of the layer at head *h*, as represented by the updated attention scores, is given by a corresponding formula Eq. (10.6) [155].

$$D_{i,h} = \frac{1}{M}\sum_{m=1}^{M}\sum_{j=1}^{M}\left[\xi A_{m,j}^{i,h} + (1-\xi)A_{m,j}^{i-1,h}\right]\|\delta_{m,j}\| \,. \tag{10.6}$$

Because earlier layers generally display more localized focus than later ones, mixing attention scores from both sources tends to produce a matrix of weights with reduced globality relative to using the current layer alone. In other words, integrating previous-layer scores into the current layer's attention through residual connections at each head moderates the rate at which the attention mechanism expands its global coverage.

10.2. Deep Residual ViT Evaluation

The quality of data classification by Deep Residual ViT is verified by calculating the same measures that were used to evaluate the ViT model, i.e. precision, recall, F1-score, specificity. Additionally, learning curves and confusion

matrices are presented. The obtained results are illustrated in Fig. 10.4–10.6 and in Table 10.1.

Table 10.1. The obtained results for RP and AVL datasets classification by Deep Residual ViT model. All values in %.

Dataset	Class	Precision	Recall	F1-score	Specificity
RP dataset	RP	98.02	96.12	97.07	97.53
	CORD	85.71	96.00	90.50	97.48
	HEALTHY	98.18	96.43	97.30	99.22
AVL dataset	AVL	85.71	94.74	89.90	98.28
	DRUSEN	98.85	98.85	98.85	99.06
	HEALTHY	100.00	97.70	98.83	100.00

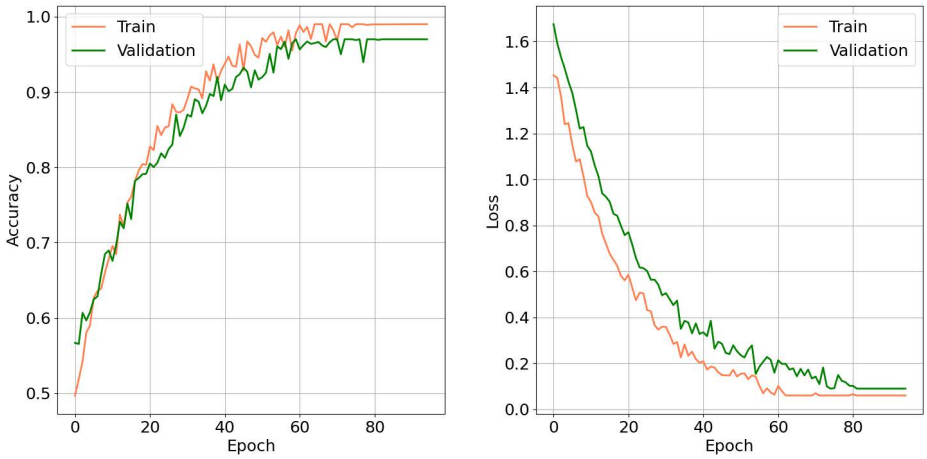


Fig. 10.4. Deep Residual ViT Training and validation accuracies (a) and loss (b) obtained for RP dataset. Compared to the baseline ViT results (see Fig. 9.20), the residual self-attention architecture yields faster convergence, lower validation loss throughout training, and improved stability of validation accuracy.

The results for Deep Residual ViT (Table 10.1) clearly indicate further improvement compared to the base ViT architecture (Table 9.1). On the RP dataset, the increase in the RP and CORD classes is particularly noticeable. For RP, recall increases from 93.20% (ViT) to 96.12%, which results in an increase in F1-score from 96.00% to 97.07%. An even greater difference is noted in the CORD class – precision increases from 74.19% to 85.71%, and recall from 92.00% to 96.00%, which translates into an F1-score of 90.50%. The classification of healthy cases also improves, with F1-score increasing from 92.86% to 97.30%, and specificity reaching almost 99.22%.

A similar trend is observed in the AVL set, where Deep Residual ViT achieves exceptionally high metrics for the DRUSEN and HEALTHY classes (F1-score 98.85% and 98.83%, respectively). The HEALTHY class achieves even 100% precision, which minimizes false alarms. Additionally, for the AVL class itself, high sensitivity (94.74%) is maintained, which prevents missing important disease cases. The analysis of the learning curves (Fig. 10.5 and Fig. 10.6) suggests that the deeper network with residual blocks does not lead to overfitting, but rather supports a better balance between precision and sensitivity.

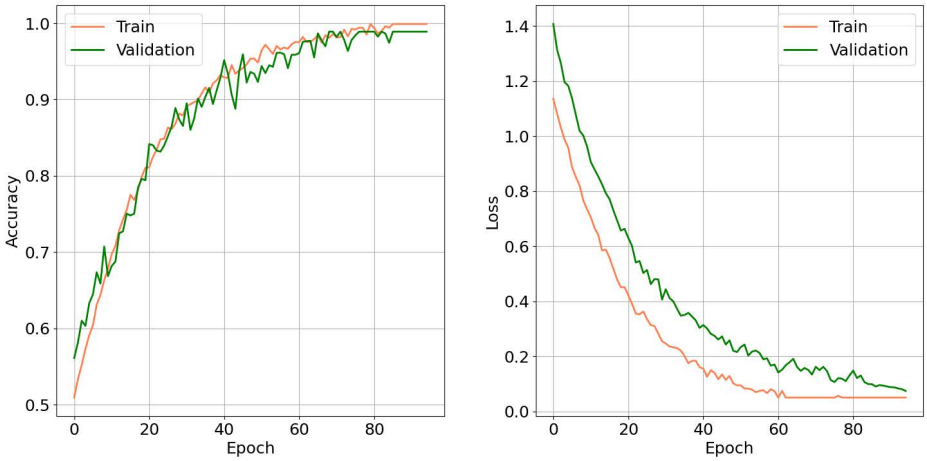


Fig. 10.5. Deep Residual ViT training and validation accuracies (a) and loss (b) obtained for AVL dataset. Compared to the baseline ViT (see Fig. 9.21), the Deep Residual ViT achieve higher accuracy results (up to 99% for validation dataset).

Compared to ViT, the residual method is characterized by faster learning stabilization, which is visible in the slight difference between training and validation accuracy in the final epochs. All this confirms that incorporating residual blocks into the Vision Transformers architecture is an effective way to achieve even more accurate classification of rare eye diseases.

A quantitative assessment of the computational cost of the proposed architecture and of the contribution of its residual attention mechanisms was carried out by means of a complexity analysis together with a targeted ablation study. In Table 10.2, a conventional Vision Transformer, used as the reference baseline, is contrasted with the proposed Deep Residual Vision Transformer. By introducing residual self-attention and cross-block attention aggregation, an increase in the number of parameters (from 86.1 M to 92.4 M) and in the inference budget (from 18.9 GFLOPs to 20.4 GFLOPs) is observed. This

increase is associated only with a modest rise in latency: a difference of a few milliseconds on an RTX 4090 GPU and a few tens of milliseconds on a single Ryzen 7950X CPU core. As a result, DR–ViT is still considered computationally feasible for near-real-time clinical use, while full spatial resolution of the input is preserved, which is required for the reliable assessment of fine retinal structure.

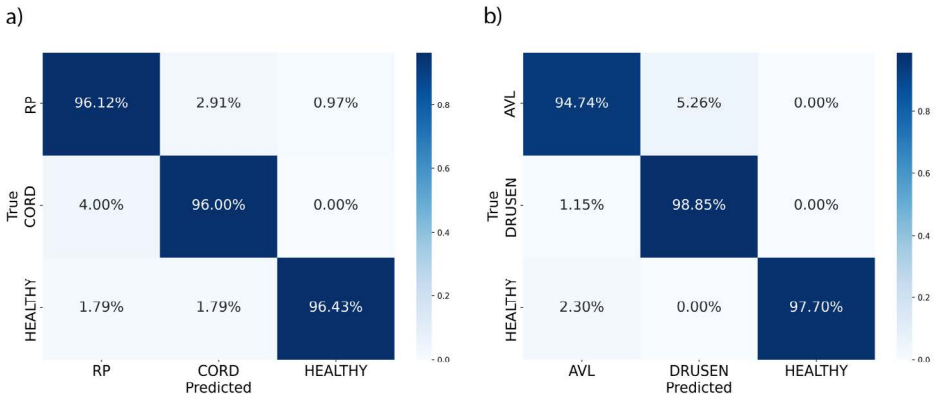


Fig. 10.6. Testing accuracies obtained for Deep Residual ViT model for a) RP dataset b) AVL dataset. Compared to the baseline ViT results (see Fig. 9.22), the residual architecture yields a consistent improvement in classification accuracy across all classes. For the RP dataset, the correct recognition rate increases from 93.20% to 96.12% for RP and from 92.86% to 96.43% for HEALTHY. Similarly, on the AVL dataset, the accuracy improves from 92.23% to 94.74% for AVL and from 96.00% to 98.85% for DRUSEN, confirming the effectiveness of residual self-attention in enhancing class separability and overall model robustness.

Table 10.2. Comparison of inference complexity and speed for the baseline ViT and Deep Residual ViT variants. Timing measurements for RTX 4090 and single-core Ryzen 7950X.

Model	Params [M]	GFLOPs	RTX 4090 [ms]	Ryzen 7950X [ms]
Baseline ViT	86.1	18.9	58.4	540
Deep Residual ViT	92.4	20.4	64.1	620

In Table 10.3, a detailed FLOPs breakdown is provided. The dominant share of computational cost is shown to be concentrated in the mid-depth and deep encoder blocks (blocks 5–8 and 9–12), where residual self–attention is applied and attention maps are propagated and fused across depth. These blocks are assigned the role of constructing global spatial context, i.e. integrating information across the entire fundus, including subtle boundaries between pathological and healthy tissue. Additional components – namely

the propagation of queries/keys (Q/K) between blocks and the hierarchical mixing of attention weights α – are shown to introduce non-trivial overhead with respect to standard self-attention. However, these mechanisms are also observed to act as a stabilizing pathway that is analogous to skip connections in convolutional networks: earlier attention structure is not discarded but instead is refined progressively across depth, thereby preventing late-layer collapse of representations.

Table 10.3. FLOPs for a single image *352×352*. Values in billions of floating-point operations (GFLOPs) Encoder block sections correspond to the grouped transformer encoder blocks: layers 1–4 are standard MHSA with MLP, layers 5–8 and 9–12 contain a residual self-attention mechanism with query/key (*Q/K*) transfer and mixing of the attention matrix between blocks. FFN stands for two-layer fully connected network.

Layer no.	Layer	Configuration/ size	FLOPs [G]	Part
Image tokenization / embedding				
1.	Patch Embed	(Conv/Linear *16×16* patch→d = 768, ~175 tokens)	1.2	5.9%
Encoder block 1–4 (high local context resolution)				
2.	MSHA		2.7	13.2%
3.	MLP	(FFN 768→3072→768)	2.1	10.3%
Encoder block 5–8 (mixed local/global context)				
4.	RSA	(MSHA + *Q/K* injection form previous blocks)	3.4	16.7%
5.	α aggregation	(attention weights of previous/ deeper layers)	1.3	6.4%
6.	MLP	(FFN 768→3072→768)	2.0	9.8%
Encoder block 9–12 (full global context with gradient stabilization)				
7.		(multi-head attention with cumulative historical attention matrices)	3.6	17.6%
8.	α aggregation	(attention matrix merging between heads/depths)	1.4	6.9%
9.	MLP	(FFN 768→3072→768)	1.9	9.3%
Sequential section and classification				
10.	Token head	(768→3)	0.8	3.9%
Total			**20.4**	**100.0%**

The most clinically relevant evidence is provided by the ablation study in Table 10.4. The full DR–ViT is reported to achieve the highest macro-averaged Precision, Recall, and F1-score on both datasets, and is shown to outperform the baseline ViT variant in which residual self–attention is not

used. When residual self–attention is fully removed, a marked drop in F1-score is observed. This degradation is interpreted as evidence that the observed gain is not solely attributable to increased capacity, but rather to the routing and reuse of information between layers. Additional ablations indicate that disabling only cross-block *Q/K* propagation or disabling only attention-weight aggregation α results in measurable performance deterioration, in particular in Recall, meaning that more pathological cases are missed. On this basis, the residual attention pathway is argued to provide clinically meaningful value: lesion sensitivity is improved at a computational cost that remains compatible with interactive or near real-time decision support.

Table 10.4. Ablation study for Deep Residual ViT model (Mean values in %).

Dataset	Retinitis Pigmentosa			AVL		
Model	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Deep Residual ViT	93.97	98.18	94.96	94.85	97.10	95.86
– RSA	88.67	92.69	90.33	92.30	93.42	92.84
– mixing attention weights α (only *Q/K* forwarding between blocks preserved)	93.50	95.70	94.50	94.44	96.50	95.40
– *Q/K* transfer between blocks (remains aggregation between layers)	92.80	95.10	93.70	93.70	95.90	94.90
Shallower encoder 6 blocks instead of 12	91.40	94.20	92.80	93.15	95.05	94.22

10.3. Chapter Summary

Deep Residual Self-Attention ViT introduces additional skip connections in the self-attention blocks. It helps to preserve feature diversity and reduce the risk of losing key image details. As observed in Subchapter 10.1, combining queries and keys from previous layers (residual self-attention) can, in some cases, decrease globality, since part of the attention is pulled from earlier, more localized contexts. This promotes a better balance between local and global features. Moreover, it effectively quells the feature-collapsing phenomenon.

Thereby, the model can hold much more distinct information from different regions of the retina, which is quite important when dealing with noisy medical images.

In the studies presented herein, five core metrics: accuracy, precision, recall, F1-score and specificity are used to measure the performance of classification. Values of precision and sensitivity were consistently high and differed just by a small percentage. This becomes very important in medical applications where the model has to be very reliable in picking up both cases of disease and health. For practical clinical utility, the balance struck is that of minimizing false negatives, (i.e., overlooking pathology), and false positives (healthy patients labelled as sick).

The obtained results confirm that Deep Residual Self-Attention ViT achieves this balance while maintaining high precision. On a higher level, the study contributes to reducing computational overhead in high-resolution ViTs, since the residual pathways guide the model more efficiently toward crucial features, even though further optimization is possible. Additionally, the residual approach enhances regularization, particularly in small, specialized medical datasets, by effectively preserving mid- and low-level features.

11. Ensemble Learning Algorithms

Ensemble learning is one of many methods for aggregating and improving classification results [156]. It has emerged as a valuable approach for improving classification performance, especially in challenging medical imaging scenarios where data can be scarce or highly imbalanced. Recent investigations into rare eye diseases have demonstrated the advantages of ensemble methods when confronted with limited datasets and heterogeneous clinical presentations [157], [158]. By combining the outputs of multiple base models, ensembles can better capture subtle disease patterns, reducing false negatives that might otherwise result from the noisy and complex nature of fundus or optical coherence tomography images. The general idea of ensemble learning process is depicted in Fig. 11.1.

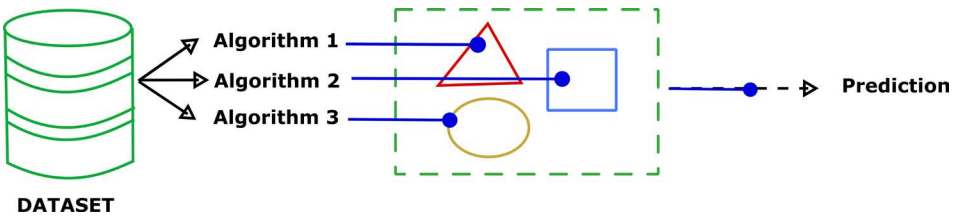


Fig. 11.1. A general scheme of ensemble learning.

One frequently employed ensemble technique is Bagging, which draws multiple bootstrap samples from the training set to train independent classifiers. This method has proven particularly adept at stabilizing performance in the presence of patient-specific variations, as each model in the ensemble captures different aspects of the underlying pathology [159]. Boosting, another well-regarded ensemble strategy, provides a complementary approach by iteratively refining weaker models. Studies focusing on rare ocular conditions have shown that boosting-based methods can focus on the most difficult-to-classify images, adjusting the learning process to correct misclassifications from previous iterations [145]. Voting ensembles use either simple majority or weighted strategies to aggregate predictions from a range of classifiers, such as convolutional neural networks and random forests, thereby leveraging consensus to bolster diagnostic accuracy [157].

11.1. Bootstrap Aggregating

Bootstrap Aggregating, often referred to as Bagging, is an ensemble method designed to enhance the stability and accuracy of machine learning models by systematically manipulating the training data [160]. Its key operation involves creating multiple bootstrap subsets from the original dataset, typically sampled with replacement, so that certain instances appear multiple times while others do not appear at all in a given subset. Each subset is then used to train an independent base learner, often of the same algorithmic type, such as decision trees or neural networks [161]. Once the individual models have been trained, Bagging combines their predictions through majority voting. The whole procedure is depicted in Fig. 11.2.

The procedure begins by partitioning the original dataset into multiple draws of approximately the same size as the initial training set, but each draw is taken randomly and with replacement. Consequently, each bootstrap draw contains parts of unique instances of the original data, and the remainder are duplicates. This approach highlights different patterns in each training subset, promoting diversity among the base models and reducing their correlation. As these individual classifiers will likely make different errors, the aggregation step leverages collective wisdom to decrease overall variance and maintain, or sometimes even lower, the bias [162]. The result is an ensemble that is more

resistant to overfitting than a single learner is, particularly in scenarios where data may be limited or noisy.

