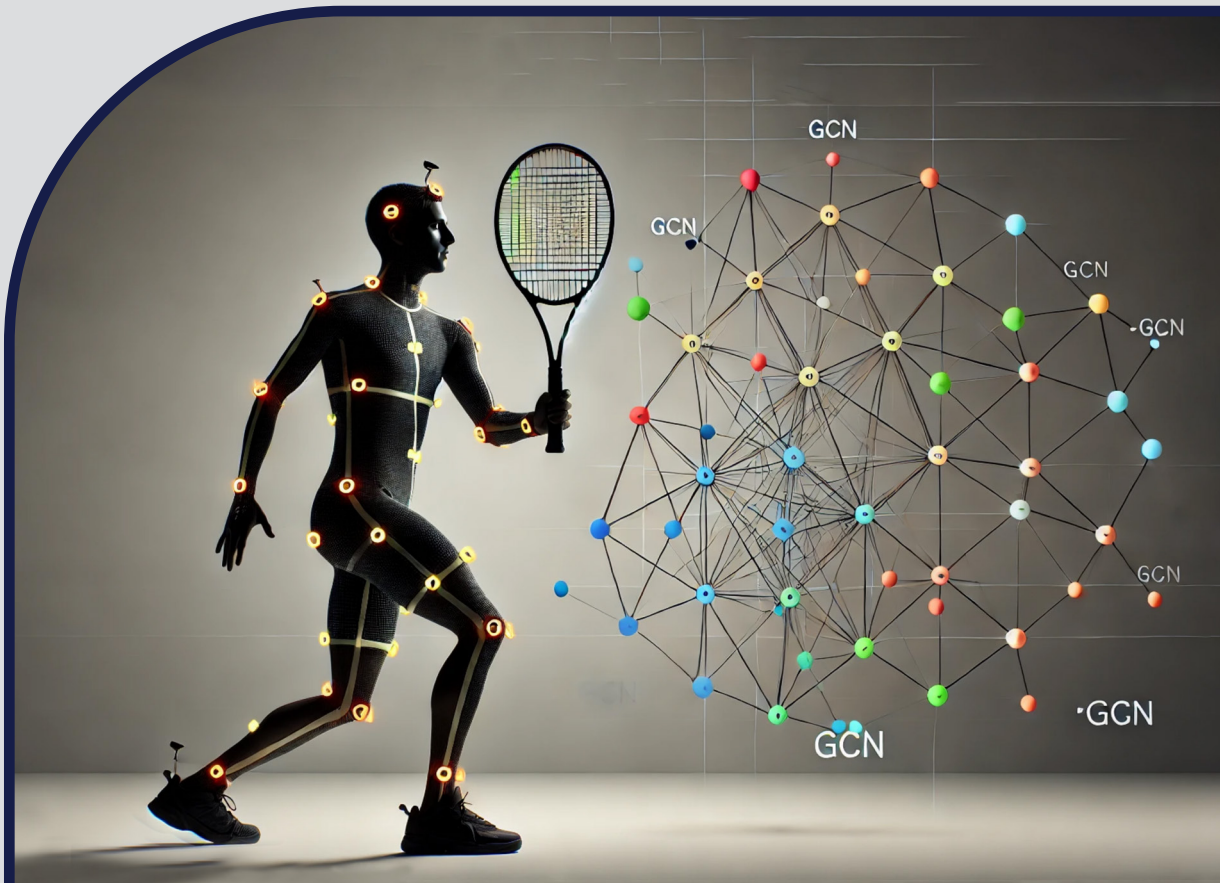
MONOGRAFIE

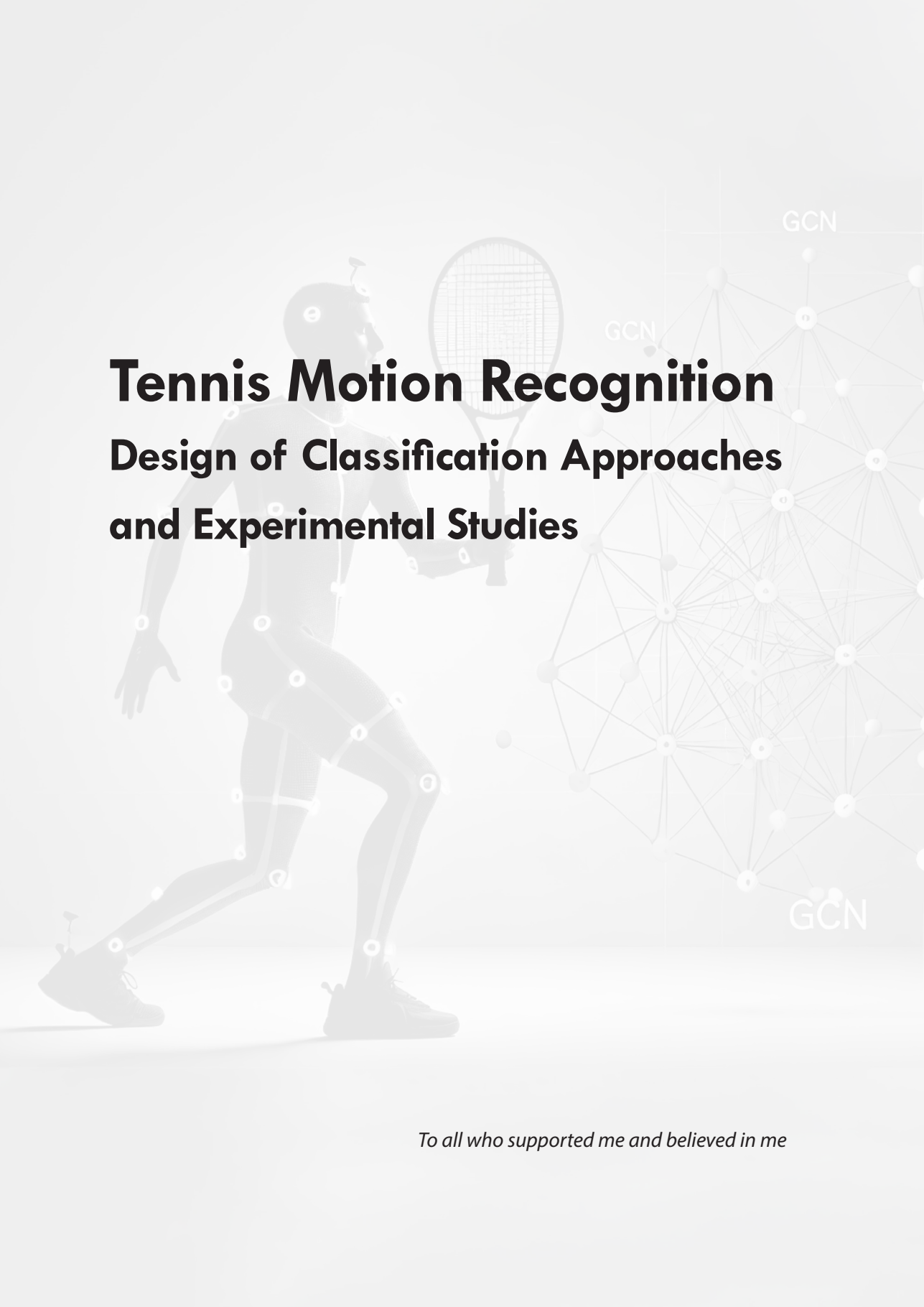
POLITECHNIKA LUBELSKA

Tennis Motion Recognition

Design of Classification Approaches and Experimental Studies

Maria Skublewska-Paszkowska





Tennis Motion Recognition

Design of Classification Approaches and Experimental Studies

To all who supported me and believed in me

MONOGRAPHS

OF LUBLIN UNIVERSITY OF TECHNOLOGY

Tennis Motion Recognition

Design of Classification Approaches and Experimental Studies

Maria Skublewska-Paszkowska

LUBLIN UNIVERSITY
OF TECHNOLOGY
PUBLISHING HOUSE

Lublin 2024

REVIEWERS:

Vimala Nunavath, PhD, Associate Professor, University of South-Eastern Norway
Prof. **Witold Pedrycz**, Ph.D., University of Alberta, Canada

AUTHOR:

Maria Skublewska-Paszkowska, Ph.D., ORCID: 0000-0002-0760-7126
Lublin University Of Technology – ROR: https://ror.org/024zjzd49

Proofreading: Ronald E. Day
Editor typesetting: Katarzyna Pełka-Smętek
Layout design of the series: Łukasz Maj

The publication was financed in the framework of the pro-quality program: "Grants for financing the cost of interdisciplinary high-scoring publication (Grant no. NBW/4/2023/P)".

The research program titled *Biomechanical parameters of athletes in individual exercises*, realized in the Laboratory of Motion Analysis and Interface Ergonomics was approved by the Commission for Research Ethics, No. 2/2016, dated 8.04.2016.

The author would like to thank the Tennis Academy POL-STAR Student Sports Club KS-WINNER Club for their support.

The 3DTennisDS dataset is publicly available at https://tennisdb.cs.pollub.pl/.

The illustration on the cover and pre-title page was generated by artificial intelligence (ChatGPT).

Unless otherwise indicated, the drawings are the author's own work of the monograph.

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

ISBN: 978-83-7947-607-7 (print version)
ISBN: 978-83-7947-608-4 (electronic version)
DOI: 10.35784/9788379476084
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

Imprint: 50 copies

Contents

ABSTRACT

Tennis Motion Recognition
Design of Classification Approaches and Experimental Studies

This monograph presents studies concerning the challenging topic of Human Action Recognition, which has recently become of increasing importance to those working in the field of Computer Vision. It contains an overview of the research literature on tennis movements recognition based on video, images, motion capture data, and data registered using various sensors.

In this monograph, three novel methods of tennis movements recognition are presented in detail with success, achieving high performance of the proposed classifiers. They all utilize Graph Convolutional Neural networks (GCN) solutions. Two methods are also applied using attention modules which improve feature extraction and thus make the classification better. The Spatial--Temporal Graph Convolutional Network, the Attention-Temporal Graph Convolutional Network and the Dual Attention Temporal Graph Convolutional Network are applied for recognition of the most frequently used tennis moves, such as: forehand, backhand, volley forehand, and volley backhand. Moreover, the forehand and backhand are further divided into two phases: 1) the preparation, starting from the player's positioning until the moment just before the ball makes contact with the racket, 2) the hit together with the racket swinging until the move is finished. These separations allow detailed analysis. For the purpose of this study, a 3DTennisDS dataset was created that contains motion capture data of the forehand, backhand, volley forehand, and volley backhand. They were registered via the Vicon motion capture system. Three proposed neural networks methods were verified based on the gathered dataset.

In this monograph, the interpretability aspect of GCN was also presented. This issue is important and was examined by presenting feature maps and heatmaps obtained after selected convolutional operations. These results showed how individual filters in GCNs extract features that are relevant for recognizing tennis movements.

Keywords: Graph Neural Networks, tennis movements recognition, fuzzification, attention modules, interpretability

List of abbreviations

A3T-GCN	– Attention Temporal Graph Convolutional Networks
CNN	– Convolution Neural Network
CRF	– Conditional Random Fields
CV	– Computer Vision
DA-GCN	– Dual Attention Graph Convolutional Network
DL	– Deep Learning
DT	– Decision Tree
FCN	– Fully Convolutional Network
FDA	– Fisher Discriminant Analysis
GCN	– Graph Convolutional Network
HAR	– Human Action Recognition
HMM	– Hidden Markov Model
IMU	– Inertial Measurement Unit
k-NN	– *k*-Nearest Neighbor
LSTM	– Long Short-Term Memory
MoCap	– Motion Capture
ML	– Machine Learning
NN	– Neural Network
RBF	– Radial Basis Function
RF	– Random Forest
ST-GCN	– Spatial-Temporal Graph Convolutional Neural Network
SVM	– Support Vector Machines
THETIS	– THree dimEnsional Tennis Shots

CHAPTER

1

Introduction

Human Action Recognition (HAR) has recently gained increasing attention in sport. Its main ideas are to identify players and to distinguish their actions or their teams' activities, usually from unknown data sequences [1] [2] [3]. HAR integrates Computer Vision (CV), Machine Learning (ML) and Deep Learning (DL) with image processing in order to extract the features that characterize the analyzed action or activities. Movement recognition of various sports has a pivotal potential in techniques analysis, sports statistics, developing virtual applications for increasing sport performance, improving learning purposes and finally in developing strategies to understand how the moves are performed [4]. This technique has also been utilized in tennis, which has gained worldwide attention recently, especially in tennis moves recognition, analyzing a player's performance and calculating match statistics.

Various HAR methods have been applied to different types of tennis data, such as videos, streaming and motion capture (MoCap) data. These sophisticated techniques enable one to collect more and more accurate data, which are further applied to obtain more precise analyses. A great number of studies have been performed using videos, mainly due to the easy access of this kind of data via webservices such as YouTube, which are inexpensive because of their low price and the high availability of cameras. Athletes' movements may be captured during their warm-ups, trainings, matches and competitions. Another type of data is gathered using various types of MoCap systems. They can be divided into: marker-based systems, markerless ones and those that operate based on the inertial measurement unit (IMU) [5]. Marker--based motion capture systems may be further categorized into optical passive and optical active. The first type utilizes the infrared cameras

to track retro-reflective markers attached to a participant's body or to additional objects, like a tennis racket. The latter type of marker-based system recognizes LED markers which reflect light from cameras. In a markerless system a video together with dedicated software are enough to track the athlete's body or additional items. The IMU systems use wearable technology and operate with inertial sensors that transmit data wirelessly to a computer or a smart device. Sport movement data are often gathered into datasets, publicly available or created for the purpose of research.

From two – or three-dimensional data feature extraction is performed in order to identify the searching object. For this purpose, deep learning, together with self-learning and transfer learning approaches, have achieved remarkable applications [1][6]. Feature maps are generated utilizing various artificial neural networks from complicated movement data. Among many deep learning methods, a very popular one which is often applied in CV is Convolution Neural Network (CNN). It plays a pivotal role in extracting features, especially from images. Due to the fact that in tennis movement recognition the previous frames of data are important, neural networks like RNN and LSTM are also often applied [4]. Taking into account a human body depicted in the form of its characteristic points (e.g., joints) and the connections between them, the application of Graph Convolutional Networks (GCNs) seems to be a natural solution for tennis recognition purposes and thus may also have a vast potential in HAR [7].

Tennis has recently become increasingly very popular. In 2021 almost 87 million people played it around the world [8]. This has resulted in increasing scientific interest through the application of new enlightened methods for investigating athletes' performance and matching statistics. Therefore, the primary aims of this study are to create and apply modern DL methods for MoCap tennis movements recognitions and to model time-series data. The main achievements are:

1. Developed novel DL methods for tennis movements recognition utilizing GCN and attention mechanisms that achieved high performance. The following tennis strokes are considered: forehand, backhand, forehand volley and backhand volley. Moreover, no--strokes are also included in the study in order to distinguish the player's movement on the court from performance of

the analyzed strokes. All tennis moves were gathered using an optical motion capture system.

2. Verification of how fuzzification of input data influenced performance of the implemented DL methods.
3. Creation a dataset of very precise tennis movements consisting of three-dimensional data, gathered using a Vicon motion capture system. All moves were performed by professional tennis coaches and players in the Motion Capture Laboratory.

In this monograph, the interesting research issues are raised. Detailed investigation has been performed in order to clarify the following current topics:

- RT.1. Determining whether Graph Convolutional Neural networks are suitable tools for tennis movement recognition based on motion data.
- RT.2. Another interesting aspect is how the type of data affects the performance of the classifier. In this study motion capture tennis moves are considered as in the forms of images as well as three-dimensional data. They include a full body tennis player model, a tennis racket model and the combination of these two types in one input.
- RT.3. Attention models and their influence on the classifier performance is another fascinating problem. Whether increasing numbers of these modules will provide extraction of the most relevant features should be investigated as well as whether they will improve the classifier performance.
- RT.4. Another aspect worth investigating is fuzzification of the input data based upon which the recognition is performed. An interesting issue is whether this process will improve the classification phases (preparation and hitting) of forehand and backhand strokes. These moves are very dynamic in which the boundary of these two phases is very difficult to determine and therefore may be incorrectly classified.

In this monograph, the overview of tennis motion recognition methods as well as the tennis movements classification based on graph convolutional networks are included. Many aspects of challenging recognition issues are investigated. Various approaches of GCN tools are proposed for motion capture tennis movement data. Each solution is described in detail. The classification results are presented and

compared for each input type. Moreover, it is of interest to investigate how the fuzzification process influenced overall outcomes. The study is performed utilizing images and three-dimensional data reflecting a tennis player model and a racket tennis model.

This comprehensive research gives insight into tennis strokes classification.

CHAPTER 2

Related works

In this chapter a thorough overview of tennis studies is presented. The literature review provides a new insight into tennis stroke detection research. Studies are divided into methods of registering various tennis strokes, such as video, motion capture systems and sensor recordings. Available datasets containing data from selected tennis strokes are also collected.

2.1. Tennis movement classification

HAR in various sport disciplines is a challenging task. Many studies involve recognition of individual actions as well as activities performed by many athletes. Recently many novel DL approaches for movement classification have been proposed. Classification of tennis movements has also gained increasing interest. These studies present various techniques applied for skeleton-based action recognition, tennis racket trajectories or combined ones taking into consideration both the whole athlete body and the tennis racket position. These studies are performed on three types of tennis data: video-based, MoCap data, and sensor data.

2.1.1. Video data

Tennis movement classification based on video data has recently been of great interest (Table 2.1). They usually combine two steps: feature extraction in order to obtain the region of interest (ROI) representing the athlete's movements and a classification method applied to distinguish the proper moves.

HAR using broadcast videos is recently a subject of extensive research. Left and right swing were recognized in [9][10][11]. A group of histograms based on optical flow (S-OFHs) represented motion. This approach was treated with spatial features. The classification was performed using Support Vector Machine (SVM), achieving 87.10% accuracy for frames [9] and 90.21% for clips [11]. The experiments were performed on broadcast video matches from various championships.

The classification of tennis-hit-the-ball, serve, and non-hit, utilizing Bags-of-visual-Words (BoW) and Spatio-Temporal Shapes (STS), was presented in [12]. The spectral regression-based implementation of the kernel Fisher discriminant analysis (FDA) was applied, obtaining up to 84.5% mean accuracy. The same tennis actions were recognized in [13] using a generative Maximum a Posteriori (MAP)-based three-class classifier, achieving 58.72% mean accuracy. The histograms of oriented gradients (HOG3D) feature vector were applied for detection of each tennis athlete's move.

Serve, left-swing topspin, left-swing backspin, right-swing topspin, right-swing backspin, as well as no-action as were classified utilizing RNN approach based on broadcast video from eighteen international competitions in [14]. The videos were divided into clips of tennis movements, which begin from the contact of a ball with a racket. Detectron2 was applied for detecting the tennis player's wrist and elbow. The presented method achieved 99.9% average precision and 99.5% average recall.

Many studies have been performed on a very-well known THETIS dataset [15] that is described in detail in 2.2 chapter. The authors of the database applied 12 non-linear SVMs for classification of all twelve tennis movements (backhand, backhand played with two hands, backhand slice, volley backhand, flat serve, forehand flat, forehand from openstand position, forehand slice, volley forehand, kick serve, slice serve, and smash). Video depth and video skeleton 3D were taken into consideration. Dense trajectory, histograms of oriented gradients (HOG), histograms of optical flow (HOF) and motion boundary histogram (MBH), as well as their combination were applied for feature extractions. Best results were achieved for the combined descriptors, obtaining 57.50% and 53.08% average accuracy for depth data and skeleton 3D, respectively. The same twelve tennis strokes were recognized in [16]. The low-level video processing, feature description

and dynamic phrase encoding were applied for feature extraction in the form of a player model. Linear-chain Conditional Random Fields (CRF) and SVM were utilized. Taking into account dynamic phrases the CRF was stated to be a better method for tennis movement recognition, achieving 86.44% average accuracy. The SVM classifier obtained much lower results, at the level of 51%.

All THETIS tennis moves were also classified [17]. For 2048 feature extraction, the Inception neural network was applied, while for classification the 3-layered LSTM network was used. The average accuracy for twelve tennis moves was 47.22%. Various types of tennis strokes (e.g., forehand flat, forehand slice) were misclassified. Another experiment concerned forehand, backhand and serve moves. Individual types of stroke data were grouped together. Average accuracy was achieved at the level of 43.19%.

The authors of [4] proposed a novel neural approach consisting of Inception V3 and historical Long Short-Term Memory (LSTM) networks for classification of all tennis strokes gathered in the THETIS dataset. The first network, pretrained on the ImageNet dataset, was applied for obtaining spatial information from RGB images as features. The 3-layer historical LSTM network generated and updated the feature vector taking into account the current frame and the previous one (at the time). In other words, the information about sequence of images was gathered. The proposed tool achieved 62% accuracy, which outperformed both the LSTM network and results obtained in [15].

Volley forehand, backhand, backhand slice, serve slice, smash, and flat service classification were performed utilizing the Attention-based LSTM network [18]. The DenseNet was applied for feature extraction from RGB data. The bi-directional LSTM network was used together with two attention modules. The first one, the channel attention module, indicated the most contributing features for learning, while, the second one, spatial attention, specified what within the particular feature was essential for learning. This novel approach was verified using the THETIS dataset, achieving an average 87.22% Recall (up to 97.67% for flat serve) and 85.63% Precision (up to 89.79% for slice serve).

Table 2.1. Tennis movements video-based classification. FH – forehand, FF– flat forehand, FO – openstand forehand, FS – forehand slice, BH – backhand, BH2 – backhand with 2 hands, BS – backhand slice, VF – volley forehand, VB – volley backhand, S – serve, SF – serve flat, SK – serve kick, SS – serve slice, SM – smash, H – hit, NH – none-hit, LS – left swing, RS – right swing

Video type	Feature extraction method	Tennis strokes	Classification method	Average accuracy	Reference
Tennis actions dataset (broadcast)	BoW STS	S, H, NH	FDA	84.5%	[12]
Tennis actions dataset (broadcast)	HOG3D	S, H, NH	generative MAP-based three-class classifier	58.72%	[13]
Broadcast video	S-OFHs	LS, RS	SVM	87.10%	[9]
Broadcast video	S-OFHs	LS, RS	SVM	90.21%	[11]
Broadcast video	Detectron2	FS, FT, BS, BT, S, NH	RNN	N/A	[14]
THETIS (depth video)	low-level video processing, feature descriptio, dynamic phrase encoding	BH, BH2, BS, VB, SF, FF, FO, FS, VF, SK, SS, SM	CRF SVM	86.44% 51.20%	[16]
THETIS (depth video)	dense trajectory MBH combination	BH, BH2, BS, VB, SF, FF, FO, FS, VF, SK, SS, SM	SVM	51.59% 54.32% 57.50%	[15]
THETIS (skeleton 3D)	dense trajectory MBH combination	BH, BH2, BS, VB, SF, FF, FO, FS, VF, SK, SS, SM	SVM	46.84% 50.78% 53.08%	[15]
THETIS (RGB video)	Inception	BH, BH2, BS, VB, SF, FF, FO, FS, VF, SK, SS, SM	LSTM	47.22%	[17]
THETIS (RGB video)	Inception	BH + BH2 +BS +VB, FF + FO + FS + VF, SF + SK+ SS, SM	LSTM	43.19%	[17]
THETIS (RGB video)	Inception	BH + BH2 +BS +VB, FF + FO + FS + VF, SF + SK+ SS, SM	Historical LSTM	62%	[4]
THETIS (RGB video)	DenseNet	VF, BH, BS, SS, SM, SF	Attention-based LSTM	N/A	[18]

2.1.2. Three-dimensional data

Three-dimensional data may be provided by recording movements with multi-cameras or using sophisticated motion capture systems.

In [19], tennis moves were captured using nine cameras with the frequency set to 50Hz. 3D coordinates were indicated based on retro-reflective markers attached to the participants' bodies and tennis rackets. The authors presented an algorithm for indicating the type of tennis motion, forehand and backhand, taking into consideration two markers placed on the hips and one on the right wrist. The second method was described for stance angle extraction that was identified by five markers attached on hips, the right wrist and two feet.

Motion capture systems, especially marker-based ones, allow for acquisition of very precise data. A great majority of studies concern various tennis strokes classification methods based on three--dimensional information (Table 2.2). Forehand, backhand, and serve were classified utilizing the Hidden Markov Model (HMM) [20]. The data were gathered using optical motion capture system with the frequency set to 100Hz. Fifteen retro-reflexive markers were attached to full bodies of five participants. In total, 500 records for each type were collected. This kind of information was transformed to discrete symbol sequences in the form of a position of the hand in each frame. The proposed approach achieved 82.14% accuracy. The methods for distinguishing the forehand swing among novice and advanced skill-level players were presented [21]. The moves were captured using a 9-camera eMotion (BTS) SMART-e 900 system with the frequency set to 50 Hz. Both body and racket were recorded with the use of 22 markers. Best results were achieved for spatiotemporal transformations using motion gradient vector flow and polynomial regression with the Radial Basis Function (RBF) classifier, reaching 84.5% accuracy for novice players and 94.6% for advance-skilled ones.

Table 2.2. Tennis movements motion capture-based classification. FH – forehand, BH – backhand, S – serve

Motion capture type	Data type	Tennis strokes	Classification method	Average accuracy	Reference
Optical	discrete sequence symbol	FH, BH, S	HMM	82.14%	[20]
9-camera eMotion (BTS) SMART-e 900, 50 Hz	3D	FH	RBF	84.5–94.6%	[21]

2.1.3. Sensor data

Tennis movement may be also recognized based on wearable sensors attached to the wrist of the tennis player [22][23][24][25][26][27][28] (Table 2.3). In [22], forehand, backhand and serve, as well as their extensions like topspin and slice, were classified utilizing the ResNet and Fully Convolutional Network (FCN). Additionally, two main groundstrokes were further divided into preparation, swing, follow-through and retraction phases. The movements were captured using SensorTile, development kit (STEVAL-STLKT01V1) of STMicroelectronics, Geneva, Switzerland. For each of the basic moves above 96% average accuracy was achieved for two DL solutions, while for each of five shots above 94% average accuracy using the ResNet was obtained (Table 2.3). For shot detection an algorithm using a finite state machine was developed. The quality of hitting load in tennis based on IMU sensors and video-based data of eight strokes (rally forehand, slice forehand, volley forehand, rally backhand, slice backhand, volley backhand, smash, serve along with false moves) was described [23]. Data were classified using six ML models. Among them two gained the best results: a cubic-kernel SVM reached 97.4% accuracy and a second cubic-kernel SVM – 93.2% accuracy. The classification of topspin forehand, slice forehand, topspin backhand, slice backhand, and smash with three IMU sensors attached to the feet and a tennis racket, respectively, was described [24]. Accuracy was reached from 94% for slice forehand up to 100% for topspin forehand and slice backhand. The longest common subsequence algorithm was applied. At the same time the player's steps were analyzed using SVM. The stroke and side steps were distinguished. Data were gathered from three right-handed male amateurs, and four left-handed male experts. Each of

the participants performed the strokes ten times. A study presenting tennis movement classification using an embedded JY-6 sensor module into a racket handle is described [25]. Forehand, topspin forehand, backhand, topspin backhand, smash and serve were taken into consideration. The study was performed using 320 records, divided into 160 strokes and 160 swinging moves. The fluctuation of acceleration based on moving windows was applied to moves recognition, achieving up to 96.8% accuracy. Forehand, backhand and smash obtained the highest accuracy.

A real-time mobile system for tennis strokes classification for training based on one SensorTile attached to a tennis racket was proposed [26]. Six tennis moves were taken into consideration, like: tennis swings: topspin forehand, subpar forehand, topspin backhand, subpar backhand, slice forehand and slice backhand. Five ML algorithms were used to classify them: SVM, NN, Decision Tree (DT), Random Forest and k-Nearest Neighbor (k-NN). 1200 sensor records were collected, 200 for each stroke. All the ML methods achieved above 93% average accuracy. The DT stated to be the weakest method gained 93.89% average accuracy, while the NN the best achieved 99.72%.

Another study concerning tennis movements classification was presented [27]. In this approach PIQ Robot, a sport tracker, was taken into account as sensor data. The Learning Vector Quantization (LVQ) was chosen as a classification tool. Ten tennis strokes were analyzed: forehand flat, forehand sliced, forehand lifted, backhand flat, backhand sliced, backhand lifted, serve, smash, volley forehand and volley backhand. The study involved only two tennis players. The obtained accuracy was up to 90%, which meant that the proposed ANN was promising for this type of data.

Table 2.3. Tennis movements classification based on sensor data. FH – forehand, FF – forehand flat, FT – topspin forehand, FS – forehand slice, FR – forehand rally, FSb – forehand subpar, FL – forehand lifted, BH – backhand, BF – backhand flat, BT – topspin backhand, BS – backhand slice, BR – backhand rally, BSb – backhand subpar, BL – backhand lifted, VF – volley forehand, VB – volley backhand, S – serve, SM – smash.