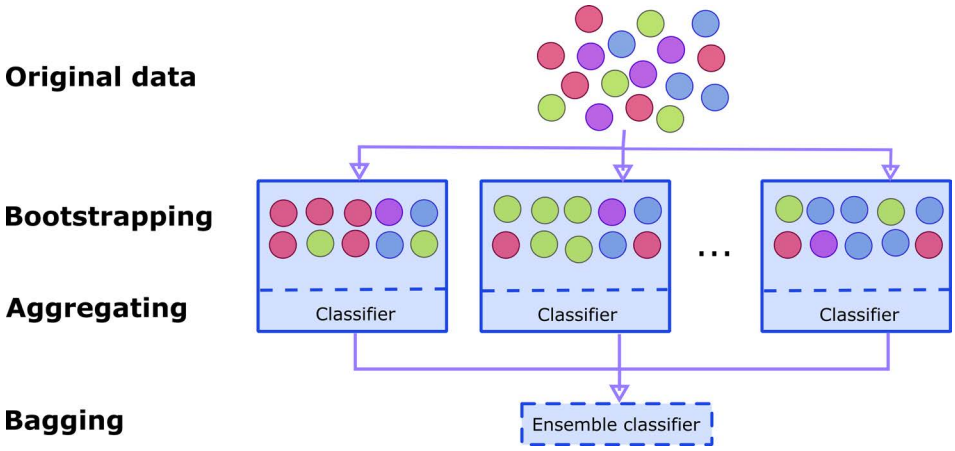


Fig. 11.2. The general scheme of a bagging technique. It should be noted that training sets for separate classifier consist of random subsets of the main set.

In medical image analysis and especially in the classification of rare diseases, Bagging proves beneficial by mitigating the risk of over-reliance on a small subset of high-prevalence examples. Through repeated sampling, it encourages each model to learn varied decision boundaries, ultimately supporting more reliable predictions across diverse patient cohorts [161]. Bagging has thus become a cornerstone in modern ensemble learning strategies, demonstrating versatility across classification, regression and anomaly detection [160].

11.2. Boosting

Another team learning technique is Boosting. It aims to convert a series of weak learners into a single, more powerful predictive model by iteratively reassigning weights to misclassified instances. Unlike Bagging, which trains multiple independent learners on different bootstrap samples, Boosting follows a sequential training procedure, where each new learner focuses on improving the errors of its predecessors [163] (see Fig. 11.3). The underlying principle is to adaptively emphasize previously misclassified examples, thereby enabling the ensemble to systematically reduce both bias and variance.

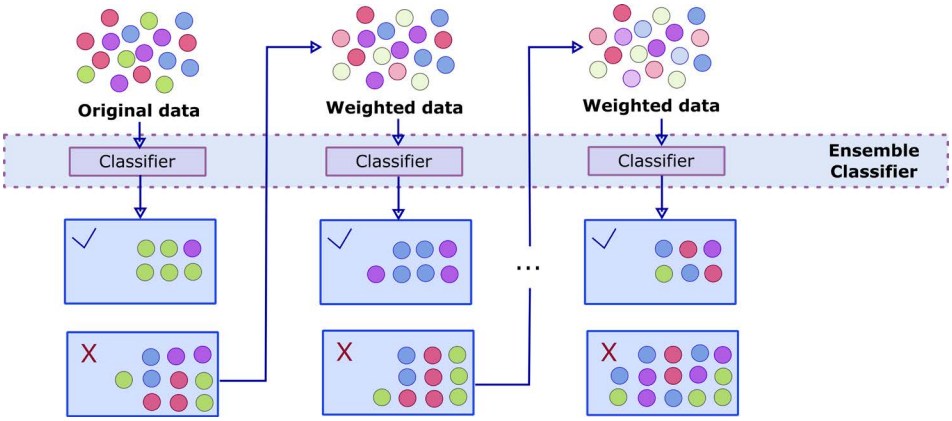


Fig. 11.3. The general scheme of a boosting technique. It should be noted that subsequent training sets do not contain data correctly classified in previous step.

A typical Boosting algorithm begins by training an initial weak model on the full training dataset. Following evaluation, higher weights are assigned to the samples that the current model misclassifies, while the weights of correctly classified samples are decreased. This weighting scheme compels the next model to devote greater attention to the harder-to-classify instances [164]. The new model is then trained on this reweighted dataset, and the process repeats for a specified number of iterations or until a desired performance threshold is reached. Thus, each iteration produces an updated learner that is better at addressing the gaps left by the previous models, leading to a cohesive ensemble of incrementally refined predictors.

The final prediction is typically determined by combining all classifiers, with each one's influence weighted according to its performance. For classification tasks, a weighted majority voting can be employed [165]. By continually adjusting the training process, Boosting is able to enhance overall accuracy, even when dealing with noisy data or limited datasets. This characteristic makes it especially valuable in medical applications, where the data often exhibits high complexity or class imbalance.

One family of Boosting algorithms is Gradient Boosting, which iteratively fits new models to the residuals of the previous ensemble's predictions. This approach optimizes a loss function by gradient descent in a function space. Algorithms like Gradient Boosted Decision Trees (GBDT) and its variants, such as XGBoost, LightGBM and CatBoost [163] are often applied.

11.3. Stacking

Stacking, often referred to as stacked generalization, is an ensemble method that combines multiple predictive models by leveraging a meta-learner trained on their collective outputs. Unlike Bagging and Boosting, which primarily focus on either sampling strategies or sequential weight adjustments, Stacking adopts a two-level process designed to capture complex relationships among different base learners [163]. The whole Stacking procedure is depicted in Fig. 11.4.

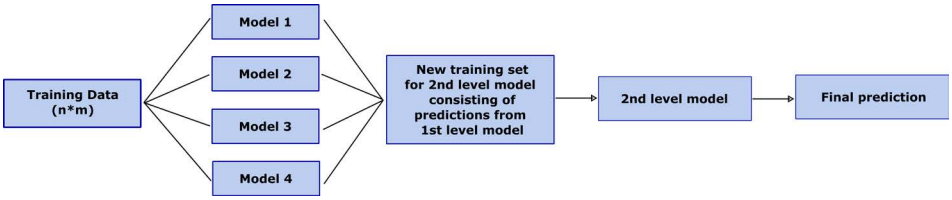


Fig. 11.4. The general scheme of a stacking technique. Stacking consists of two-levels estimators. The first level is built from all the base models that are used to predict the results on the test data sets. The second level consists of a Meta-Classifier that takes all the predictions of the base models as input and generates new predictions.

The first level consists of several base models – these can be of the same type (e.g., multiple neural networks) or entirely different algorithms (e.g., decision trees, support vector machines and logistic regression). Each base model is trained independently on the same training dataset, allowing each to learn diverse characteristics and patterns. Once these base models have been trained, they generate predictions for each sample in the training set. For classification tasks, these predictions can be class probabilities or discrete class labels [166].

These first-level predictions then form a new training set for the second level, sometimes called the meta-level or meta-learner. In this step, each instance in the meta-level dataset comprises the predictions from all base models as inputs, alongside the true labels from the original dataset. By analysing how the first-level models perform collectively, the meta-learner can discern intricate patterns, effectively learning the optimal way to aggregate the base models' outputs [167]. This structure allows the stacked ensemble to rectify the limitations of any single model by exploiting their complementary strengths.

Once training is complete, the meta-learner produces the final prediction. The key advantage is that Stacking provides an expressive framework for combining different algorithms, which can be particularly beneficial in domains characterized by heterogeneity or high dimensionality, such as genomics,

medical image analysis. By capitalizing on the diversity of base models, Stacking often achieves higher accuracy and better generalization compared to simpler ensembles [163], [164], [166].

11.4. Voting

Voting is a fundamental ensemble learning technique that combines the predictions of multiple models to produce a more stable and often more accurate final output. In contrast to Boosting, which builds models sequentially while emphasizing misclassified instances, and Bagging, which samples data to reduce variance, voting focuses on aggregating different perspectives by simply pooling the decisions of various independently trained base learners [168]. The rationale behind this method is that individual biases or errors in each model can be mitigated or cancelled out when the predictions are combined.

The voting procedure typically starts with training multiple base classifiers on the same dataset. These base classifiers can be homogeneous, such as several decision trees or neural networks, or heterogeneous, like a mix of decision trees, support vector machines and logistic regression [169]. After the base learners are fitted, each of them predicts the class. The predictions from all classifiers are then aggregated to deliver a consensus decision.

Two main types of voting approaches exist: hard voting and soft voting. In hard voting, each model's prediction counts as a single vote, and the final output is determined by the majority or plurality of votes. While straightforward, hard voting can disregard useful information about classifier confidence. Soft voting, on the other hand, integrates class probabilities or confidence scores, weighting each classifier's vote according to how certain it is about its prediction. This approach can lead to more nuanced ensemble decisions, particularly for datasets with overlaps or class imbalances [170].

11.5. Classification Results

In order to aggregate the results, the datasets described in Chapter 3 are divided into 5 bootstrap samples. The models that achieve the highest classification performance are selected for aggregation, i.e. ViT, Deep Residual ViT, CNN-GRU

with Kronecker convolution, GRU U-Net with Kronecker convolution, RAN with Kronecker convolution. Each model is trained on each bootstrap set. Due to the similar quality of classification parameters, the final response of the bootstrap model is determined by majority voting. The aggregated results are presented in Table 11.1.

Table 11.1. The obtained results for RP and AVL datasets classification by bootstrapping ensemble model. All values in %.

Dataset	Class	Accuracy	Precision	Recall	F1-score	Specificity
RP dataset	RP	99.03	99.03	100.00	99.51	98.78
	CORD	100.00	100.00	96.15	98.04	100.00
	HEALTHY	98.21	98.21	98.21	98.21	99.22
AVL dataset	AVL	100.00	100.00	100.00	100.00	100.00
	DRUSEN	98.85	98.85	98.85	98.85	99.06
	HEALTHY	98.85	98.85	98.85	98.85	99.06

By analysing, obtained in these studies, the confusion matrices, one can deduce which classes are particularly hard to distinguish for individual models. In the case of the RP dataset, the most difficult class is CORD and HEALTHY, which are often confused with each other. For example, the CNN-GRU and ViT models have relatively frequent errors in recognizing these two classes. In the case of the AVL dataset, on the other hand, the classification of the DRUSEN class and partially HEALTHY is much more difficult. For example, CNN-GRU and GRU U-Net sometimes incorrectly assign images of the DRUSEN class to HEALTHY and vice versa. When constructing the first base model, the one that achieves the best results is used – Deep Residual ViT. For the RP dataset it has the highest recall in all classes (e.g. CORD: 96%, RP: 96.12%, HEALTHY: 96.43%), similarly for the AVL dataset (AVL: 94.74%, DRUSEN: 98.85%, HEALTHY: 97.70%). After training the first classifier, the weights of the training samples that are incorrectly classified (based on the classification on the validation set) are increased. Particular attention is paid to the CORD class samples (for the RP dataset) and AVL and DRUSEN (for the AVL dataset), because these classes caused problems for other models. Then, the next models are trained in a similar way, increasing the weights of the examples that were incorrectly classified by the first model. Weighted voting is used to aggregate the results of all models. The applied boosting ensemble procedure is presented in Algorithm 11.1. The obtained results are presented in Table 11.2.

Table 11.2. The obtained results for RP and AVL datasets classification by bagging ensemble model. All values in %.

Dataset	Class	Accuracy	Precision	Recall	F1-score	Specificity
RP dataset	RP	100.00	100.00	100.00	100.00	100.00
	CORD	96.00	96.00	96.00	96.00	99.37
	HEALTHY	98.21	98.21	98.21	98.21	99.22
AVL dataset	AVL	100.00	100.00	100.00	100.00	100.00
	DRUSEN	98.85	98.85	98.85	98.85	99.06
	HEALTHY	98.85	98.85	98.85	98.85	99.06

Another method for aggregating classification results that has been explored is stacking. Here, XGBoost, described in Chapter 4, is applied as a metamodel. In this method, cross-validation is used to generate predictions for training the metamodel. Finally, the stacking model is tested on the same data set as all models in this monograph. The stacking results are presented in Table 11.3.

The meta-classifier will learn to use the strengths of each model, e.g.: Deep Residual ViT classifies DRUSEN very well, while RAN or GRU U-Net classify CORD more effectively. Due to stacking, the meta-model will automatically learn, for example, that for a sample suspected of DRUSEN, it is worth trusting the Deep Residual ViT prediction more, and for a sample suspected of CORD, e.g., it is worth trusting the RAN or GRU U-Net model more.

Table 11.3. The obtained results for RP and AVL datasets classification by stacking ensemble model. All values in %.

Dataset	Class	Accuracy	Precision	Recall	F1-score	Specificity
RP dataset	RP	100.00	100.00	100.00	100.00	100.00
	CORD	96.00	96.00	96.00	96.00	99.37
	HEALTHY	100.00	100.00	100.00	100.00	100.00
AVL dataset	AVL	100.00	95.00	100.00	97.40	99.43
	DRUSEN	98.85	100.00	98.85	99.40	100.00
	HEALTHY	100.00	100.00	100.00	100.00	100.00

It should be noted that all ensemble methods give very similar results. Relatively "low values", around 96%, are largely due to a small test set. It should be emphasized that one incorrectly classified image can reduce the values of the obtained parameters by even a few percentage points.

Algorithm 11.1. Proposed boosting ensemble procedure with weighted voting.

Require: Train set D_{train}, Validation set D_{val}, Test set D_{test}
Eqnure: The final ensemble classifier $M_{ensemble}$

1: Step 1: Class-Level Error Analysis

2: Step 2: Training the First Base Model

3: 2.1. Train the first classifier, e.g. *Deep Residual ViT*, on the set D_{train} in the standard way.

4: 2.2. Evaluate the model on the validation set D_{val} and note which examples were misclassified.

5: 2.3. The analysis at this stage does not concern the test set D_{test} – it remains only for final evaluation.

6: Step 3: Increase the weights of misclassified examples

7: 3.1. Raise the weights w_i of those samples that were misidentified by the model in Step 2. This can be done e.g. by:
$w_i \leftarrow w_i \times \alpha, \alpha > 1$ is the coefficient boosting the weight.

8: 3.2. Special attention is paid to samples from the most problematic classes (e.g. CORD or DRUSEN), giving them even higher weights if necessary.

9: Step 4: Train subsequent models iteratively

10: 4.1. On the modified set (updated weights in D_{train}, train another model.

11: 4.2. Re-evaluate the newly trained model on D_{val}, identify mispredictions, and increase the weights of misclassified samples if necessary.

12: 4.3. Train the next models in the same way. (e.g. *GRU U-Net*}, *RAN*, etc.), each time:

Increasing the weights of examples incorrectly classified by previous models,

Training the next model on the updated set,

Analyzing the results on validation.

13: Step 5: Weighted Voting Based on the Confusion Matrix

14: 5.1. There is a set of trained classifiers $M_1, M_2, \ldots, M_k$. Each model M_j may have different performance in different classes – this should be determined from the confusion matrix on D_{val}.

15: 5.2. Set the weights w_j^c for model M_j with respect to class c:

The highest weight *Deep Residual ViT* in class DRUSEN,

The average weights for models *GRU U-Net* and *RAN*,

Lower weights for *ViT* and *CNN-GRU* in those classes where they are often confused (e.g. DRUSEN, CORD).

16: 5.3. Create the final team prediction:

$$\hat{y} = \arg\max_{c} \sum_{j=1}^{k} \left(w_j^{(c)} \cdot p_j\left(c \mid x\right) \right),$$

where $p_j(c|x)$ is the probability of assigning class c by model M_j.

17: Step 6: Evaluate the quality of the final model on test set

18: return: The final ensemble classifier $M_{ensemble}$.

12. Interpretability and Explainability in Image Classification Models

Interpreting the internal workings of deep neural networks has become a critical research focus, particularly in image classification tasks where opaque black-box models often hamper user trust and effective decision-making [171], [172]. As the popularity of deep learning has increased, concerns over its lack of transparency have simultaneously intensified. Even in fields where model outputs can have life-critical implications, such as medical imaging, deep learning systems frequently provide limited insight into why certain predictions are made [173].

In response to this challenge, Explainable Artificial Intelligence (XAI) methods have been developed to reveal which features or image regions contribute most strongly to a model's decision [174]. Among many proposed techniques, Gradient-weighted Class Activation Mapping (Grad-CAM) and SHapley Additive exPlanations (SHAP) are two complementary approaches that empower practitioners to better understand and interpret computer vision models [175], [176]. These techniques not only aid in debugging complex architectures and improving model reliability, but they also enhance user confidence and open the door to more ethically guided AI solutions.

12.1. Grad-CAM

Deep learning architectures often pose challenges when attempting to elucidate the rationale behind their outputs, particularly in tasks like object classification or localization. Grad-CAM is introduced as a method to address these challenges by illuminating the regions of an image that most strongly contribute to a particular prediction [177]. In Grad-CAM, the gradient of the target class score is computed with respect to the feature maps in the final convolutional layer, which typically captures high-level, semantically meaningful patterns. These gradients are then averaged and used to weigh the corresponding feature maps. The resulting heatmap is typically overlaid on the original image to visually represent which areas of the image are most significant for the model's decision. Warmer colours (like red and yellow) indicate higher importance, while cooler colours (like blue) indicate lower importance.

Once the process of assigning weights to the corresponding feature maps is completed, the resulting maps are summed and passed through a *ReLU* function, thereby discarding negative contributions and focusing on positively influential areas. The outcome is a heatmap that can be superimposed on the original image, illustrating the features or regions that drive the classification. This selective highlighting provides insights into whether the model is attending to relevant aspects of the image or potentially focusing on artifacts.

To generate a Grad-CAM visualization, a specific target class is selected to interpret the model's decision. Grad-CAM applies the gradients of the class score (the output for the selected class) with respect to the feature maps of a chosen convolutional layer, often the last convolutional layer, during a backward pass. The derivative of the class score with respect to these feature maps indicates how the final prediction changes if the feature map values are adjusted, thus revealing each feature map's contribution to the output. The gradients are then globally averaged (pooled) across the height and width of the feature maps, producing a weight for each feature map. These weights form the basis for creating a coarse localization map Eq. (12.1) [178].

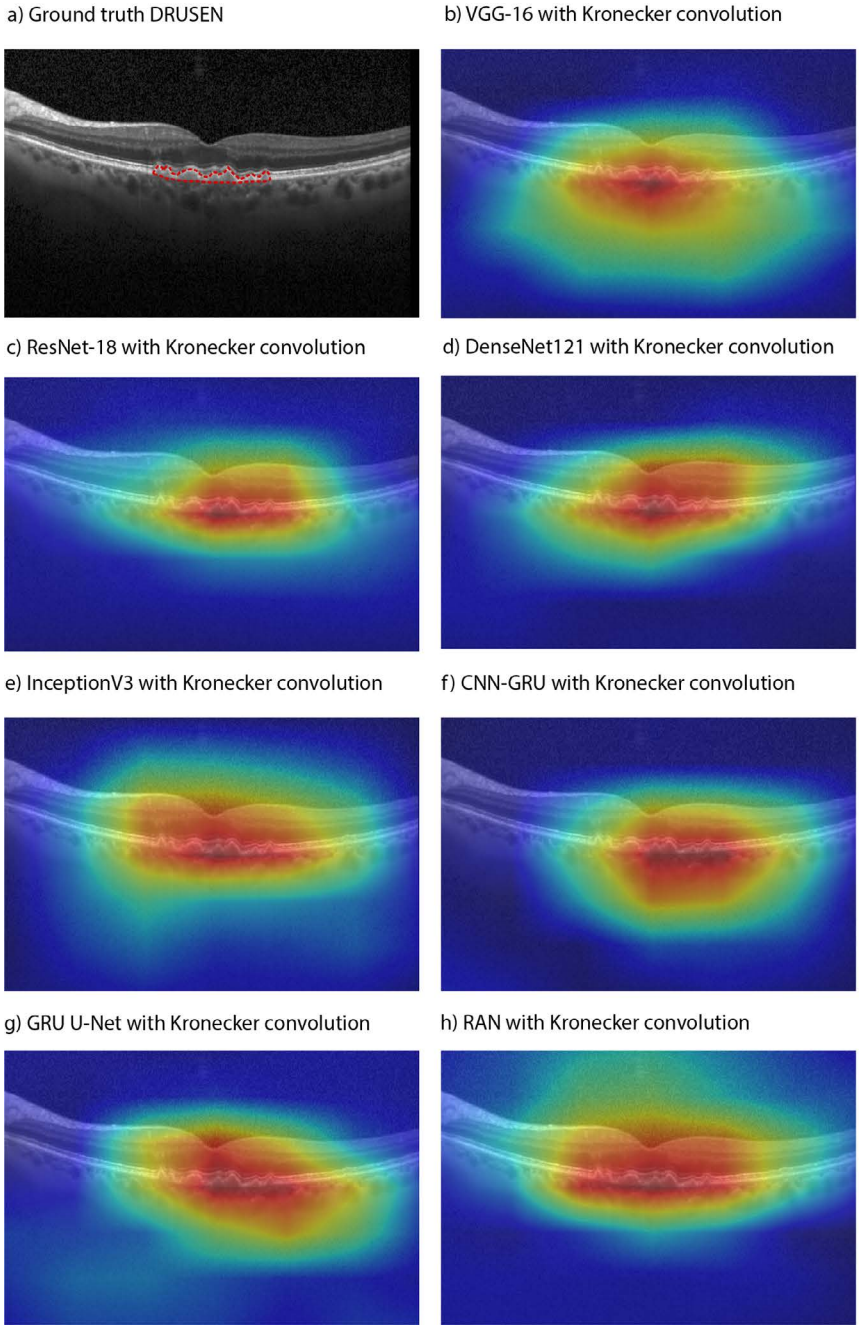


Fig. 12.1. An example of Grad-CAM images generated for different models. More red regions indicate more attention of the model. Image a) shows the reference OCT image with clinically relevant regions annotated by an expert. Images b)–h) present Grad-CAM heatmaps produced by VGG-16, ResNet-18, DenseNet121, InceptionV3, CNN–GRU, GRU U-Net, RAN, respectively, all incorporating Kronecker convolution. Warmer colours (red/yellow) indicate regions that contribute most strongly to the model's prediction.

$$\alpha_i = \frac{1}{N} \sum_k \sum_j \frac{\partial y^c}{\partial A^i_{kj}}. \tag{12.1}$$

In Eq. 12.1 α_i indicates the weight for feature map i. N is a number of pixels in the map, y^c represents the score value for class c and A^i_{kj} denotes the activation of i-th feature map at location (k, j). A weight for each feature map that represents its importance in the context of the target class. Each feature map is multiplied by its respective weight, and the weighted feature maps are summed to obtain a single two-dimensional heatmap Eq. (12.2) [178].

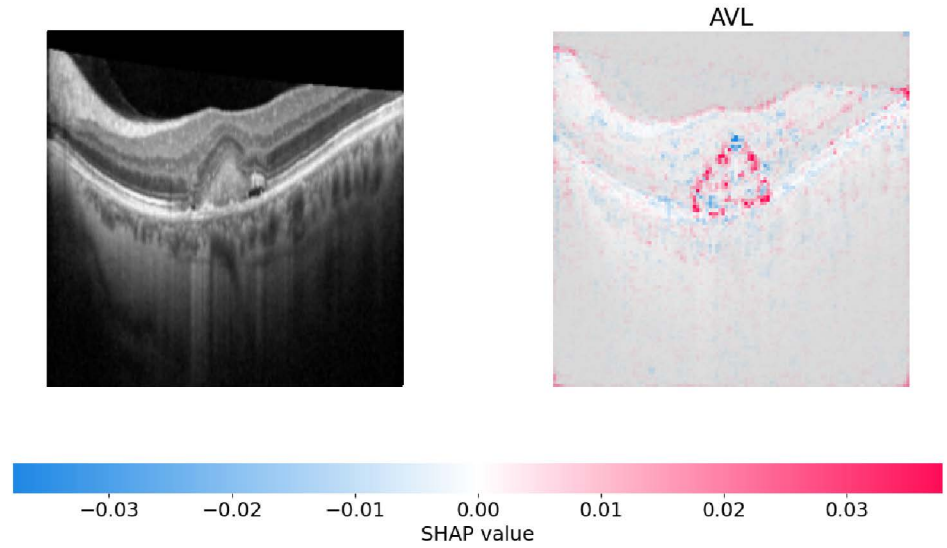
$$\text{Grad-CAM} = ReLU\left(\sum_i \alpha_i A^k\right). \tag{12.2}$$

A *ReLU* activation function is used to emphasize positive contributions, generating a heatmap that highlights key regions for the target class. This heatmap can be normalized to map its values to the range [0,1], enhancing its visualization. Afterward, the heatmap is resized to align with the dimensions of the original input image. Finally, it is superimposed on the original image, often using a colour map to represent areas of greater significance. Example Grad-CAM images for the models used in this monograph are presented in Fig. 12.1.

12.2. SHAP Value

While Grad-CAMs focus on visually localizing important image regions, SHAP derives from cooperative game theory, offering a more generalizable and model-agnostic perspective [176]. In essence, SHAP values quantify the contribution of each input feature (or image segment in a computer vision context) by considering how the prediction changes when that feature is included versus when it is withheld. Formally, for a set of features F, the contribution Φ_i of the i-th feature, is computed by averaging its marginal contribution over all possible subsets $S \subseteq F \setminus \{1\}$ Eq. (12.3) [176].

a) An example of SHAP values image for Deep Residual Vision Transformers



b) An example of SHAP values image for GRU U-Net with Kronecker convolution

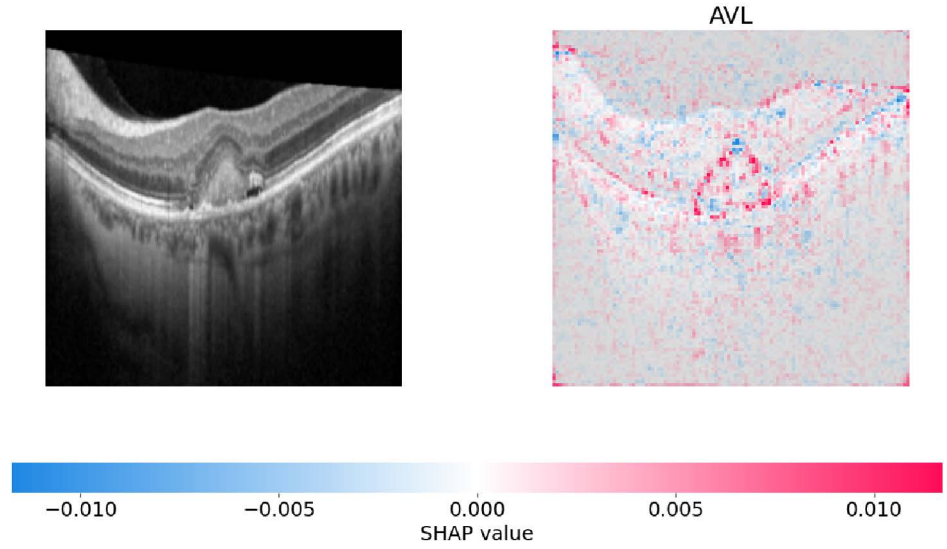
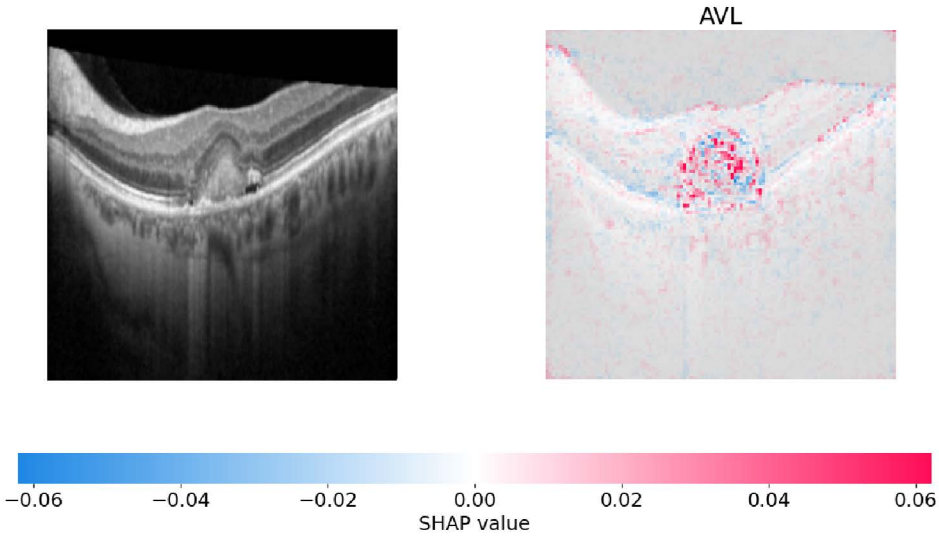


Fig. 12.2. An example of SHAP values images generated for different models. Red regions indicate a positive contribution of a given image region to the model decision, whereas blue regions correspond to a negative contribution.

a) CNN-GRU with Kronecker convolution



b) Residual Attention Network with Kronecker convolution

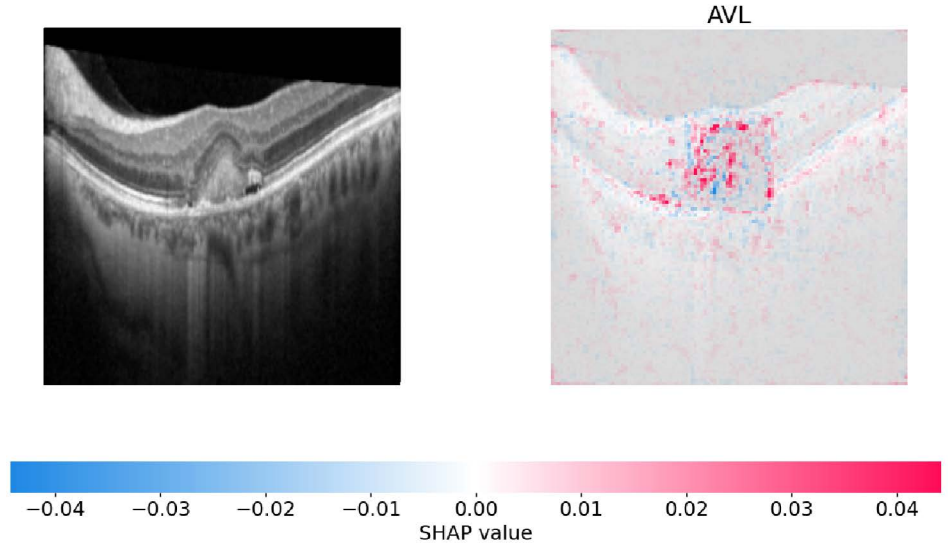


Fig. 12.3. An example of SHAP values images generated for different models. Red regions indicate a positive contribution of a given image region to the model decision, whereas blue regions correspond to a negative contribution.

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f(S \cup \{i\}) - f(S)\ ]\ . \tag{12.3}$$

Where *f(•)* is the model's prediction function. By translating this framework to image classification, one can interpret feature "presence" as including a particular pixel region or latent representation. Such a method enables a granular view of how various segments collectively drive the outcome, making it suitable for complex architectures and heterogeneous datasets [179]. When integrated with domain knowledge, SHAP values can pinpoint subtle predictors, such as small lesions in medical imaging, that Grad-CAMs alone might only partially highlight, thereby furnishing a comprehensive XAI toolkit for deep learning practitioners. The examples of *GradientExplainer* are presented in Fig. 12.2 and Fig. 12.3.

Grad-CAM and SHAP provide complementary forms of model explainability and are designed for different practical purposes. Grad-CAM focuses on spatial localization. It highlights image regions that contribute most to the prediction of a given class. This makes the method intuitive and easy to interpret. The method is computationally efficient and well-suited for rapid model inspection. However, its main limitation is its model dependence and its use primarily in convolutional architectures.

On the other hand, SHAP provides a model-independent explanation framework based on game theory. It quantifies how individual features or image regions influence the final prediction. SHAP presents both positive and negative contributions, allowing for a more detailed and quantitative interpretation of model behaviour. SHAP is particularly useful for analysing subtle image patterns or small pathological changes. However, it is more computationally demanding.

In practice, Grad-CAM is more effective for visual verification, while SHAP is better suited for in-depth analysis and inter-model comparisons. Combining both methods provides a more comprehensive understanding of the decision-making process.

13. Statistical Significance of the Obtained Results

The classical methods for establishing the statistical significance of differences between classifiers treat each prediction as the basic experimental unit and formulate a null hypothesis that two systems have identical expected error. Because the same set of instances is evaluated by every model, observations are naturally paired, violating the independence assumption of simple two-sample proportion tests, however, specialised paired-sample methods correct for this dependence.

McNemar's test addresses the case of two competing classifiers under a dichotomous success-failure view. This test addresses the question of whether two classifiers evaluated on the same set of instances differ in their accuracy.

Consider a binary indicator of correctness for each model on every instance, forming a matched-pairs *2×2* contingency matrix whose cells are traditionally denoted n_{11} (both correct), n_{10} (model A only correct), n_{01} (model B only correct) and n_{00} (both wrong). n_{11} and n_{00} represent agreement, thus they carry no information about relative performance. Under the null hypothesis $H_0: \pi A = \pi B$ that the two marginal success probabilities are equal, the unordered vector (n_{10}, n_{01}) is conditionally distributed as Eq. 13.1 [180]:

$$(n_{10}, n_{01}) \mid m \sim Binomial\left(m, \frac{1}{2}\right), \tag{13.1}$$

where $m = n_{10}+n_{01}$ is the total number of discordant pairs. Large-sample inference applies the Pearson approximation to the difference of the two counts, producing the continuity-corrected statistic (Eq. 13.2) [181]:

$$\chi^2 = \frac{(|n_{10} - n_{01}| - 1)^2}{n_{10} + n_{01}}, \tag{13.2}$$

which, for $m \geq 25$ is asymptotically $X^2_{(1)}$ -distributed. When m is small, the discrete nature of the data renders the approximation inaccurate; an exact test therefore computes the upper-tail probability (Eq. 13.3) [181]:

$$p_{\text{exact}} = 2 \sum_{k=0}^{\min(n_{10},n_{01})} \binom{m}{k} \left(\frac{1}{2}\right)^m, \tag{13.3}$$

optionally reduced by the mid-P adjustment $\tfrac{1}{2}\, Pr(N = k)$ to mitigate conservatism. Rejection of H_0 implies that the classifiers differ significantly in accuracy, with the sign of n_{10}–n_{01} indicating which system prevails. The test assumes that the paired observations are independent across instances, that each outcome is strictly dichotomous, and that the discordant categories are interchangeable. This property is violated by repeated measures or clustered data, for which a clustered McNemar is required. For multiclass predictions the Stuart–Maxwell extension compares the k-dimensional vector of marginal differences and evaluates (Eq. 13.4) [181]:

$$X^2 = \left(d^{\mathrm{T}} \Sigma^{-1} d\right), \tag{13.4}$$

where d collects the $k-1$ independent row-column differences and Σ is their covariance, yielding a $X^2_{(k-1)}$ reference distribution.

Because the McNemar framework is conditioned on the discordant set, it exploits the full pairing of observations and achieves substantially higher power than comparisons based on independent proportions [182].

Effect sizes may be summarised by the paired difference in accuracy, $\hat{\Delta} = \dfrac{n_{10} - n_{01}}{N}$, with 95% Wilson score confidence interval, or by the paired odds ratio $\hat{\theta} = \dfrac{n_{10}}{n_{01}}$ [183]. The statistical significance, demonstrated by the McNemar test, of the classification results obtained using the proposed models is presented in Tables 13.1–13.4.

Table 13.1. McNemar test results: CNN-GRU model vs each baseline models. n_{10} denotes CNN-GRU corrects, baseline model wrong, n_{01} CNN-GRU wrong, baseline model corrects.

Dataset	Baseline model	n_{10}	n_{01}	p-value
RP dataset	DensNet121	4	1	0.37500
	InceptionV3	6	0	0.03125
	ResNet-18	7	0	0.01563
	VGG16	20	0	$2*10^{-6}$
AVL dataset	DensNet121	4	1	0.37500
	InceptionV3	6	0	0.03125
	ResNet-18	7	0	0.01563
	VGG16	20	0	$2*10^{-6}$

The comparison of CNN-GRU model against established convolutional architectures reveals that its gains in overall accuracy are not to be attributable to random variation. Applying McNemar's exact test to the paired correctness indicators on the shared test set shows (Table 13.1) that CNN-GRU model significantly outperforms Inception ($p = 0.0313$), ResNet ($p = 0.0156$) and VGG16 ($p \approx 2*10^{-6}$), while the difference relative to DenseNet does not reach significance ($p = 0.375$). Since McNemar's test conditions on the total number of discordant outcomes, it removes the contribution of cases on which both models agree and thus affords greater sensitivity to true differences in classifier behaviour [181].

Table 13.2. McNemar test results: GRU U-Net model vs each baseline models. n_{10} denotes GRU U-Net corrects, baseline model wrong, n_{01} GRU U-Net wrong, baseline model corrects.