Sensor type	Number of sensors	Tennis strokes	Classification method	Average accuracy	Reference
SensorTile	1	FH, BH, S	FCN ResNet	96.34% 96.69%	[22]
SensorTile	1	FT, FS, BT, BS, S	ResNet	94.37%	[22]
SensorTile	1	FT, FSb, BT, BSb, FS, BS	NN DT RF *k*-NN SVM	99.72% 93.89% 98.61% 99.44% 98.06%	[26]
IMU	1	FH, BH	cubic kernel SVM	97.40%	[23]
IMU	1	FR, FS, VF, BR, BS, VB, S, SM	cubic kernel SVM	93.20%	[23]
IMU	1	FH+FT, FH+FS BH+BS, BH+BT, SM	longest common subsequence algorithm	94.00%	[24]
IMU	6 (full body)	FH, BH, S	SVM *k*-NN	82.43% 93.44%s	[28]
IMU	1 (right arm)	FH, BH, S	SVM *k*-NN	88.36% 89.41%	[28]
JY-6 sensor	1	FH, FT, BH, BT, S, SM	acceleration fluctuations	90.94%	[25]
PIQ Robot	1	FF, FS, FL, BF, BS, BL, S, SM, VF, VB	LVQ	0-90%	[27]

The comparison of classification methods based on video and IMUs was described [28]. Classification was performed using six IMU sensors attached to the full body as well as only one sensor attached to the right arm. SVM and *k*-NN methods were applied to forehand, backhand, and serve shots classification. The *k*-NN algorithm stated to be the most suitable method for this purpose, acquired 89.41% for one IMU and 93.44% for six IMUs, respectively. A very interesting approach was proposed for fusion of video-based and IMU-based data utilizing a *k*-NN algorithm. For one IMU accuracy was reached at a level of 98.68% and at 98.00% for back view video and side view video, respectively. Taking into consideration full body sensors, data of 100% and 99.67% accuracy was obtained from the back view video and

the side view video, respectively. The study was performed on a dataset containing 300 strokes.

2.2. Tennis datasets

Many studies, concerning tennis movements analysis, have been performed on existing datasets and are available online. They were gathered using various types of data, like video, sensors and motion capture data. They vary in the frequency of data recording, as well as the type of performed tennis moves.

The dataset consisting of primitive tennis actions, titled Tennis actions dataset, like serve, hit (defined as the moment of contact with a ball, except during a serve), and non-hit was proposed [12]. It was created based on two broadcast tennis female games, one single and one double, obtained from the Australian Open championships. Additionally, statistics of the above-mentioned actions were added. 122 serves, 386 hits and 2294 non-hits were gathered. In total, 65 minutes of tennis play occurred.

The THree dimEnsional Tennis Shots (THETIS) is the most well-known MoCap dataset [15]. 31 amateur players and 24 experienced ones performed twelve tennis movements that were recorded using Kinect's depth sensor in two various indoor rooms. In total, 8734 avi videos were obtained.

The Mimetics dataset [29] was also created using a Kinect camera. It consisted of 713 video clips representing 50 various mimed human actions, including playing tennis. Movements were performed by mime artists or by people from everyday life. However, this dataset does not include the specified tennis strokes performed by professional players.

The tennis dataset collecting tennis forehand, backhand, serve, volley forehand, volley backhand, and smash was recorded using optical motion capture, Optitrack Flex V100 with the frequency set to 100 Hz [30]. Body joints were captured using 34 retro-reflective markers. 17 tennis players from the Caldas-Colombia tennis league were participants.

A dataset was presented [22] containing sensor data of five tennis strokes: forehand slice, forehand topspin, backhand slice, backhand topspin and serve gathered while in training or in matches, mainly for

amateurs (6 male and 10 female). Another dataset containing 28,582 strokes from IMU was described [23]. A dataset containing video-based data as well as IMU-based data was presented [28]. Data were captured from five elite, nationally ranked, tennis players. Participants performed forehand, backhand, and serve movements. In total, 300 strokes were obtained. Nine cameras and six IMU sensors were applied.

CHAPTER

3

Methodology of creating 3DTennisDS

In this chapter an original methodology for creating a new tennis stroke dataset is described. It includes recordings of tennis strokes using a passive optical motion capture system. The preparation of the system and the participants are described. For the purpose of this study a new racket model is defined. Finally, the preparation of tennis data for publication is shown.

In scientific papers, only a few datasets containing three-dimensional data of tennis movements are widely utilized. The most well-known and frequently applied is THETIS that was gathered using Kinect sensors. Apart from markerless motion capture systems, marker-based ones are more accurate. However, only Tennis-Mocap, available as a publicity dataset, was created using an optical motion capture system, Optitrack Flex V100 [30]. It provided data in bvh (Biovision Hierarchy) format, which described data in a hierarchical way, in which the header defines the order of the data while the section contains the motion capture data.

However, there is a lack of available publicity datasets, where tennis movements can be stored using c3d format. These files keep three-dimensional coordinates of markers. Additionally, analog data, as muscles activity or ground reaction forces, may be stored as well. This format is very easy to use and modify. It combines a binary format together with its content using the characteristics of a database. c3d files have minimal demands on storage. Taking into account the above advantages, a new dataset of tennis movements was created using the optical Vicon motion capture system.

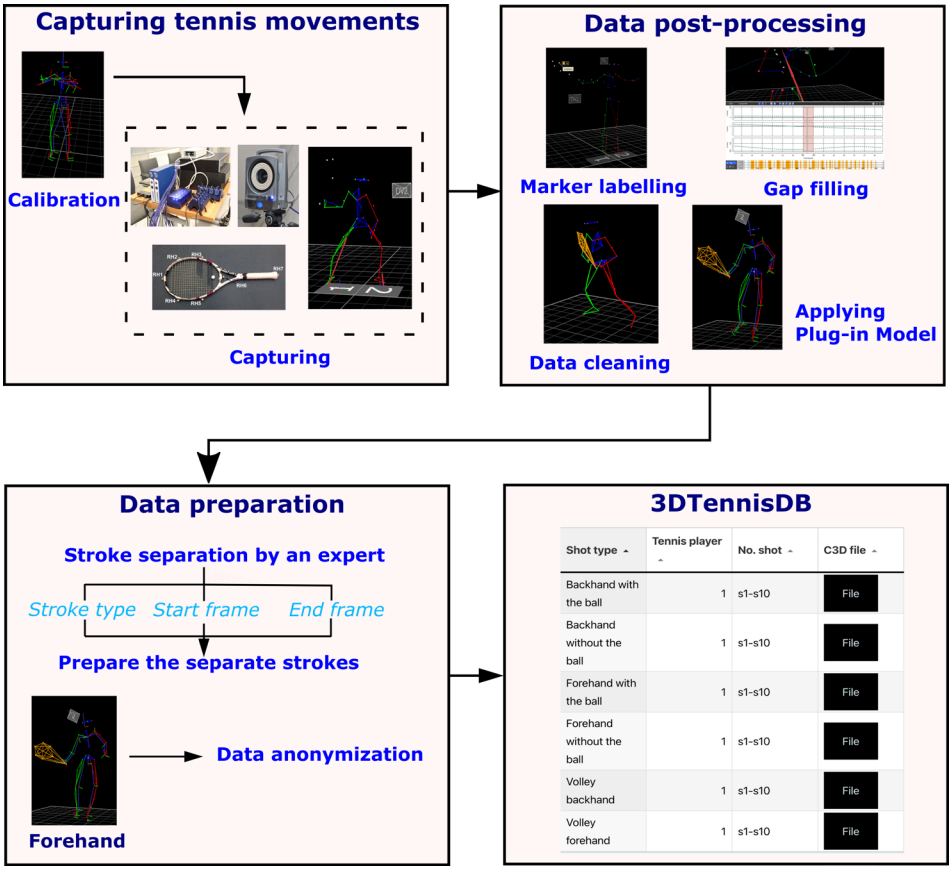


Fig. 3.1. Schema of developing 3DTennisDS [31]

The development of a new and sophisticated motion capture dataset, the 3DTennisDS, is very time-consuming and very demanding. The following steps for creating this dataset are depicted in Fig. 3.1. It consists of four main tasks. First, the motion sessions were prepared, during which, the most important movements of tennis players were recorded. Second, all registered data were post-processed. Third, the data were prepared for publication. Finally, the dataset was made public (at https://tennisdb.cs.pollub.pl/).

3.1. Capturing tennis movements

3.1.1. Motion capture system

The movements capturing was performed using the optical, 8-camera Vicon motion capture system (Vicon Oxford, UK). It consisted of T40S cameras, two Bonita reference cameras, Giganet hub, a desktop computer, and a set of accessories. The motion capture cameras operate in near infrared, up to 512Hz. However, the tennis moves were registered with the frequency set to 100Hz. The recording was performed in a special room without windows (which may produce additional reflections during a session). On each wall two cameras were located so that the final capturing area has a dimension of 3.4 by 5.4 meters. The placement of the motion capture cameras is presented in Fig. 3.2. The cameras register the position based on retro-reflective markers that are attached to participants or objects that are being recorded. During the movements recording a marker must be visible by at least two cameras. Only when such a condition is met, are its three--dimensional data registered. Two reference Bonita cameras are for digital video capturing. They are synchronized with the whole system, so it is very helpful in post-processing while labeling the markers and checking whether all markers are properly placed. Based on these data video files containing integrated video and a biomechanical, post--processed model of a participant may also be generated. The Giganet hub creates a link between motion capture cameras, digital cameras and a PC computer utilizing a 5-port Ethernet switch. The main idea of this device is data synchronization for all sources.

The Vicon's Nexus is an integral part of the motion capture system. This software is used to calibrate the system, register data and perform data post-processing. The set of accessories (like a calibration wand, retro-reflective markers and double-sided tapes) is vital for preparing the system and capturing the movements.

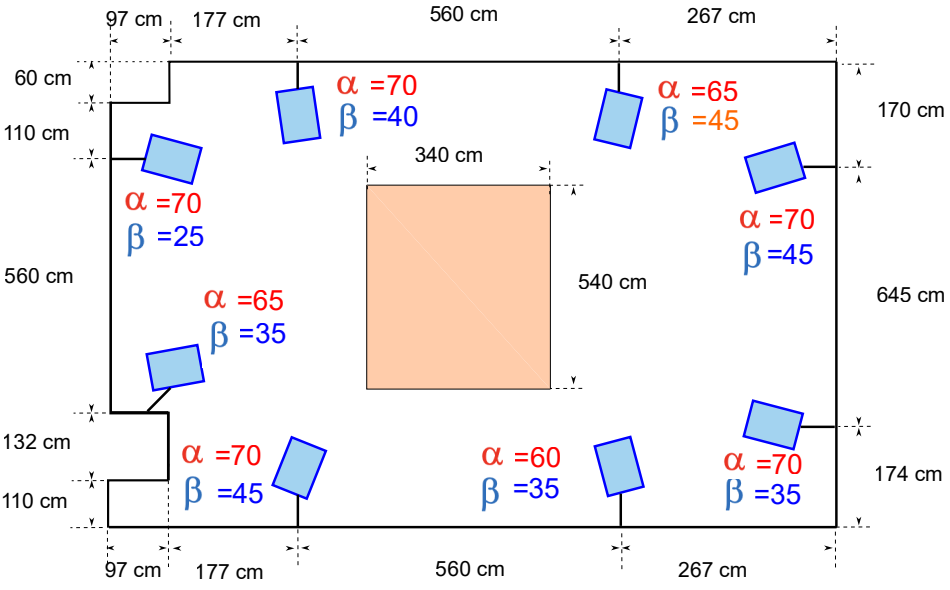


Fig. 3.2. The deployment of motion capture cameras α stands for the angle in the OX plane between the floor and the camera axis, β while for the angle in the OY plane between the floor and the camera axis

3.1.2. System calibration

The process of system calibration consisted of two stages: camera calibration and setting the volume origin. For the first part a special wand is used. While changing the position and orientation of this device, the Vicon system calculates the position and orientation of all cameras. The system learns how the cameras are relative to each other. It is essential for combining real and virtual worlds. Also, optical distortion of the camera lenses and any other non-linearity in the system are calculated. The correction matrix is computed.

In the second stage, a calibration device is measured and the global coordinate system is set. The coordinate system is essential for proper display and visualization of recorded data.

The global origin (0,0,0) is indicated by the calibration wand that is usually placed on the floor. It specifies the center of the capture volume, and the global axes (X,Y,Z). The axes indicate the three-dimensional plane.

The calibration process indicates the system accuracy. Therefore, performing this step accurately is crucial for the correct mapping of movements. The result of the calibration process is presented in Fig. 3.3.

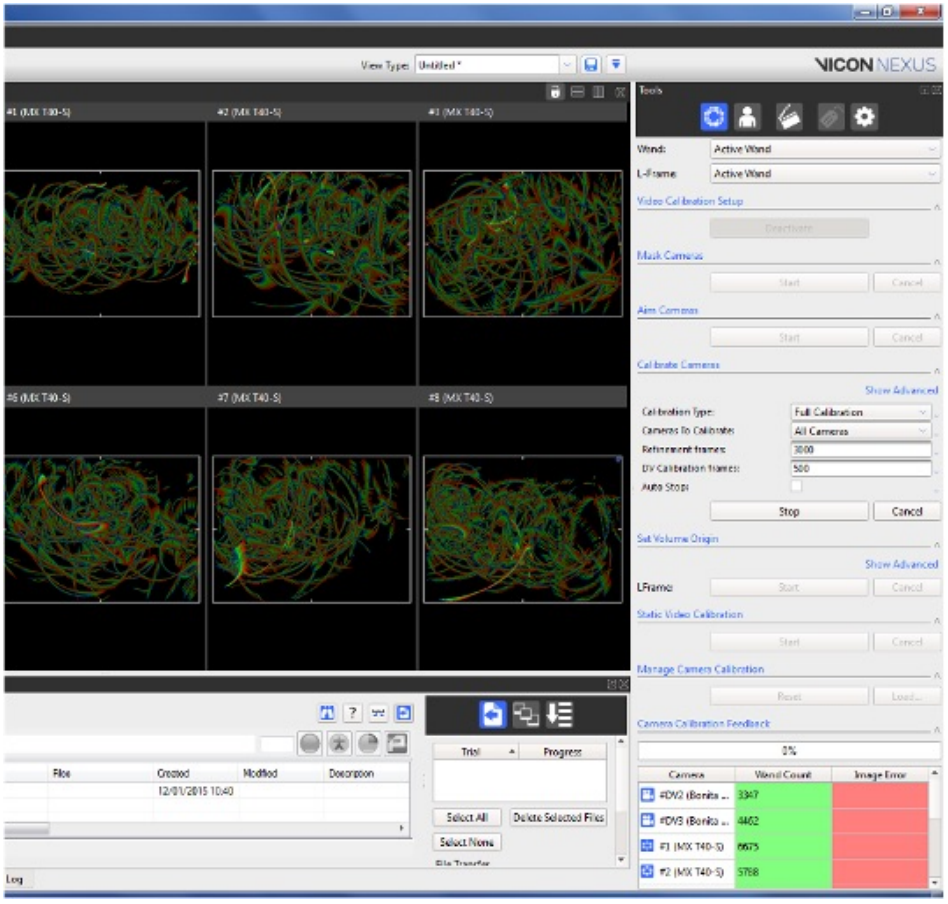


Fig. 3.3. System calibration result

3.1.3. Participants

Seven male and three female tennis coaches (age 23.7±4.58, height 1.77±0.13 m, weight 71.65±10.68 kg) were the participants of the study. They all signed consent for movement capturing. They are professionally active. They also have taken part in various tennis tournaments.

Each participant was prepared according to the Plug-in Gait model [32]. Thirty-nine retro-reflective markers were attached to the participant's body. One of the participants is presented in Fig. 3.4.

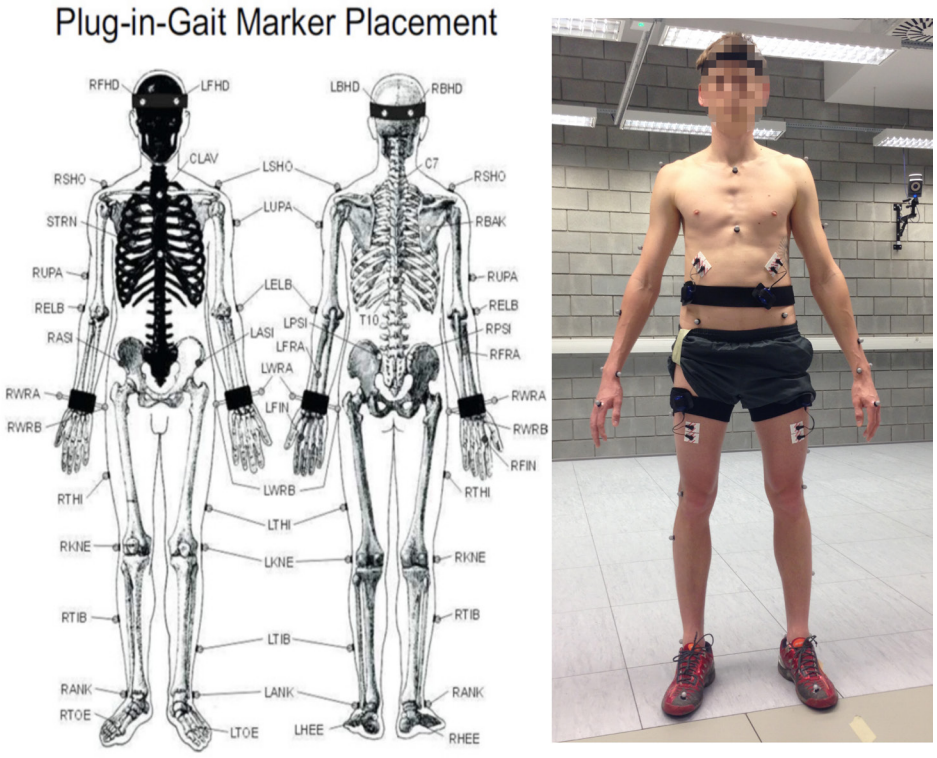


Fig. 3.4. Marker placement according to Plug-in Gait model

3.1.4. Motion capture session

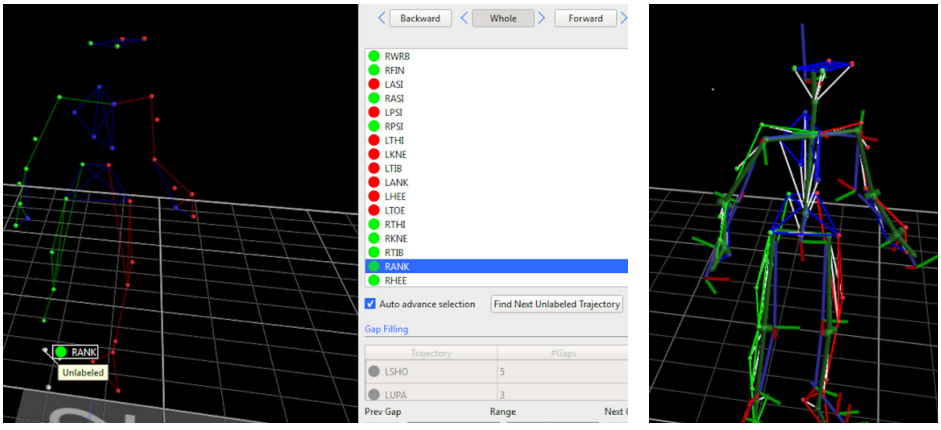
Motion capture session consists of calibration of the participant, creating a new model for a tennis racket and finally recording the movements.

3.1.4.1. Participant calibration

After attaching markers, each participant was measured for the purpose of calibration in the Nexus software. The following measures were performed for the left and right sides of the body: leg length, knee width, ankle width, shoulder offset, elbow width, wrist width, and hand width.

Additionally, the participant's height and weight were gathered. These values were entered into Nexus software and a new model was created.

The participant was recorded in a "motor bike pose" for a few seconds. This step is extremely important. First, verification of whether all markers attached in their proper positions is performed based on labeling. Second, the skeleton calibration is done. This process cannot be performed if the distance between two markers is too large, according to the Plug-in Gait model. Finally, the Static Plug-in Gait Model is applied. A high value of covariance ellipsoids indicates that the calibration is poor. Participant calibration is presented in Fig. 3.5.



a) b)

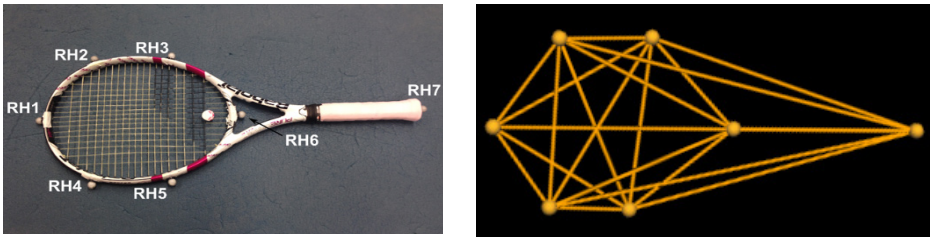
Fig. 3.5. a) Subject labeling according to Plug-in Gait model, b) Subject after applying Static Plug-in Gait

3.1.4.2. Creating a tennis racket model

While performing tennis strokes, racket position is a very important issue that should be taken into consideration in movement analysis. That is why capturing a tennis player moves should be done together with the racket so that the recorded strokes will be as natural as possible during real playing conditions.

Tennis player movements are captured using the Plug-in Gait model that is utilized into the Nexus software. However, for the purpose of recording racket movements, a dedicated tennis racket model needs to be created. Its shape was indicated by seven retro-reflective markers

(Fig. 3.6). One marker was attached to the top of the racket head, two on both sides of the racket head, one at the bottom of the racket head, and one to the bottom of the racket handle. This kind of marker deployment allows for further examination of the racket's trajectory.



a) b)

Fig. 3.6. a) Tennis racket with attached retro-reflective markers [31], b) tennis model

3.1.4.3. Capturing tennis movements

Participants of the study performed four various types of tennis strokes: forehand, backhand, forehand volley, and backhand volley. The first two were performed while moving and avoiding a small bollard placed on the floor so that the movements were as similar as possible to those performed on the court during a game or training. First, the participants did ten forehand strokes followed by ten backhand strokes without the ball. The tennis coach performed the movements in such a way that they were carried out from the same place, position. The moves were repeatable. Then, the same moves were performed with a ball that was thrown so that the stroke was carried out in the same position. A tennis net was set up in the laboratory to make it easier for players to perform tennis strokes (Fig. 3.7).

However, volley strokes were played while the athlete was standing in front of the tennis net. The ball was thrown alternately, once from the right and once from the left side. This ensured forehand and backhand volley movements. The participant had to approach the net, get into the appropriate position and perform a proper type of volley. Each stroke was recorded into a separate file. The capturing was performed with the frequency set to 100 Hz.

Fig. 3.7. Placement a tennis net while performing forehand and backhand strokes

3.2. Data post-processing

Each recording needed post-processing that consisted of four steps: marker labeling, gap filling, data cleaning, and in the case of a human subject, applying the Plug-in Gait model. Due to dynamic movements, some markers were covered, incorrectly registered or swapped with others. In the first stage the proper names (labels) must be specified in every frame for all markers. For the player model the markers were consistent with the Plug-in Gait model, while for the tennis racket model the markers were consistent with the designated names of the created tennis racket model. This part of post-processing was time--consuming because it required viewing the entire recording, frame by frame. This process is depicted in Fig. 3.8.

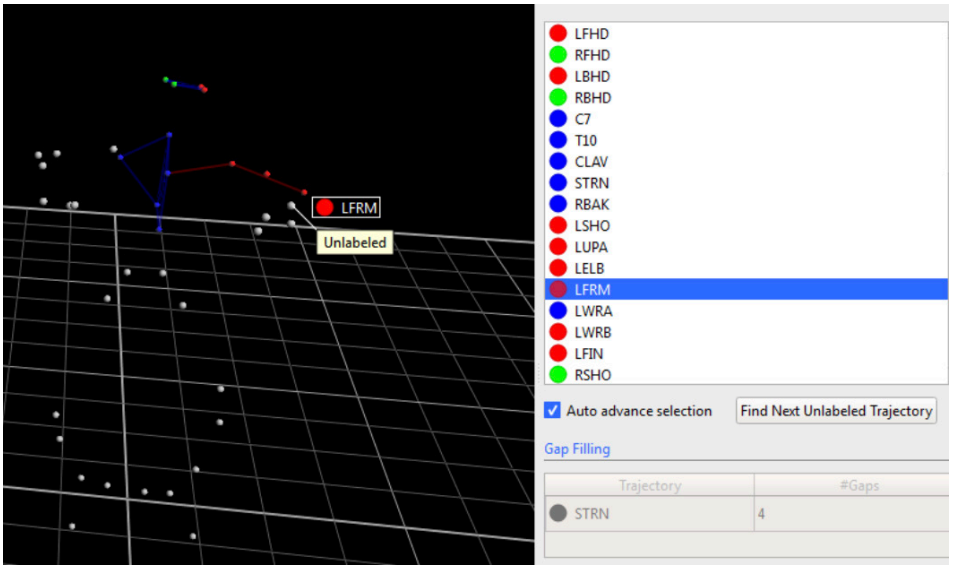


Fig. 3.8. Labeling the markers according to Plug-in Gait model

After capturing the movements, some of the markers were missing in part of the recordings. The idea of the second stage of post--processing was to interpolate the 3D positions of the marker based on its position before and after the gap. Various interpolated methods were integrated in the Vicon Nexus software. The most proper one was chosen in order to ensure the smoothness of the whole markers' trajectories. For the full body model various types of interpolations were selected, depending on the region of the body. However, for the tennis racket model the rigid body was the most frequently chosen algorithm. It is worth mentioning that these algorithms were appropriate only if the gap was not at the beginning nor at the end of the recordings. The process of interpolating a gap is presented in Fig. 3.9.

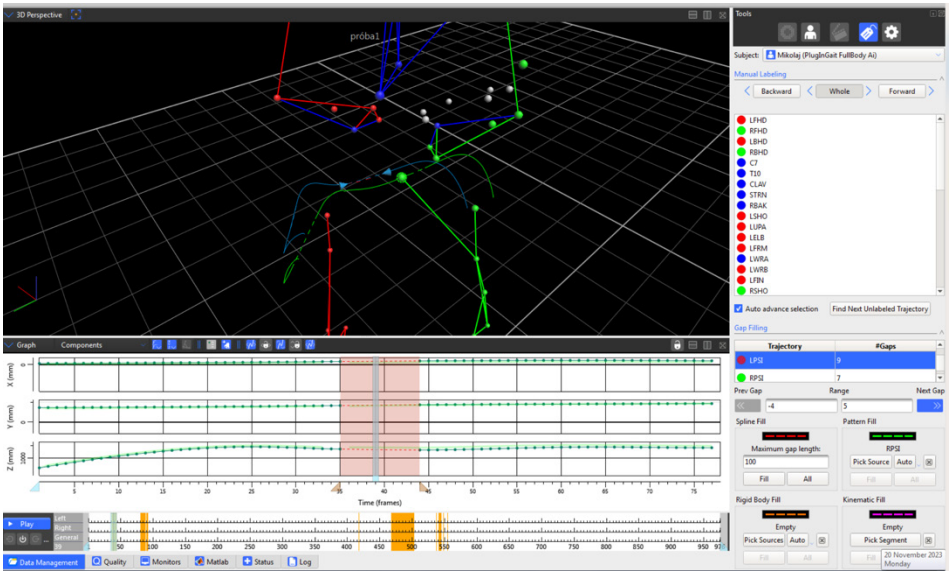


Fig. 3.9. Interpolating RASI marker utilizing pattern fill method

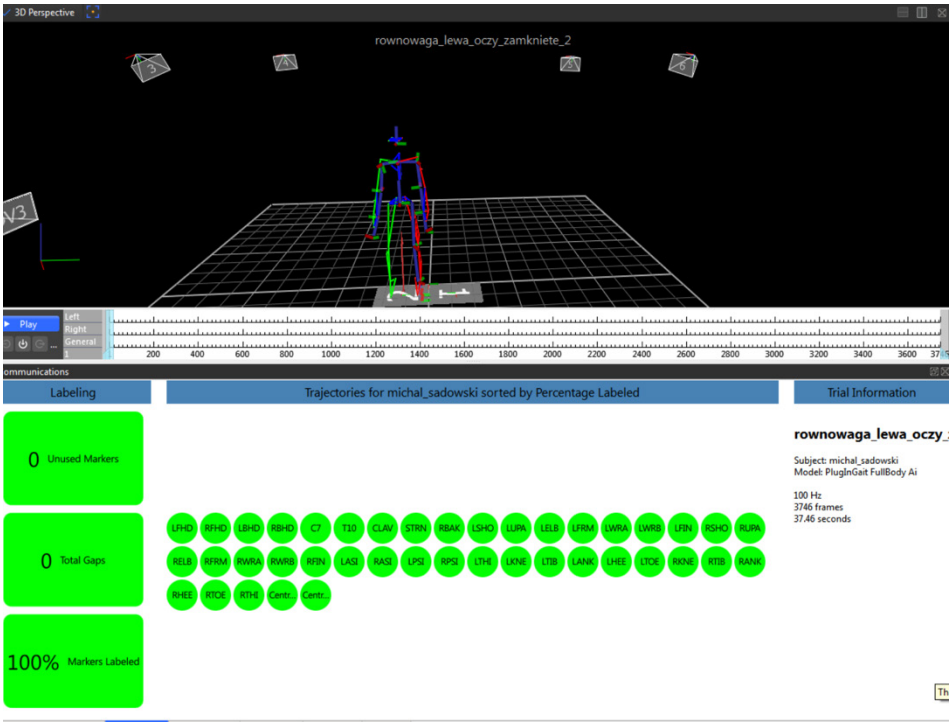


Fig. 3.10. 3D model after full gap feeling and applying the Plug-in Gait model

During the recordings some additional markers may appear, for example as a result of reflections. When the gaps interpolation perform inappropriately, additional markers may appear. In the third step of the post-processing, all of this kind of data were deleted. Finally, for the human-based model the Plug-in Gait model was applied. It generated body angles, forces, and moments. The post-processed tennis data is depicted in Fig. 3.10.