Dataset	Baseline model	n_{10}	n_{01}	p-value
RP dataset	DensNet121	5	1	0.21875
	InceptionV3	7	0	0.01563
	ResNet-18	8	0	0.00781
	VGG16	21	0	$9.53*10^{-7}$
AVL dataset	DensNet121	7	0	0.01562
	InceptionV3	11	0	0.00097
	ResNet-18	14	0	0.00012
	VGG16	25	0	$5.96*10^{-8}$

On the RP test set GRU U-Net demonstrates a statistically significant gain in marginal accuracy relative to Inception, ResNet, and VGG16. The counts of discordant pairs, instances correctly classified by GRU U-Net but misclassified by the baseline, are 7 : 0 for Inception ($p = 0.0156$), 8 : 0 for ResNet ($p = 0.00781$),

and 21 : 0 for VGG16 ($p = 9.53*10^{-7}$), whereas the comparison with DenseNet yields 5 : 1 discordance and a non-significant $p = 0.2188$.

On the AVL dataset, GRU U-Net demonstrates a statistically significant accuracy improvement over all four baselines. The absence of any cases misclassified by GRU U-Net but correctly classified by the others ($n_{01} = 0$ across the board) and the monotonic increase in discordant n_{10} – from 7 against DenseNet to 25 against VGG16, translate into progressively more extreme p-values under the exact binomial null of equal marginal accuracy.

Table 13.3. McNemar test results: RAN model vs each baseline models. n_{10} denotes RAN corrects, baseline model wrong, n_{01} RAN wrong, baseline model corrects.

Dataset	Baseline model	n_{10}	n_{01}	p-value
RP dataset	DensNet121	11	1	0.00634
	InceptionV3	13	0	0.00024
	ResNet-18	14	0	0.00012
	VGG16	27	0	$1.49*10^{-8}$
AVL dataset	DensNet121	10	0	0.00195
	InceptionV3	14	0	0.00012
	ResNet-18	17	0	0.00001
	VGG16	28	0	$7.45*10^{-9}$

Considering RP dataset, the McNemar exact test outcomes for RAN compared to each convolutional baseline. RAN achieves 11 : 1 discordance against DenseNet (exact $p = 0.00635$), 13 : 0 against Inception ($p = 0.000244$), 14 : 0 against ResNet ($p = 0.000122$), and 27:0 against VGG16 ($p \approx 1.49*10^{-8}$). The significant p-values for all comparisons indicate that RAN's superior accuracy is not random.

In case of AVL dataset, RAN demonstrates a highly significant advantage over each of the four convolutional architectures tested. The absence of any instances on which a baseline model outperforms RAN ($n_{01} = 0$) and the increasing discordant counts (n_{10} from 10 against DenseNet up to 28 against VGG16) yield exact p-values well below the conventional $\alpha = 0.05$ threshold.

With eight instances correctly classified, on RP dataset, by Deep Residual ViT but only one instance correctly classified by ViT, a discordance ratio of 8 : 1, the exact McNemar test yields a two-sided p-value of 0.0391 under the null hypothesis of equal marginal accuracy. This p-value is below the conventional 0.05 threshold, indicating that the observed advantage of Deep Residual ViT is not obtained by chance.

Table 13.4. McNemar test results: Deep Residual ViT model vs each baseline models. n_{10} denotes Deep Residual ViT corrects, baseline model wrong, n_{01} Deep Residual ViT wrong, baseline model corrects.

Dataset	Baseline model	n_{10}	n_{01}	*p*-value
RP dataset	ViT	8	1	0.03910
AVL dataset	ViT	4	0	0.12500

Although all four discordant cases favour Deep Residual ViT, in case of AVL dataset, the exact McNemar two-sided *p*-value of 0.125 exceeds the conventional significance threshold of 0.05. Based on the null hypothesis, observing four wins for Deep Residual ViT and none for ViT on the paired test set is not sufficiently unlikely to reject parity. For both analysed datasets (RP and AVL), statistically significant advantages of models with recurrent or hybrid components over conventional convolutional networks were demonstrated. CNN-GRU, GRU U-Net and RAN significantly outperform InceptionV3, ResNet-18 and VGG16. The only exception is the lack of significance in relation to DenseNet121 on RP in the case of CNN-GRU and GRU U-Net. Deep Residual ViT outperforms classic ViT in RP ($p \approx 0.04$), but not in AVL, where the difference turned out to be insignificant. Together, the results confirm that the inclusion of recurrent memory or residual mechanisms translates into a real increase in accuracy, and the applied test allows us to demonstrate this advantage while maintaining statistical rigor.

The monograph also proposes to replace the standard convolution operation with Kronecker convolutions. This change results in the development of new neural models, which are tested on the discussed data sets. The statistical significance of the obtained results is presented in Tables 13.5–13.7.

Table 13.5 presents the paired McNemar analysis contrasting CNN-GRU with four widely used convolutional backbones on the identical test set. The discordant cell counts reveal a consistent pattern in which CNN-GRU corrects substantially more samples than every baseline misclassifies. While the opposite discordance is either absent or limited to a single instance. Gather values indicate that the observed advantages of CNN-GRU over all four comparators are unlikely to have arisen by chance.

The exact McNemar tests presented in Table 13.6 show that the recursively extended U-Net architecture, equipped with Bidirectional GRU units, achieves statistically significant improvement in marginal accuracy on every convolutional baseline tested. In every comparison, the asymmetric distribution of inconsistent predictions favours GRU U-Net.

Table 13.5. McNemar test results: CNN-GRU with Kronecker convolution model vs each baseline models also with Kronecker convolution. n_{10} denotes CNN-GRU corrects, baseline model wrong, n_{01} CNN-GRU wrong, baseline model corrects.

Dataset	Baseline model	n_{10}	n_{01}	*p*-value
RP dataset	DensNet121	6	0	0.03125
	InceptionV3	7	0	0.01562
	ResNet-18	12	1	0.003418
	VGG16	21	0	$9.53 * 10^{-7}$
AVL dataset	DensNet121	6	0	0.03130
	InceptionV3	7	0	0.01560
	ResNet-18	12	1	0.00342
	VGG16	21	0	$9.54 * 10^{-7}$

Table 13.6. McNemar test results: GRU U-Net with Kronecker convolution model vs each baseline models also with Kronecker convolution. n_{10} denotes GRU U-Net corrects, baseline model wrong, n_{01} GRU U-Net wrong, baseline model corrects.

Dataset	Baseline model	n_{10}	n_{01}	*p*-value
RP dataset	DensNet121	8	0	0.00781
	InceptionV3	9	0	0.00391
	ResNet-18	14	1	0.00098
	VGG16	23	0	$2.38 * 10^{-7}$
AVL dataset	DensNet121	12	2	0.01294
	InceptionV3	13	2	0.00738
	ResNet18	18	2	0.00041
	VGG16	22	2	$3.58 * 10^{-5}$

Table 13.7. McNemar test results: RAN with Kronecker convolution model vs each baseline also with Kronecker convolution models. n_{10} denotes RAN corrects, baseline model wrong, n_{01} RAN wrong, baseline model corrects.

Dataset	Baseline model	n_{10}	n_{01}	*p*-value
RP dataset	DensNet121	9	0	0.00391
	InceptionV3	10	0	0.00195
	ResNet-18	15	1	0.00052
	VGG16	24	0	$1.19 * 10^{-7}$
AVL dataset	DensNet121	13	0	0.00024
	InceptionV3	14	0	0.00012
	ResNet-18	19	0	$3.81 * 10^{-5}$
	VGG16	23	0	$2.38 * 10^{-7}$

Exact McNemar testing on the paired prediction set (Table 13.7) shows that the dedicated Recurrent Attention Network attains a statistically significant improvement in marginal accuracy over every convolutional baseline evaluated. The asymmetry of discordant pairs consistently favours RAN: between nine and twenty-four images are classified correctly by RAN but not by the comparator, whereas at most a single image shows the reverse pattern. All calculated results are below the conventional $\alpha = 0.05$ threshold and therefore incompatible with the null hypothesis of equal error probabilities.

14. Discussion

This monograph has investigated various deep learning models' ability to detect uncommon retinal diseases, mainly concentrating on various convolutional neural networks and vision transformers. These latest neural models are driven by the need to precisely search for small pathological features of retinal diseases: retinitis pigmentosa, acquired vitelliform lesions, as well as, drusen. The chapter on proposed convolutional classifiers displays models that fused CNNs with gated recurrent units, enhancing the model's capability to capture spatial as well as layers dimensions in OCT images.

Deep CNN-GRU and Convolutional Gated Recurrent Units U-Net show significant improvements in classification accuracy when benchmarked against traditional CNN models. Another major advancement comes from Residual Attention Networks which in turn offers highlighting of important image regions by the networks and ignoring irrelevant features. Attention mechanisms within residual blocks support these models in their pursuit of increased depth along with expressiveness without the usual degradation added to performance when going deeper into a network.

Furthermore, Vision Transformer architectures, by design, assign attention weights to different image regions through the self-attention mechanism. It allows the model to capture long-range dependencies between spatially distant areas of the image.

Another major contribution of this study is the detailed comparison of these state-of-the-art models. While traditional CNN-based approaches achieved high accuracy in local feature extraction, Residual Attention Vision Transformers significantly outperformed them in the identification and comprehension of complicated global patterns, direct determinants of infrequent disease recognition.

The observed incremental attention mechanism implementations, in both CNNs and ViTs diagnostic accuracies considered, imply that attention-based approaches are central to further development of ophthalmic automatic diagnostic tools.

In Chapter 6 a comprehensive analysis of selected deep learning models used in medical image classification is performed. Particular emphasis is placed on assessing their effectiveness in diagnosing retinal diseases. The conducted studies compare architectures based on residual networks, densely connected networks, as well as, Inception-type networks. A detailed analysis of the results is performed, taking into account metrics such as accuracy, precision, recall, and specificity.

Based on the results, it can be seen that ResNet-type models show high effectiveness in analysing medical data, especially in the context of subtle pathological changes. The accuracy value of these networks ranges from 88.45% to 92.78%, which indicates their stability in recognizing normal and pathological structures in OCT images. The precision for these models ranged from 87.12% to 91.23%, indicating that the models rarely generated false positive diagnoses, which is of significant clinical importance. In turn, the recall (sensitivity) for residual networks reached values from 86.47% to 90.54%, suggesting that these models effectively identify most pathological cases. The specificity for ResNets is exceptionally high, ranging from 89.65% to 93.47%, which means a low level of misclassification of healthy cases as pathological.

In the case of DenseNet-type models, connections between layers allow for the efficient transfer of information between different layers of the network, which favour better use of image features. The classification accuracy of these networks ranges from 89.78% to 93.12%, slightly exceeding these results obtained by residual models. The precision of the DenseNet121 network reaches a level ranging 88.50% to 92.90%, which indicates its high potential in minimizing false positive errors. It is worth noting that the sensitivity of these models is also very high, from 87.23% to 91.45%, confirming their ability to detect subtle pathological changes. The specificity, similarly to the residual models, obtains a high value between 90.12% and 94.23%.

Due to their unique structure based on modules processing data in parallel, the Inception-type models achieve very good results in the context of analysing multi-scale image information. The accuracy of these networks ranges from 90.15% to 93.67%, indicating their superiority in classifying images containing diverse pathological changes. In the case of precision, the InceptionV3 models reach values from 89.75% to 93.33%, emphasizing their efficiency in avoiding false positives. Sensitivity, on the other hand, ranges between 88.67% and 92.25%, indicating that these models are particularly effective in detecting positive cases. Specificity of Inception models varies from 90.45% to 94.65%, confirming their ability to correctly classify healthy images.

Detailed analysis shows that confusion matrices also play an important role, providing additional information about the classification errors of individual models. Analysis of the matrices show that most errors resulted from confusing subtle pathologies with healthy images, which is a typical challenge in the diagnosis of rare retinal diseases. ResNet models show a lower number of false positive errors compared to DenseNet and Inception, while Inception models are more effective in minimizing the number of false negative errors.

Other contributions in the field of computer vision in the case of rare eye diseases detection are presented in the chapter "Proposed Convolutional Classifiers". It presents advanced neural architectures designed to improve the detection efficiency of OCT medical images. In this context, diagnostic accuracy and reliability in identifying subtle pathological changes that can be easily missed in traditional diagnostic approaches are crucial. The proposed architectures include three main approaches: the Deep CNN-GRU model integrating convolutional networks with GRU-type units, the Bidirectional Gated Recurrent Units U-Net model using U-Net-type structures enriched with bidirectional GRU modules and the specially designed Residual Attention Network combining residual blocks with dedicated attention modules. All models are subjected to thorough analysis, taking into account both the results of detailed classification quality metrics and additional aspects such as confusion matrix analysis and the dynamics of the learning process presented by learning curves.

Analysis of classification quality metrics shows clear differences between the individual architectures. The Deep CNN-GRU model achieves accuracy ranging from 87.45% to 92.56%, indicating a solid ability to identify pathological images. The model's precision values, which range from 86.50% to 91.40%, suggest a limited number of false positives, which is crucial in clinical practice. Sensitivity rates ranging from 85.67% to 90.75% demonstrate the model's high

efficiency in detecting real pathological changes, while specificity ranging from 88.92% to 93.68% confirms the model's good ability to correctly classify healthy images.

Even more promising results are achieved by the Bidirectional GRU U-Net architecture, which achieves accuracy from 89.70% to 93.85%. The high precision of this method is particularly noteworthy, oscillating from 88.90% to 93.20%, which indicates a very good level of reliability in diagnosis. It is worth noting that the recall of this approach reaches an equally high level (87.85–92.45%), due to which GRU U-Net effectively minimizes the risk of overlooking even subtle pathological symptoms. The specificity of the model ranges from 90.75% to 94.60%, which indicates its excellent ability to avoid misclassification of healthy cases as pathological.

However, the highest results are achieved by the model based on the Residual Attention Network, which, due to the integration of dedicated attention mechanisms with residual layers, is able to effectively focus on the most important features of OCT images. The accuracy of this architecture ranges from 91.25% to 94.32%, which is the best result among all the convolutional models tested. Similarly high is precision (90.85–93.75%) and sensitivity (89.90–93.15%), which proves a very high diagnostic reliability and effectiveness in detecting pathologies. Specificity reaches a high level of from 91.20% to 95.25%, thus confirming the exceptional effectiveness of the Residual Attention Network model in reducing false positive errors.

The confusion matrix analysis indicates some subtle but significant differences between the discussed architectures. The Residual Attention Network turns out to be the most effective model in minimizing both false positive and false negative errors. In turn, the CNN-GRU model shows a higher number of false negative errors, which suggests its lower effectiveness in detecting more subtle pathological changes. GRU U-Net, on the other hand, balances both types of errors well, presenting itself as the most stable model and well-adapted to diverse diagnostic requirements.

Learning curves provide important information about the dynamics and efficiency of the training process of individual models. In the case of the CNN-GRU and GRU U-Net models, the stabilization of results and achievement of optimal accuracy occur relatively quickly, which suggests their high computational efficiency and the possibility of effective use with a limited number of epochs. On the other hand, the Residual Attention Network model, although ultimately achieving the best results, requires a longer training

period, thus indicating greater computational complexity and potentially higher hardware requirements for implementing this solution.

The Chapter 8 entitled "Kronecker Convolution" presents an advanced analysis of an alternative method of implementing convolutional operations, in which case an innovative Kronecker convolution is used. This solution aims both to improve the classification efficiency of medical OCT images and to optimize the computational costs resulting from traditional convolutional approaches. The experimental study includes both adaptations of commonly used CNN architectures and the evaluation of custom architectures proposed in this chapter, all incorporating the Kronecker convolution mechanism.

A number of observations should be emphasized. First, the results obtained by adapting standard CNN architectures, such as ResNet-18, InceptionV3, EfficientNet and DenseNet121, are compared, replacing their standard convolutional layers with Kronecker convolution. A detailed analysis of the tables containing quality metrics (accuracy, precision, recall, specificity) reveals that the use of Kronecker convolution significantly increases the classification quality. For the ResNet model, the classification accuracy ranges from 89.20% to 91.15%, while the use of Kronecker convolution increases this result to the range of 91.50% to 93.20%. Similar trends are observed for the DenseNet models, where the use of the new mechanism results in an increase in accuracy from 88.75–90.35% to 91.45–93.80%. The results for subsequent models indicate a significant improvement in classification efficiency, especially in the case of subtle retinal pathologies.

Second, the precision results also confirm the superiority of the Kronecker convolution models. The ResNet models initially achieve precision in the range of 88.10–90.05%, while the introduction of Kronecker convolution allows achieving results from 90.35% to 92.55%. Similarly, the DenseNet models show an increase in precision from 87.50–89.65% to 90.75–93.25%. Such a significant improvement in precision in all tested models indicates that the Kronecker convolution models are better at minimizing false positive errors.

Third, the recall rate analysis also shows that the models using Kronecker convolution better identify real-world disease cases. ResNet models with traditional convolution have recall values ranging from 86.90% to 89.40%, while the implementation of Kronecker convolution increases recall to the range of 89.20–91.75%. For DenseNet121 and InceptionV3 models, similar improvements in recall are noticeable. These results indicate that Kronecker Convolution enables better detection of real-world pathologies, reducing the number of cases in which the disease can be missed.

Fourth, the specificity of the models is also an important issue in the diagnostic context. The beneficial effect of the Kronecker convolution is visible. The specificity for the classic ResNet and DenseNet architectures is 89.50–91.40% and 89.25–91.15%, respectively. After implementing the Kronecker convolution, these values increase to the range of 91.70–93.80% and 91.45–94.10%, respectively, indicating a significant reduction in the misclassification of healthy images as pathological. Similar trends are observed for the other proposed classic models.

The conducted analysis of the confusion matrix shows clear differences between the classic models and those based on Kronecker convolution. The standard models show a higher number of errors, especially false negatives, while the models using Kronecker convolution clearly reduce the number of these errors. This effect is particularly noticeable in the classes of rare pathologies, where the standard models tend to miss subtle changes. The confusion matrices indicate that the models based on Kronecker convolution identify these subtle disease cases much more effectively.