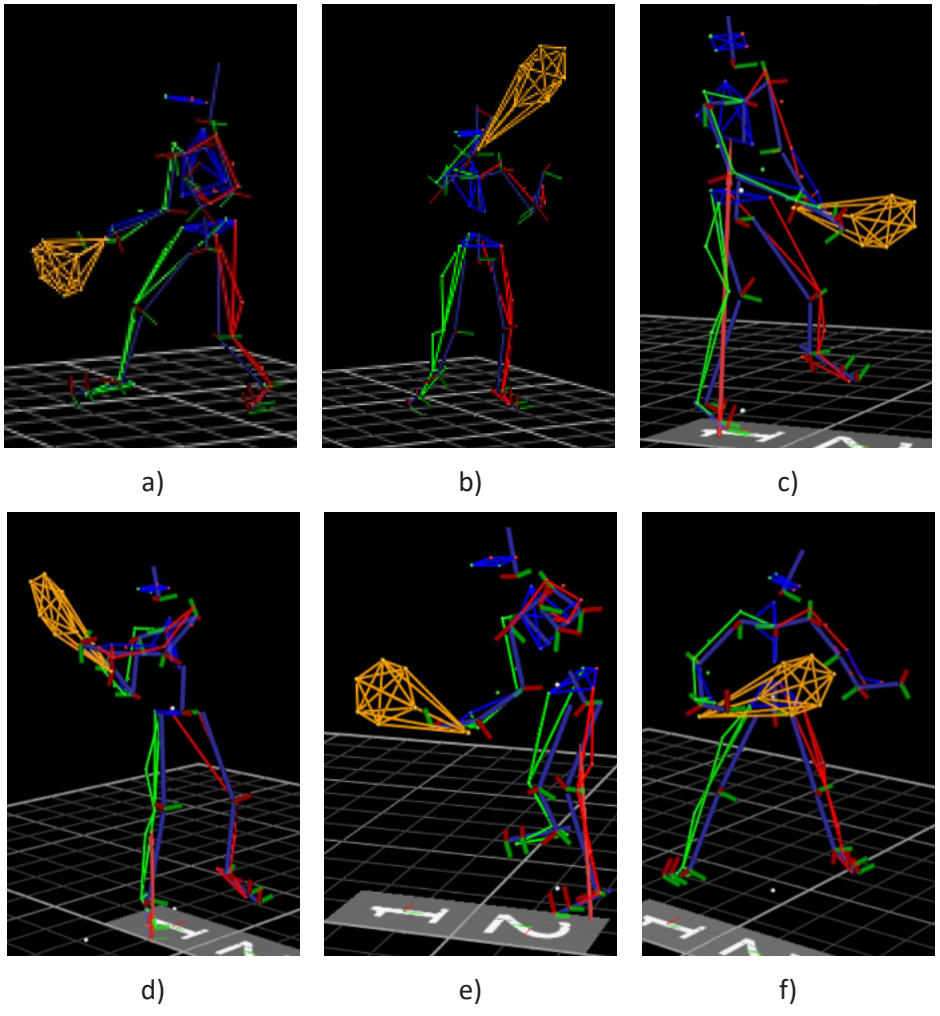


Fig. 3.11. Tennis strokes captured using the Vicon motion capture system: a) forehand preparation phase, b) forehand hit with swing, c) backhand preparation phase, d) backhand hit with swing, e) forehand volley, f) backhand volley

3.3. Data preparation

When the captured tennis movements were post-processed, the recordings were verified by experts. All incorrectly performed movements were removed. From the whole recordings the individual tennis strokes were separated indicating the beginning and the end frames based on the position of the tennis racket as well as the position of the player's body. The following types were obtained: forehand, backhand, volley forehand and volley backhand. Additionally, forehand and backhand strokes were further divided into phases of the moves. The preparation phase started from the moment the player raised the racket at the highest point until the moment right before the racket hit the ball. The hitting phase began from the moment the ball contacted the racket through the swing until the movement ended and the racket was positioned behind the player's body. Due to the fact that the volley strokes are short, they were not divided into phases. The whole stroke separation was performed in the Vicon Nexus software. The following tennis strokes are depicted in Fig. 3.11.

Another important issue was to make all recordings anonymous. In data the name of the participant was stored in each marker as well as in the analog data. That is why all sensitive data were removed to protect privacy.

3.4. 3DTennisDS

The 3DTennisDS dataset is available for public use at https://tennisdb.cs.pollub.pl/ [31]. The whole dataset was created in the form of a table. The strokes were divided into the tennis participants and the type of moves. For each player the number of strokes is specified. Additionally, the link to the Plug-in Gait model was made available. The racket's model is also available for downloading in order to incorporate it into the project.

CHAPTER 4

Data classification – selected aspects

The chapter contains theoretical foundations of data classification, including Convolutional Neural Networks, Spatial-Temporal Graph Convolutional Neural Networks, and attention modules. All these topics are essential to better understand studies involving tennis stroke recognition.

Data classification is one of the tasks of CV. It involves image analysis and assigning one of the fixed category labels to the picture. Because an image is represented as a great variety of numbers, a set of various computations have to be computed. There are many challenges that data classification must face, such as: various views of the image, changes in illumination, deformation of the objects, occlusion that occurs when an image contains only partial information, similarity between object and its background. For image classification, Convolutional Neural Networks (CNNs) have been stated to achieve the highest performance.

4.1. Convolutional Neural Networks

CNNs have obtained impressive results in CV, especially in image – and signal-based analysis, recently [33][34]. The idea of this technique has been known for several dozen years, but the development of computer science has allowed it to be used in practice in many two – and three-dimensional data processing.

Yan LeCun is one of the precursors of this deep learning approach. In [35] the researcher (together with B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel) proposed an architecture of the Convolution Neural Network for handwritten zip code digits classification. The dataset was constructed using images of a great variety of sizes, written styles and instruments of handwritten zip codes of U.S. mail that passed through the Buffalo, NY post office.

7291 images were used for training the network and 2007 for testing and calculating its performance. All the digits were normalized to fit an image into a 16x16 pixel matrix. The image was input to the network, while the output consisted of ten units that corresponded to ten defined classes. The proposed network architecture consisted of three hidden layers. The first layer had 12 groups of 64 units that were represented as 12 independent 8 by 8 feature maps, the second one as 12 feature maps of 16 units arranged in 4 by 4, and finally the third layer consisted of 30 units that were fully connected to the previous layer. Each unit of feature maps in the hidden layers took 5 by 5 neighborhood on the input plane.

In [36], another one of the first CNN architectures was presented which consisted of eight layers combined with the ReLu function. Five convolutional layers and three fully-connected ones were applied. Each convolutional layer was followed by a max pooling method to reduce the feature maps resolutions. In the end of the network architecture 1000-way Softmax was used to generate 1000 classes as the network results. In this approach the overlapping max pooling was introduced instead of average pooling. The following number of neurons in the subsequent layers were: 253, 440–186, 624–64, 896–64, 896–43, 2644096–4096–1000.The first layer consisted of 96 kernels of 11x11x3 size, the second of 256 kernels of 5x5x48 size, the third of 384 kernels of 3x3x256 size, the fourth of 384 kernels of 3x3x192 size, and finally the fifth of 256 kernels of 3x3x192 size.

The CNN approach has got many advantages [34]. First, local connections among neurons are applied. It means that only selected neurons from the previous layer are connected to the proper neuron. It not only reduces the parameters, but also speeds up the convergence. Second, a weight sharing technique is applied. Therefore, the same weights can be assigned to the group of connections and as a result the parameters are also decreased. Finally, down-sampling of the image reduces the data processed but retains relevant information.

The basic architecture of CNN is constructed with three layers (Fig. 4.1): Convolutional, Pooling and Fully Connected layers.

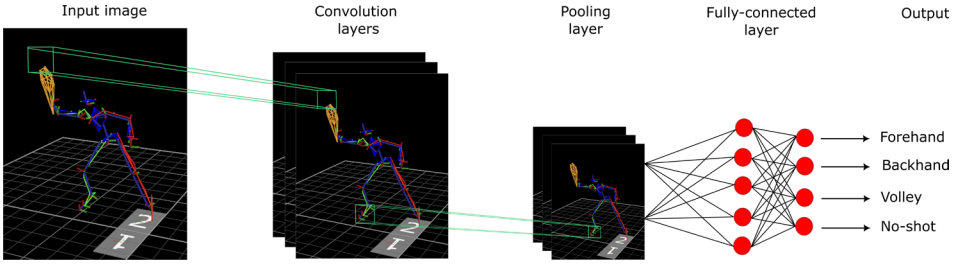


Fig. 4.1. 3-layer CNN architecture for four types tennis movements

The first layer obtains feature representations (feature maps) of the input using several convolution kernels (also called filters). Each kernel corresponds to a specific feature map. It is computed based on a region of neighboring neurons (neuron's receptive field) from the previous layer. A new feature map is indicated in two steps. First, a convolution operation of the input with the learnt kernel is computed (Fig. 4.2). The filter is put over the input and the multiplication operation is performed between the kernel and the portion of an input (neighborhood) over which the filter is covered. The filter is then moved with a set stride from left to right and from top towards bottom of the input. After the filter is applied to the whole input, the output image is obtained with decreased dimensions (width and height). Second, a nonlinear activation function is applied to the obtained result. It is worth noting that a particular feature map is achieved using various kernels and is given by the formula 4.1 [37]:

$$z^l_{i,j,k} = {w^l_k}^{\mathrm{T}} x^l_{i,j} + b^l_k \tag{4.1}$$

where w^l_k and b^l_k denote weight vector and bias for the k-th filter on the l-th layer, respectively, while $x^l_{i,j}$ stands for the input location at (i,j) coordinates on the l-th layer. A very important factor is that a particular weight vector is shared, which results in reducing the complexity of the model as well as providing its easier training. In order to indicate the nonlinear features, the activation functions,

defined as $a(\cdot)$ are applied. The activation function for convolutional feature $z_{i,j,k}^{l}$is given by formula 4.2 [37]:

$$a_{i,j,k}^{l} = a\left(z_{i,j,k}^{l}\right) \tag{4.2}$$

There are three most-commonly applied functions: sigmoid (4.3), tanh (4.4) and $ReLu$ (4.5). They are depicted in Fig. 4.2.

$$sigm = \frac{1}{1 + e^{-z_{i,j,k}^{l}}} \tag{4.3}$$

$$tanh = \frac{e^{z_{i,j,k}^{l}} - e^{-z_{i,j,k}^{l}}}{e^{z_{i,j,k}^{l}} + e^{-z_{i,j,k}^{l}}} \tag{4.4}$$

$$ReLu = max\left(z_{i,j,k}^{l}, 0\right) \tag{4.5}$$

There are many modifications of the ReLu function, such as leaky $ReLu$ (4.6), parametric $ReLu$ (4.7) or randomized $ReLu$ (4.8) [37].

In order to eliminate one of the $ReLu$ zero gradient disadvantages, the Leaky $ReLu$ was proposed [38]:

$$LReLu = max\left(z_{i,j,k}^{l}, 0\right) + \lambda min\left(z_{i,j,k}^{l}, 0\right) \tag{4.6}$$

where λ stands for a predefined parameter in range [0–1].

In parametric $ReLu$, instead of a predefined value, a parameter was applied which changes values in order to improve accuracy [39].

$$PReLu = max\left(z_{i,j,k}^{l}, 0\right) + \lambda_{k} min\left(z_{i,j,k}^{l}, 0\right) \tag{4.7}$$

where λ_k stands for the indicated parameter of the *k*-th channel.

Randomized $ReLu$ is another modification of the $ReLu$ method [40]. In this solution the parameter is randomly selected in the training process and additionally improved in testing.

$$RReLu = max\left(z_{i,j,k}^{(n)}, 0\right) + \lambda_k{}^{(n)} min\left(z_{i,j,k}^{(n)}, 0\right) \tag{4.8}$$

where $z_{i,j,k}^{(n)}$ stands for input at the (i,j) position on the k-th channel of the n-th example, while $\lambda_k{}^{(n)}$ for the corresponding sampled value.

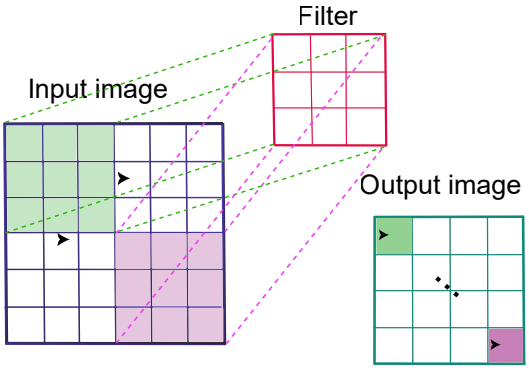


Fig. 4.2. Convolution operation

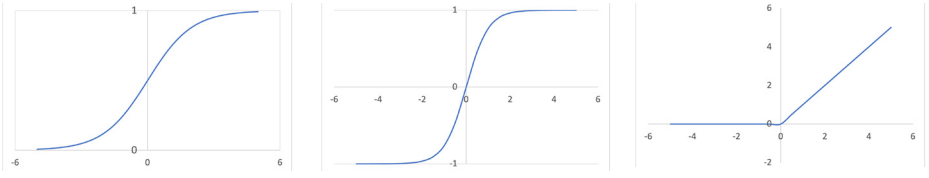


Fig. 4.3. Activation functions: a) sigmoid, b) tanh, c) $ReLu$

Extracting features is a very crucial aspect of CNN. The first layer of convolution operations extracts low-level features (e.g., color, edges or textures), while the subsequent layers extract more conceptual characteristics, namely high-level features.

The second layer of CNN architecture, the Pooling layer, is responsible for downsampling the features obtained in the convolutional

layer. It is usually located between two convolutional layers. The pooling function is defined as [37]:

$$y^l_{i,j,k} = pool(a^l_{m,n,k}), \forall (m,n) \in \mathscr{R}_{ij} \tag{4.9}$$

where $\mathscr{R}_{ij}$ denotes the local neighborhood around the (i,j) position and $a^l_{m,n,k}$ is the feature map localized at the (m,n) position of the l-th layer obtained using the k-th filter. There are two types of pooling functions very often applied: max pooling (4.10) [37][41], and average pooling (4.11) [37][42]. The first one indicates the most characteristic features, while the second one the average of all features.

$$y^l_{i,j,k} = \max_{(m,n)\in\mathscr{R}_{ij}} a^l_{m,n,k} \tag{4.10}$$

$$y^l_{i,j,k} = \sum_{(m,n)\in\mathscr{R}_{ij}} a^l_{m,n,k} \tag{4.11}$$

L_p is another example of pooling function that was stated to obtain better generalization [37][43][44]. It is given by formula 4.12. For p equal to one the L_p corresponds to average pooling, while for p equal to infinity, corresponds to max pooling.

$$y^l_{i,j,k} = \left[\sum_{(m,n)\in\mathscr{R}_{ij}} {a^l_{m,n,k}}^p \right]^{1/p} \tag{4.12}$$

In [37][45] a mixed pooling was proposed that combined both max and average pooling:

$$y^l_{i,j,k} = \lambda \max_{(m,n)\in\mathscr{R}_{ij}} a^l_{m,n,k} + (1-\lambda)\frac{1}{\lfloor \mathscr{R}_{ij} \rfloor} \sum_{(m,n)\in\mathscr{R}_{ij}} a^l_{m,n,k} \tag{4.13}$$

where λ stands for indicating one of the pooling types, which is a random value equal to one or zero.

Usually, the CNN architecture contains several convolutional layers followed by pooling layers. The last layer has one or more fully-connected layers that give high-level interpretation [37][46][47][48]. This part performs the classification using the features that are extracted in the previous layers. In this layer each node of the output layer is connected to all nodes from the previous layer. The final part of this layer has an operator that at the end gives a probability in range [0,1]. The Softmax or SVM are very commonly used functions.

The optimization of the CNN, which means finding the best network parameters, is about minimizing the loss function, which is given by the formula 4.14 [37][46].

$$\mathscr{L} = \frac{1}{N}\sum_{n=1}^{N} \ell\left(\theta; y^{(n)}, o^{(n)}\right) \tag{4.14}$$

where θ denotes CNN parameters (weight vectors and bias values), N – number of relations between n-*th* input and corresponding n-*th* output in a form of $\{x^{(n)}, y^{(n)}\}$; $n \in [1, ..., N]$, and finally $o^{(n)}$ stands for output of the network. Four examples of loss functions are defined [37][46]: Hinge (4.15), Softmax (4.16), Contrastive (4.19) and Triplet (4.21). The Hinge loss is usually applied in the SVM classifier.

$$\mathscr{L}_{Hinge} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K}\left[max\left(0,1-\delta\left(y^{(i)}, j\right)w^{T}x_i\right)\right]^{p} \tag{4.15}$$

where w stands for weight vector, x_i for *i-th* input and $y^{(i)} \in [1, ..., K]$ – is the correct class label from the set of K classes. It should be noted that $\delta\left(y^{(i)}, j\right) = 1$ for $y^{(i)} = j$ and otherwise $\delta\left(y^{(i)}, j\right) = -1$. The parameter $p = 2$ indicates the squared function, which is used in deep networks.

Softmax is one of the most commonly applied functions in CNN approaches. It calculates the prediction from the [0–1] range. It is given by formula 4.16.

$$\mathscr{L}_{Softmax} = -\frac{1}{N}\left[\sum_{i=1}^{N}\sum_{j=1}^{K} 1\left\{y^{(i)} = j\right\} log p_j^{(i)}\right] \tag{4.16}$$

where $p_j^{(i)}$ stands for the prediction of the *j-th* class for the *i-th* input (4.17),

$$p_j^{(i)} = \frac{e^{z_j^{(i)}}}{\sum_{l=1}^{K} e^{z_l^{(i)}}} \tag{4.17}$$

The $z_j^{(i)}$ is the activation of the densely connected layer, given by formula 4.18.

$$z_j^{(i)} = w_j^T a^{(i)} + b_j \tag{4.18}$$

The contrastive loss function is applied in weakly-supervised networks based on matching and non-matching pairs of data (4.19) [37][46]. The pairs may be defined as input data $\left(x_\alpha^{(i)}, x_\beta^{(i)}\right)$ and the corresponding output data at the *l-th* layer $\left(y_\alpha^{(i,l)}, y_\beta^{(i,l)}\right)$, where $l \epsilon [1, ..., L]$.

$$\mathscr{L}_{Contrastve} = \frac{1}{2N} \sum_{i=1}^{N} (p) d^{(i,L)} + (1-p) max\left(m - d^{(i,L)}, 0\right) \tag{4.19}$$

where d stands for the Euclidean distance between features (4.20), p for the parameter that is equal to one if $\left(x_\alpha^{(i)}, x_\beta^{(i)}\right)$ is a matching pair, otherwise p is equal to zero, while m is the margin parameter having an influence on non-matching pairs.

$$d^{(i,L)} = ||y_\alpha^{(i,L)} - y_\beta^{(i,L)}||_2^2 \tag{4.20}$$

Triplet is another example of loss function [37][46]. The triplet is defined as $\left(x_\alpha^{(i)}, x_p^{(i)}, x_n^{(i)}\right)$, where the first element stands for anchor instance, the second and the third ones for positive and negative instances from the same class of $x_\alpha^{(i)}$, respectively. The triplet function is given by formula 4.21.

$$\mathscr{L}_{Triplet} = \frac{1}{N}\sum_{i=1}^{N} max\left\{d_{(\alpha,p)}^{(i)} - d_{(\alpha,n)}^{(i)} + m, 0\right\} \tag{4.21}$$

where the distances are defined as [37][46]:

$$d_{(\alpha,p)}^{(i)} = ||y_{\alpha}^{(i)} - y_{p}^{(i)}||_2^2 \tag{4.22}$$

$$d_{(\alpha,n)}^{(i)} = ||y_{\alpha}^{(i)} - y_{n}^{(i)}||_2^2 \tag{4.23}$$

Since the first early attempts of CNN, that is the LeNet-5 in 1998 and the AlexNet in 2012, many CNN approaches have been developed. The VGGNets, created by the Visual Geometry Group from Oxford University, are definitely worth mentioning [34][46]. They include the following networks: VGG-11, VGG-13, VGG-16 or VGG-19. They achieved greater depths of CNN and thus improved network performance. 3x3 kernels instead of 5x5 were applied. The total parameters were reduced by about 45% [49].

GoogleLeNet solutions, like Inception (v1, v2, v3 and v4) and ResNet also obtained excellent results in image classification tasks [34][49]. They applied different kernel sizes that enabled one to extract feature maps of various scales of an image. The number of channels were reduced, which in turn lowered computational costs. Some of the networks eliminated the covariate shift problem by utilizing batch normalization. New solutions contained more and more deep layers. This caused a vanishing or exploring gradient and, as a result, an increase in training and test errors. The Residual Networks, created in 2015, reduced the number of parameters as well as improved their nonlinearity by applying 1x1 convolution kernels. These approaches introduced two – or three – layer residual blocks [50] that allowed skipping the connections between layers. Some layers were omitted. An advantage of this adjustment is that all layers are important and necessary to obtain high network performance.

Light models of CNN, called MobileNets, have also been developed, mainly dedicated to resource-limited devices, such as smartphones [34]. They utilized depth-wise separable convolutions together with

other techniques, such as: a multiplier for the number of channel reductions, inverted residual blocks, lightweight attention mode or hard swish activation function to make computation costs lower with minimum loss of accuracy [51][52]. The ShuffleNets are other implementations for mobile devices [53][54]. These solutions utilized pointwise group convolution. In order to ensure extracting proper features, the channel shuffle was also necessary to allow for feature flow among channels. The reduction of computation costs is mainly obtained by combining 3x3 average pooling layers in shortcut paths, reducing elementwise operations (like $ReLu$) as well as adjusting the number of convolution groups.

LetNet-5 and AlexNet introduced very important foundations on the basis of which solutions for image segmentation and classification have been and are still being developed. Increasingly deeper and deeper neural networks allow for effective image and video processing. Deep learning made it possible to perform challenging image analysis tasks. The CNN has performed excellently in CV due to the ability of extracting high-level featuring and thus performing pattern recognition issues.

4.2. Graph Convolutional Neural Networks

In traditional CNNs there is a difficulty of carrying out the local features stored in the form of a graph [55]. This problem appears to be caused mainly by unstructured features as well as the variable number of neighbors of a given node. However, in images, where there is a constant number of neighboring pixels, CNN approaches have been implemented successfully.

Graphs are widely applied in a great variety of areas mainly because of their power. They allow for the storing of complex data together with their relationships. There are a large variety of studies concerning data represented as graphs, such as interactions between users and products for e-commerce recommendations, drug recognition based on molecules or citation network of papers. For these kind of structures, also skeletal ones, that store irregular types of data, GCNs have been applied with great success due to their high performance [56]. GCNs leverage the structure of the graph and aggregate the information from nodes' neighbors using a convolutional way. This model

has obtained much attention as a result of various node classification, prediction or recommendation tasks. Because GCN is growing developing research area, many studies have resulted in implementing new models including graph combining with auto-encoder [57][58][59] recurrent neural networks [57][60][61] attention models [57][62] [63] and generative models [57][64][65][66].

Graph G (4.24) may be represented as a set of nodes V, edges E and the adjacency matrix A. The number of nodes is set to n, while the number of edges to m [57].

$$G = \{V, E, A\} \tag{4.24}$$

The adjacency matrix $A(k,l)$ between node k and l has a value equal to zero if there is no connection between these nodes, otherwise the value is set to one.

A diagonal matrix D, the degree matrix of adjacency matrix, is given by formula 4.25.

$$D(k,k) = \sum_{l=1}^{n} A(k,l) \tag{4.25}$$

The Laplacian matrix of A is defined as 4.26 and the symmetrically normalized Laplacian matrix as 4.27.

$$L = D - A \tag{4.26}$$

$$\widetilde{L} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \tag{4.27}$$

where I stands for the identity matrix.

A graph data of nodes is defined as a vector $x \in \mathbb{R}^n$ where $x(k)$ is the data of the *k*-th node. The node attribute matrix can be expressed as $X \in \mathbb{R}^{nxd}$, where d stands for a graphcal data.

An example of architecture of the GCN is depicted in Fig. 4.4. It consisted of the input channel (X), the hidden layers, the calculated feature maps (Z) and the output layer (Y).

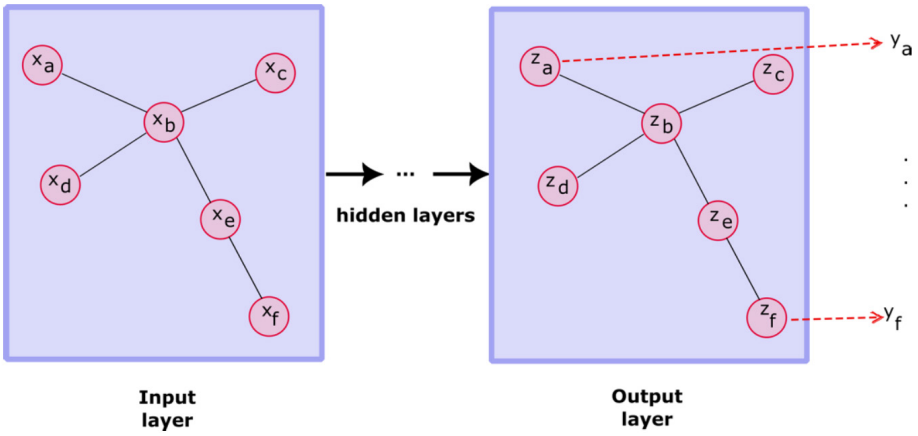


Fig. 4.4. GCN architecture [58]

As in GNN the feature maps are calculating and utilizing the convolutional operation. In GCN the convolutional layer was defined based on the Chebyshev recursive polynomial T_s (4.28) to reduce computational complexity [57]. In order to provide the eigenvalues in range [-1,1] , a filter was defined according to formula 4.29. In this configuration the convolution layer is defined as in formula 4.30 [57].

$$T_s(x) = 2xT_{s-1}(x) - T_{s-2}(x), \text{ for } T_0 = 1, T_1(x) = 1 \tag{4.28}$$

$$\widetilde{\lambda}_k = 2\frac{\lambda_k}{\lambda_{max}} \tag{4.29}$$

$$X^{p+1}(:,j) = \sigma\left(\sum_{i=1}^{d_p}\sum_{s=0}^{S-1}\left(\Theta_{i,j}^{p}\right)(s+1)T_s(\widetilde{L})X^p(:,i)\right), \quad \forall_j = 1,\ldots,d_{p+1} \tag{4.30}$$

where $X^{p+1}(:,j)$ stands for the *j-th* column of feature map obtained for the *p-th* layer, $\Theta_{i,j}^{p}$ denotes a parameter of *S*-dimensional vector for the *i-th* column of the input feature map $X^p(:,i)$.

In semi-supervised node classification in [58] the Chebyshev of the first-order polynomial was proposed. Additionally, the set

$\Theta_{i,j}(1) = -\Theta_{i,j}(2) = \Theta_{i,j}$ was defined. Moreover, the $\lambda_{max} = 2$ was used which yielded $-1 \leq \widetilde{\lambda}_k \leq 1, \forall_j = 1, ..., n$. As a result, the eigenvalues of the symmetrically normalized Laplacian are in range [0,2]. The convolutional layer (convolved signal matric) is then defined according to formula 4.31.

$$X^{p+1} = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}X^{p}\Theta^{p}\right) \tag{4.31}$$

where $\widetilde{A} = I + A$ stands for adjacency matrix A together with self-connections in the form of an identity matrix I, $\widetilde{D}$ for the diagonal degree of $\widetilde{A}$ matrix and finally Θ^p denotes the matrix of filter parameters of the dimension $d_{p+1} \times d_p$. This solution aggregates data from directly neighboring nodes, and also provides a gap between spectral and spatial methods. Because the convolution is computed in the Laplacian matrix, obtaining spectral data is not an easy task. For this purpose, classical CNN methods are applied, like propagation-based models.