Learning curve analysis will provide additional information on the course and efficiency of the training process of the models under this study. Models using Kronecker convolution achieve stability and satisfactory classification results after a relatively small number of epochs, which indicates their high computational efficiency. In addition, the learning curves show that the learning process in the case of Kronecker convolution models is characterized by a more stable course, with fewer fluctuations than in the case of traditional models, which require a longer time to achieve optimal efficiency.

This monograph also provides a detailed analysis of the effectiveness of using innovative transformer architectures (ViT) and their extension with residual connections (Deep Residual ViT) in the task of OCT image classification. The described studies focus primarily on the assessment of diagnostic effectiveness, measured using standard metrics: accuracy, precision, recall and specificity, as well as a detailed analysis of the error matrix and the course of the learning process.

Analysis of the results in the tables shows that transformer architectures, including the basic ViTs, offer significant progress compared to traditional convolutional models. The accuracy of OCT image classification achieved by the basic ViT models ranges from 90.45% to 93.75%. This confirms their high efficiency in medical applications. The precision values for these models range from 89.55% to 92.85%. This indicates a limited number of false positives. The recall results for the ViT models are equally promising and range from 88.90% to 92.45%. This indicates a high level of efficiency in identifying real

pathologies. However, the specificity, oscillating between 90.95% and 94.20%, confirms the very good ability of the models to distinguish healthy from pathological images.

The use of the more advanced Deep Residual Vision Transformer architecture results in further significant improvement of results. These models are equipped with residual connections, which facilitate gradient flow and prevent the problems of vanishing gradients. The accuracy of this architecture increases to the range from 92.65% to even 95.85%. This means that the residual extension of the transformer model allows for more effective identification of diagnostically difficult pathological changes. The precision rates for Deep Residual ViT are exceptionally high, ranging from 91.75% to 94.65%. This means that these models generate few false alarms. Similarly, recall (from 90.55% to 94.20%) indicates high efficiency in detecting even subtle disease symptoms. The specificity of these models reaches values from 92.35% to 96.20%, indicating very effective minimization of false positive cases.

Detailed analysis of the confusion matrix provides further important conclusions regarding the quality of classification offered by both discussed architectures. Vision transformer, despite very good overall results, shows a tendency to make a slightly higher number of false negative errors in the case of subtle pathologies. This suggests the need for additional optimization of this architecture in the case of, for example, very early stages of the disease. It is worth noting that Deep Residual Vision Transformer significantly reduces the number of these errors, which allows for more effective detection of early stages of pathology. As a result, the number of false negative cases decreases on the average by several percent points compared to the standard ViT model.

The analysis of the learning curves presented for both types of models shows significant differences in the dynamics of network learning. The standard ViT is characterized by a relatively stable learning process. It achieves satisfactory results in a moderate number of epochs, but there are visible periodic fluctuations resulting probably from local optimization problems related to the operation of the attention mechanism. In the case of Deep Residual Vision Transformer, the learning process is much more stable. The model quickly reaches a plateau at a very high level of efficiency. Due to the use of residual connections, the model effectively copes with optimization, reaching the optimal point faster and avoiding long periods of stagnation.

Table 14.1. The comparison with the state-of-the-art. Acc stands for Accuracy, Prec. for Precision.

Model	Classes no.	Dataset size	Dataset	Metrics	Ref.
OCCNet	4	OCT2017	84 484	Acc: 99.69%	[184]
ResNet50				Acc: 99.40%	[131]
VGG16				Acc: 98.60%	[185]
Hybrid (VGG16, VGG19, InceptionV3)				Acc: 98.30%	[186]
CNN				Acc: 96.50%	[187]
DMF-CNN				Acc: 96.03%	[188]
IFCNN				Acc: 85.15%	[66]
MHA-Net				Acc: 96.51%	[189]
RepVGG				Acc: 94.45%	
Res2Net50				Acc: 95.47%	
SENet				Acc: 95.24%	
SKNet				Acc: 95.40%	
Deep CNN-GRU				Acc: 98.90%	[45]
OCTNet				Acc: 98–99%	[190]
ROCT-Net				Acc: 98–99%	[157]
MsTGANet		Own OCT	n/d	Acc: 96–98%	[191]
Dual Guidance Net				Acc: 95–97%	[192]
Label Smoothing GAN				Acc: 96–98%	[193]
Multi-scale Fusion CNN				Acc: 96–98%	[66]
EfficientNet-B7	multi	UWFI	574	Acc: 96.44–99.32%	[194]
	2	Fundus HR	n/d	Acc: 85.45%	[195]
Random Forest	1	RIPS+OCT	n/d	Acc: 97.99–98.45%	[196]
AdaBoost				Acc: 98.99–99.45%	[197]
Convolutional GRU U-Net	3	UWFP+UWFAF	744	Acc: 91.00–97.90%	[30]
ViT	4	OCT2017	84 484	Acc: 93.34–98.10%	[53], [54], [56]
SViT				Acc: 97.50%	[198]
Mobile ViT				Acc: 99.17%	[55]
HCTNet				Acc: 91.56%	[199]
Conv-ViT				Acc: 92.37%	[77]
SwinPolyT				Acc: 99.82%	[59]
MedViT	3	NEH-v2	16 822	Acc: 99.70%	[200]
	4	UCSD	108 312	Acc: 84.10%	[201]
RS-A ViT	3	Own OCT	1 280	Acc: 96.92%	[155]
Multi-scale CNN	4	Own OCT	n/d	Acc: 96–98%	[202]
M3 (multimodal)	2	Own OCT AMD)	n/d	AUC: 0.94–0.98%	[158]
Biomarker detection DL	2	Own OCT	n/d	Acc: 96–97%	[203]
SVM, NB, DT, RF, NN, Adaptive Boostiog	4	Multi medical data sets	80 – 245 057	Acc: 62–99%	[204]
Explainable Hybrid		Own OCT	n/d	Prec: 94.32–99.59%	[206]
Federated AMD DL	2	OCT 2017	84 484	Acc: 90.94–99.45%	[205]

While analysing the literature on automatic classification of retinal diseases in OCT and UFWAW images it can be stated that most studies are conducted on the OCT2017 dataset (Tab. 14.1). Methods operating on this dataset achieve accuracies in a wide range, from 85.15% for IFCNN [66] to 99.69% for OCCNet [184] and 99.40% for ResNet50 [131]. High accuracy values were also achieved for VGG16 [185] architectures and models combining multiple convolutional networks, such as VGG16, VGG19, and InceptionV3 [186]. Variant variants based on RepVGG, Res2Net50, SENet, and SKNet achieve accuracies in the 94–95% range [189]. The Deep CNN-GRU method achieved 98.90% accuracy [45], which places it in the upper range of results reported for the four-class classification of CNV, DME, DRUSEN, and NORMAL. OCTNet and ROCT-Net report similar performance [157], [190].

Models using multi-scale or generatively supported approaches, such as MsTGANet or Label Smoothing GAN, achieve 96–98% [191],[193]. In light of these results, it is important to emphasize that the high accuracy achieved on large datasets is partially a consequence of sample size and class uniqueness. In studies described in [158] the significant impact of dataset size on classification quality was demonstrated. This is particularly observable in medical applications. Therefore, interpreting a high accuracy metric requires considering the number of classes, data balance, and image heterogeneity.

In case of retinitis pigmentosa, often in studies defined as diabetic retinopathy, which is not a rare disease, the literature presents a more complex information. Random Forest and AdaBoost achieve 98.99–99.45% accuracy [196], [197] on the RIPS and OCT datasets, but these are single-disease tasks. Convolutional GRU U-Net achieves 91.00–97.90% accuracy [30] in three-class classification. EfficientNet-B7 reports 85.45% accuracy in the configuration including RP [195], and 96.44–99.32% accuracy [194] in the multi-disease analysis on the UWFI dataset. The ViT model proposed in the monograph for the RP problem achieves an F1 value of 96.00% with a specificity of 98.77%. These values remain competitive with classical methods [196], [197], but include a multi-class task. For the CORD class, recall of 92.00% and an F1-score of 82.14% were achieved.

Transformer architectures dominate in the AMD and vitelliform lesions domains. SViT achieves 99.90% [198], MobileViT 99.17% [55], and SwinPolyT 99.82% [59]. MedViT achieves 99.70% for the NEH-v2 dataset [200], but only 84.10% for the UCSD dataset [201]. HCTNet and Conv-ViT achieve 91.56% and 92.37%, respectively [77], [199]. In AMD biomarker tasks, multi-task and multimodal models, such as M3 [158] and segmentation approaches [203],

achieve high AUC and accuracy values. Federated and few-shot solutions [204], [205] indicate a trend toward greater generalizability. The transformer solutions presented in the monograph, for data from the AVL dataset, achieve competitive and often superior values for the basic metrics compared to the models mentioned above. These results are comparable to the best hybrid and transformer models [200]–[206], while maintaining high selectivity. In summary, the data in Table 14.1 show that the highest accuracy values are most often reported for large four-class datasets.

15. Conclusions and Future Works

Results presented in this monograph demonstrate the potential that deep learning methods can have for tackling important challenges in diagnosing rare retinal diseases. The use of DCGANs, working basically on medical data augmentation, helped improve generalization of the model somewhat by increasing classification accuracy to over 91%, much higher than the 73% (more or less) achieved with standard techniques of augmentation. Notably, the study shows that CNNs enhanced by new attention modules, like residual attention networks and Convolutional Block Attention Modules, significantly surpass standard CNN architectures in accuracies by 5% to 10%. These modules effectively steer the model's vision towards key image parts and therefore greatly enhance the detection of fine pathological changes characteristic of RP, CORD as well as AVL and Drusen.

Furthermore, bidirectional convolutional GRUs and variations of U-Net in CNN architectures also show substantial incremental gains in accuracy around 7–12% through GRUs capturing both spatial and layer dependencies in OCT imaging data.

The novel application of Kronecker convolutions in place of standard convolutions is another promising result, which shows a significant decrease in computational complexity and better efficacy in feature extraction.

These kinds of methodological progress underline the key role that architectural innovation plays in the adaptation of deep learning methodologies

to complex medical imaging tasks where conditions exhibit very subtle and intricate pathological manifestations.

Incorporation of Vision Transformers is another key advancement that this monograph has examined. ViTs, more so when fused with the residual self-attention mechanism, usually achieve good classification performance, reaching accuracy rates of about 98%, highlighting their usefulness in ophthalmic diagnostics. Unlike CNNs, the Vision Transformers models inherently have interpretable attention maps which can bring extra clinical trust due to clearer insights into model decision-making processes. This study also pushes for the fusion of ViT architectures with residual networks, which has impressive diagnostic accuracy and interpretability.

Ensemble learning methods, like stacking, boosting and voting, have been good at achieving better diagnostic accuracy (much over 90%) and strength against overfitting which is the key in situations with small and unbalanced data. Group methods combine well different forecasts using the powers of single models and greatly better diagnostic results.

Lastly, this work emphasizes the importance of XAI methodologies. Techniques like Grad-CAM and SHAP play a significant role in showcasing important areas that impact model predictions. Thus, they help in transparent talk with clinicians. Explainable AI not only enhances understanding of model behaviour but also provides valuable insights for model refinement and performance improvement.

To sum up, the methods made and tested in this book show important steps towards automatic, exact, and understandable diagnostic tools for uncommon eye diseases. Future studies should keep looking into new ways to add data, mixed model designs, and refine AI to achieve even more accurate results. Such ongoing growth is key for moving these advanced models from objectives for study to regular clinical use.

Future directions for studies resulting from these presented analyses and experiments may focus on several important areas that will allow further improvement of the effectiveness, efficiency and clinical utility of the presented models.

First, researchers should undertake studies on the further development of data augmentation methods, especially based on generative models such as GAN and DCGAN. It will be possible to create adaptive techniques for generating synthetic images that will better reflect individual patient characteristics or specific pathological changes. This may include the development of conditional

GAN models that generate OCT images adapted to specific types of retinal pathologies.

Second, an important area for exploration is the further development of Kronecker convolution-based approaches. Future work may focus on the analysis of other types of unusual convolutional operations and their integration with advanced attention modules. Of particular interest would be the application of mechanisms for dynamically selecting convolutions, depending on the analysed image, as well as testing the applicability of Kronecker convolutions in other datasets.

Third, studies related to ViTs architectures and their residual variants should be continued by analysing new tokenization techniques and attention mechanisms. This will allow for the more effective capture of subtle pathological features. Particularly promising solutions seem to be hierarchical Vision Transformers, which allow for analysis of OCT images at multiple levels of resolution.

Fourth, the application of methods for aggregating results also seems to be a promising direction of future study. Aggregations based on adaptive attention models or the Choquet integral seem to be suitable.

In summary, future studies should focus on further improvement of data augmentation techniques, development of innovative neural network architectures, especially transformer ones, optimization of ensemble algorithms and development of advanced methods of interpretability of AI models. At the same time, clinical verification and integration with real medical systems are necessary, which will allow the full realisation of the potential of these presented solutions.

References

[1] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, 'A survey on deep learning in medical image analysis', *Medical Image Analysis*, vol. 42, pp. 60–88, Dec. 2017, doi: 10.1016/j.media.2017.07.005.

[2] A. Esteva et al., 'A guide to deep learning in healthcare', *Nat Med*, vol. 25, no. 1, pp. 24–29, Jan. 2019, doi: 10.1038/s41591-018-0316-z.

[3] A. S. Panayides et al., 'AI in Medical Imaging Informatics: Current Challenges and Future Directions', *IEEE J. Biomed. Health Inform.*, vol. 24, no. 7, pp. 1837–1857, July 2020, doi: 10.1109/JBHI.2020.2991043.

[4] J. De Fauw et al., 'Clinically applicable deep learning for diagnosis and referral in retinal disease', *Nat Med*, vol. 24, no. 9, pp. 1342–1350, Sept. 2018, doi: 10.1038/s41591-018-0107-6.

[5] D. S. W. Ting et al., 'Artificial intelligence and deep learning in ophthalmology', *Br J Ophthalmol*, vol. 103, no. 2, pp. 167–175, Feb. 2019, doi: 10.1136/bjophthalmol-2018-313173.

[6] X. Leng, R. Shi, Y. Wu, S. Zhu, X. Cai, X. Lu, and R. Liu, 'Deep learning for detection of age-related macular degeneration: A systematic review and meta-analysis of diagnostic test accuracy studies', *PLoS ONE*, vol. 18, no. 4, p. e0284060, Apr. 2023, doi: 10.1371/journal.pone.0284060.

[7] P. M. Burlina, N. Joshi, K. D. Pacheco, D. E. Freund, J. Kong, and N. M. Bressler, 'Use of Deep Learning for Detailed Severity Characterization and Estimation of 5-Year Risk Among Patients With Age-Related Macular Degeneration', *JAMA Ophthalmol*, vol. 136, no. 12, p. 1359, Dec. 2018, doi: 10.1001/jamaophthalmol.2018.4118.

[8] K. B. Freund, K. Laud, L. H. Lima, R. F. Spaide, S. Zweifel, and L. A. Yannuzzi, 'Acquired Vitelliform Lasions: Correlation of Clinical Findings and Multiple Imaging Analyses', *Retina*, vol. 31, no. 1, pp. 13–25, Jan. 2011, doi: 10.1097/IAE.0b013e3181ea48ba.

[9] D. T. Hartong, E. L. Berson, and T. P. Dryja, 'Retinitis pigmentosa', *The Lancet*, vol. 368, no. 9549, pp. 1795–1809, Nov. 2006, doi: 10.1016/S0140-6736(06)69740-7.

[10] T. Y. Wong and C. Sabanayagam, 'Strategies to Tackle the Global Burden of Diabetic Retinopathy: From Epidemiology to Artificial Intelligence', *Ophthalmologica*, vol. 243, no. 1, pp. 9–20, 2020, doi: 10.1159/000502387.

[11] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth, 'f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks', *Medical Image Analysis*, vol. 54, pp. 30–44, May 2019, doi: 10.1016/j.media.2019.01.010.

[12] M. S. Meor Yahaya and J. Teo, 'Data augmentation using generative adversarial networks for images and biomarkers in medicine and neuroscience', *Front. Appl. Math. Stat.*, vol. 9, p. 1162760, May 2023, doi: 10.3389/fams.2023.1162760.

[13] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, 'Continual lifelong learning with neural networks: A review', *Neural Networks*, vol. 113, pp. 54–71, May 2019, doi: 10.1016/j.neunet.2019.01.012.

[14] R. Rocha Bastos, C. S. Ferreira, E. Brandão, F. Falcão-Reis, and Â. M. Carneiro, 'Multimodal Image Analysis in Acquired Vitelliform Lesions and Adult-Onset Foveomacular Vitelliform Dystrophy', *Journal of Ophthalmology*, vol. 2016, pp. 1–6, 2016, doi: 10.1155/2016/6037537.

[15] A. Choudhary, S. Ahlawat, S. Urooj, N. Pathak, A. Lay-Ekuakille, and N. Sharma, 'A Deep Learning-Based Framework for Retinal Disease Classification', *Healthcare*, vol. 11, no. 2, p. 212, Jan. 2023, doi: 10.3390/healthcare11020212.

[16] C. S. Lee, D. M. Baughman, and A. Y. Lee, 'Deep Learning Is Effective for Classifying Normal versus Age-Related Macular Degeneration OCT Images', *Ophthalmology Retina*, vol. 1, no. 4, pp. 322–327, July 2017, doi: 10.1016/j.oret.2016.12.009.

[17] A. J. Sreejith Kumar et al., 'Evaluation of Generative Adversarial Networks for High-Resolution Synthetic Image Generation of Circumpapillary Optical Coherence Tomography Images for Glaucoma', *JAMA Ophthalmol*, vol. 140, no. 10, p. 974, Oct. 2022, doi: 10.1001/jamaophthalmol.2022.3375.

[18] C. Zheng et al., 'Assessment of Generative Adversarial Networks Model for Synthetic Optical Coherence Tomography Images of Retinal Disorders', *Trans. Vis. Sci. Tech.*, vol. 9, no. 2, p. 29, May 2020, doi: 10.1167/tvst.9.2.29.

[19] H. Zafar, J. Zafar, and F. Sharif, 'GANs-Based Intracoronary Optical Coherence Tomography Image Augmentation for Improved Plaques Characterization Using Deep Neural Networks', *Optics*, vol. 4, no. 2, pp. 288–299, Mar. 2023, doi: 10.3390/opt4020020.