The idea of GCN convolution is depicted in Fig. 4.5. The new value of the node is calculated based on the corresponding node value together with its all neighbors based on the given rule. For calculating the value of the node depicted as purple, 15, 9, 12, and 23 nodes are required. As a result of the convolution a new graph is created.

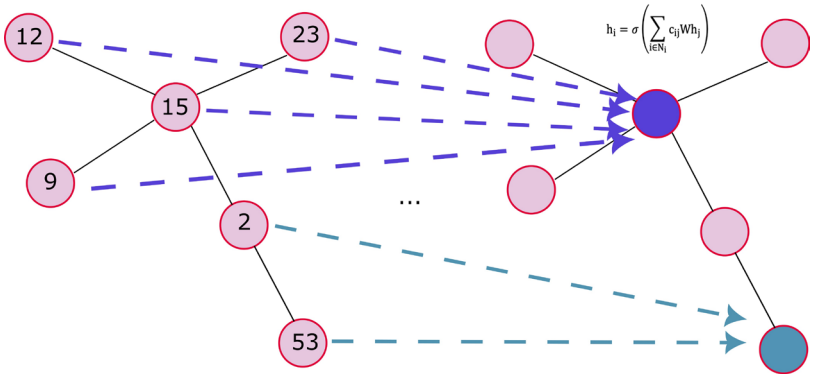


Fig. 4.5. An example of graph convolution

Many studies have considered new models for spatial graph convolutions including propagation and aggregation representation methods from neighboring nodes in the vertex domain. One example

of convolution for the *t-th* node at the *p-th* layer was proposed by the authors of [67]. It is given in formulas 4.32 and 4.33.

$$x^{p}{}_{N(t)} = X^{p}(t,:) + \sum_{v \in N(t)} X^{p}(v,:) \tag{4.32}$$

$$X^{p+1}(t,:) = \sigma\left(x^{p}_{N(t)} \Theta^{p}_{|N(t)|}\right) \tag{4.33}$$

where $\Theta^{p}_{|N(t)|}$ stands for weight matrix for nodes with the degree equal to $|N(t)|$ for *p-th* layer.

In order to take into consideration, data associated with edges of the graph, an edge-conditioned convolution operation was proposed in [68] based on a dynamic filter network. It is given in formula 4.34.

$$X^{p+1}(t,:) = \frac{1}{|N(t)|} \sum_{v \in N(t)} \Theta^{p}_{t,u} X^{p}(u,:) + b^{p} \tag{4.34}$$

where $\Theta^{p}_{t,u}$ stands for the edge weight matrix for *p-th* layer between *t-th* and *u-th* nodes and for the bias and filtering-generating network $F^{p} : \mathbb{R}^{s} \rightarrow \mathbb{R}^{d_{p+1} x d_{p}}$.

The feature maps for an example of GCN consisting of two layers for node classification are defined in the formula 4.35 [58].

$$Z = f(X, A) = Softmax\left(\hat{A} ReLu\left(\hat{A} X W^{0}\right) W^{1}\right) \tag{4.35}$$

where W^{0} stands for the weight matrix that is an input to the hidden layer, W^{1} for the weight matrix for the output layer. This is presented in Fig. 4.6.

In the case of graph classification, the result of a specific class (graph representation) should be indicated. In this approach a pooling mechanism is applied to create a coarser graph that is further reduced.

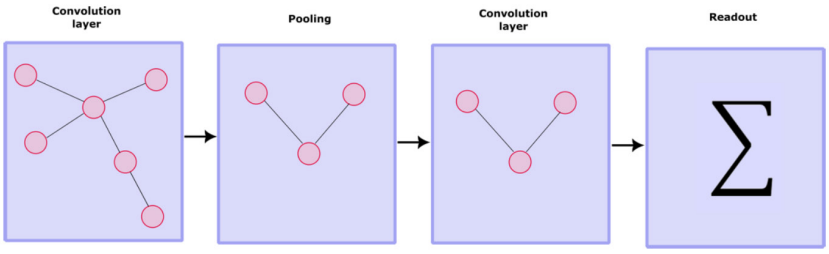


Fig. 4.6. GCN with pooling

After the convolutional and pooling layers, the extracted features are mapped into a vector using a global pooling layer (Fig. 4.7). At the end the fully-connected network is applied to obtained the classification tasks.

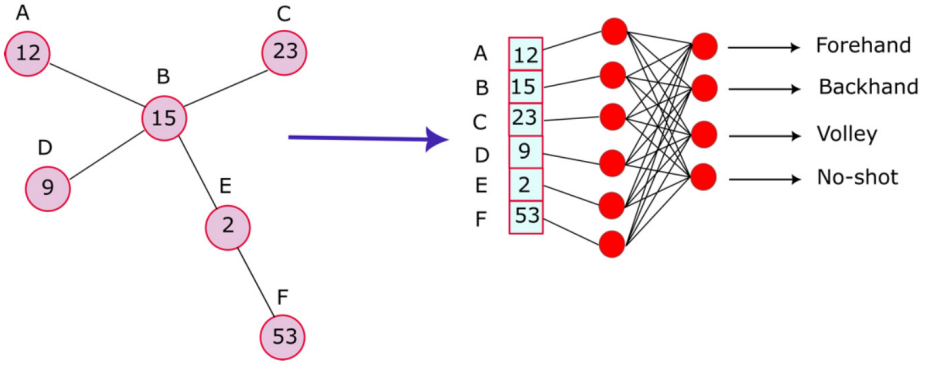


Fig. 4.7. Construction of a vector from a graph

GCNs have achieved impressive results in many real-life applications, such as object detection in images and videos, image classification, medical diagnoses, point cloud classification and segmentation, shape completion, text classification as well as prediction.

4.3. Spatial-Temporal Graph Convolutional Neural Networks

Graph Convolutional Neural Networks (GCNs) is a very fast developing field of study, so much so that a great variety of up-to-date models have recently been created. For the purpose of video or motion capture processing, where skeleton-based data can be obtained, a new model, the Spatial-Temporal Graph Convolutional Neural Network (ST-GCN), was developed. This AI method takes into account spatial dependency and temporal dependency at the same time [69]. This model is able to leverage latent patterns from spatial-temporal graph structures.

The ST-GCN was proposed for the first time in 2018 [70] for human action recognition purposes. The graph, reflecting the data, was created based on human topology. Two types of edges can be specified: spatial and temporal. In the case of the first character, each node in the graph corresponded to a proper joint from the human body, while bones connected to these joints were represented as edges. The second type corresponds to the same joints but in consecutive frames (time intervals). In this solution combining the multi layers resulted in leveraging and integrating both spatial and temporal information. The important advantage of this model is that the created model did not take into account hand-crafted feature-based methods, which allowed for a better representation of activities.

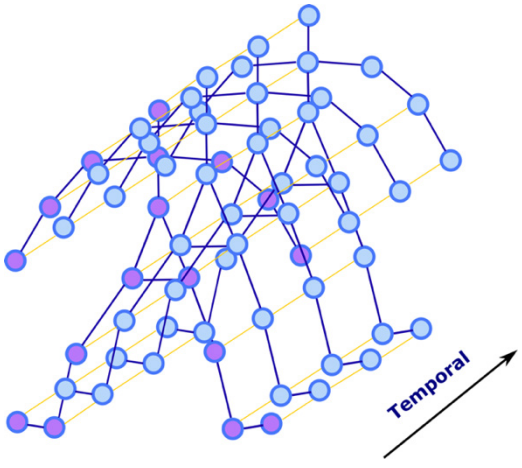


Fig. 4.8. The structure of spatial temporal graph, based on [70]

The node structure of the graph $G(V,E)$ in the ST-GCN is represented as [70]:

$$V = \{v_{ti} | t = 1, ..., T, i = 1, ...N\} \tag{4.36}$$

where v_{ti} denotes the coordinate vector of the *i-th* joint in the *t-th* frame. This kind of construction involves the information on all joints and their sequence in time. Each node, in the current frame, is connected to the corresponding nodes using edges according to the specificity of the data, and then each node is connected to its corresponding node in the consecutive frame. This structure is illustrated in Fig. 4.8. Blue lines indicate the edges between nodes in a graph in a current frame, while the yellow lines connect the corresponding nodes between frames. The edges between *i-th* and *j-th* nodes in a frame at time τ can be defined as [70]:

$$E(\tau) = \{v_{ti}v_{tj} | t = \tau, (i,j) \in H\} \tag{4.37}$$

In the ST-GCN the implementation of convolutional operation is similar to 4.31, where adjacency and identity matrices are utilized for connections between joints in a specific frame. The convolution operation may be written as [70]:

$$X^{j+1} = \sum_j \Lambda_j^{-\frac{1}{2}} A_j \Lambda_j^{-\frac{1}{2}} X^j W_j \tag{4.38}$$

where similarly $\Lambda_j^{ii} = \sum_k (A_j^{ik}) + \alpha$, with $\alpha = 0.001$, W_j stands for the weight matrix. The convolution operation is performed utilizing 2D convolution and then multiplies the results by the normalized adjacency matrix $\Lambda^{-\frac{1}{2}}(A+I)\Lambda^{-\frac{1}{2}}$ in the second dimension. It must be stressed that the adjacency matrix is disassembled into matrix A_j. Then $A + I = \sum_j A_j$.

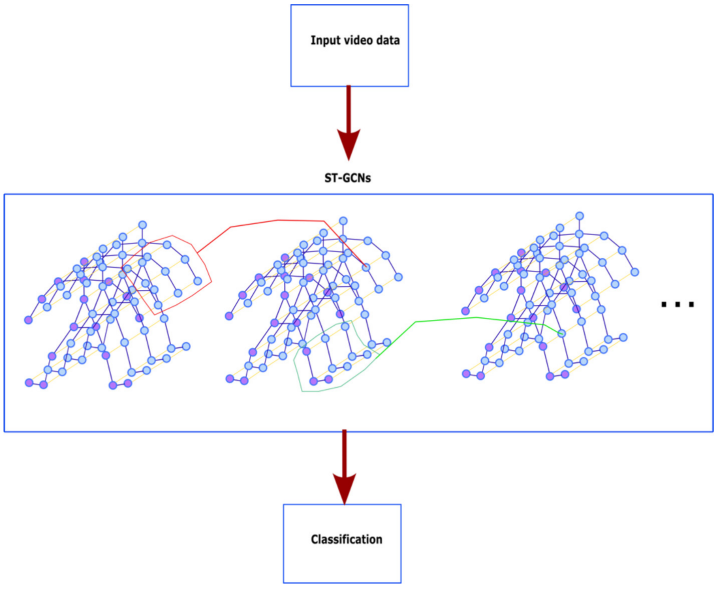


Fig. 4.9. The ST-GCN classifier [70]

The structure of the ST-GCN classifier is depicted in Fig. 4.9. Input to the classifier is a sequence of data, for example video, based on which the graph representing human topology is created. Multiple layers of graph convolutions are applied. As a result, high-level feature maps are obtained. The next step is a classification process utilizing Softmax for specified output classes.

The ST-GCN has found many applications, such as human action recognition [70] or traffic speed forecasting [71].

4.4. Attention Modules

CNN models are still being improved. Deeper and deeper CNNs have been developed such as VGGNet, ResNet or GoogLeNet for boosting spatial correlations [39][46][49]. Attention models have found many applications in CV [73][74][75]. They obtain information from a small visual area [73]. The selected region from which information is obtained is indicated by specific patterns of activities, textures, colors or shapes. There are three types of attention mechanism: soft, hard and those combining several attention modules into one framework [76].

4.4.1. Soft attention modules

The soft attention module is a fully differentiable deterministic mechanism that can be utilized in a neural network. The gradients are disseminated by the attention mechanism at the same time as they are propagated through the rest of the network [76]. The soft attention mechanism can be further divided into spatial attention and channel attention modules.

One of the examples of a spatial attention module is the residual attention network, dedicated to image classification [77]. It can be applied in any neural network structure. It utilized several attention modules, each of which consisted of a mask branch and trunk branch. The output of the whole attention module is specified as [77]:

$$H_{i,c}(x) = M_{i,c}(x) * T_{i,c}(x) \tag{4.39}$$

where i denotes spatial positions, $c \in \{1, ..., C\}$ is the channel index, $M_{i,c}$ is the mask branch, and $T_{i,c}$ – the trunk branch. The structure may be trained end to end.

The mask branch serves two functions: as a feature selection and for updating the filter during the backpropagation process. In this part of attention module, the gradient of the mask for the input features can be defined as [77]:

$$\frac{\partial M(x, \Theta) T(x, \phi)}{\eth \phi} = M(x, \Theta) \frac{\partial T(x, \phi)}{\eth \phi} \tag{4.40}$$

where Θ stands for the mask branch parameters, while ϕ for the trunk branch parameters.

In the Residual Attention Module, the trunk and mask branches were located next to each other. This resulted in a learning attention that was connected to its features.

Image classification usually deals with channel convolutional operations that lead to obtaining various channel information [76]. By adding the weights to the channel, the meaningful areas are indicated and the key information is extracted.

One of the channel-based attention models applied with a great success in classification purposes is Squeeze-and-Excitation Network

(SeNet) [78]. In this approach, after obtaining the feature maps using convolutional operation, the channel dependencies are obtained utilizing squeeze and excitation steps. Due to the fact that each filter operates in a specific area, the first step is to squeeze the global spatial information into the channel descriptor. Global average pooling is performed to determine statistics for individual channels. The statistic is generated by reducing the feature maps (U) through its spatial dimensions (H x W). These elements are calculated as [78]:

$$z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i,j) \tag{4.41}$$

In the next step, the dependencies between the channels are determined. It is stated that the function must be flexible (it must be able to learn non-linear interaction between channels), and it must also learn a relationship that is non-mutually-exclusive. In this way, multiple channels are highlighted. For this purpose, the gating mechanism and sigmoid activation functions are applied [78]:

$$s = F_{ex}(z, W) = \sigma(g(z, W)) = \sigma(W_2 \delta(W_1 z)) \tag{4.42}$$

where σ denotes the ReLu function, $W_1 \in \mathbb{R}^{\frac{C}{r} \times C}$ and $W_2 \in \mathbb{R}^{C \times \frac{C}{r}}$. Additionally, r is the reduction ratio parameter.

Finally, the output of the network is calculated as [78]:

$$\widetilde{x_c} = F_{scale}(u_c, s_c) = s_c u_c \tag{4.43}$$

where $X = [\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_C]$, F_{scale} denotes the channel-wise multiplication between s_c and the feature map.

4.4.2. Hard attention modules

In contrast to soft attention modules, hard ones perform the analysis of input images using subregions, rather than using the whole image [76][79][80][81]. These approaches utilize various stochastic models as well as reinforcement learning. The model of hard attentions is random.

In [80] a new model with a hard attention-based converter for a CNN was presented. In this solution pixel-based analysis was performed in order to create a new image. For each pixel one feature is selected after model training. Then the CNN was performed for the prediction based on the obtained image. Each pixel of an image is calculating using an attention mechanism [80]:

$$P_{i,j}(x) = \langle a^{i,j}, x \rangle := \sum_{k=1}^{d} a_k^{i,j} x_k \tag{4.44}$$

where $P \in \mathbb{R}^{H \times W}$, $i = 1,...,H$, $j = 1,...,W$, a stands for the attention weight for a pixel. The domain of $a_{i,j}$ is a d–1 dimensional simplex $\Delta^{d-1} = \{a \in \mathbb{R}^d | a_k \in [0,1], \sum_{k=1}^{d} a_k = 1\}$, $x \in \mathbb{R}^d$.

The representation z of the category is calculated as [80]:

$$z_k = \begin{cases} 1 & if \quad k = \underset{j \in (1,...,m)}{argmax} log(p_j) + g_j \\ 0 & otherwise \end{cases} \tag{4.45}$$

where $g_j \sim Gumbel(0,1)$ for $j = 1,...,m$ is computed as $g_j = -log(u_j)$ using uniform distribution and denotes the probabilities of the Gumbel-Max categorical distribution.

The Gumbel-Softmax is applied to compute the attention weight [80]:

$$a_k^{i,j} = \frac{exp\left(\left(log\left(p_k^{i,j}\right) + g_k^{i,j}\right)/\tau\right)}{\sum_{s=1}^{d} exp\left(\left(log\left(p_s^{i,j}\right) + g_s^{i,j}\right)/\tau\right)} \tag{4.46}$$

for $k = 1,...,d$, where τ denotes the temperature parameter controlling sample distribution. Controlling this parameter is necessary for the training process. As a result, a new image is generated, in which each pixel corresponds to one feature that was selected by the convector.

4.4.3. Convolution Block Attention Module

A third type of attention modules is the kind that combines both the spatial attention module and the channel one. In 2018 the Convolutional Block Attention Module (CBAM) was proposed as an enhancement for generating feature maps [82]. It was specified for feed forward CNNs. The main advantage of this solution was obtaining the information in which features are important as well as their locations. The obtained attention maps utilized convolution activations. As a result, the most principal features are extracted.

Due to the specificity of convolution operations in neural networks, which are performed on small areas of the analyzed image, and due to the fact that the spot extracts both cross-channel and spatial information, the proposed CBAM model consists of two parts: the channel attention and the spatial attention modules (Fig. 4.10). After the channel attention module, the intermediate features are generated [82]:

$$F' = M_c(F) \otimes F \tag{4.47}$$

where F is the input feature map, M_c denotes a one-dimensional channel attention map and F' is the intermediate feature map.

The channel attention module is presented in Fig. 4.11. This module indicates what is extremely important from the input image. In order to obtain spatial information both max-pooling and average-pooling operations are applied simultaneously. This combination greatly improves the network performance compared to their separate use [82]. As a results of these operations two spatial context descriptors are achieved, F^c_{max} and F^c_{avg}, respectively. The extracted max-pooled and average-pooled are input to the multilayer perceptron network. The outputs of the MLP network are summed. The equation representing the channel attention calculation is described as eq. 4.48 [82]:

$$\begin{aligned} M_c(F) &= \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) = \\ &= \sigma(W_1(W_0(F^c_{avg}) + W_1(W_0(F^c_{max}))) \end{aligned} \tag{4.48}$$

where σ denotes the sigmoid function, W_1 and W_2 are the MLP weight.

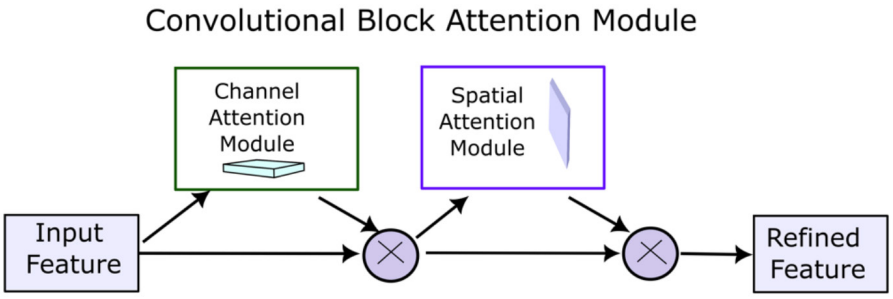


Fig. 4.10. Convolutional Block Attention Module [82]

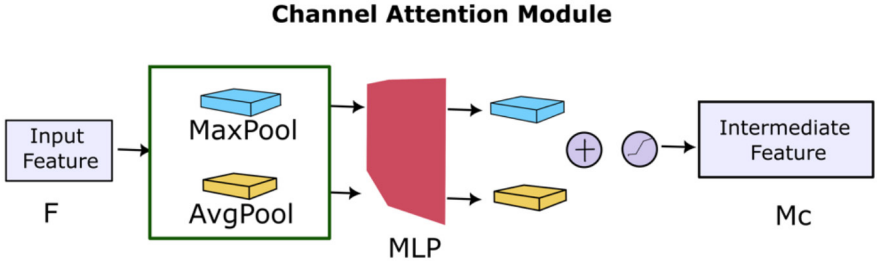


Fig. 4.11. Channel Attention Module [82]

After calculating intermediate features, a two-dimensional spatial attention map is generated (M_s), which is further used to obtained refined feature map (eq. 4.49).

$$F'' = M_s(F') \otimes F' \tag{4.49}$$

where F' is the intermediate feature map, M_s denotes a two-dimensional spatial attention map and F'' is the output feature map.

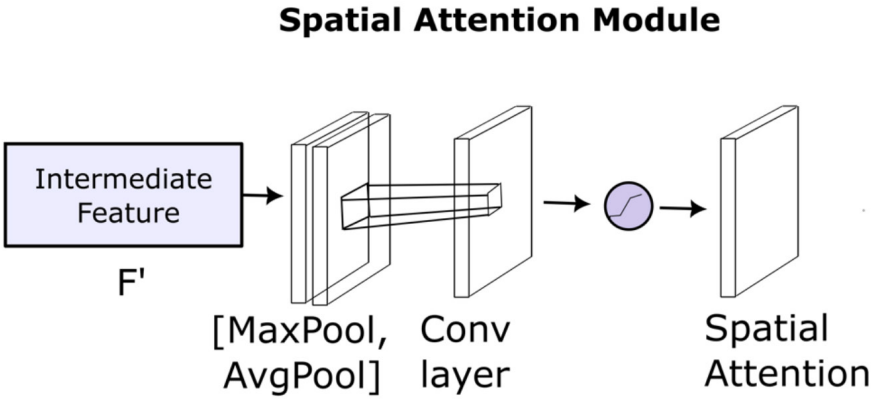


Fig. 4.12. Spatial Attention Module [82]

The spatial attention module indicates the locations which correspond to the channel attention (Fig. 4.12). In order to emphasize the dedicated regions, first average-pooling and then max-pooling are applied in order to obtain two feature descriptors: F^s_{avg} and F^s_{max}, respectively. The next step is to apply a standard convolutional layer in order to compute a two-dimensional spatial attention map (M_s). The spatial attention map is calculated as [82]:

$$M_c(F) = \sigma(f^{7x7}([AvgPool(F); MaxPool(F)])) = \\ = \sigma(f^{7x7}([F^s_{avg}; F^s_{max}])) \tag{4.50}$$

where σ denotes the sigmoid function while the f^{7x7} is a convolution operation with a 7x7 filter size.

It has been stated that the method and the order of arranging the modules influences classification efficiency [82]. The best results are obtained by placing the modules sequentially, with the channel attention being the first one.

Spatial-Temporal Graph Convolutional Networks for tennis movements recognition

CHAPTER 5

This chapter describes a new classifier, Spatial-Temporal Graph Convolutional Networks, developed for data representing the silhouette of a tennis player and a tennis racket, both in the form of images and 3D data. Moreover, the fuzzification of input data is applied in order to verify how it affects classification performance. Extensive results of the classification of tennis strokes in the form of accuracy, recall, precision, F1 score and confusion matrix are also presented.

HAR is a very challenging task in the CV field. ST-GCN is the first approach implemented in this study for tennis movement classification using data gathered via the Vicon motion capture system [7]. Two types of data have been taken into consideration, both of these obtained from the created 3DTennisDS [31]. First, images of a three-dimensional subject representing a tennis player with a tennis racket reflecting the proper tennis moves while performing actions. Second, three--dimensional data representing the tennis player model and a tennis racket. The 3D data were gathered utilizing retro-reflecting markers attached to the body and to the racket.

5.1. Image-based classification

5.1.1. Data preparation

From the post-processed motion capture data, reflecting tennis movements, like forehand, backhand, volley forehand, and volley backhand, videos were generated using the Vicon Nexus 2.0 software. Each recording of forehand and backhand moves presented the tennis player model

holding a tennis racket and performing strokes. Between each tennis move, the tennis player ran and avoided a bollard placed on the floor. He/she returned to the specified position from which the next move was performed. As for the video presenting volley strokes, the latter are performed alternately, e.g., the volley forehand preceded the volley backhand. The images were generated from videos using the ffmpeg tool with a sampling equal to 0.1 s. As a result, the following categories of tennis moves were collected: forehand preparation phase, forehand shot, backhand preparation phase, backhand shot, forehand volley, backhand volley, and no-stroke. Each move, except the last one, is defined as beginning at the moment the ball is hit and ending at the moment the stroke is finished. The last move does not correspond to tennis strokes. It was used in order to distinguish tennis movements from other activities presenting a tennis player holding a tennis racket.

Each image was further transformed to obtain a graph structure. First, the size of the images was reduced. Second, the background of an image was changed to white. From an image prepared like this the skeleton of a tennis player was indicated. The markers attached to the tennis player's body corresponded to nodes of the future graph. Moreover, some simplifications were performed for the tennis racket model and the head of the tennis model. Instead of seven markers from racket model, only two were considered as most important. The first marker was attached to the top of the racket head and the second to the end of the racket handle. These two nodes and an edge created between them were enough to indicate the position of the racket in reference to the tennis player. Another simplification was performed for the head and pelvis of the tennis player model. Four markers representing the head or pelvis were replaced by one marker. It was indicated using interpolation methods. Some markers from front and back, such as on the spine and on the chest, or two markers attached to wrists were interpolated to one. The markers from lower and upper limbs that were not attached to joints were also removed. Totally, 19 nodes were obtained for the player model, and the tennis racket model. The described operations did not affect classification quality. Additionally, these transformations allowed reduction in the number of input points and also acceleration of the calculations.

For the classification of tennis strokes, indicating the temporal features are very important factors. That is why a spatial temporal graph was created utilizing the transformed motion capture data. The structure was represented as a graph $G = (V, E)$, which consisted of N nodes and their change in time. The set of nodes were given as in eq. 4.36. An example of the input data are presented in Fig. 5.1. In order to indicate the moves, three frames were taken into consideration: preparation phase, the hit, and the swinging the racket.

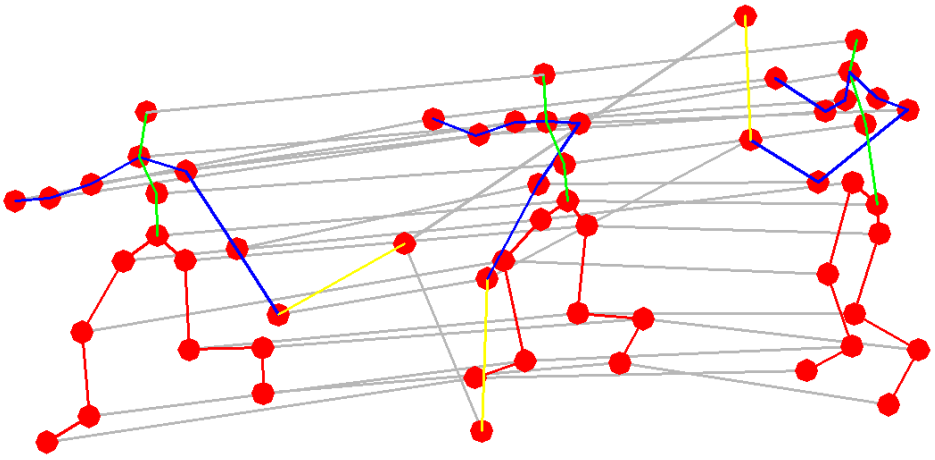


Fig. 5.1. A spatial temporal graph for forehand groundstroke performed by left-handed tennis player [7]

In Fig. 5.1, red dots represent the nodes corresponding to both the tennis player model and to the racket model, red lines denote the edges between the nodes of the lower limbs, green edges of the upper limbs, while yellow are the edges between two nodes of the tennis racket.

5.1.2. ST-GCN classifier

While performing tennis strokes, joints and corresponding parts of the body change position. These data are adequate for motion recognitions [83][84]. The representation of changes in model over time allowed one to create the hierarchical structure for classification purposes. Vectors containing joint coordinates of the graph nodes were input to the classifier. Coordinates were indicated starting from the left bottom corner of the image. Input data was transformed

utilizing spatial-temporal graph convolutional operations in multiple layers of the proposed neural network. The classification was performed in the last layer utilizing the standard Softmax function for the tennis moves classification. The neural network solution learning process was performed using a stochastic gradient descent algorithm.

The convolution operation for image utilizing $K \times K$ – size kernel may be defined as [7]:

$$f_{out} = \sum_{1}^{K} \sum_{1}^{K} f_{in} p(x) * \varpi \tag{5.1}$$

where p stands for sampling function $Z^2 \times Z^2 \to Z^2$, ϖ is the weight function $Z^2 \to R^2$, f_{in} and f_{out} are input and features at the spatial location x, respectively. Due to the fact that the data were presented in the form of a graph, the convolution operation was performed on the neighbor set [7]:

$$G(\nu_{ti}) = \{\nu_{tj} | d(\nu_{tj}, \nu_{ti}) \leq D\} \tag{5.2}$$

where $d(\nu_{tj}, \nu_{ti})$ stands for minimum length path between ν_{tj} node and ν_{ti} node. In this study the D was set to 1. The sampling function was defined as [7]:

$$p : \mathrm{G}(\nu_{ti}) \to V \tag{5.3}$$

The convolution operation (5.1) was defined as:

$$f_{out} = \sum_{\nu_{tj} \in G(\nu_{ti})} f_{in}(\nu_{tj}) * \varpi(\nu_{tj}) \tag{5.4}$$

According to (4.38) the convolution operation for a single frame may be defined as:

$$f_{out} = \Lambda^{-\frac{1}{2}}(A + I)\Lambda^{-\frac{1}{2}} f_{in} W \tag{5.5}$$

The input vector was defined as ($C,\ V,\ T$), where C denoted the dimension of each node, V the number of nodes, and T the number of time steps. In this study C was equal to 2, V was 19, and T was set

to 14. The convolution operation was implemented as a two-dimensional convolution, multiplied by a normalized adjacency matrix on the second dimension.

The proposed classifier for tennis movement recognition was presented in Fig. 5.2. It was composed with a ST-GCN classifier, a two-dimensional average pooling layer, a one-dimensional convolutional layer, an active features knowledge base and a Fully-connected layer with the Softmax function for tennis stroke recognition.

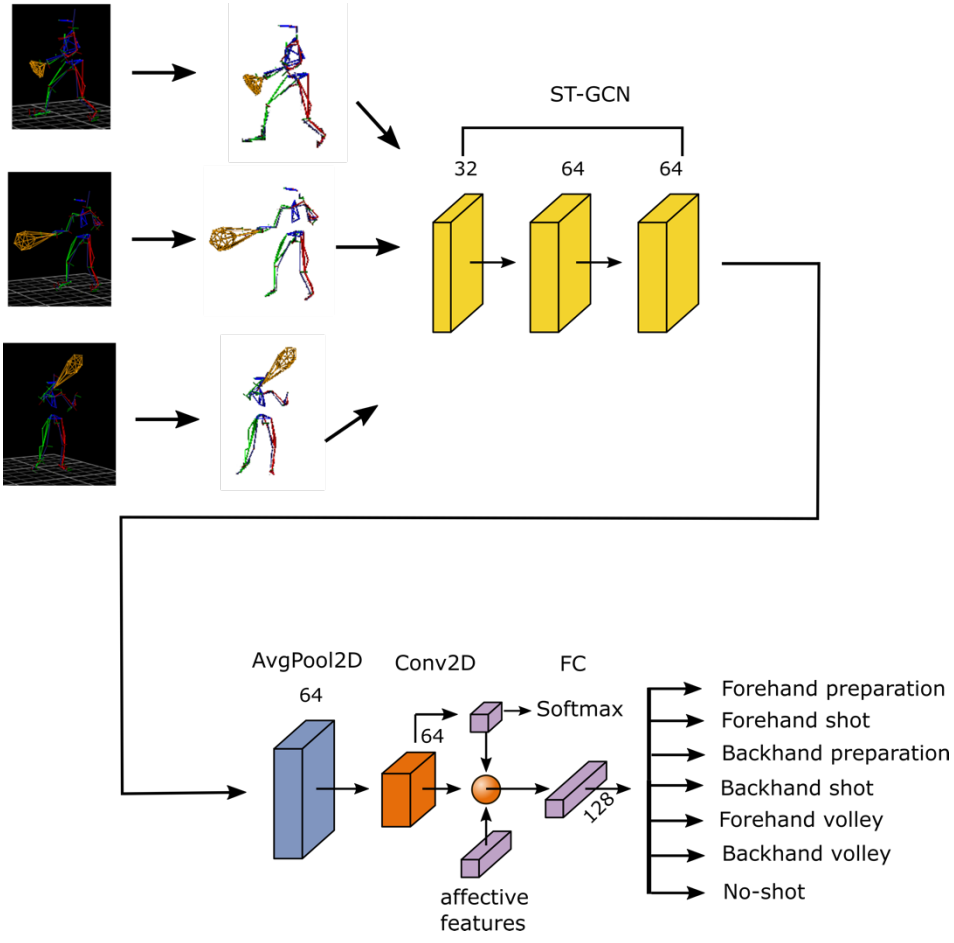


Fig. 5.2. The classifier for tennis movements recognition based on images utilizing the ST-GCN and affective features modules [7]

The ST-GCN part consisted of three spatial-temporal layers. They were made up of 32, 64, and 64 kernels, respectively. The output from the ST-GCN classifier was then passed to the average pooling layer,

where the average values of joints and temporal directions were calculated. The convolutional operation was performed on the obtained values using 64x64 kernels. The final layer was defined as a seven-dimensional vector, which corresponded to the specified types of tennis movements: forehand preparation phase, forehand shot, backhand preparation phase, backhand shot, forehand volley, backhand volley, and no-shot.

Each of the ST-GCN layers was followed by ReLu nonlinearity. After each layer, except the Fully-connected one, a BatchNorm layer was added.

The active feature part introduced two types of important information to the proposed classifier: movement parameters and posture data. The first one consisted of the acceleration and velocity of each joint. The second are the distance and angles between joints in two triangular areas for lower (left and right feet, and spine) and upper limb (left and right hand, and head). These types of data were combined with the data obtained from the ST-GCN classifier and finally passed to the Fully-connected layer.

5.1.3. Tennis strokes recognition

In total, 44,834 RGB 800x600 images were used in this study, separated into seven categories: forehand preparation phase (9,082), forehand hit together with racket swinging (9,716), backhand preparation phase (7,309), backhand hit together with racket swinging (7,709), forehand volley (6,676), backhand volley (6,055), and no-stroke (8,055). The whole dataset was divided into 80% (35,867) and 20% (8,967), for training and testing, respectively. Ten tests were performed.

Classifier performance was evaluated using the following measures: accuracy (5.6), precision (5.7), recall (5.8), and F1 score (5.9). In this monograph, all presented results were obtained based on testing data.

$$Accuracy = \frac{True_{positive} + True_{negative}}{True_{positive} + True_{negative} + False_{positive} + False_{negative}} \tag{5.6}$$

$$Precision = \frac{True_{positive}}{True_{positive} + False_{positive}} \tag{5.7}$$

$$Recall = \frac{True_{positive}}{True_{positive} + False_{negative}} \tag{5.8}$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{5.9}$$

Accuracy gives one knowledge of how the neural network model deals with all specified classes. This measure is defined as the ratio between the correct predictions to all (including true and false) predictions. It should be used, especially when all classes are equally important.

Precision defines the accuracy of the model for positive classification. It is calculated as the ration between positive samples that are classified correctly to the samples recognized as positive, both correctly and incorrectly. The high value of precision is achieved when a lot of positive classifications are obtained or when there are only a few incorrect positive classifications. Recall gives knowledge of how many positive samples are recognized by the model. If more positive samples are classified, a higher recall value is obtained.

F1 score is the harmonic mean of precision and recall measures. The higher the value of this score, the better the classifier performance is. When the precision and recall results are similar to each other, the higher the value of this score is obtained.

The obtained results for the proposed classifier are gathered in Tables 5.1–5.5.

Table 5.1. Obtained accuracy results for the **ST-GCN** classifier for image input for all classes

Mean	*Max*	*Min*	*±SD*
81.57%	84.70%	77.11%	5.52%

The results presented in Table 5.1 indicates that the mean obtained accuracy for all classes exceeds 81.50%, which is satisfactory. The achieved values are within the range 77.11–84.70%. It can be concluded that, based on the sequence of images showing subsequent parts of the same stroke, using the ST-GCN network is possible to classify tennis strokes and their phases with high accuracy. It should be emphasized that the low value of the average standard deviation determines the stability of the developed model.

Table 5.2. Obtained accuracy results for the **ST-GCN** classifier for image input for individual classes FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	80.12%	84.58%	77.16%	5.56%
FHS	79.35%	83.84%	77.73%	6.63%
BHP	84.52%	85.84%	77.15%	5.51%
BHS	84.70%	85.24%	77.11%	6.24%
VFH	84.67%	87.49%	77.18%	5.92%
VBH	84.69%	86.89%	77.47%	6.21%
NST	83.75%	84.62%	77.13%	6.52%

Accuracy results for individual classes are gathered in Table 5.2. The minimum results are higher than 77.12%, while the maximum values exceed 83%. The highest accuracy values are achieved for volley strokes, both forehand and backhand. Forehand strokes as well as the forehand preparation phase were recognized with the lower results. However, it should be stressed that the mean accuracy values are not lower than 79%. Also, no-stroke class concerning running with the tennis racket is recognized with high accuracy, exceeding the 83% mean value.

Table 5.3. Obtained precision results for the **ST-GCN** classifier for image input for individual classes FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

Class	Mean	Max	Min	±SD
FHP	82.34%	85.41%	78.55%	8.68%
FHS	81.32%	84.38%	77.61%	7.54%
BHP	87.48%	89.68%	84.82%	7.50%
BHS	86.78%	89.11%	83.62%	7.37%
VFH	88.67%	90.21%	86.77%	8.27%
VBH	88.38%	90.32%	85.92%	9.39%
NST	85.57%	88.01%	82.67%	8.41%

Precision results for individual classes are presented in Table 5.3. As in the case of accuracy, the highest precision values are achieved

for volley strokes. Slightly lower values are obtained for backhand strokes (both preparation phase and the hit with racket swinging). Classification for forehand strokes obtains the lowest precision values. It should be stressed that the lowest mean values are higher than 81%, which proves that the proposed ST-GCN model has the ability of positive classification.

Table 5.4. Obtained recall results for the **ST-GCN** classifier for image input for individual classes FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	92.67%	94.13%	89.95%	6.61%
FHS	86.11%	88.44%	83.17%	7.47%
BHP	84.21%	86.88%	81.16%	8.22%
BHS	84.51%	87.42%	81.24%	8.37%
VFH	83.21%	85.36%	80.60%	8.38%
VBH	84.75%	87.14%	81.58%	7.46%
NST	85.12%	87.51%	81.93%	7.82%

Recall results for individual classes are gathered in Table 5.4. The highest ability of the model to detect the positive tennis movements are obtained for forehand strokes. Recall values are also high for backhand strokes. About one percent less values are obtained for volley strokes. Also, movements that do not correspond to tennis strokes receive high recall results.

Table 5.5. Obtained F1 results for the **ST-GCN** classifier for image input for individual classes FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

Class	Mean	Max	Min	±SD
FHP	87.19%	89.56%	84.39%	8.62%
FHS	83.64%	86.36%	80.29%	8.43%
BHP	85.81%	88.26%	82.95%	6.33%
BHS	85.63%	88.26%	82.65%	8.34%
VFH	85.85%	87.72%	83.58%	11.32%
VBH	86.53%	88.70%	83.71%	9.41%
NST	85.35%	87.76%	82.33%	9.61%

The results of the F1 score for individual classes are gathered in Table 5.5. The obtained values indicate that the proposed model is well-balanced. This means that the analyzed classifier achieves both high precision and high recall results. The mean values are between 83.64% and 87.19%. The classifier obtains minimum F1 scores exceeding 80%. The highest mean F1 score, 87.19%, is obtained for the forehand preparation phase. The remaining classes have slightly worse results. However, they are above 83%. The lowest value reached is for the forehand stroke.

The confusion matrix is depicted in Fig. 5.3. In most cases, the ST-GCN model correctly classifies all tennis strokes and their phases. Incorrect classification occurred up to 3.52%, which proves that the model achieves high performance.

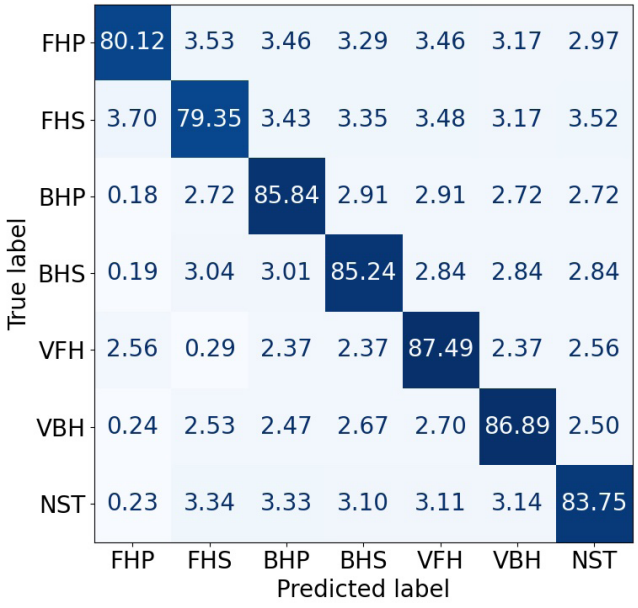


Fig. 5.3. Confusion matrix (in%) for tennis movement classification for the **ST-GCN** model FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No-Stroke

The forehand preparation phase was most often confused with the forehand stroke, the backhand preparation phase and the volley forehand. There is very little difference between the end of the preparation phase and the beginning of the execution phase. Additionally, the forehand volley contains similar elements to the forehand

preparation phase. This phase may have been incorrectly classified with the backhand preparation phase due to the fact that the dataset includes strokes performed by both right-handed and left-handed tennis players. The forehand stroke was most often confused with the preparatory phase of the same tennis move, as well as with the no-stroke. Racket positioning during the running may be confused with racket positioning during racket swinging.

The backhand preparation phase was misclassified with the backhand stroke and the volley forehand move. As in the case of the forehand, the end of the backhand preparation phase may be confused with the beginning of the backhand hit and with the volley forehand. The volley forehand was confused with the forehand preparation phase due to the racket positioning and similarly performed movements. Similarly, the volley backhand was most often confused with the backhand shot. Due to right-handed and left-handed tennis players, both types of volley shots were also confused with each other. Racket positioning while running (no-stroke) could be confused with the forehand stroke and the backhand preparation phase.

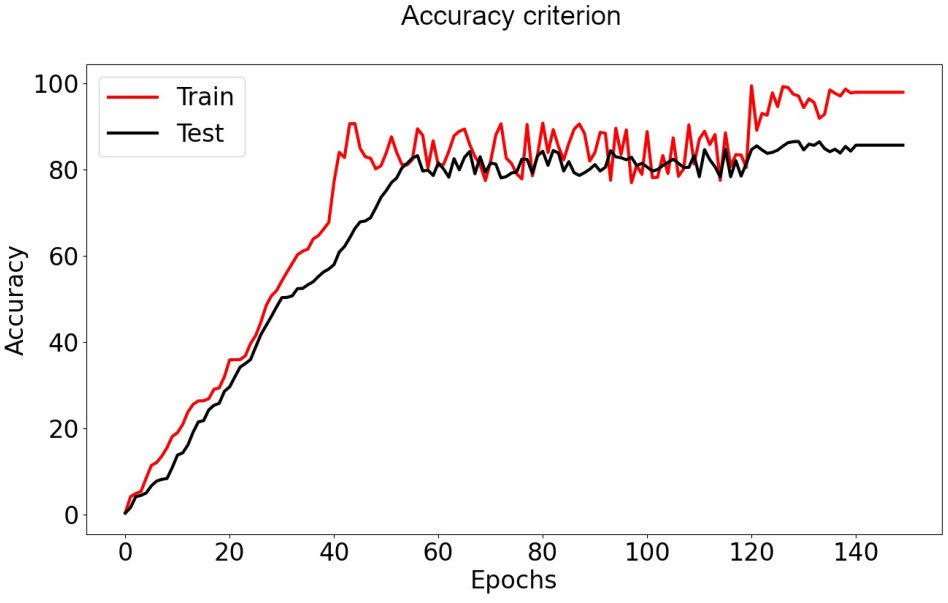


Fig. 5.4. Learning accuracy for the **ST-GCN** classifier for image input

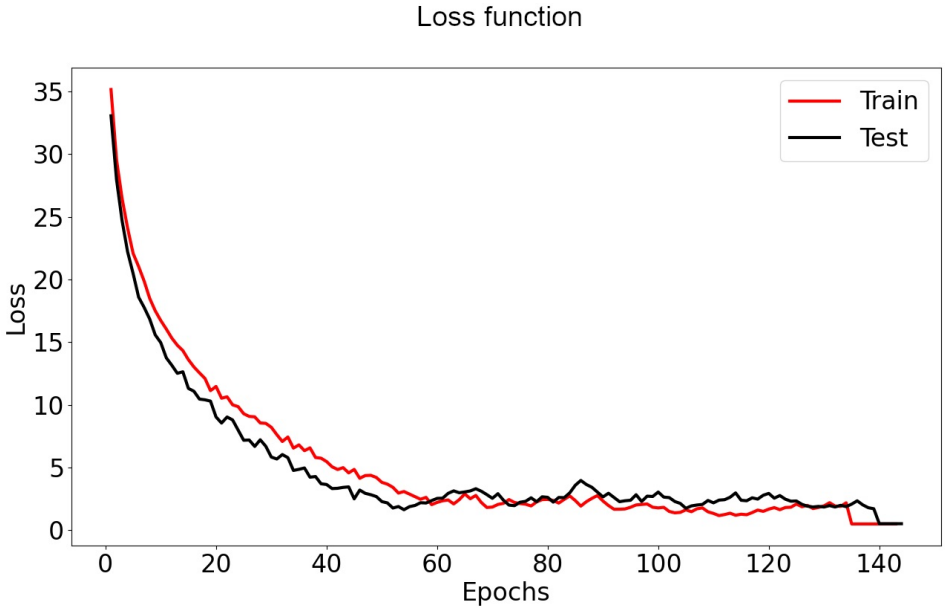


Fig. 5.5. Loss function for the **ST-GCN** classifier for image input

The process of learning the ST-GCN model is presented in Fig. 5.4 and 5.5. Accuracy values for training and testing datasets are depicted in Fig. 5.4. It can be noticed that reaching 140 epochs is enough to reach satisfactory accuracy. The loss function is presented in Fig. 5.5. It can be observed that the sum of the errors for each sample systematically decreases in subsequent epochs. After exceeding 140 epochs, the value of the loss function becomes the minimal value and the model reaches its highest performance.

5.2. 3D-based classification

In this subsection the recognition of tennis movements is presented using three-dimensional data. They are obtained from motion capture recordings, gathered for the purpose of this study. In order to conduct comprehensive studies, the classification of tennis strokes was carried out based on the tennis player model, the tennis player model with a tennis racket model and the tennis racket model itself.

5.2.1. Data preparation

Motion capture data consist of three-dimensional coordinates of thirty-nine markers attached to the body of the tennis player and seven markers attached to the tennis racket. These coordinates were read from c3d files using Python software with the c3d library. These data were divided into seven categories representing six tennis movements and one no-stroke. For the purpose of this study three types of spatial-temporal graphs were created, representing the whole tennis player model, the tennis player model holding a tennis racket and the same tennis racket model, respectively.

5.2.2. ST-GCN classifier

The structure of the ST-GCN classifier is the same as presented in Fig. 5.2. One difference is the type of input data, which is depicted in Fig. 5.6. From three various phases of tennis movements, the corresponding three-dimensional points were obtained for the defined frames. The input vector, defined as (C, V, T) was also modified. C was equal to 3. V was set to 39 in the case of input defined as a whole tennis player model, 46 in case of the tennis player model with a tennis racket model or 7 when only the tennis racket model was taken into consideration. T was set to 14. Another difference is that the convolution operation was implemented as a three-dimensional convolution, multiplied by the normalized adjacency matrix on the second dimension.

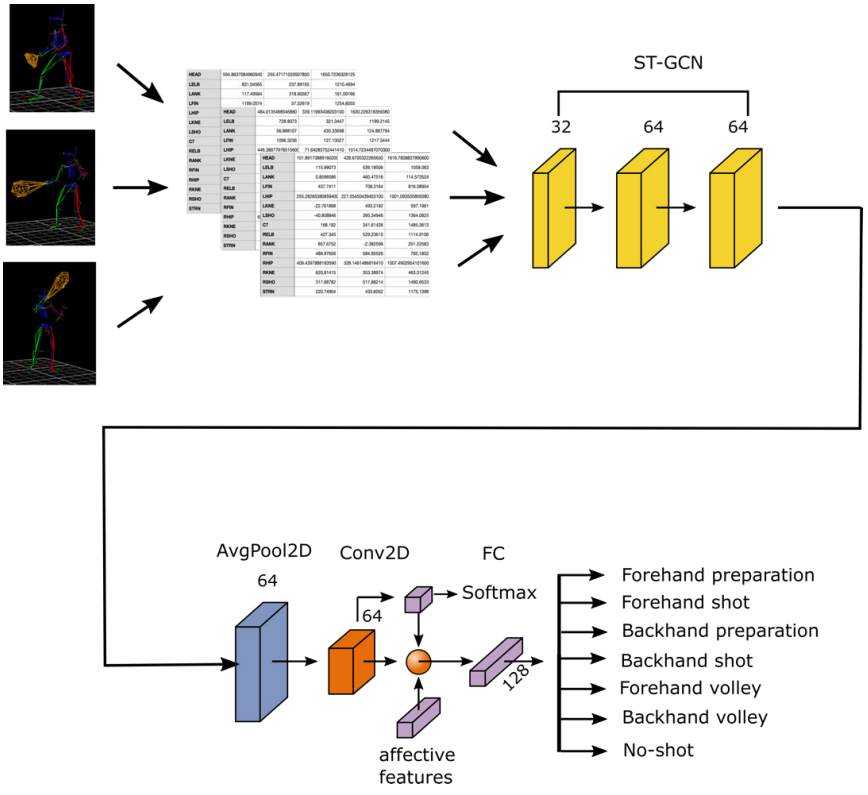


Fig. 5.6. The classifier for tennis movements recognition based on 3D data utilizing the **ST-GCN** and affective features modules, based on [7]

5.2.3. Tennis strokes recognition based on a tennis player model

In total, 1150 three-dimensional files were used in this study, separated into seven categories, as was described in 5.1.3. For the following tennis moves data were gathered: forehand preparation phase (178), forehand hit with racket swinging (178), backhand preparation phase (190), backhand hit with racket swinging (190), volley forehand (125), volley backhand (125), and no-stroke (164). Each data consists of 39 three-dimensional points that represent the movements of a tennis player model. In each file the parts consisting of 25 frames were gathered for the classification. The whole dataset was divided into 80% (920 files) and 20% (230 files) for training and testing, respectively. Ten tests were performed.

The classifier performance was evaluated using the following measures: accuracy (5.6), precision (5.7), recall (5.8), and F1 score (5.9).

The obtained results for the proposed classifier for the three--dimensional data of a tennis player model are gathered in Tables 5.6–5.10.

It can be observed that the ST-GCN classifier obtained better accuracy results for input data in the form of three-dimensional coordinates than in the form of images (Table 5.6). The mean accuracy for all classes exceeds 82.25%, which is satisfactory. The achieved values are within the range 78.10–88.70%. It can be concluded that, based on the sequence of three-dimensional data of a tennis player model, using the ST-GCN network, it is possible to classify tennis strokes and their phases with high accuracy. It should be emphasized that the low value of the average standard deviation determines the stability of the developed model.

Table 5.6. Obtained accuracy results for the **ST-GCN** classifier for 3D tennis player model input for all classes

Mean	Max	Min	±SD
82.26%	88.70%	78.10%	2.82%

Table 5.7. Obtained accuracy results for the **ST-GCN** classifier for 3D tennis player model input for individual classes. FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

Class	Mean	Max	Min	±SD
FHP	78.62%	87.70%	78.30%	2.76%
FHS	77.43%	88.70%	79.00%	2.93%
BHP	84.73%	87.90%	78.10%	3.21%
BHS	83.98%	87.86%	78.63%	3.11%
VFH	86.73%	87.62%	78.89%	2.91%
VBH	85.80%	87.30%	78.20%	2.58%
NST	83.48%	87.40%	78.30%	2.59%

Accuracy results for individual classes are presented in Table 5.7. The minimum results exceed 78%, while the maximum values exceed 87.29%. The best mean accuracy values are achieved for volley strokes, both forehand and backhand. Slightly worse results are obtained for backhand strokes, both the preparation phase, the hit with racket swinging and the no-stroke. The worst values are for both types of forehand strokes. It should be emphasized that the difference between the highest (volley forehand) and the lowest (forehand hit with racket

swinging) results is 9.3%. The mean standard deviation values do not exceed 3.21%, which proves that the neural network is stable.

Table 5.8. Obtained precision results for the **ST-GCN** classifier for 3D tennis player model input for individual classes: FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

Class	Mean	Max	Min	±SD
FHP	96.55%	97.67%	95.83%	5.45%
FHS	96.77%	97.82%	96.09%	5.09%
BHP	97.21%	98.18%	96.61%	4.68%
BHS	96.92%	97.94%	96.33%	4.91%
VFH	97.49%	98.16%	97.06%	3.29%
VBH	97.61%	98.32%	97.10%	3.78%
NST	97.10%	98.10%	96.51%	4.67%

The precision results for individual classes for 3D tennis player model input data are presented in Table 5.8. As in the case of accuracy, the highest precision values are achieved for volley strokes. Slightly lower values are obtained for backhand strokes and no-stroke. The classification for forehand strokes obtained the lowest values, however not lower than 96.55%. The obtained results indicate that the proposed ST-GCN model has the ability of positive classification based on 3D tennis player model input data.

Table 5.9. Obtained recall results for the **ST-GCN** classifier for 3D tennis player model input for individual classes: FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

Class	Mean	Max	Min	±SD
FHP	96.54%	97.67%	95.95%	5.29%
FHS	97.24%	98.05%	96.62%	4.41%
BHP	96.92%	97.99%	96.24%	5.24%
BHS	97.82%	98.49%	97.33%	3.79%
VFH	99.41%	99.64%	99.21%	1.24%
VBH	97.34%	98.25%	96.84%	4.16%
NST	94.72%	96.56%	93.58%	8.88%

Recall results for individual classes for 3D tennis player model input data are gathered in Table 5.9. The highest ability of the model to

detect the positive tennis movements is obtained for volley forehand. The rest classes are also received high values from 94.72% to 97.34%.

Table 5.10. Obtained F1 score results for the **ST-GCN** classifier for 3D tennis player model input for individual classes: FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

Class	Mean	Max	Min	±SD
FHP	96.54%	97.64%	95.89%	5.35%
FHS	97.01%	97.93%	96.35%	4.74%
BHP	97.06%	98.09%	96.42%	4.94%
BHS	97.37%	98.22%	96.83%	4.34%
VFH	98.44%	98.81%	98.17%	1.75%
VBH	97.47%	98.29%	96.97%	3.91%
NST	95.89%	97.32%	95.02%	6.81%

The results for the F1 score for individual classes for 3D silhouette input data are presented in Table 5.10. The obtained values indicate that the proposed model is very well-balanced. It means that the analyzed classifier achieves both high precision and high recall results. The mean values are between 95.89% and 98.44%. The classifier obtains minimum F1 scores exceeding 95%. The two highest mean F1 scores, 98.44% and 97.47%, are obtained for volley strokes. The remaining classes have slightly worse results but on the same level. The lowest value reached is for no-stroke movement.

The confusion matrix is depicted in Fig. 5.7. Misclassifications in the matrix have low value, up to 4.00%, as is the case with the classification of tennis movements based on images. It proves that the model achieves high performance.

The forehand preparation phase was most often confused with the first phase of the backhand stroke and with the forehand stroke itself. It was misclassified with the preparation phase of the same move. The backhand preparation phase was confused with the backhand stroke and the volley forehand, while the backhand stroke with the forehand move and the backhand preparation phase. Both types of volleys were confused with no-stroke. However, the latter one was misclassified with the backhand preparation phase in most cases.

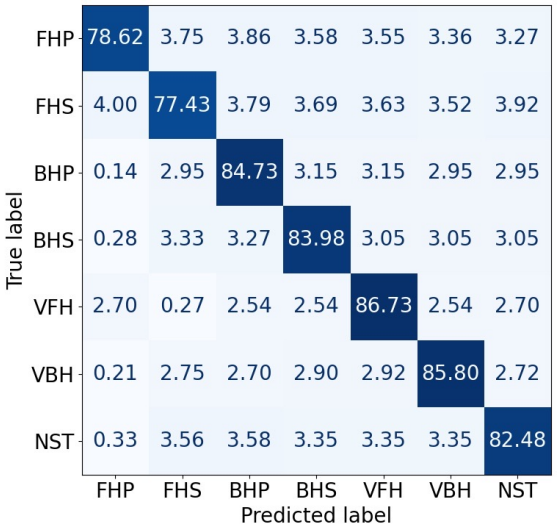


Fig. 5.7. Confusion matrix (in%) for tennis movement classification for the **ST-GCN** model for 3D tennis player model input data FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

All incorrect classifications result from the fact that the end of the preparation phase and the beginning of the actual forehand and backhand stroke do not have a fixed boundary. Some strokes are also misrecognized due to stroke data performed by both right – and left-handed tennis players.