[20] H. Abdelmotaal, M. Sharaf, W. Soliman, E. Wasfi, and S. M. Kedwany, 'Bridging the resources gap: deep learning for fluorescein angiography and optical coherence

tomography macular thickness map image translation', *BMC Ophthalmol*, vol. 22, no. 1, p. 355, Sept. 2022, doi: 10.1186/s12886-022-02577-7.

[21] C. Zheng et al., 'Assessment of Generative Adversarial Networks for Synthetic Anterior Segment Optical Coherence Tomography Images in Closed-Angle Detection', *Trans. Vis. Sci. Tech.*, vol. 10, no. 4, p. 34, Apr. 2021, doi: 10.1167/tvst.10.4.34.

[22] P. Powroznik, M. Skublewska-Paszkowska, K. Nowomiejska, A. Aristidou, A. Panayides, and R. Rejdak, 'Deep convolutional generative adversarial networks in retinitis pigmentosa disease images augmentation and detection', *Adv. Sci. Technol. Res. J.*, vol. 19, no. 2, pp. 321–340, Jan. 2025, doi: 10.12913/22998624/196179.

[23] R. Remtulla, A. Samet, M. Kulbay, A. Akdag, A. Hocini, A. Volniansky, S. Kahn Ali, and C. X. Qian, 'A Future Picture: A Review of Current Generative Adversarial Neural Networks in Vitreoretinal Pathologies and Their Future Potentials', *Biomedicines*, vol. 13, no. 2, p. 284, Jan. 2025, doi: 10.3390/biomedicines13020284.

[24] H. Danesh, D. H. Steel, J. Hogg, F. Ashtari, W. Innes, J. Bacardit, A. Hurlbert, J. C. A. Read, and R. Kafieh, 'Synthetic OCT Data Generation to Enhance the Performance of Diagnostic Models for Neurodegenerative Diseases', *Trans. Vis. Sci. Tech.*, vol. 11, no. 10, p. 10, Oct. 2022, doi: 10.1167/tvst.11.10.10.

[25] H. Danesh, K. Maghooli, R. Kafieh, and A. Dehghani, 'Automatic Production of Synthetic Labeled OCT Images Using Active Shape Model', *IET Image Processing,* vol.14 no.15, doi: 10.1049/iet-ipr.2020.0075

[26] K. Han, Y. Yu, and T. Lu, 'Transfer Learning and Interpretable Analysis-Based Quality Assessment of Synthetic Optical Coherence Tomography Images by CGAN Model for Retinal Diseases', *Processes*, vol. 12, no. 1, p. 182, Jan. 2024, doi: 10.3390/pr12010182.

[27] S. Moon, Y. Lee, J. Hwang, C. G. Kim, J. W. Kim, W. T. Yoon, and J. H. Kim, 'Prediction of anti-vascular endothelial growth factor agent-specific treatment outcomes in neovascular age-related macular degeneration using a generative adversarial network', *Sci Rep*, vol. 13, no. 1, p. 5639, Apr. 2023, doi: 10.1038/s41598-023-32398-7.

[28] D. Romo-Bucheli, P. Seeböck, J. I. Orlando, B. S. Gerendas, S. M. Waldstein, U. Schmidt-Erfurth, and H. Bogunović, 'Reducing image variability across OCT devices with unsupervised unpaired learning for improved segmentation of retina', *Biomed. Opt. Express*, vol. 11, no. 1, p. 346, Jan. 2020, doi: 10.1364/BOE.379978.

[29] A. Lang, A. Carass, B. M. Jedynak, S. D. Solomon, P. A. Calabresi, and J. L. Prince, 'Intensity inhomogeneity correction of SD-OCT data using macular flatspace', *Medical Image Analysis*, vol. 43, pp. 85–97, Jan. 2018, doi: 10.1016/j.media.2017.09.008.

[30] M. Skublewska-Paszkowska, P. Powroznik, R. Rejdak, and K. Nowomiejska, 'Application of Convolutional Gated Recurrent Units U-Net for Distinguishing between Retinitis Pigmentosa and Cone–Rod Dystrophy', *Acta Mechanica et Automatica*, vol. 18, no. 3, pp. 505–513, Sept. 2024, doi: 10.2478/ama-2024-0054.

[31] A. Plaksyvyi, M. Skublewska-Paszkowska, and P. Powroznik, 'A Comparative Analysis of Image Segmentation Using Classical and Deep Learning Approach', *Adv. Sci. Technol. Res. J.*, vol. 17, no. 6, pp. 127–139, Dec. 2023, doi: 10.12913/22998624/172771.

[32] K. Liang, X. Liu, S. Chen, J. Xie, W. Qing Lee, L. Liu, and H. Kuan Lee, 'Resolution enhancement and realistic speckle recovery with generative adversarial modeling of micro-optical coherence tomography', *Biomed. Opt. Express*, vol. 11, no. 12, p. 7236, Dec. 2020, doi: 10.1364/BOE.402847.

[33] T. D. Le, Y.-J. Lee, E. Park, M.-S. Kim, T. J. Eom, and C. Lee, 'Synthetic polarization-sensitive optical coherence tomography using contrastive unpaired translation', *Sci Rep*, vol. 14, no. 1, p. 31366, Dec. 2024, doi: 10.1038/s41598-024-82839-0.

[34] T. K. Yoo, J. Y. Choi, and H. K. Kim, 'Feasibility study to improve deep learning in OCT diagnosis of rare retinal diseases with few-shot classification', *Med Biol Eng Comput*, vol. 59, no. 2, pp. 401–415, Feb. 2021, doi: 10.1007/s11517-021-02321-1.

[35] G. S. Liu, M. H. Zhu, J. Kim, P. Raphael, B. E. Applegate, and J. S. Oghalai, 'ELHnet: a convolutional neural network for classifying cochlear endolymphatic hydrops imaged with optical coherence tomography', *Biomed. Opt. Express*, vol. 8, no. 10, p. 4579, Oct. 2017, doi: 10.1364/BOE.8.004579.

[36] J. Wang, Z. Wang, F. Li, G. Qu, Y. Qiao, H. Lv, and X. Zhang, 'Joint retina segmentation and classification for early glaucoma diagnosis', *Biomed. Opt. Express*, vol. 10, no. 5, p. 2639, May 2019, doi: 10.1364/BOE.10.002639.

[37] N. Motozawa et al., 'Optical Coherence Tomography-Based Deep-Learning Models for Classifying Normal and Age-Related Macular Degeneration and Exudative and Non-Exudative Age-Related Macular Degeneration Changes', *Ophthalmol Ther*, vol. 8, no. 4, pp. 527–539, Dec. 2019, doi: 10.1007/s40123-019-00207-y.

[38] Ramya Mohan, Kirupa Ganapathy, and Rama Arunmozhi, 'Comparison of the proposed DCNN model with standard CNN architectures for retinal diseases classification', *JPTCP*, vol. 29, no. 3, Jan. 2022, doi: 10.47750/jptcp.2022.945.

[39] S. S. Mishra, B. Mandal, and N. B. Puhan, 'Multi-Level Dual-Attention Based CNN for Macular Optical Coherence Tomography Classification', *IEEE Signal Process. Lett.*, vol. 26, no. 12, pp. 1793–1797, Dec. 2019, doi: 10.1109/LSP.2019.2949388.

[40] J. Yoon, J. Han, J. I. Park, J. S. Hwang, J. M. Han, J. Sohn, K. H. Park, and D. D.-J. Hwang, 'Optical coherence tomography-based deep-learning model for detecting central serous chorioretinopathy', *Sci Rep*, vol. 10, no. 1, p. 18852, Nov. 2020, doi: 10.1038/s41598-020-75816-w.

[41] D. Jee, J. H. Yoon, H. Ra, J. Kwon, and J. Baek, 'Predicting persistent central serous chorioretinopathy using multiple optical coherence tomographic images by deep learning', *Sci Rep*, vol. 12, no. 1, p. 9335, June 2022, doi: 10.1038/s41598-022-13473-x.

[42] N. Y. Kang, H. Ra, K. Lee, J. H. Lee, W. K. Lee, and J. Baek, 'Classification of pachychoroid on optical coherence tomography using deep learning', *Graefes Arch Clin Exp Ophthalmol*, vol. 259, no. 7, pp. 1803–1809, July 2021, doi: 10.1007/s00417-021-05104-4.

[43] T. Tsuji et al., 'Classification of optical coherence tomography images using a capsule network', *BMC Ophthalmol*, vol. 20, no. 1, p. 114, Dec. 2020, doi: 10.1186/s12886-020-01382-4.

[44] G. Barbosa, E. Carvalho, A. Guerra, S. Torres-Costa, N. Ramião, M. L. P. Parente, and M. Falcão, 'Deep Learning to Distinguish Edema Secondary to Retinal Vein Occlusion and Diabetic Macular Edema: A Multimodal Approach Using OCT and Infrared Imaging', *JCM*, vol. 14, no. 3, p. 1008, Feb. 2025, doi: 10.3390/jcm14031008.

[45] P. Powroznik, M. Skublewska-Paszkowska, R. Rejdak, and K. Nowomiejska, 'Automatic Method of Macular Diseases Detection Using Deep CNN-GRU Network in OCT Images', *Acta Mechanica et Automatica*, vol. 18, no. 4, pp. 697–706, Dec. 2024, doi: 10.2478/ama-2024-0074.

[46] Z. Xiao, X. Zhang, R. Higashita, and J. Liu, 'MM-UNet: A Mixed MLP Architecture for Improved Ophthalmic Image Segmentation', in *Ophthalmic Medical Image Analysis*, vol. 15188, A. Bhavna, H. Chen, H. Fang, H. Fu, and C. S. Lee, Eds., in Lecture Notes in Computer Science, vol. 15188. Cham: Springer Nature Switzerland, 2025, pp. 63–72. doi: 10.1007/978-3-031-73119-8_7.

[47] W. J. Jaimes, W. J. Arenas, H. J. Navarro, and M. Altuve, 'Detection of retinal diseases from OCT images using a VGG16 and transfer learning', *Discov Appl Sci*, vol. 7, no. 3, p. 160, Feb. 2025, doi: 10.1007/s42452-025-06565-6.

[48] P. Shourie, V. Anand, D. Upadhyay, S. Devliyal, and S. Gupta, 'OCTNet: Leveraging VGG16 for Automated Classification of Retina Health from OCT Scans', in *2024 First International Conference on Pioneering Developments in Computer Science & Digital Technologies (IC2SDT)*, Delhi, India: IEEE, Aug. 2024, pp. 338–343. doi: 10.1109/IC2SDT62152.2024.10696685.

[49] M. Kirankumar, 'Automated Classification of Coloboma Subtypes Using InceptionV3 Algorithm on Optical Coherence Tomography Images', *jisem*, vol. 10, no. 9s, pp. 128–134, Feb. 2025, doi: 10.52783/jisem.v10i9s.1173.

[50] E. Yusufoğlu, H. Fırat, H. Üzen, S. T. A. Özçelik, İ. B. Çiçek, A. Şengür, O. Atila, and N. H. Guldemir, 'A Comprehensive CNN Model for Age-Related Macular Degeneration Classification Using OCT: Integrating Inception Modules, SE Blocks, and ConvMixer', *Diagnostics*, vol. 14, no. 24, p. 2836, Dec. 2024, doi: 10.3390/diagnostics14242836.

[51] T. Bahr, T. A. Vu, J. J. Tuttle, and R. Iezzi, 'Deep Learning and Machine Learning Algorithms for Retinal Image Analysis in Neurodegenerative Disease: Systematic Review of Datasets and Models', *Trans. Vis. Sci. Tech.*, vol. 13, no. 2, p. 16, Feb. 2024, doi: 10.1167/tvst.13.2.16.

[52] L. Hamid, A. Elnokrashy, E. H. Abdelhay, and M. M. Abdelsalam, 'A deep learning LSTM-based approach for AMD classification using OCT images', *Neural Comput & Applic*, vol. 36, no. 31, pp. 19531–19547, Nov. 2024, doi: 10.1007/s00521-024-10149-7.

[53] L. Cai, C. Wen, J. Jiang, C. Liang, H. Zheng, Y. Su, and C. Chen, 'Classification of diabetic maculopathy based on optical coherence tomography images using a Vision Transformer model', *BMJ Open Ophth*, vol. 8, no. 1, p. e001423, Dec. 2023, doi: 10.1136/bmjophth-2023-001423.

[54] J. Wu, R. Hu, Z. Xiao, J. Chen, and J. Liu, 'Vision Transformer-based recognition of diabetic retinopathy grade', *Medical Physics*, vol. 48, no. 12, pp. 7850–7863, Dec. 2021, doi: 10.1002/mp.15312.

[55] S. Akça, Z. Garip, E. Ekinci, and F. Atban, 'Automated classification of choroidal neovascularization, diabetic macular edema, and drusen from retinal OCT images using vision transformers: a comparative study', *Lasers Med Sci*, vol. 39, no. 1, p. 140, May 2024, doi: 10.1007/s10103-024-04089-w.

[56] Z. Jiang, L. Wang, Q. Wu, Y. Shao, M. Shen, W. Jiang, and C. Dai, 'Computer-aided diagnosis of retinopathy based on vision transformer', *J. Innov. Opt. Health Sci.*, vol. 15, no. 02, p. 2250009, Mar. 2022, doi: 10.1142/S1793545822500092.

[57] L. Cai, C. Wen, J. Jiang, H. Zheng, Y. Su, and C. Chen, 'Automated classification of a new grading system for diabetic maculopathy based on optical coherence tomography by deep learninng', In Review, June 2023, doi: 10.21203/rs.3.rs-3012804/v1.

[58] W. Huang, C. Liu, R. Li, and P. Chen, 'DAT-SegNet: a medical multi-organ segmentation network based on deformation attention and transformer', in *Third International Conference on Computer Vision and Pattern Analysis (ICCPA 2023)*, L. Shen and G. Zhong, Eds., Hangzhou, China: SPIE, 2023, p. 81. doi: 10.1117/12.2684274.

[59] Z. Ma, Q. Xie, P. Xie, F. Fan, X. Gao, and J. Zhu, 'HCTNet: A Hybrid ConvNet-Transformer Network for Retinal Optical Coherence Tomography Image Classification', *Biosensors*, vol. 12, no. 7, p. 542, July 2022, doi: 10.3390/bios12070542.

[60] Z. Zhou, C. Niu, H. Yu, J. Zhao, Y. Wang, and C. Dai, 'Diagnosis of retinal diseases using the vision transformer model based on optical coherence tomography images', in *SPIE-CLP Conference on Advanced Photonics 2022*, X. Liu, X. Yuan, and A. Zayats, Eds., China: SPIE, 2023, p. 5, doi: 10.1117/12.2665918.

[61] J. Zhao, J. Zhu, J. He, G. Cao, and C. Dai, 'Multi-label classification of retinal diseases based on fundus images using Resnet and Transformer', *Med Biol Eng Comput*, vol. 62, no. 11, pp. 3459–3469, Nov. 2024, doi: 10.1007/s11517-024-03144-6.

[62] N. Sengar, R. C. Joshi, M. K. Dutta, and R. Burget, 'EyeDeep-Net: a multi-class diagnosis of retinal diseases using deep neural network', *Neural Comput & Applic*, vol. 35, no. 14, pp. 10551–10571, May 2023, doi: 10.1007/s00521-023-08249-x.

[63] A. Rajpoot and K. R. Seeja, 'Enhancing rare retinal disease classification: a few-shot meta-learning framework utilizing fundus images', *Multimed Tools Appl*, vol. 83, no. 18, pp. 55731–55749, Nov. 2023, doi: 10.1007/s11042-023-17691-x.

[64] S. A. Hussein, A. A. Farouk, and M. M. Saeid, 'Intelligent retinal disease detection using deep learning', *Sci Rep*, vol. 15, no. 1, p. 43282, Dec. 2025, doi: 10.1038/s41598-025-28376-w.

[65] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, 'Gradient-based learning applied to document recognition', *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.

[66] L. Fang, Y. Jin, L. Huang, S. Guo, G. Zhao, and X. Chen, 'Iterative fusion convolutional neural networks for classification of optical coherence tomography images', *Journal of Visual Communication and Image Representation*, vol. 59, pp. 327–333, Feb. 2019, doi: 10.1016/j.jvcir.2019.01.022.

[67] H. Masumoto, H. Tabuchi, S. Nakakura, H. Ohsugi, H. Enno, N. Ishitobi, E. Ohsugi, and Y. Mitamura, 'Accuracy of a deep convolutional neural network in detection of retinitis pigmentosa on ultrawide-field images', *PeerJ*, vol. 7, p. e6900, May 2019, doi: 10.7717/peerj.6900.

[68] A. Oishi, M. Miyata, S. Numa, Y. Otsuka, M. Oishi, and A. Tsujikawa, 'Wide-field fundus autofluorescence imaging in patients with hereditary retinal degeneration: a literature review', *Int J Retin Vitr*, vol. 5, no. S1, p. 23, Dec. 2019, doi: 10.1186/s40942-019-0173-z.

[69] A. G. Robson, C. A. Egan, V. A. Luong, A. C. Bird, G. E. Holder, and F. W. Fitzke, 'Comparison of Fundus Autofluorescence with Photopic and Scotopic Fine-Matrix Mapping in Patients with Retinitis Pigmentosa and Normal Visual Acuity', *Invest. Ophthalmol. Vis. Sci.*, vol. 45, no. 11, p. 4119, Nov. 2004, doi: 10.1167/iovs.04-0211.

[70] S. Schmitz-Valckenberg, F. G. Holz, A. C. Bird, and R. F. Spaide, 'Fundus Autofluorescence Imaging: Review and Perspectives', *Retina*, vol. 28, no. 3, pp. 385–409, Mar. 2008, doi: 10.1097/IAE.0b013e318164a907.

[71] G. Querques, R. Forte, L. Querques, N. Massamba, and E. H. Souied, 'Natural Course of Adult-Onset Foveomacular Vitelliform Dystrophy: A Spectral-Domain Optical Coherence Tomography Analysis', *American Journal of Ophthalmology*, vol. 152, no. 2, pp. 304–313, Aug. 2011, doi: 10.1016/j.ajo.2011.01.047.