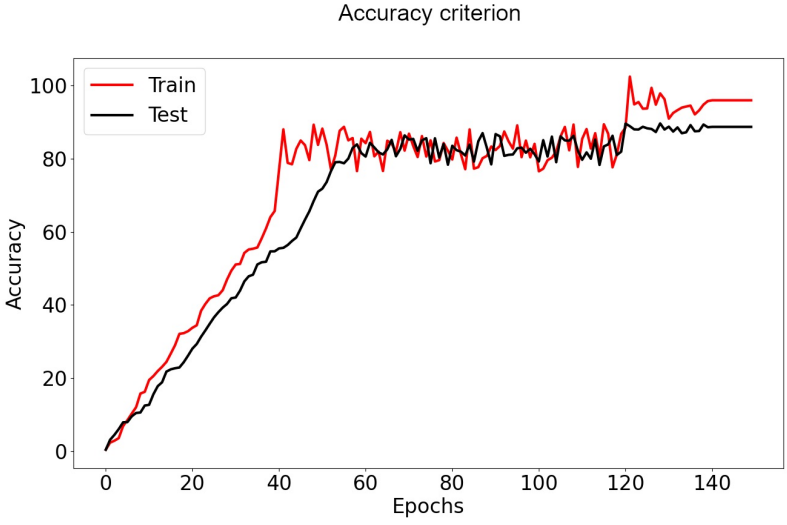


Fig. 5.8. Learning accuracy for the **ST-GCN** classifier for 3D tennis player model input data

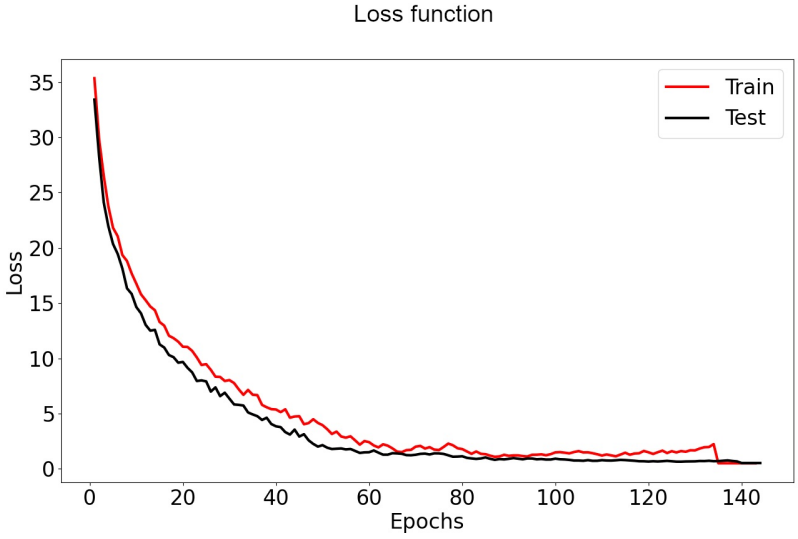


Fig. 5.9. Loss function for the **ST-GCN** classifier for 3D tennis player model input data

The learning parameters for the ST-GCN model for 3D silhouette input data are presented in Fig. 5.8 and 5.9. It can be noticed that reaching 140 epochs is enough to reach satisfactory accuracy and loss value. This model reaches high performance.

5.2.4. Tennis strokes recognition based on a tennis racket model

In total, 1150 three-dimensional data were used in this study, separated into seven categories, as was described in 5.1.3. Each data consists of 7 three-dimensional points that represent the movements of a tennis racket. The whole dataset was divided into 80% and 20% for training and testing, respectively. Ten tests were performed. Classifier performance was evaluated using the following measures: accuracy (5.6), precision (5.7), recall (5.8), and F1 score (5.9).

Table 5.11. Obtained accuracy results for the **ST-GCN** classifier for 3D racket model input for all classes

Mean	Max	Min	±SD
76.75%	81.91%	70.21%	13.45%

The obtained results for the proposed classifier for three-dimensional data of a tennis racket are gathered in Tables 5.1–5.15. It can be stated that the ST-GCN classifier obtained lower accuracy results for input data in the form of three-dimensional coordinates for a tennis racket model than for images or for 3D data of a tennis player model (Table 5.11). The mean accuracy is equal to 76.75%. The range of the obtained values, 70.21–81.91%, proves that the model performs well. The standard deviation is higher than for other input data and thus this model has less stability.

Table 5.12. Obtained accuracy results for the **ST-GCN** classifier for 3D tennis racket model input data for individual classes: FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	72.03%	81.19%	70.21%	9.11%
FHS	70.60%	81.75%	71.56%	5.99%
BHP	79.81%	81.91%	70.90%	8.08%
BHS	78.93%	81.58%	70.29%	5.28%
VFH	82.49%	81.77%	70.53%	9.85%
VBH	81.12%	81.59%	70.94%	11.75%
NST	77.11%	81.51%	73.31%	7.35%

Accuracy results for individual classes for 3D tennis input data are presented in Table 5.12. The minimum results exceed 70%, while the maximum values exceed 81%. The best mean accuracy values were achieved for volley strokes, both forehand and backhand, 82.49% and 81.12%, respectively. Slightly worse results were obtained for backhand strokes, both the preparation phase and the hit with racket swinging and for no-stroke. The worst values are for both types of forehand strokes, 72.03% and 72.03%. It should be emphasized that the difference between the highest and the lowest results is 11.89%. The mean standard deviation values do not exceed 11.75%, which prove that the neural network is stable.

Table 5.13. Obtained precision results for the **ST-GCN** classifier for 3D tennis racket model input data for individual classes: FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	77.16%	82.53%	71.48%	8.21%
FHS	75.89%	81.38%	70.11%	9.24%
BHP	83.62%	87.45%	79.48%	8.18%
BHS	82.91%	86.93%	78.62%	9.02%
VFH	85.70%	88.64%	82.18%	8.88%
VBH	84.79%	88.41%	80.77%	6.95%
NST	81.34%	85.56%	76.74%	9.07%

Precision results for individual classes for 3D tennis racket model input data are presented in Table 5.13. As in the case of accuracy, the highest precision values are achieved for volley strokes. Slightly lower values are obtained for backhand strokes and no-stroke. The classification for forehand strokes obtained the lowest values, however not lower than 77.16%. The obtained results, the mean ranging from 77.16% to 85.70%, indicate that the proposed ST-GCN model can be relied upon for positive classification based on 3D tennis racket model input data.

Table 5.14. Obtained recall results for the **ST-GCN** classifier for 3D tennis racket model input data for individual classes: FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	90.52%	92.56%	87.66%	7.22%
FHS	81.77%	86.02%	77.05%	11.89%
BHP	79.61%	84.47%	74.65%	9.37%
BHS	80.01%	84.54%	75.03%	10.13%
VFH	79.06%	83.17%	74.42%	9.82%
VBH	80.23%	84.76%	75.22%	8.11%
NST	80.70%	85.50%	75.76%	10.61%

Recall results for individual classes for 3D tennis input data are gathered in Table 5.14. The highest ability of the model to detect the positive tennis movements, 90.52%, is obtained for the forehand preparation phase. The remaining classes also received high values,

from 79.06.% to 81.77%. The obtained values, however, are lower than recall results for images and 3D tennis player model input data.

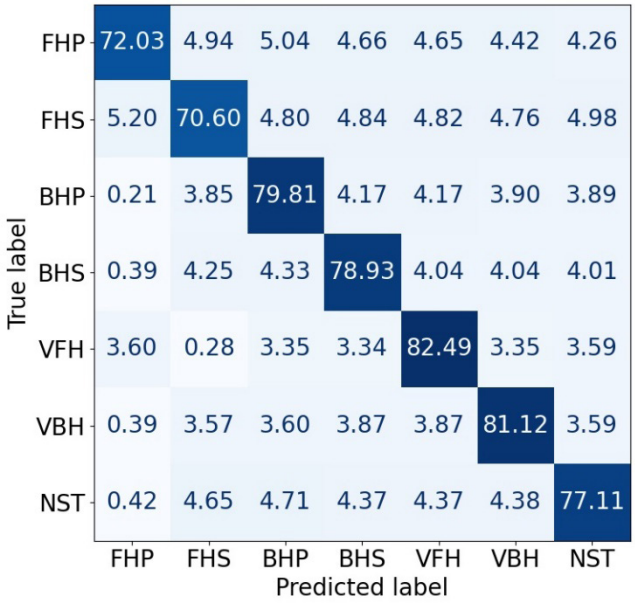


Fig. 5.10. Confusion matrix (in%) for tennis movement classification for the **ST-GCN** model for 3D tennis racket model input data: FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

Table 5.15. Obtained F1 score results for the **ST-GCN** classifier for 3D tennis racket model input data for individual classes: FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

Class	Mean	Max	Min	±SD
FHP	83.29%	87.12%	78.75%	9.53%
FHS	78.72%	83.60%	73.45%	8.86%
BHP	81.57%	85.93%	76.99%	8.09%
BHS	81.44%	85.70%	76.78%	9.15%
VFH	82.24%	85.82%	78.11%	7.22%
VBH	82.44%	86.54%	77.93%	8.30%
NST	81.02%	85.53%	76.25%	9.66%

Results for F1 score for individual classes for 3D tennis racket model input data are presented in Table 5.15. The obtained values indicate that the proposed model is well-balanced. It means that

the analyzed classifier achieves both high precision and high recall results. The mean values are between 81.02% and 83.29%. The classifier obtains minimum F1 scores exceeding 73%.

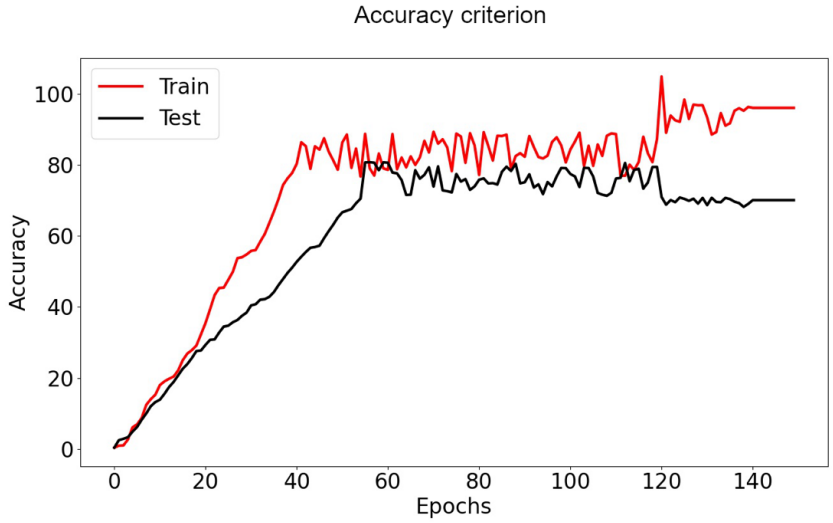


Fig. 5.11. Learning accuracy for the **ST-GCN** classifier for 3D tennis racket model input data

The three highest mean F1 scores, 83.29%, 82.44%, and 82.24%, are obtained for the forehand preparation phase and for both types of volley strokes. The remaining classes have slightly worse results but are on the same level. The lowest value reached is for forehand stroke. It must be emphasized that these results are lower than for image and 3D tennis player model input data.

The confusion matrix for 3D tennis racket input data is presented in Fig. 5.10. The highest level of misclassification is 5.20 when the forehand preparation phase was confused with the next phase of that tennis move. Similarly, to the ST-GCN classifier for other input data types, the forehand and the backhand stroke phases were confused, as well as the volley movements with each other. The same situation can be observed for the forehand, the backhand phases and the no-stroke moves. However, the percentages are small, but higher than in case of input data represented as images or 3D data of the tennis player model.

The learning parameters for the ST-GCN model for 3D tennis racket model input data are presented in Fig. 5.11 and 5.12. It can

be noticed that reaching 140 epochs is enough to reach satisfactory accuracy and loss value, similarly to as the ST-GCN for other input data types.

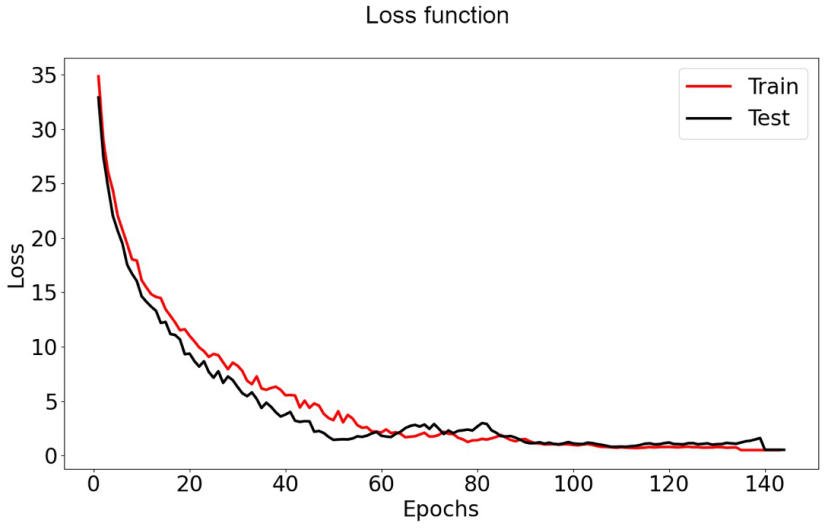


Fig. 5.12. Loss function for the **ST-GCN** classifier for 3D tennis racket model input data

Based on the data obtained, it can be concluded that the use of the tennis racket model alone as the input data in the ST-GCN classifier allows for lower recognition of tennis movements than in case of images or the 3D tennis player model input data.

5.2.5. Tennis strokes recognition based on a combined tennis player and racket model

In total, 1150 three-dimensional data were used in this study, separated into seven categories, as was described in 5.1.3. Each data consists of 39 three-dimensional points that represent the movements of a tennis player model and an additional 7 points for tennis racket moves. The whole dataset was divided into 80% and 20% for training and testing, respectively. Ten tests were performed. Classifier performance was evaluated using the following measures: accuracy (5.6), precision (5.7), recall (5.8), and F1 score (5.9).

Table 5.16. Obtained accuracy results for the **ST-GCN** classifier for 3D tennis player model with tennis racket model input for all classes

Mean	*Max*	*Min*	*±SD*
83.52%	88.91%	79.10%	9.82%

Obtained results for the proposed classifier for the three-dimensional data of a tennis player model with a tennis racket model are gathered in Tables 5.16–5.5.20.

The ST-GCN classifier obtained better accuracy results for input data in the form of three-dimensional coordinates for both a tennis player model and a racket model, which results are higher in comparison to other input types (see Tables 5.1, 5.6, 5.11, and 5.16). The values for accuracy range from 79.10% to 88.91%, which means that three-dimensional data with the combined models are the most adequate for classification purposes. Moreover, in this case, the model is stable as evidenced by the low value of the standard deviation.

Table 5.17. Obtained accuracy results for the **ST-GCN** classifier for 3D tennis player model and tennis racket model input data for individual classes: FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	81.36%	88.91%	79.10%	8.31%
FHS	80.29%	88.10%	80.30%	7.55%
BHP	86.71%	88.90%	79.11%	10.22%
BHS	86.10%	88.30%	79.45%	11. 38%
VFH	88.14%	88.81%	79.38%	9.42%
VBH	87.66%	88.55%	79.43%	7.14%
NST	84.78%	88.90%	79.15%	8.04%

Accuracy results for individual classes for 3D combined input data are presented in Table 5.17. These values are the highest compared to other types of input data, both for the mean (80.29–88.14%), the minimum (79.10–80.30%) and the maximum (88.10–88.91%) values. As in the case of the previous results (Table 5.2, 5.7), the highest values were obtained for volley strokes, and then for both phases of the backhand, no-stroke, and both phases of the forehand moves, respectively. The mean standard deviation values do not exceed 11.38%, which prove that the ST-GCN model is stable.

Table 5.18. Obtained precision results for the **ST-GCN** classifier for 3D tennis player model and tennis racket model input data for individual classes: FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	84.05%	89.07%	79.32%	8.44%
FHS	83.11%	88.24%	78.37%	9.42%
BHP	88.68 %	92.17%	85.32%	9.56%
BHS	88.07%	91.83%	84.74%	10.32%
VFH	89.60%	92.27%	87.13%	6.43%
VBH	89.39%	92.25%	86.44%	8.82%
NST	86.94%	90.87%	83.16%	11.21%

Precision results for individual classes for 3D combined input data are presented in Table 5.18. Precision results are also higher for this type of data than for other input types. The distribution of the received results, starting with the highest values, is the same as for accuracy. The obtained mean results, ranging from 83.11% to 89.60%, indicate that the proposed ST-GCN model is reliable for positive classification based on a 3D tennis player model with tennis racket model input data.

Table 5.19. Obtained Recall results for the **ST-GCN** classifier for 3D tennis player model and tennis racket model input data for individual classes: FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	93.36%	95.31%	91.64%	8.67%
FHS	87.45%	91.41%	83.82%	5.76%
BHP	85.73%	90.17%	81.50%	9.83%
BHS	86.09%	90.47%	82.10%	10.23%
VFH	84.45%	88.43%	81.06%	11.26%
VBH	86.05%	89.80%	82.14%	9.32%
NST	86.59%	90.70%	82.58%	8.27%

Recall results for individual classes for 3D combined input data are gathered in Table 5.19. The highest ability of the model to detect the positive tennis movements, 93.36%, is obtained for the forehand preparation phase, as in the case previously for other types of input. The remaining classes are also received high values, reaching from 84.45% to 87.45%. Obtained values are higher than recall results for images, 3D tennis player model, and 3D tennis racket model input data.

Table 5.20. Obtained F1 score results for the **ST-GCN** classifier for 3D tennis player model and tennis racket model input data for individual classes: FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	88.45%	92.08%	85.04%	8.75%
FHS	85.22%	89.78%	81.00%	7.83%
BHP	87.18%	91.16%	83.37%	9.46%
BHS	87.07%	91.14%	83.40%	9.78%
VFH	86.94%	90.28%	83.99%	8.36%
VBH	87.69%	91.00%	84.23%	7.59%
NST	86.76%	90.79%	82.87%	9.78%

Results for F1 score for individual classes for 3D silhouette with tennis racket input data are presented in Table 5.20. Obtained values indicate that the proposed model is well-balanced. The mean values were between 85.22% and 88.45%. The classifier obtained minimum F1 scores exceeding 80%. The results were higher than for image and 3D racket model input types, however lower than for 3D tennis player model data.

The four highest mean F1 scores, 88.45%, 87.69%, 87.18%, and 87.07% were obtained for the forehand preparation phase, the volley backhand, and both phases of the backhand move. The remaining classes have slightly worse results but on the same level. The lowest value reached is for the forehand stroke. It must be emphasized that these results are lower than for image and 3D tennis player model input data, however higher than for 3D tennis racket model input data.

The confusion matrix for 3D combined input data is presented in Fig. 5.13. Similarly, to earlier confusion matrices, the forehand and backhand phases were confused with each other. The backhand move is also misclassified with the forehand. Volley strokes are confused with each other and the backhand. No-stroke was most often misclassified with both volleys and the backhand stroke. It must be emphasized that confused values are very low, up to 3.49%.

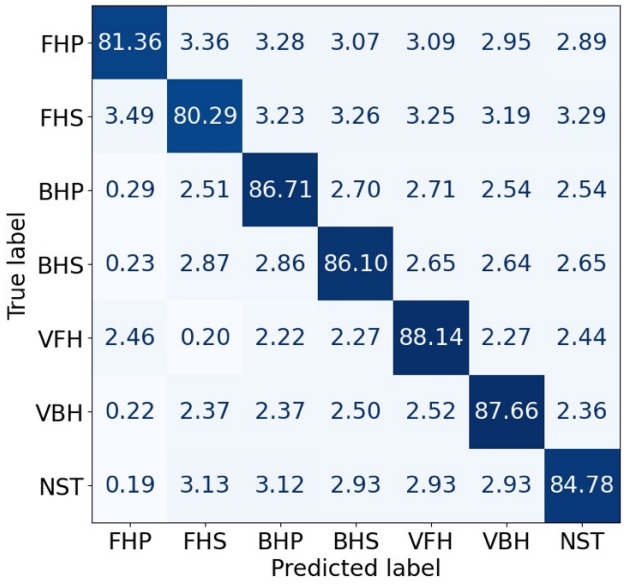


Fig. 5.13. Confusion matrix (in%) for tennis movement classification for the **ST-GCN** for 3D tennis player model and tennis racket model input data: FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

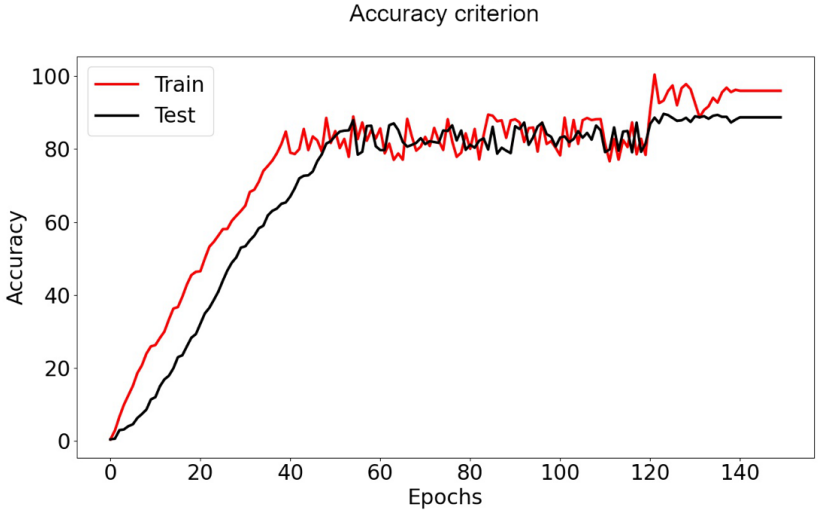


Fig. 5.14. Learning accuracy for the **ST-GCN** classifier for 3D tennis player model and tennis racket model input data

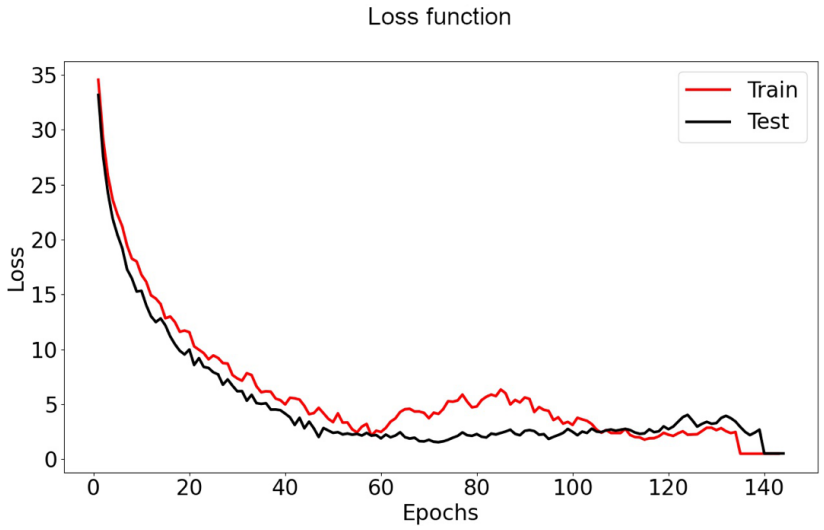


Fig. 5.15. Loss function for the **ST-GCN** classifier for 3D combined input data

The learning parameters for the ST-GCN model for the 3D tennis player model with tennis racket model input data are presented in Fig. 5.14, 5.15. It can be noticed that reaching 140 epochs is enough to reach satisfactory accuracy and loss value, similarly to the ST-GCN classifier for other input data types.

5.3. Classification utilizing fuzzification of input data

Analyzing the results obtained for the ST-GCN classifier in Sections 5.1 and 5.2, it can be noticed that one of the main problems is confusing the end of the preparation phase of backhand and forehand strokes with the actual hit together with racket swinging. Due to the lack of clearly defined boundaries, it was decided to verify how the fuzzification of the input data will affect the quality of tennis stroke classification.

Definition 5.1.

A fuzzy subset of a non-empty set f_s is a mapping $\sigma : f_s \longmapsto [0,1]$ which assigns to each element x in f_s a degree of membership, $0 \leq \sigma(x) \leq 1$ [85].

Definition 5.2.

A fuzzy relation on f_s is a fuzzy subset of $f_s \times f_s$. A fuzzy relation η_{fs} on f_s is a fuzzy relation on the fuzzy subset σ, *if* $\eta_{f_s}(x,y) \leq \sigma(x) \Lambda \sigma(y)$, for all x,y from f_s and Λ stands for the minimum. A fuzzy relation η_{fs} on f_s is said to be symmetric if $\eta_{fs}(x,\ y) = \eta_{fs}(y,x)$ for $x, y \in f_s$ [85].

Definition 5.3.

A fuzzy graph is a pair G: (σ, η_{fs}) where σ is a fuzzy subset of f_s ,η_{fs}, is a symmetric fuzzy relation on σ [85].

Definition 5.4.

(σ', η_{f_s}') is a fuzzy subgraph of (σ, η_{fs}) if $\sigma' \subseteq \sigma$ and $\eta_{fs}' \subseteq \eta_{fs}$ that is if $\sigma'(\eta_{fs}) \leq \sigma'(\eta_{fs})$ for every $\eta_{f_s} \in f_s$ and η_{fs}' (e)$\leq \eta_{fs}$ (e) [85].

The fuzzy approach is widely utilized in many fields of study, such as data analysis, decision making and classification purposes [86][87][88]. In order to model "uncertain" phenomena, fuzzy logic for the proposed the ST-GCN solution is applied [89]. This attitude allows for imitating the way in which the environment is perceived. Sharp boundaries between the analyzing sets are blurred. The process of fuzzification as well as defuzzification transforms sets from one state to another.

Membership functions are crucial parts of fuzzy models, because they define their behavior by assigning to an object a grade in the range between zero and one [90][91]. The higher the value that is obtained the higher the degree of membership is achieved. Determining the proper membership function influences the fuzzification process. Different types of membership functions are applied, such as: linear, exponential, piecewise linear, S-curve or hyperbolic [86]. Among them, the triangular and trapezoidal functions are the most often used [90]. They are both easy to apply and not computationally expensive. Although the trapezoidal function is more computationally complex compared to the triangular one, the obtained classification performance is higher [92]. The trapezoidal function is also stated to be more reliable than the triangular one [93]. Taking into the consideration the above-mentioned issues, in this study the following memberships functions are applied (Fig. 5.16):

$$\eta_{Rfs}(x)\begin{cases} 0, & x > d \\ \frac{d-x}{d-b}, & b \leq x \leq d \\ 1, & x < b \end{cases} \tag{5.10}$$

$$\eta_{Lfs}(x)\begin{cases} 0, & x < a \\ \frac{x-a}{c-a}, & a \leq x \leq c \\ 1, & x > c \end{cases} \tag{5.11}$$

where $a,\ b,\ c,\ d$ are parameters of the trapezoidal function and $a < b < c < d$.

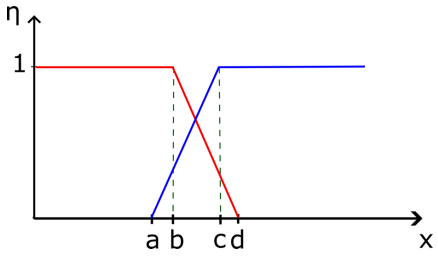


Fig. 5.16. Memberships functions

The input fuzzification was performed in the Matlab environment. This process was done for forehand, backhand and volley strokes separately. The following steps were included:

1. Prepare datasets for fuzzification.
2. Start Fuzzy Logic Designer.
3. Choose Mamdani Type-1 as a template for Fuzzy Inference System.
4. Define two trapezoidal membership functions. Generate the rules automatically.
5. Set parameters (a = 0.2, b = 0.4, c = 0.6, d = 0.8) and ranges for each function.
6. Loading the input data.
7. Define output.
8. Run the fuzzification process.
9. Export fuzzified data.

As the results, two new datasets were obtained for which the input data have been assigned. The fuzzification process does not influence the network structure. The number of elements in the data subsets could be modified.

The results for accuracy for all classes for the ST-GCN with fuzzification of input are gathered in Table 5.21. It must be emphasized that adding fuzzification improves the mean accuracy of the ST-GCN classifier, for image input by 3,06%, for 3D tennis player model by 2,88%, for 3D tennis racket model by 1,51%, and for 3D combined data by 2,39%. The highest values are obtained for input represented by the 3D tennis player model with tennis racket, and the lowest for 3D tennis racket model.

Table 5.21. Obtained accuracy results for the **ST-GCN** classifier for input fuzzification for all classes

Input type	Mean	Max	Min	±SD
image	84.63%	89.89%	79.12%	8.23%
3D silhouette	85.14%	94.45%	78.18%	6.51%
3D racket	78.26%	82.31%	72.79%	9.02%
3D silhouette with racket	85.91%	94.38%	77.52%	6.93%

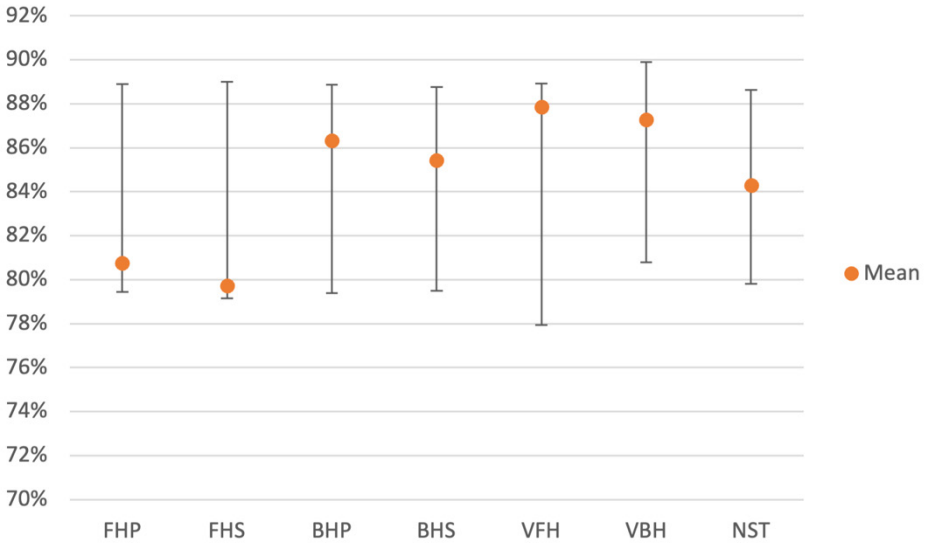


Fig. 5.17. Obtained Accuracy results for the **ST-GCN** classifier for image input fuzzification for successive classes

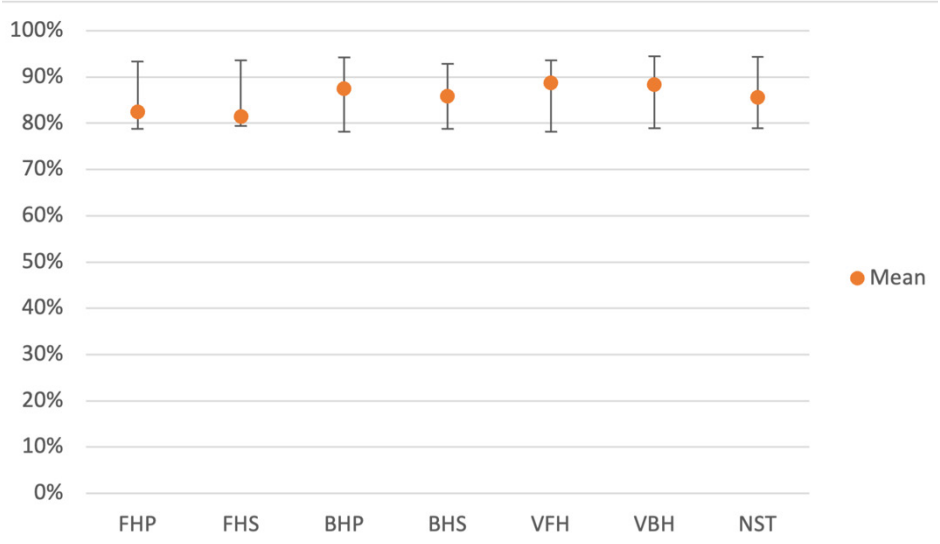


Fig. 5.18. Obtained Accuracy results for the **ST-GCN** classifier for 3D tennis player model input fuzzification for successive classes

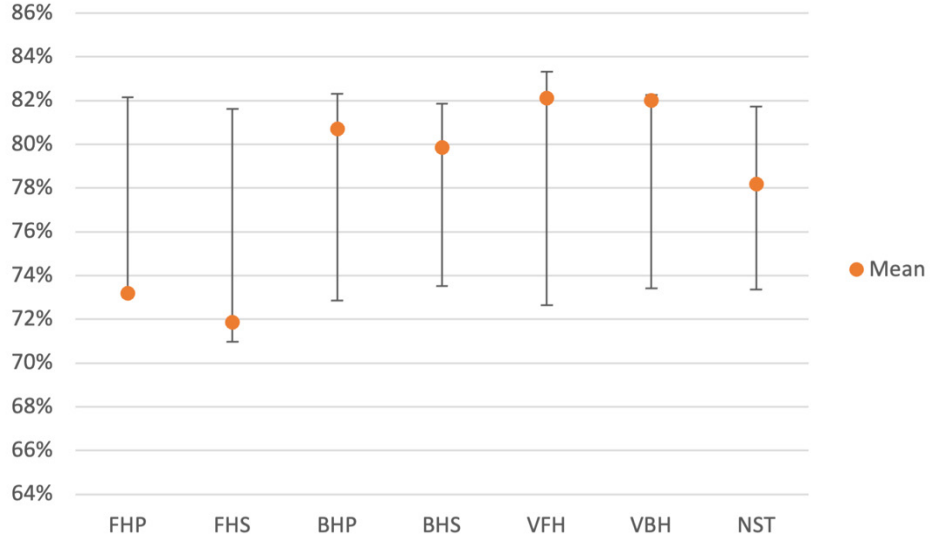


Fig. 5.19. Obtained Accuracy results for the **ST-GCN** classifier for 3D tennis racket model input fuzzification for successive classes

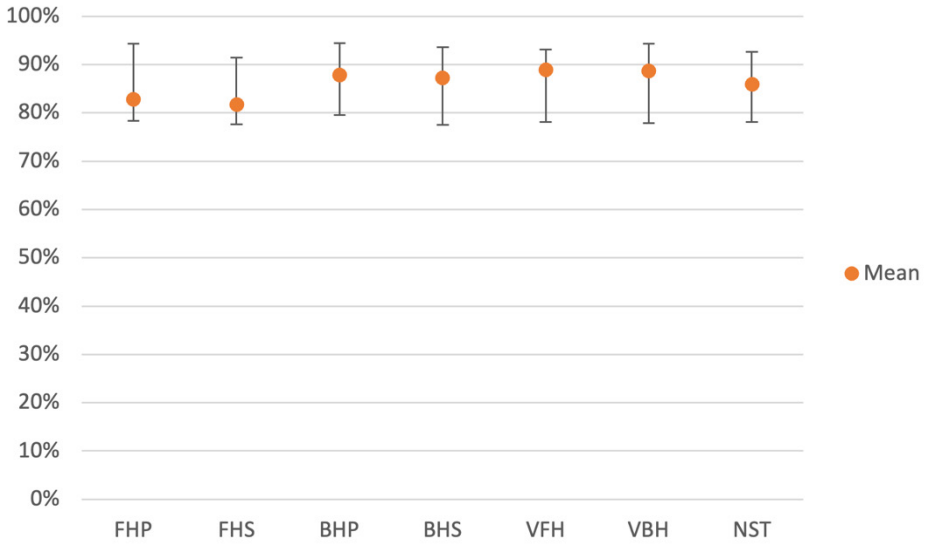


Fig. 5.20. Obtained Accuracy results for the **ST-GCN** classifier for 3D tennis player and racket model input fuzzification for successive classes

The results for accuracy for all tennis moves (classes) for the ST-GCN with fuzzification of various input types are presented in Fig. 5.17–5.20. For each type of input the applying the fuzzification caused an improvement of the ST-GCN accuracy. For the input represented in the form of image, accuracy was improved, ranging from 0.2% to 0.64%. Tennis movements in which classification was improved the most were the forehand preparation phase, no-stroke and the backhand preparation phase, by 0.64%, 0.54% and 0.49%, respectively. Volleys were improved by approximately 0.4%. The lowest improvement was registered for the forehand and the backhand hits with racket swinging.

Regarding 3D point inputs for the tennis player model, the improvement was more significant, going from 1.89% to 4.05%. In this case, the largest increase in accuracy was recorded for both phases of the forehand move, by 4.05% and 3.79% for the preparation phase and the hit together with racket swinging phase, respectively. Accuracy of the preparation phase of the backhand stroke was improved by 2.75%, volley backhand moves by 2.51%, and no-stroke by 2.14%. The lowest improvement was achieved for the backhand and volley forehand strokes by 1.83% and 1.99%, respectively.

Taking into account input data from three-dimensional points of a tennis racket, accuracy improvements were obtained from 0.82% for the volley forehand to 1.28% for the forehand hit with racket swinging. Moreover, an improvement of over 1% was obtained for the forehand preparation phase – 1.17% and for no-stroke – 1.08%. The lowest improvement values were registered for both backhand phases by 0.93%, 0.9%, respectively.

In the case of three-dimensional data representing a tennis player model with a tennis racket, accuracy improvements were achieved from 0.75% to 1.46%. The forehand hit and its preparation phase reached the highest improvements, of 1.46% and 1.43%, respectively. Improvements at the same level were obtained for the backhand hit, the backhand preparation phase, and no-stroke, by 1.14%, 1,08%, and 1.10%, respectively.

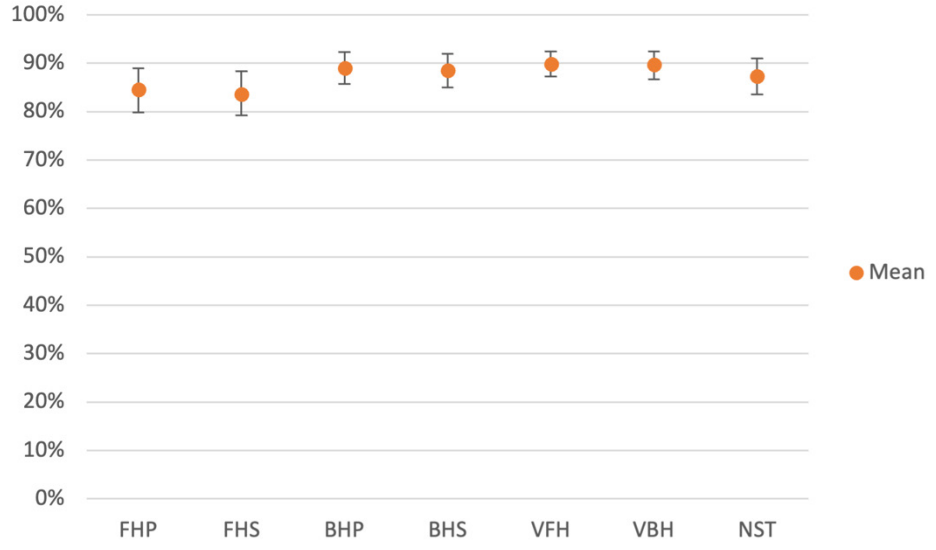


Fig. 5.21. Obtained Precision results for the **ST-GCN** classifier for image input fuzzification for successive classes

Results for precision for all classes for the ST-GCN with fuzzification of various input types are presented in Fig. 5.21, 5.24. These obtained results show that the ability of positive classification for the proposed neural network was improved for the following input type: image – ranging from 1.16% to 2.30%, 3D tennis racket – from 0.95% to 1.54%, and 3D tennis player model with tennis

racket – from 1.41% to 2.63%. In each case, the forehand move was the best improved. However, the precision results were worsened for 3D tennis player model input, up to 12.04%.

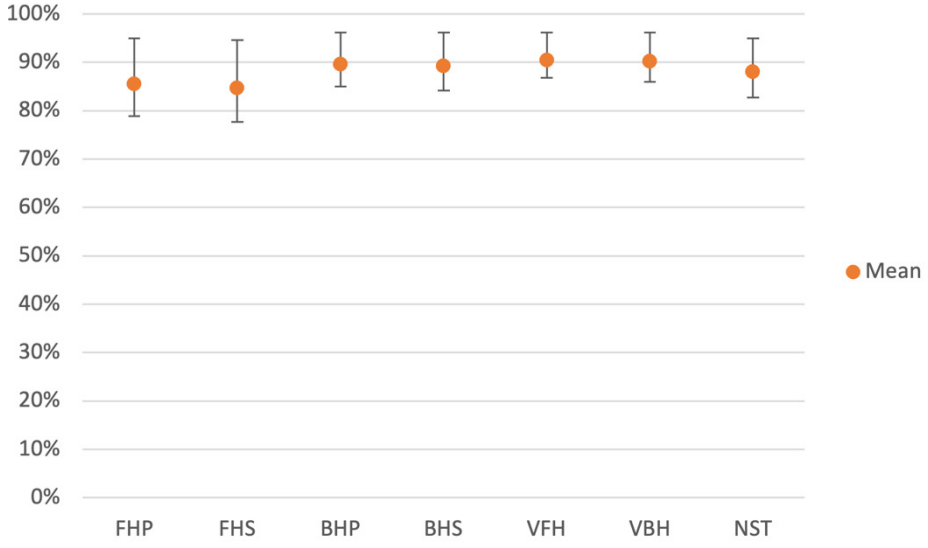


Fig. 5.22. Obtained Precision results for the **ST-GCN** classifier for 3D tennis player model input fuzzification for successive classes

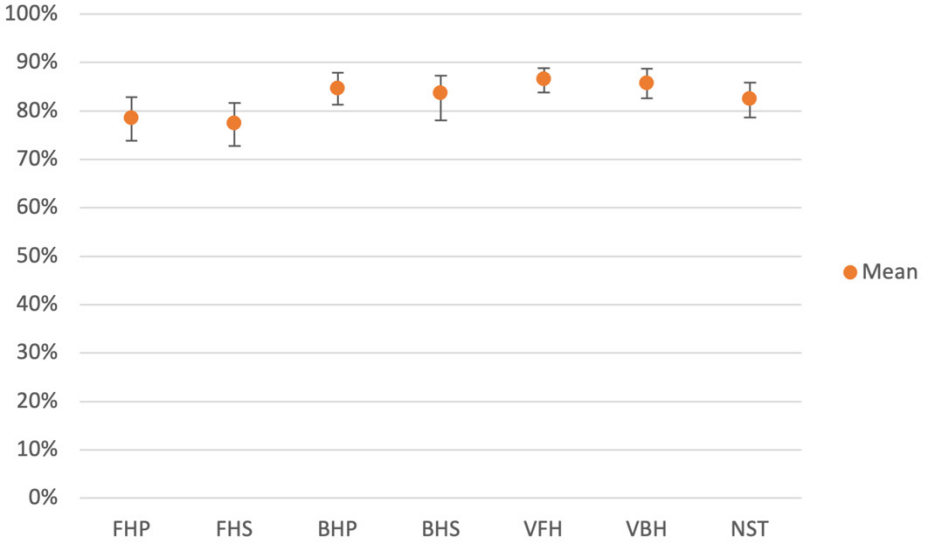


Fig. 5.23. Obtained Precision results for the **ST-GCN** classifier for 3D tennis racket model input fuzzification for successive classes

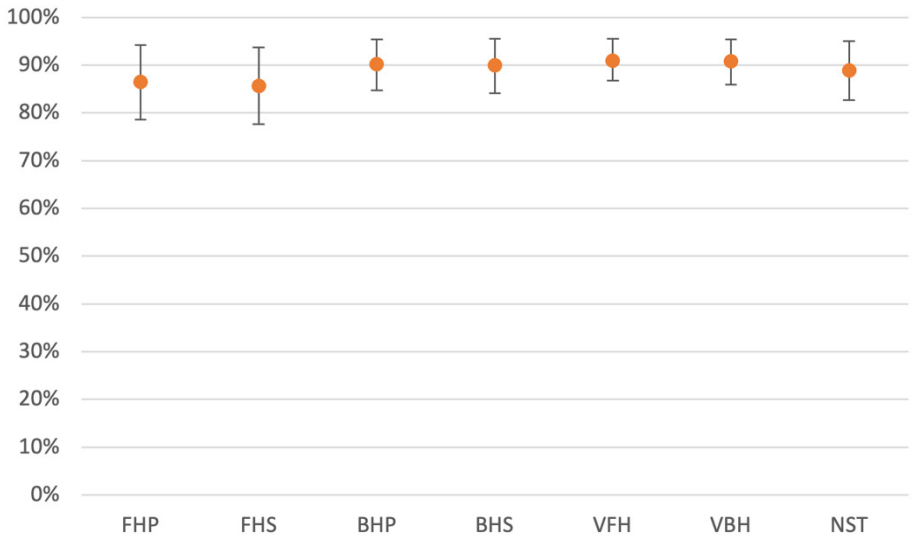


Fig. 5.24. Obtained Precision results for the **ST-GCN** classifier 3D tennis player and racket model input fuzzification for successive classes

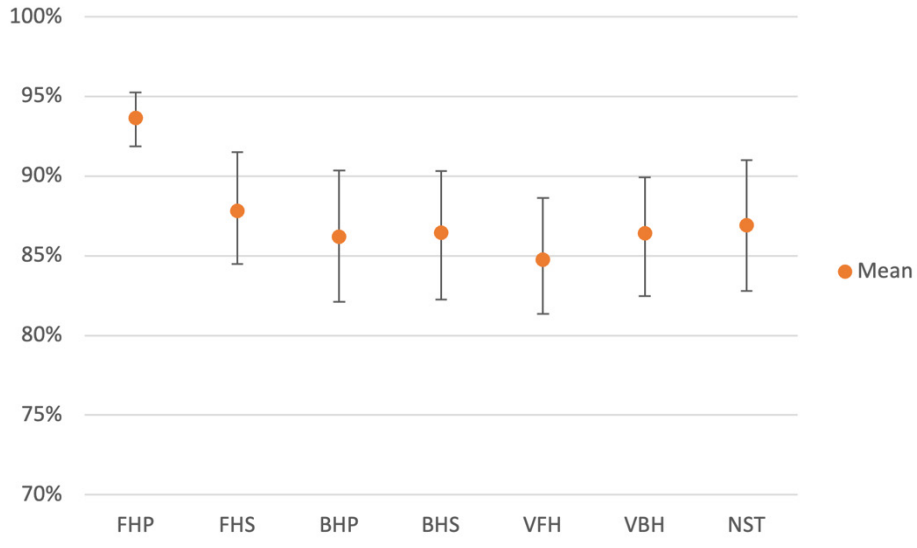


Fig. 5.25. Obtained Recall results for the **ST-GCN** classifier for image input fuzzification for successive classes

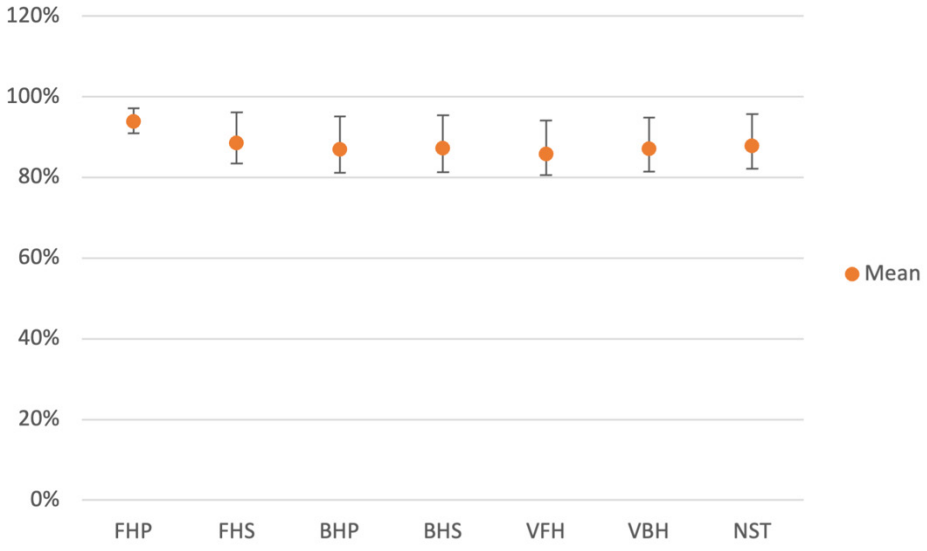


Fig. 5.26. Obtained Recall results for the **ST-GCN** classifier for 3D tennis player model input fuzzification for successive classes

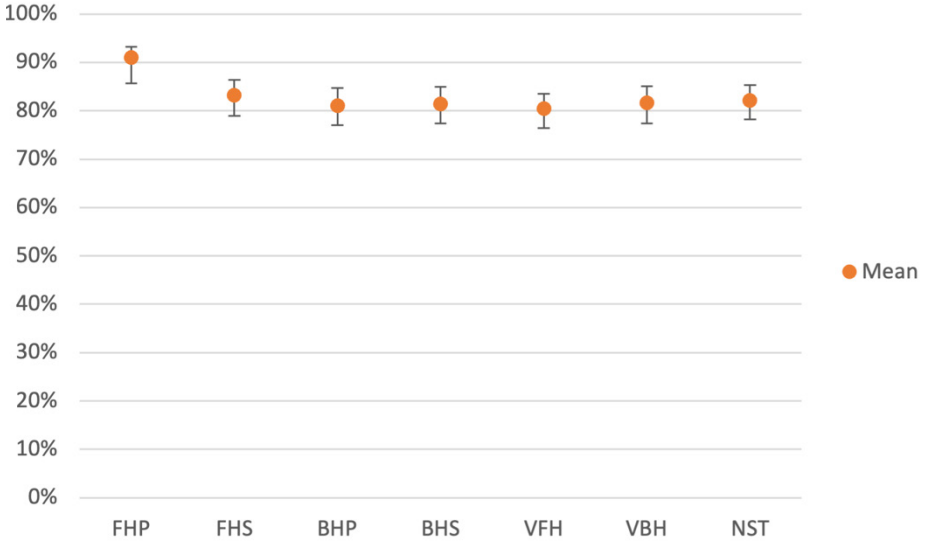


Fig. 5.27. Obtained Recall results for the **ST-GCN** classifier for 3D tennis racket model input fuzzification for successive classes

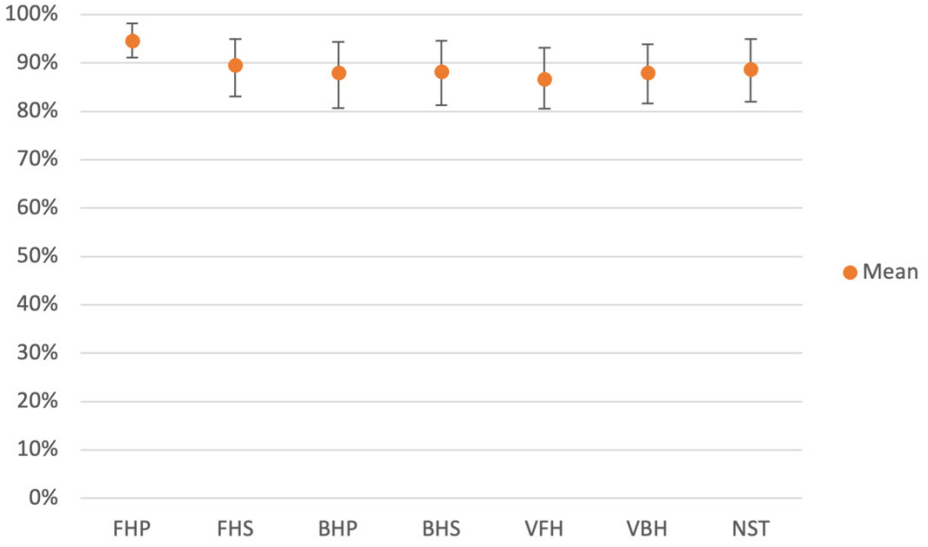


Fig. 5.28. Obtained Recall results for the **ST-GCN** classifier for 3D tennis player and racket model input fuzzification for successive classes

Results for recall for all classes for the ST-GCN with fuzzification of various input types are presented in Fig. 5.25–5.28. The recall results were improved for the following input data types: images, ranging from 0.99% to 1.99%, 3D tennis racket model ranging from 0.41% to 1.40%, and 3D tennis player model with a tennis racket – from 1.09% to 2.18%. The highest improvements were obtained for backhand movements. As in the case of precision results, the fuzzification process did not improve the classification for 3D tennis player model data.

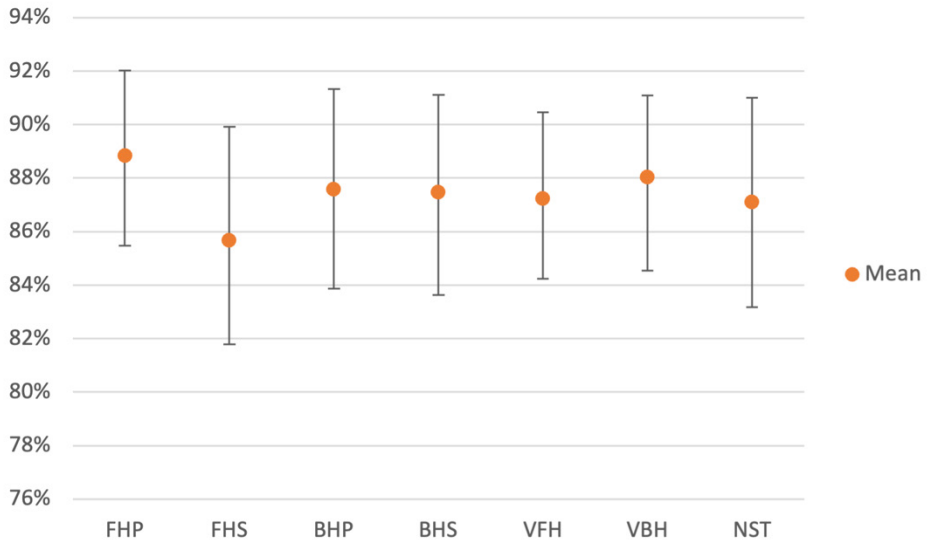


Fig. 5.29. Obtained F1 score results for the **ST-GCN** classifier for image input fuzzification for successive classes

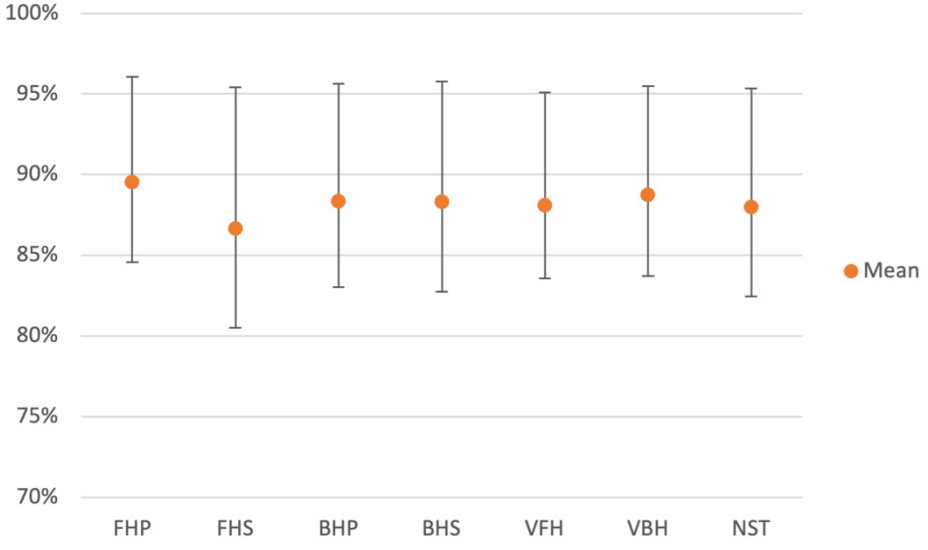


Fig. 5.30. Obtained F1 score results for the **ST-GCN** classifier for 3D tennis player model input fuzzification for successive classes

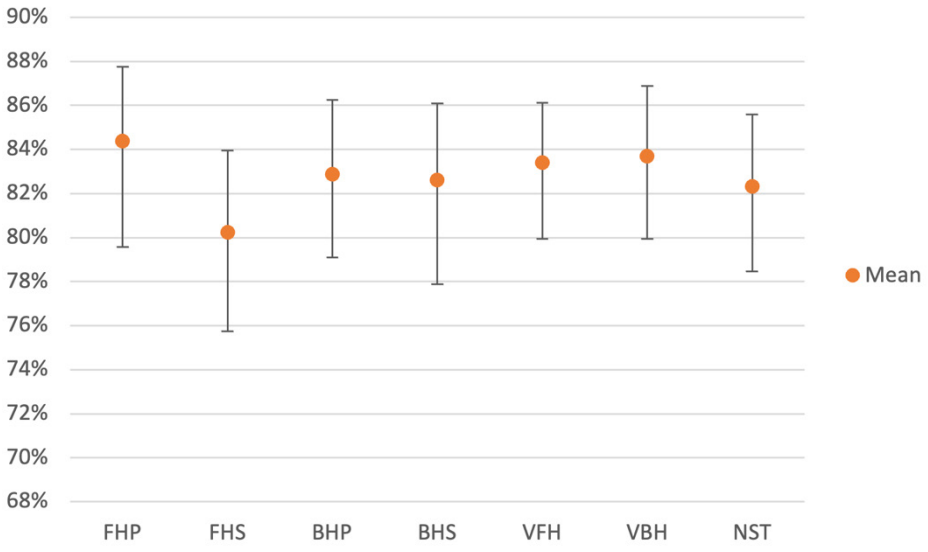


Fig. 5.31. Obtained F1 score results for the **ST-GCN** classifier for 3D tennis racket model input fuzzification for successive classes

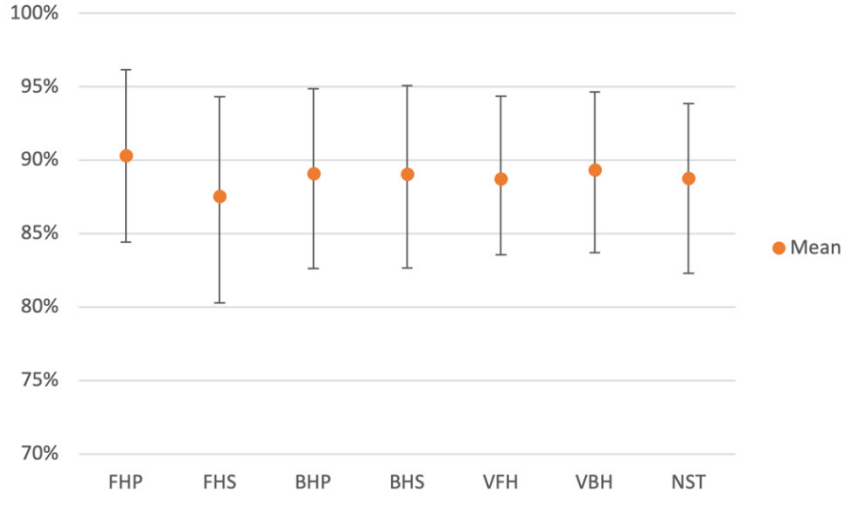


Fig. 5.32. Obtained F1 score results for the **ST-GCN** classifier for 3D tennis player and racket model input fuzzification for successive classes

The F1 score results for all classes for the ST-GCN with fuzzification of various input types are presented in Fig. 5.29–5.32. Similar to the precision and recall results, fuzzification improved the effectiveness of the ST-GCN classifier for all types of input data except

for the three-dimensional data of the tennis player model. The F1 score improvement was from 1.08% to 2.30%. The highest result was obtained for 3D tennis player model and tennis racket model data. Both phases of the forehand, the backhand, as well as no-stroke achieved the most dramatic improvement.