[72] C. Balaratnasingam, Q. V. Hoang, M. Inoue, C. A. Curcio, R. Dolz-Marco, N. A. Yannuzzi, E. Dhrami-Gavazi, L. A. Yannuzzi, and K. B. Freund, 'Clinical Characteristics, Choroidal Neovascularization, and Predictors of Visual Outcomes in Acquired Vitelliform Lesions', *American Journal of Ophthalmology*, vol. 172, pp. 28–38, Dec. 2016, doi: 10.1016/j.ajo.2016.09.008.

[73] L. H. Lima, K. Laud, K. B. Freund, L. A. Yannuzzi, and R. F. Spaide, 'Acquired Vitelliform Lesion Associated with Large Drusen', *Retina*, vol. 32, no. 4, pp. 647–651, Apr. 2012, doi: 10.1097/IAE.0b013e31823fb847.

[74] K. M. Gehrs, D. H. Anderson, L. V. Johnson, and G. S. Hageman, 'Age-related macular degeneration—emerging pathogenetic and therapeutic concepts', *Annals of Medicine*, vol. 38, no. 7, pp. 450–471, Jan. 2006, doi: 10.1080/07853890600946724.

[75] P. Powroznik, M. Skublewska-Paszkowska, K. Nowomiejska, B. Gajda-Deryło, M. Brinkmann, M. Concilio, M. D. Toro, and R. Rejdak, 'Residual self-attention vision transformer for detecting acquired vitelliform lesions and age-related macular drusen', *Sci Rep*, vol. 15, no. 1, p. 17107, May 2025, doi: 10.1038/s41598-025-02299-y.

[76] M. Kameoka, 'masumoto RP data RP optos'. figshare, p. 5391652918 Bytes, 2018. doi: 10.6084/M9.FIGSHARE.7403831.V1.

[77] D. S. Kermany et al., 'Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning', *Cell*, vol. 172, no. 5, pp. 1122-1131.e9, Feb. 2018, doi: 10.1016/j.cell.2018.02.010.

[78] W. Liang, Y. Liang, and J. Jia, 'MiAMix: Enhancing Image Classification through a Multi-stage Augmented Mixed Sample Data Augmentation Method', Process, vol. 11, no. 12, p. 3284, Nov. 2023, *arXiv*. doi: 10.48550/ARXIV.2308.02804.

[79] P. Chlap, H. Min, N. Vandenberg, J. Dowling, L. Holloway, and A. Haworth, 'A review of medical image data augmentation techniques for deep learning applications', *J Med Imag Rad Onc*, vol. 65, no. 5, pp. 545–563, Aug. 2021, doi: 10.1111/1754-9485.13261.

[80] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, 'Generative adversarial networks', *Commun. ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020, doi: 10.1145/3422622.

[81] A. Mao, M. Mohri, and Y. Zhong, 'Cross-Entropy Loss Functions: Theoretical Analysis and Applications', 2023, *arXiv*. doi: 10.48550/ARXIV.2304.07288.

[82] Z. Zhao, J. C. Ye, and Y. Bresler, 'Generative Models for Inverse Imaging Problems: From mathematical foundations to physics-driven applications', *IEEE Signal Process. Mag.*, vol. 40, no. 1, pp. 148–163, Jan. 2023, doi: 10.1109/MSP.2022.3215282.

[83] Y. Wang, 'A Mathematical Introduction to Generative Adversarial Nets (GAN)', 2020, *arXiv*. doi: 10.48550/ARXIV.2009.00169.

[84] A. Radford, L. Metz, and S. Chintala, 'Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks', 2015, *arXiv*. doi: 10.48550/ARXIV.1511.06434.

[85] T. Chen and C. Guestrin, 'XGBoost: A Scalable Tree Boosting System', 2016, doi: 10.48550/ARXIV.1603.02754.

[86] A. H. Basori, S. J. Malebary, and S. Alesawi, 'Hybrid Deep Convolutional Generative Adversarial Network (DCGAN) and Xtreme Gradient Boost for X-ray Image Augmentation and Detection', *Applied Sciences*, vol. 13, no. 23, p. 12725, Nov. 2023, doi: 10.3390/app132312725.

[87] A. Żmudzińska et al., 'An innovative approach to intelligent management of retinal pathology recognition based on deep learning and intuitionistic fuzzy sets', in *Intelligent Management and Artificial Intelligence: Trends, Challenges, and Opportunities, Vol. 2*, K. Beyer, M. Łatuszyńska, K. Nermend and M. Piwowarski, Eds., Szczecin, Wydawnictwo Naukowe Uniwersytetu Szczecińskiego, 2025, pp. 314–327. doi: 10.18276/978-83-8419-053-1-22.

[88] Y. LeCun, Y. Bengio, and G. Hinton, 'Deep learning', *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[89] Z. Li, Y. He, S. Keel, W. Meng, R. T. Chang, and M. He, 'Efficacy of a Deep Learning System for Detecting Glaucomatous Optic Neuropathy Based on Color Fundus Photographs', *Ophthalmology*, vol. 125, no. 8, pp. 1199–1206, Aug. 2018, doi: 10.1016/j.ophtha.2018.01.023.

[90] V. Gulshan et al., 'Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs', *JAMA*, vol. 316, no. 22, p. 2402, Dec. 2016, doi: 10.1001/jama.2016.17216.

[91] K. He, X. Zhang, S. Ren, and J. Sun, 'Deep Residual Learning for Image Recognition', in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, June 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[92] M. Rohm et al., 'Predicting Visual Acuity by Using Machine Learning in Patients Treated for Neovascular Age-Related Macular Degeneration', *Ophthalmology*, vol. 125, no. 7, pp. 1028–1036, July 2018, doi: 10.1016/j.ophtha.2017.12.034.

[93] J. Cheng et al., 'Superpixel Classification Based Optic Disc and Optic Cup Segmentation for Glaucoma Screening', *IEEE Trans. Med. Imaging*, vol. 32, no. 6, pp. 1019–1032, June 2013, doi: 10.1109/TMI.2013.2247770.

[94] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, 'Squeeze-and-Excitation Networks', 2017, *arXiv*. doi: 10.48550/ARXIV.1709.01507.

[95] J. Gu et al., 'Recent advances in convolutional neural networks', *Pattern Recognition*, vol. 77, pp. 354–377, May 2018, doi: 10.1016/j.patcog.2017.10.013.

[96] F. Yu and V. Koltun, 'Multi-Scale Context Aggregation by Dilated Convolutions', 2015, *arXiv*. doi: 10.48550/ARXIV.1511.07122.

[97] J. Long, E. Shelhamer, and T. Darrell, 'Fully Convolutional Networks for Semantic Segmentation', 2014, *arXiv*. doi: 10.48550/ARXIV.1411.4038.

[98] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, 'Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)', 2015, *arXiv*, doi: 10.48550/ARXIV.1511.07289.

[99] A. Kiersztyn, R. Lopucki, K. Kiersztyn, P. Karczmarek, P. Powroznik, D. Czerwinski, and W. Pedrycz, 'A Comprehensive Analysis of the Impact of Selecting the Training Set Elements on the Correctness of Classification for Highly Variable Ecological Data', in *2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Luxembourg, Luxembourg: IEEE, July 2021, pp. 1–6. doi: 10.1109/FUZZ45933.2021.9494399.

[100] P. Powroźnik, 'Polish Emotional Speech Recognition using Artificial Neural Network', *Adv. Sci. Technol. Res. J.*, vol. 8, pp. 24–27, Dec. 2014, doi: 10.12913/22998624/562.

[101] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, 'CBAM: Convolutional Block Attention Module', 2018, *arXiv*. doi: 10.48550/ARXIV.1807.06521.

[102] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, 'Recurrent Models of Visual Attention', 2014, *arXiv*. doi: 10.48550/ARXIV.1406.6247.

[103] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, 'Residual Attention Network for Image Classification', in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA: IEEE, July 2017, pp. 6450–6458. doi: 10.1109/CVPR.2017.683.

[104] J. Schlemper, O. Oktay, M. Schaap, M. Heinrich, B. Kainz, B. Glocker, and D. Rueckert, 'Attention gated networks: Learning to leverage salient regions in medical images', *Medical Image Analysis*, vol. 53, pp. 197–207, Apr. 2019, doi: 10.1016/j.media.2019.01.012.

[105] M.-H. Guo et al., 'Attention mechanisms in computer vision: A survey', *Comp. Visual. Med.*, vol. 8, no. 3, pp. 331–368, Sept. 2022, doi: 10.1007/s41095-022-0271-y.

[106] X. Zhang and T. Chen, 'Attention U-net for Interpretable Classification on Chest X-ray Image', in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Seoul, Korea (South): IEEE, Dec. 2020, pp. 901–908. doi: 10.1109/BIBM49941.2020.9313354.

[107] O. Oktay et al., 'Attention U-Net: Learning Where to Look for the Pancreas', 2018, *arXiv*. doi: 10.48550/ARXIV.1804.03999.

[108] E. Jang, S. Gu, and B. Poole, 'Categorical Reparameterization with Gumbel-Softmax', 2016, *arXiv*. doi: 10.48550/ARXIV.1611.01144.

[109] A. Papadopoulos, P. Korus, and N. Memon, 'Hard-Attention for Scalable Image Classification', 2021, *arXiv*. doi: 10.48550/ARXIV.2102.10212.

[110] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, 'Proximal Policy Optimization Algorithms', 2017, *arXiv*. doi: 10.48550/ARXIV.1707.06347.

[111] G. W. Lindsay, 'Attention in Psychology, Neuroscience, and Machine Learning', *Front. Comput. Neurosci.*, vol. 14, p. 29, Apr. 2020, doi: 10.3389/fncom.2020.00029.

[112] C. J. Maddison, A. Mnih, and Y. W. Teh, 'The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables', 2016, *arXiv*. doi: 10.48550/ARXIV.1611.00712.

[113] T. Shan and J. Yan, 'SCA-Net: A Spatial and Channel Attention Network for Medical Image Segmentation', *IEEE Access*, vol. 9, pp. 160926–160937, Dec. 2021, doi: 10.1109/ACCESS.2021.3132293.

[114] X. Zhang, S. Shang, X. Tang, J. Feng, and L. Jiao, 'Spectral Partitioning Residual Network With Spatial Attention Mechanism for Hyperspectral Image Classification', *IEEE Trans. Geosci. Remote Sensing*, vol. 60, pp. 1–14, 2022, doi: 10.1109/TGRS.2021.3074196.

[115] K. Nowomiejska, P. Powroźnik, M. Skublewska-Paszkowska, K. Adamczyk, M. Concilio, L. Sereikaite, R. Zemaitiene, M. D. Toro, and R. Rejdak, 'Residual Attention Network for distinction between visible optic disc drusen and healthy optic discs', *Optics and Lasers in Engineering*, vol. 176, p. 108056, May 2024, doi: 10.1016/j.optlaseng.2024.108056.

[116] Z. Gao, L. Zhou, W. Ding, and H. Wang, 'A retinal vessel segmentation network approach based on rough sets and attention fusion module', *Information Sciences*, vol. 678, p. 121015, Sept. 2024, doi: 10.1016/j.ins.2024.121015.

[117] Q. H. Vuong, 'Likelihood Ratio Tests for Model Selection and Non-Nested Hypotheses', *Econometrica*, vol. 57, no. 2, p. 307, Mar. 1989, doi: 10.2307/1912557.

[118] C. M. Bishop, *Pattern recognition and machine learning*. in Information science and statistics. New York: Springer, 2006.

[119] R. K. Srivastava, K. Greff, and J. Schmidhuber, 'Highway Networks', 2015, *arXiv*. doi: 10.48550/ARXIV.1505.00387.

[120] K. Zhang, M. Sun, T. X. Han, X. Yuan, L. Guo, and T. Liu, 'Residual Networks of Residual Networks: Multilevel Residual Networks', *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 6, pp. 1303–1314, June 2018, doi: 10.1109/TCSVT.2017.2654543.

[121] S. Ioffe and C. Szegedy, 'Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift', 2015, *arXiv*. doi: 10.48550/ARXIV.1502.03167.

[122] A. Krizhevsky, I. Sutskever, and G. E. Hinton, 'ImageNet classification with deep convolutional neural networks', *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[123] V. Alex, M. Khened, S. Ayyachamy, and G. Krishnamurthi, 'Medical image retrieval using Resnet-18 for clinical diagnosis', in *Medical Imaging 2019: Imaging Informatics for Healthcare, Research, and Applications*, P. R. Bak and P.-H. Chen, Eds., San Diego, United States: SPIE, Mar. 2019, p. 35. doi: 10.1117/12.2515588.

[124] Y. Li, Z. Ding, C. Zhang, Y. Wang, and J. Chen, 'SAR Ship Detection Based on Resnet and Transfer Learning', in *IGARSS 2019 – 2019 IEEE International Geoscience and Remote Sensing Symposium*, Yokohama, Japan: IEEE, July 2019, pp. 1188–1191. doi: 10.1109/IGARSS.2019.8900290.

[125] D. Milea et al., 'Artificial Intelligence to Detect Papilledema from Ocular Fundus Photographs', *N Engl J Med*, vol. 382, no. 18, pp. 1687–1695, Apr. 2020, doi: 10.1056/NEJMoa1917130.

[126] Y. M. Ro, W.-H. Cheng, J. Kim, W.-T. Chu, P. Cui, J.-W. Choi, M.-C. Hu, and W. De Neve, 'Correction to: MultiMedia Modeling', in *MultiMedia Modeling*, vol. 11962, Y. M. Ro, W.-H. Cheng, J. Kim, W.-T. Chu, P. Cui, J.-W. Choi, M.-C. Hu, and W. De Neve, Eds., in Lecture Notes in Computer Science, vol. 11962. , Cham: Springer International Publishing, 2020, pp. C1–C1. doi: 10.1007/978-3-030-37734-2_75.

[127] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, 'Densely Connected Convolutional Networks', 2016, *arXiv*. doi: 10.48550/ARXIV.1608.06993.

[128] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, 'Rethinking the Inception Architecture for Computer Vision', in *2016 IEEE Conference on Computer Vision and*

Pattern Recognition (CVPR), Las Vegas, NV, USA: IEEE, June 2016, pp. 2818–2826. doi: 10.1109/CVPR.2016.308.

[129] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, 'Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning', *AAAI*, vol. 31, no. 1, Feb. 2017, doi: 10.1609/aaai.v31i1.11231.

[130] S. F. Rabbi, Md. A. Mamun, Md. F. Faruk, S. M. Mahedy Hasan, and A. Y. Srizon, 'A Multi-branch and Attention Based CNN Architecture for the Classification of Retinal Diseases from OCT Images', in *2023 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*, Dhaka, Bangladesh: IEEE, Sept. 2023, pp. 36–40. doi: 10.1109/ICICT4SD59951.2023.10303384.

[131] S. Asif, K. Amjad, and Qurrat-ul-Ain, 'Deep Residual Network for Diagnosis of Retinal Diseases Using Optical Coherence Tomography Images', *Interdiscip Sci Comput Life Sci*, vol. 14, no. 4, pp. 906–916, Dec. 2022, doi: 10.1007/s12539-022-00533-z.

[132] Md. Z. Islam, Md. M. Islam, and A. Asraf, 'A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images', *Informatics in Medicine Unlocked*, vol. 20, p. 100412, 2020, doi: 10.1016/j.imu.2020.100412.

[133] O. Ronneberger, P. Fischer, and T. Brox, 'U-Net: Convolutional Networks for Biomedical Image Segmentation', 2015, *arXiv*. doi: 10.48550/ARXIV.1505.04597.

[134] Y. Eroğlu, M. Yildirim, and A. Çinar, 'Convolutional Neural Networks based classification of breast ultrasonography images by hybrid method with respect to benign, malignant, and normal using mRMR', *Computers in Biology and Medicine*, vol. 133, p. 104407, June 2021, doi: 10.1016/j.compbiomed.2021.104407.

[135] R. Azad, M. Asadi-Aghbolaghi, M. Fathy, and S. Escalera, 'Bi-Directional ConvLSTM U-Net with Densley Connected Convolutions', in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, Seoul, Korea (South): IEEE, Oct. 2019, pp. 406–415. doi: 10.1109/ICCVW.2019.00052.

[136] H. Song, W. Wang, S. Zhao, J. Shen, and K.-M. Lam, 'Pyramid Dilated Deeper ConvLSTM for Video Salient Object Detection', in *Computer Vision – ECCV 2018*, vol. 11215, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., in Lecture Notes in Computer Science, vol. 11215. , Cham: Springer International Publishing, 2018, pp. 744–760. doi: 10.1007/978-3-030-01252-6_44.

[137] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. Wong, and W. Woo, 'Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting', 2015, *arXiv*. doi: 10.48550/ARXIV.1506.04214.

[138] Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzel, 'Learning to Diagnose with LSTM Recurrent Neural Networks', 2015, *arXiv*. doi: 10.48550/ARXIV.1511.03677.

[139] Z. Zhang and M. Wang, 'Convolutional Neural Network with Convolutional Block Attention Module for Finger Vein Recognition', 2022, *arXiv*. doi: 10.48550/ARXIV.2202.06673.

[140] J.-H. Kim, S.-W. Lee, D.-H. Kwak, M.-O. Heo, J. Kim, J.-W. Ha, and B.-T. Zhang, 'Multimodal Residual Learning for Visual QA', 2016, *arXiv*. doi: 10.48550/ARXIV.1606.01455.

[141] K. Simonyan and A. Zisserman, 'Very Deep Convolutional Networks for Large-Scale Image Recognition', 2014, *arXiv*. doi: 10.48550/ARXIV.1409.1556.

[142] A. Newell, K. Yang, and J. Deng, 'Stacked Hourglass Networks for Human Pose Estimation', 2016, *arXiv*. doi: 10.48550/ARXIV.1603.06937.

[143] Y. Deng and J. Ma, 'ResMatch: Residual Attention Learning for Local Feature Matching', 2023, *arXiv*. doi: 10.48550/ARXIV.2307.05180.

[144] Y. Liu, Z. Zhang, C. Thompson, R. Leibbrandt, S. Qin, and A. J. Yepes, 'Stacked Attention-based Networks for Accurate and Interpretable Health Risk Prediction', in *2023 International Joint Conference on Neural Networks (IJCNN)*, Gold Coast, Australia: IEEE, June 2023, pp. 1–8. doi: 10.1109/IJCNN54540.2023.10191099.

[145] S. Zhou, J.-N. Wu, Y. Wu, and X. Zhou, 'Exploiting Local Structures with the Kronecker Layer in Convolutional Networks', 2015, *arXiv*. doi: 10.48550/ARXIV.1512.09194.

[146] Z. Tang, F. Jiang, M. Gong, H. Li, Y. Wu, F. Yu, Z. Wang, and M. Wang, 'SKFAC: Training Neural Networks with Faster Kronecker-Factored Approximate Curvature', in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA: IEEE, June 2021, pp. 13474–13482. doi: 10.1109/CVPR46437.2021.01327.

[147] K. K. Patro, J. P. Allam, B. C. Neelapu, R. Tadeusiewicz, U. R. Acharya, M. Hammad, O. Yildirim, and P. Pławiak, 'Application of Kronecker convolutions in deep learning technique for automated detection of kidney stones with coronal CT images', *Information Sciences*, vol. 640, p. 119005, Sept. 2023, doi: 10.1016/j.ins.2023.119005.

[148] M. J. Ali, B. Raza, and A. R. Shahid, 'Multi-level Kronecker Convolutional Neural Network (ML-KCNN) for Glioma Segmentation from Multi-modal MRI Volumetric Data', *J Digit Imaging*, vol. 34, no. 4, pp. 905–921, Aug. 2021, doi: 10.1007/s10278-021-00486-7.

[149] A. Meda, L. Nelson, and M. Jagdish, 'DKCN-Net: Deep kronecker convolutional neural network-based lung disease detection with federated learning', *Computational Biology and Chemistry*, vol. 116, p. 108376, June 2025, doi: 10.1016/j.compbiolchem.2025.108376.

[150] Y. Zhang, Y. Li, S. Wang, and C.-X. Wang, 'Image Restoration Based on the Convolution Kernel Matrix Reconstructed by Kronecker Product', in *2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP)*, Nanjing, China: IEEE, Oct. 2020, pp. 185–189. doi: 10.1109/ICSIP49896.2020.9339404.

[151] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, 'Attention Augmented Convolutional Networks', 2019, *arXiv*. doi: 10.48550/ARXIV.1904.09925.

[152] A. Dosovitskiy et al., 'An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale', 2020, *arXiv*. doi: 10.48550/ARXIV.2010.11929.

[153] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z. Jiang, F. E. Tay, J. Feng, and S. Yan, 'Tokens-to-Token ViT: Training Vision Transformers from Scratch on ImageNet', 2021, *arXiv*. doi: 10.48550/ARXIV.2101.11986.

[154] E. Shabaninia, H. Nezamabadi-pour, and F. Shafizadegan, 'Transformers in Action Recognition: A Review on Temporal Modeling', 2023, *arXiv*. doi: 10.48550/ARXIV.2302.01921.

[155] P. Powroznik, M. Skublewska-Paszkowska, K. Nowomiejska, B. Gajda-Deryło, M. Brinkmann, M. Concilio, M. D. Toro, and R. Rejdak, 'Residual self-attention vision transformer for detecting acquired vitelliform lesions and age-related macular drusen', *Sci Rep*, vol. 15, no. 1, p. 17107, May 2025, doi: 10.1038/s41598-025-02299-y.

[156] P. Karczmarek, M. Dolecki, P. Powroznik, Z. A. Łagodowski, A. Gregosiewicz, Ł. Gałka, W. Pedrycz, D. Czerwinski, and K. Jonak, 'Quadrature-Inspired Generalized Choquet Integral in an Application to Classification Problems', *IEEE Access*, vol. 11, pp. 124676–124689, Nov. `2023, doi: 10.1109/ACCESS.2023.3330245.

[157] A. Wali, Z. Suhail, S. Naz, and I. Younas, 'An ensemble deep learning model for OCT Image Detection and Classification', *In Review*, Sept. 2024, doi: 10.21203/rs.3.rs-4923941/v1.

[158] Q. Chen et al., 'Multimodal, multitask, multiattention (M3) deep learning detection of reticular pseudodrusen: Toward automated and accessible classification of age-related macular degeneration', *Journal of the American Medical Informatics Association*, vol. 28, no. 6, pp. 1135–1148, June 2021, doi: 10.1093/jamia/ocaa302.

[159] M. Arsalan, A. Haider, C. Park, J. S. Hong, and K. R. Park, 'Multiscale triplet spatial information fusion-based deep learning method to detect retinal pigment signs with fundus images', *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108353, July 2024, doi: 10.1016/j.engappai.2024.108353.

[160] J. Błaszczyński and J. Stefanowski, 'Actively Balanced Bagging for Imbalanced Data', in *Foundations of Intelligent Systems*, vol. 10352, M. Kryszkiewicz, A. Appice, D. Ślęzak, H. Rybinski, A. Skowron, and Z. W. Raś, Eds., in Lecture Notes in Computer Science, vol. 10352. , Cham: Springer International Publishing, 2017, pp. 271–281. doi: 10.1007/978-3-319-60438-1_27.

[161] M. Hasyim, D. S. Rahayu, N. E. Muliawati, D. Hayuhantika, R. Puspasari, D. Anggreini, R. C. Hastari, S. Hartanto, and F. H. Utomo, 'Bootstrap Aggregating Multivariate Adaptive Regression Splines (Bagging MARS) to Analyse the Lecturer Research Performance in Private University', *J. Phys.: Conf. Ser.*, vol. 1114, p. 012117, Nov. 2018, doi: 10.1088/1742-6596/1114/1/012117.

[162] A. Biswas and R. Banik, 'CNN Fusion: A Promising Technique for Ophthalmic Disorder Diagnosis', *Procedia Computer Science*, vol. 233, pp. 411–421, 2024, doi: 10.1016/j.procs.2024.03.231.

[163] G. Velarde, M. Weichert, A. Deshmunkh, S. Deshmane, A. Sudhir, K. Sharma, and V. Joshi, 'Tree boosting methods for balanced and imbalanced classification and their robustness over time in risk assessment', *Intelligent Systems with Applications*, vol. 22, p. 200354, June 2024, doi: 10.1016/j.iswa.2024.200354.

[164] M. A. Al-antari, 'Advancements in Artificial Intelligence for Medical Computer-Aided Diagnosis', *Diagnostics*, vol. 14, no. 12, p. 1265, June 2024, doi: 10.3390/diagnostics14121265.

[165] S. Wassan, B. Suhail, R. Mubeen, B. Raj, U. Agarwal, E. Khatri, S. Gopinathan, and G. Dhiman, 'Gradient Boosting for Health IoT Federated Learning', *Sustainability*, vol. 14, no. 24, p. 16842, Dec. 2022, doi: 10.3390/su142416842.

[166] S. K. Satapathy, A. K. Bhoi, D. Loganathan, B. Khandelwal, and P. Barsocchi, 'Machine learning with ensemble stacking model for automated sleep staging using dual-channel EEG signal', *Biomedical Signal Processing and Control*, vol. 69, p. 102898, Aug. 2021, doi: 10.1016/j.bspc.2021.102898.

[167] T. Yoon and D. Kang, 'Multi-Modal Stacking Ensemble for the Diagnosis of Cardiovascular Diseases', *JPM*, vol. 13, no. 2, p. 373, Feb. 2023, doi: 10.3390/jpm13020373.

[168] S. Saleh, B. Cherradi, O. El Gannour, S. Hamida, and O. Bouattane, 'Predicting patients with Parkinson's disease using Machine Learning and ensemble voting technique', *Multimed Tools Appl*, vol. 83, no. 11, pp. 33207–33234, Sept. 2023, doi: 10.1007/s11042-023-16881-x.

[169] V. C. Osamor and A. F. Okezie, 'Enhancing the weighted voting ensemble algorithm for tuberculosis predictive diagnosis', *Sci Rep*, vol. 11, no. 1, p. 14806, July 2021, doi: 10.1038/s41598-021-94347-6.

[170] G. P. Miriyala and A. K. Sinha, 'Voting Ensemble Learning Technique with Improved Accuracy for the CAD Diagnosis', in *ICDSMLA 2021*, vol. 947, A. Kumar, S. Senatore, and V. K. Gunjan, Eds., in Lecture Notes in Electrical Engineering, vol. 947., Singapore: Springer Nature Singapore, 2023, pp. 477–487. doi: 10.1007/978-981-19-5936-3_45.

[171] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, D. Pedreschi, and F. Giannotti, 'A Survey Of Methods For Explaining Black Box Models', 2018, *arXiv*. doi: 10.48550/ARXIV.1802.01933.

[172] E. Tjoa and C. Guan, 'A Survey on Explainable Artificial Intelligence (XAI): Towards Medical XAI', 2019, doi: 10.48550/ARXIV.1907.07374.

[173] W. Samek, T. Wiegand, and K.-R. Müller, 'Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models', 2017, *arXiv*. doi: 10.48550/ARXIV.1708.08296.

[174] M. Sundararajan, A. Taly, and Q. Yan, 'Axiomatic Attribution for Deep Networks', 2017, *arXiv*. doi: 10.48550/ARXIV.1703.01365.

[175] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, 'Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization', in 2017 IEEE International Conference on Computer Vision (ICCV) Venice, Italy IEEE, 2017, pp. 618-626, doi: 10.48550/ARXIV.1610.02391.

[176] S. Lundberg and S.-I. Lee, 'A Unified Approach to Interpreting Model Predictions', 2017, *arXiv*. doi: 10.48550/ARXIV.1705.07874.

[177] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, 'Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization', *Int J Comput Vis*, vol. 128, no. 2, pp. 336–359, Feb. 2020, doi: 10.1007/s11263-019-01228-7.

[178] S. Suara, A. Jha, P. Sinha, and A. A. Sekh, 'Is Grad-CAM Explainable in Medical Images?', in Computer Vision and Image Processing, CVIP 2023, Communications Science vol. 2009, H. Kaur, V. Jakhetiya, P. Goyal, P. Khanna, B. Raman, S. Kumar, Eds., Springer, Cham 2024, doi: 10.48550/ARXIV.2307.10506.

[179] A. Messalas, Y. Kanellopoulos, and C. Makris, 'Model-Agnostic Interpretability with Shapley Values', in *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, PATRAS, Greece: IEEE, July 2019, pp. 1–7. doi: 10.1109/IISA.2019.8900669.

[180] N. E. Hawass, 'Comparing the sensitivities and specificities of two diagnostic procedures performed on the same group of patients.', *The British Journal of Radiology*, vol. 70, no. 832, pp. 360–366, Apr. 1997, doi: 10.1259/bjr.70.832.9166071.

[181] M. W. Fagerland, S. Lydersen, and P. Laake, 'The McNemar test for binary matched-pairs data: mid-p and asymptotic are better than exact conditional', *BMC Med Res Methodol*, vol. 13, no. 1, p. 91, Dec. 2013, doi: 10.1186/1471-2288-13-91.

[182] A. E. Maxwell, 'Comparing the Classification of Subjects by Two Independent Judges', *Br J Psychiatry*, vol. 116, no. 535, pp. 651–655, June 1970, doi: 10.1192/bjp.116.535.651.

[183] J. M. Lachin, 'Biostatistical Methods: The Assessment of Relative Risks. in Wiley series in probability and statistics', v. 509. Hoboken: John Wiley & Sons, Inc., 2009.

[184] A. M. Alqudah, M. AlTantawi, and A. Alqudah, 'Artificial Intelligence Hybrid System for Enhancing Retinal Diseases Classification Using Automated Deep Features Extracted from OCT Images', *IJISAE*, vol. 9, no. 3, pp. 91–100, Sep. 2021, doi: 10.18201/ijisae.2021.236.

[185] F. Li, H. Chen, Z. Liu, X. Zhang, and Z. Wu, 'Fully automated detection of retinal disorders by image-based deep learning', *Graefes Arch Clin Exp Ophthalmol*, vol. 257, no. 3, pp. 495–505, Mar. 2019, doi: 10.1007/s00417-018-04224-8.

[186] J. Kim and L. Tran, 'Retinal Disease Classification from OCT Images Using Deep Learning Algorithms', in 2021 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), Melbourne, Australia: IEEE, Oct. 2021, pp. 1–6. doi: 10.1109/CIBCB49929.2021.9562919.

[187] A. Tayal, J. Gupta, A. Solanki, K. Bisht, A. Nayyar, and M. Masud, 'DL-CNN-based approach with image processing techniques for diagnosis of retinal diseases', Multimedia Systems, vol. 28, no. 4, pp. 1417–1438, Aug. 2022, doi: 10.1007/s00530-021-00769-7.

[188] V. Das, S. Dandapat, and P. K. Bora, 'Automated Classification of Retinal OCT Images Using a Deep Multi-Scale Fusion CNN', IEEE Sensors J., vol. 21, no. 20, pp. 23256–23265, Oct. 2021, doi: 10.1109/JSEN.2021.3108642.

[189] L. Xu, L. Wang, S. Cheng, and Y. Li, 'MHANet: A hybrid attention mechanism for retinal diseases classification', PLoS ONE, vol. 16, no. 12, p. e0261285, Dec. 2021, doi: 10.1371/journal.pone.0261285.

[190] A. P. Sunija, K. Saikat, S. Gayathri, V. P. Gopi, P Palanisamy, 'OctNET: A Lightweight CNN for Retinal Disease Classification from Optical Coherence Tomography Images', *Computer Methods and Images', IEEE Trans. Med. Imaging Programs in Biomedicine*, vol. 200, p. 105877, Mar. 2021, doi: 10.1016/j.cmpb.2020.105877.

[191] M. Wang et al., 'MsTGANet: Automatic Drusen Segmentation From Retinal OCT Images', *IEEE Trans. Med. Imaging*, vol. 41, no. 2, pp. 394–406, Feb. 2022, doi: 10.1109/TMI.2021.3112716.

[192] S. Diao et al., 'Classification and segmentation of OCT images for age-related macular degeneration based on dual guidance networks', Biomedical Signal Processing and Control, vol. 84, p. 104810, Jul. 2023, doi: 10.1016/j.bspc.2023.104810.

[193] X. He, L. Fang, H. Rabbani, X. Chen, and Z. Liu, 'Retinal optical coherence tomography image classification with label smoothing generative adversarial network', *Neurocomputing*, vol. 405, pp. 37–47, Sep. 2020, doi: 10.1016/j.neucom.2020.04.044.

[194] G. Sun et al., 'Deep Learning for the Detection of Multiple Fundus Diseases Using Ultra-widefield Images', Ophthalmol Ther, vol. 12, no. 2, pp. 895–907, Apr. 2023, doi: 10.1007/s40123-022-00627-3.

[195] H. Das, A. Saha, and S. Deb, 'An expert system to distinguish a defective eye from a normal eye', in 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), Ghaziabad, India: IEEE, Feb. 2014, pp. 155–158. doi: 10.1109/ICICICT.2014.6781270.

[196] N. Brancati et al., 'Learning-based approach to segment pigment signs in fundus images for Retinitis Pigmentosa analysis', *Neurocomputing*, vol. 308, pp. 159–171, Sep. 2018, doi: 10.1016/j.neucom.2018.04.065.

[197] N. Brancati, M. Frucci, D. Gragnaniello, D. Riccio, V. Di Iorio, and L. Di Perna, 'Automatic segmentation of pigment deposits in retinal fundus images of Retinitis Pigmentosa', *Computerized Medical Imaging and Graphics*, vol. 66, pp. 73–81, Jun. 2018, doi: 10.1016/j.compmedimag.2018.03.002.

[198] G. R. Hemalakshmi, M. Murugappan, M. Y. Sikkandar, S. S. Begum, and N. B. Prakash, 'Automated retinal disease classification using hybrid transformer model (SViT) using optical coherence tomography images', *Neural Comput & Applic*, vol. 36, no. 16, pp. 9171–9188, Jun. 2024, doi: 10.1007/s00521-024-09564-7.

[199] M. M. Azizi, S. Abhari, and H. Sajedi, 'Stitched vision transformer for age-related macular degeneration detection using retinal optical coherence tomography images', *PLoS ONE*, vol. 19, no. 6, p. e0304943, Jun. 2024, doi: 10.1371/journal.pone.0304943.

[200] P. Dutta, K. A. Sathi, Md. A. Hossain, and M. A. A. Dewan, 'Conv-ViT: A Convolution and Vision Transformer-Based Hybrid Feature Extraction Method for Retinal Disease Detection', *J. Imaging*, vol. 9, no. 7, p. 140, Jul. 2023, doi: 10.3390/jimaging9070140.

[201] J. He, J. Wang, Z. Han, J. Ma, C. Wang, and M. Qi, 'An interpretable transformer network for the retinal disease classification using optical coherence tomography', *Sci Rep*, vol. 13, no. 1, p. 3637, Mar. 2023, doi: 10.1038/s41598-023-30853-z.

[202] S. Sotoudeh-Paima, A. Jodeiri, F. Hajizadeh, and H. Soltanian-Zadeh, 'Multi-scale convolutional neural network for automated AMD classification using retinal OCT images', Computers in Biology and Medicine, vol. 144, p. 105368, May 2022, doi: 10.1016/j.compbiomed.2022.105368.

[203] R. Schwartz et al., 'A Deep Learning Framework for the Detection and Quantification of Reticular Pseudodrusen and Drusen on Optical Coherence Tomography', *Trans. Vis. Sci. Tech.*, vol. 11, no. 12, p. 3, Dec. 2022, doi: 10.1167/tvst.11.12.3.

[204] A. Althnian et al., 'Impact of Dataset Size on Classification Performance: An Empirical Evaluation in the Medical Domain', *Applied Sciences*, vol. 11, no. 2, p. 796, Jan. 2021, doi: 10.3390/app11020796.

[205] S. Gholami, J. I. Lim, T. Leng, S. S. Y. Ong, A. C. Thompson, and M. N. Alam, 'Federated learning for diagnosis of age-related macular degeneration', *Front. Med.*, vol. 10, p. 1259017, Oct. 2023, doi: 10.3389/fmed.2023.1259017.

[206] Y. Wang, M. Lucas, J. Furst, A. A. Fawzi, and D. Raicu, 'Explainable Deep Learning for Biomarker Classification of OCT Images', in 2020 IEEE 20th International Conference on Bioinformatics and Bioengineering (BIBE), Cincinnati, OH, USA: IEEE, Oct. 2020, pp. 204–210. doi: 10.1109/BIBE50027.2020.00041.