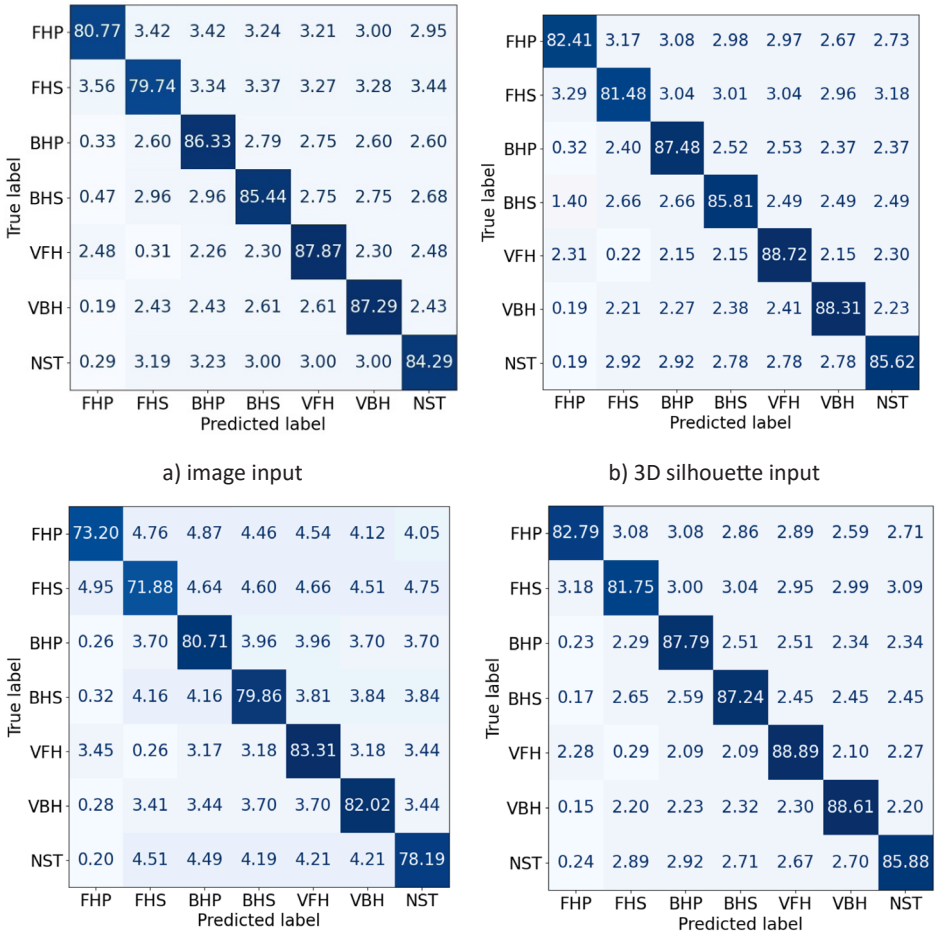
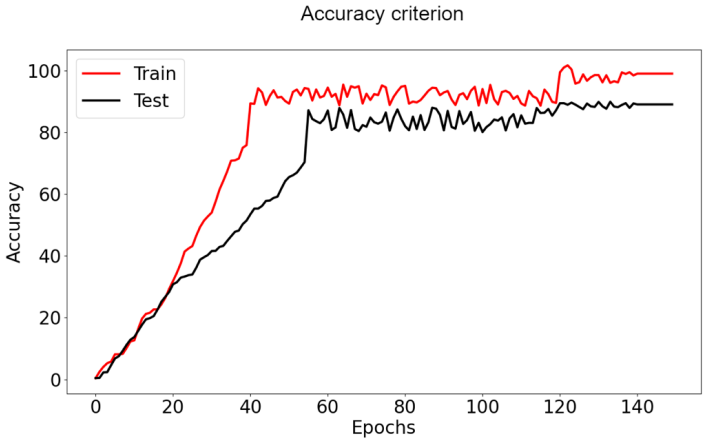


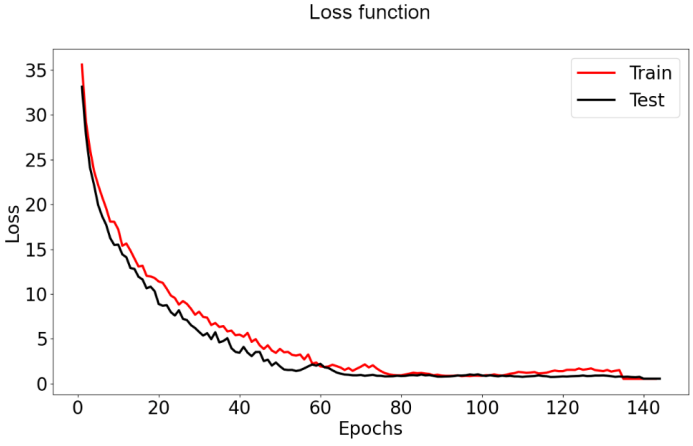
Fig. 5.33. Confusion (in%) for tennis movement the **ST-GCN** classification with fuzzification for various motion capture input data: FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

Confusion matrices for the ST-GCN classifier with fuzzification for various input data are presented in Fig. 5.33. Similar to results without fuzzification, the same moves were misclassified. However, the values of

the confused classification decreased, which proves that the input fuzzification process improves the effectiveness of tennis movement recognition.



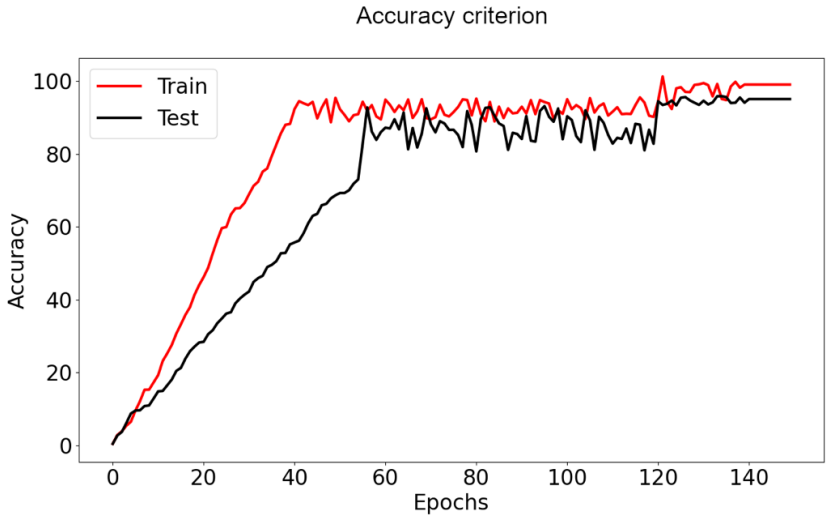
a)



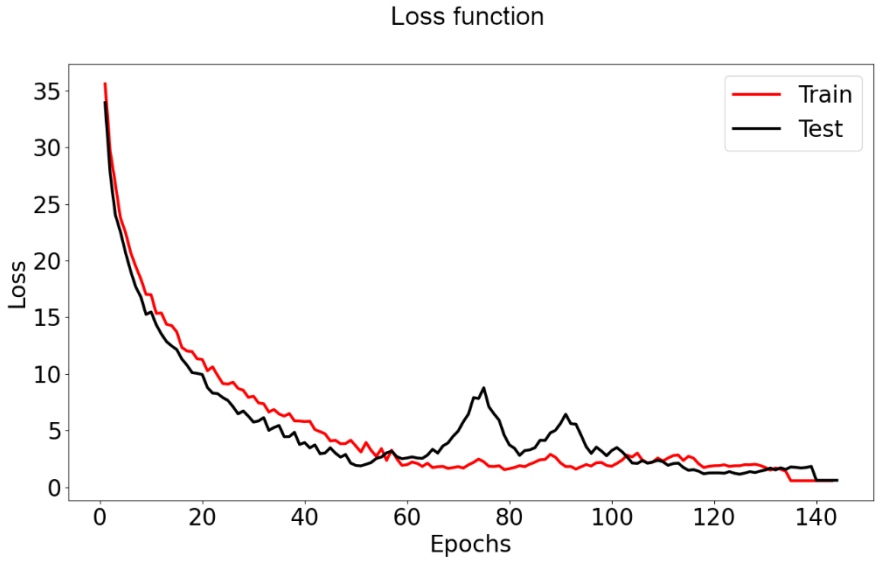
b)

Fig. 5.34. Learning parameters for the **ST-GCN** classifier with fuzzification for image input data a) accuracy, b) loss function

The learning parameters for the ST-GCN with fuzzification process for various input data are presented in Fig. 5.34–5.37. They all vary in the shapes of their functions. However, they reached 140 epochs, which are sufficient to obtain a stable model.

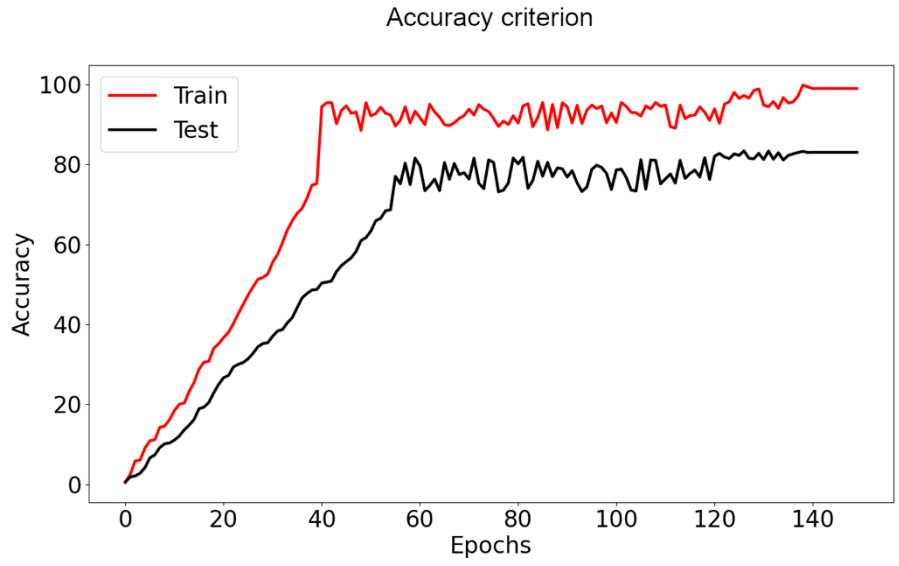


a)

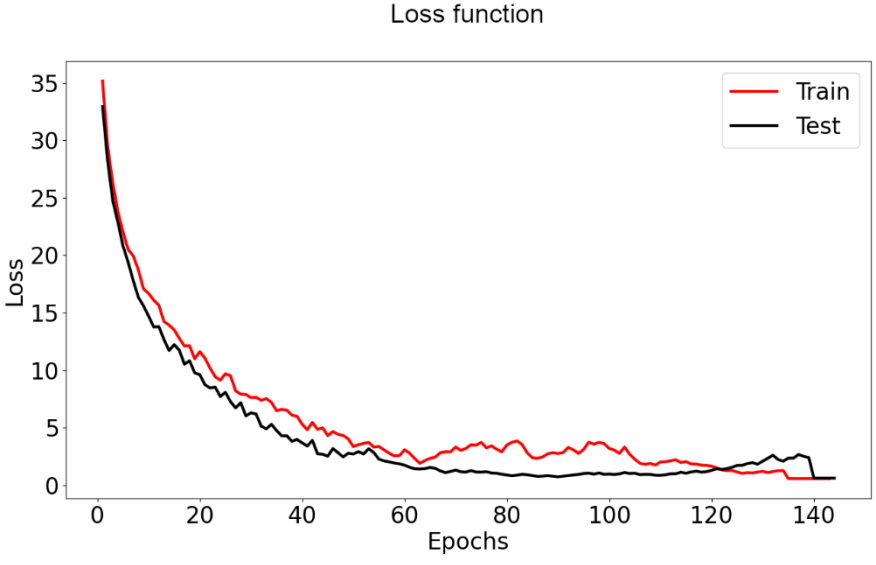


b)

Fig. 5.35. Learning parameters for the **ST-GCN** classifier with fuzzification for 3D tennis player model input data a) accuracy, b) loss function

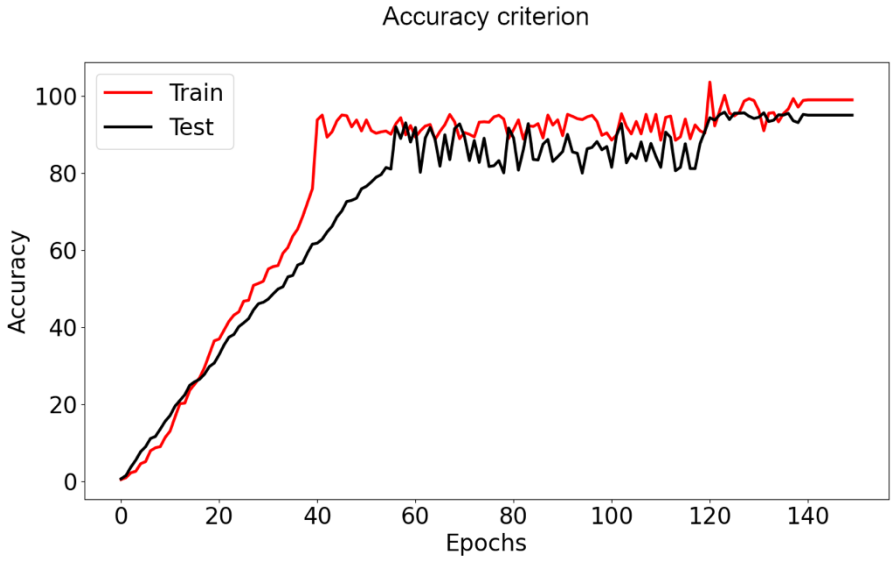


a)

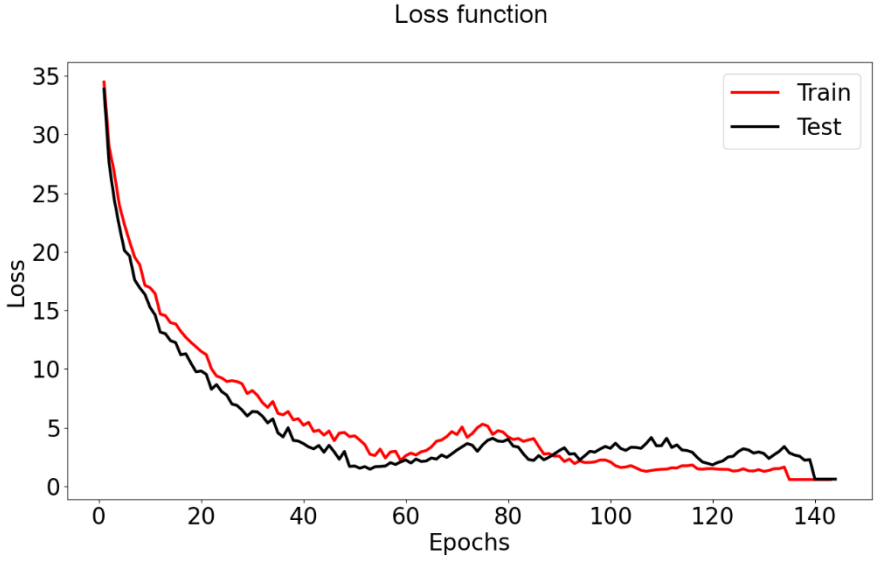


b)

Fig. 5.36. Learning parameters for the **ST-GCN** classifier with fuzzification for 3D tennis racket model input data a) accuracy, b) loss function



a)



b)

Fig. 5.37. Learning parameters for the **ST-GCN** classifier with fuzzification for 3D tennis player model and tennis racket model input data a) accuracy, b) loss function

Based on the obtained results, one can conclude that the fuzzification process improves the ST-GCN classifier performance.

Attention Temporal Graph Convolutional Networks for tennis movements recognition

CHAPTER 6

The Attention Temporal Graph Convolutional Network (A3T-GCN) is the second proposed classifier dedicated to tennis movements recognition [94]. In this approach an attention module is applied for indicating relevant information for classification. Data gathered from the 3DTennisDS are inputted to the proposed solution. Two types of classification are considered: image-based and 3D-based. In both cases data are prepared as described in 5.1.1 and 5.2.1 for image and three-dimensional data, respectively.

6.1. Image-based classification

6.1.1. A3T-GCN classifier

The structure of the A3T-GCN is presented in Fig. 6.1. It consists of five elements: input, spatial components, temporal components, attention model, and finally a classification part.

As it was presented in 5.1.1 section, from the sequences of images representing the tennis player model holding a tennis racket, an input is specified. Three phases of a specific tennis movement were considered. Based on 2D data the spatial temporal graph was generated with nodes and the edges corresponding to time representation. The graph $G = (V, E)$ consist of N nodes and their change in time. The set of nodes were given as eq. 4.36. The GCN was applied to indicate the spatial features from motion capture data. A Fourier filter was utilized to determine the spatial relationships, according to eq. 4.31. In this approach the 3-layer GCN model was applied, as identified in [94]:

$$f(X, A) = \sigma\Big(\hat{A}ReLu\Big(\Big(\hat{A}XW_0\Big)\hat{A}XW_1\Big)W_2\Big) \tag{6.1}$$

where σ stands for Softmax function, $\hat{A} = \widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}$ is pre-processing step, $W_0 \in R^{P\times H}$ denotes weight matrix from input layer to hidden one, while P corresponds to length of the feature matrix, and H to hidden unit numbers. $W_1, W_2 \in R^{H\times T}$ denote weight metrices from hidden to output layer. The output is $f(X, A) \in R^{N\times T}$, where T corresponds to output length and N to the number of nodes.

For determining temporal features the Gated Recurrent Unit (GRU) model was applied, which is the improved version of the RNN. The GRU calculates the information about a tennis player in t time taking into consideration phases of the strokes specified in time t–1 and t–2. Data about previous stroke phases are essential for indicating a tennis player position in current time. Temporal dependence is obtained.

The structure of the GRU is defined as follow [95][96]:

$$u_t = \sigma(W_u * [X_t, h_{t-1}] + b_u) \tag{6.2}$$

$$r_t = \sigma(W_r * [X_t, h_{t-1}] + b_r) \tag{6.3}$$

$$c_t = tanh(W_c * [X_t, (r_t * h_{t-1})] + b_c) \tag{6.4}$$

$$h_t = u_t * h_{t-1} + (1 - u_t) * c_t \tag{6.5}$$

where u_t and r_t stand for the update gate and reset gate, respectively. c_t denotes the memory context, h_{t-1} – the hidden state at time t–1, h_t is the output state at the current time, and X_t is the current information about the tennis player's posture. In this case the reset gate output is equal to zero, the information from the previous moment is skipped over. Otherwise, the information is considered essential data for the next moment of the stroke. The update gate indicates how much information from the previous time moments have to be transformed into current ones. This gate is dedicated to controlling the state information quantity.

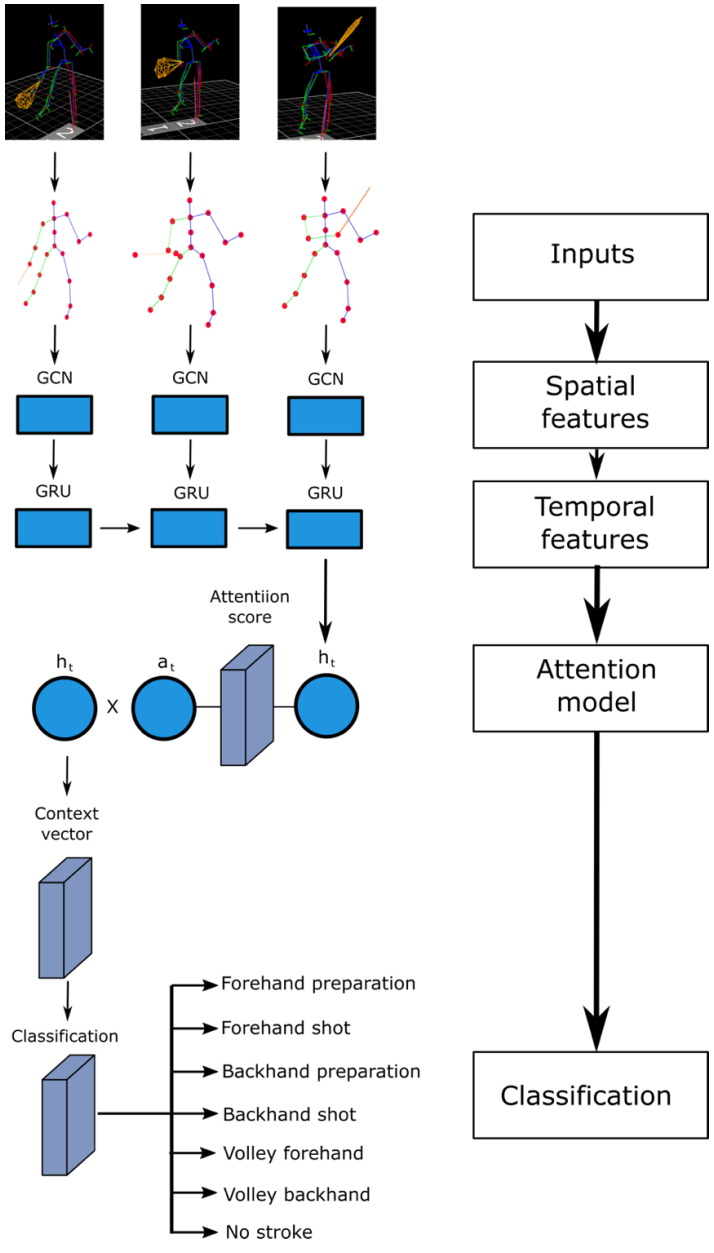


Fig. 6.1. **A3T-GCN** structure for image input [94]

The attention model applied in the A3T-GCN is a soft one. It is implemented as an encoder-decoder. This model is used for indicating the essential information of tennis movement at every moment.

As a result, a context vector is obtained, whose function is to foresee the future of the tennis player's positions. The attention module incorporates several stages. First, the hidden states h_i = (i = 1,2, ...n) of the time series given by states X_i = (i = 1,2, ...n) at various moments are determined utilizing RNN, where n is the length of the series. Then, the output of the RNN is expressed as H = h_1, h_2, ..., h_n. Finally, the context vector (C_v) is calculated.

Output is determined based on two hidden layers, whose weights are specified by the Softmax function [96]:

$$e_i = w_{(2)}\left(w_{(1)}H + b_{(1)}\right) + b_{(2)} \tag{6.6}$$

$$\alpha_i = \frac{exp(e_i)}{\sum_{k=1}^{n} exp(e_k)} \tag{6.7}$$

$$C_v = \sum_{i=1}^{n} \alpha_i * h_i \tag{6.8}$$

where $w_{(1)}$, $b_{(1)}$, $w_{(2)}$, $b_{(2)}$, denote the weights and the biases for the first and the second layer, respectively.

The final classification process was performed using the two-layer Multilayer perceptron. The Softmax function was applied as an activation function of neuron in the first layer and consisted of one element. The second layer has seven neurons that correspond to the final classes of the tennis movements. This linear function was applied for the activation process.

Although GCNs have been widely applied with great success in many fields, their weakness is in revealing features that correspond to their neighbors. As a result, this type of neural network is not able to express the long-range dependencies in graphs, which are crucial for differentiating tennis strokes. Volley forehand movements are similar to some part of the forehand moves. The same is for volley backhand strokes and backhand movements. In order to eliminate the abovementioned difficulties, while designing the A3T-GCN structure for tennis movement recognition, the soft attention mechanism is proposed in which continuous weights are assigned. The main motivation is to capture the internal dependencies of nodes.

6.1.2. Tennis stroke recognition

The methodology of this study is the same as described in 5.1.3. The performance of the A3T-GCN was verified using accuracy, precision, recall, and F1 score measures (eq. 5.6–5.9).

The mean accuracy for the A3T-GCN classifier for all classes is presented in Table 6.1.

Table 6.1. Obtained accuracy results for the **A3T-GCN** classifier for all classes

Mean	*Max*	*Min*	*±SD*
86.12%	88.70%	80.10%	2.69%

The obtained mean accuracy is 4.55% higher compared to the mean accuracy received for ST-GCN classifier. The proposed model is also more stable, as evidenced by almost as twice as low as the standard deviation value. The implementation of the attention model improved classification performance for the image input.

Accuracy results for the A3T-GCN classifier for all classes are gathered in Table 6.2.

Table 6.2. Obtained accuracy results for the **A3T-GCN** classifier for all classes

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	85.34%	88.70%	80.11%	6.89%
FHS	84.31%	89.11%	81.80%	4.29%
BHP	88.00%	89.56%	80.44%	4.67%
BHS	88.40%	89.09%	80.10%	5.96%
VFH	88.71%	90.24%	81.40%	9.61%
VBH	88.90%	90.20%	80.59%	5.46%
NST	87.87%	88.84%	81.83%	9.21%

The obtained accuracy results for all classes are about 5% higher than the Accuracy values calculated for the ST-GCN. For the A3T-GCN the mean outcomes are between 85.34% and 90.24%, while the maximum ones between 88.70% and 90.24%. Precision results for the A3T-GCN classifier for all classes are presented in Table 6.3.

Table 6.3. Obtained precision results for the **A3T-GCN** classifier for all classes

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	84.47%	87.84%	80.89%	6.59%
FHS	83.59%	87.09%	79.71%	6.49%
BHP	88.98%	91.48%	86.38%	6.43%
BHS	88.48%	90.96%	85.65%	6.39%
VFH	89.81%	91.66%	87.85%	6.33%
VBH	89.61%	91.63%	87.32%	6.43%
NST	87.28%	89.96%	84.17%	6.39%

These values are also about 2% higher than ST-GCN precision results. The model's ability to achieve positive classification is the best for the volley forehand and the backhand. Slightly less ability is achieved for no-stroke. The least values are obtained for both phases of the forehand. It should be indicated that mean precision calculations are very satisfactory, ranging between 83.59% and 89.81%.

Recall results for the A3T-GCN classifier for all classes are gathered in Table 6.4. They also exceed recall results for the ST-GCN model. The highest results were obtained for the forehand movements. Slightly lower values were for no-stroke, the volley backhand and for the backhand moves. The lowest recall results were for the volley forehand. However, the values are very high, reaching from 84.72% to 93.55%.

Table 6.4. Obtained recall results for the **A3T-GCN** classifier for all classes

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	93.55%	95.03%	92.01%	6.55%
FHS	87.80%	90.48%	84.73%	6.51%
BHP	86.18%	89.26%	82.99%	6.38%
BHS	86.41%	89.10%	83.11%	6.52%
VFH	84.72%	87.37%	82.05%	7.41%
VBH	86.43%	88.87%	83.11%	7.85%
NST	86.97%	90.01%	83.95%	7.93%

The F1 score results for the A3T-GCN classifier for all classes are presented in Table 6.5. The mean values range from 85.64% for the forehand hit with the racket swinging up to 88.77% in the forehand preparation phase. The remaining scores are on a similar level.

Table 6.5. Obtained F1 score results for the **A3T-GCN** classifier for all classes

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	88.77%	91.30%	86.18%	6.57%
FHS	85.64%	88.75%	82.15%	7.25%
BHP	87.56%	90.36%	84.65%	6.74%
BHS	87.43%	90.02%	84.36%	6.86%
VFH	87.19%	89.47%	84.85%	8.23%
VBH	87.99%	90.23%	85.16%	8.24%
NST	87.13%	89.99%	84.06%	7.76%

The confusion matrix for the A3T-GCN for image input data is depicted in Fig. 6.2. Obtained values are lower than those for the ST-GCN model. Misclassifications are very low; they do not exceed 2.79%.

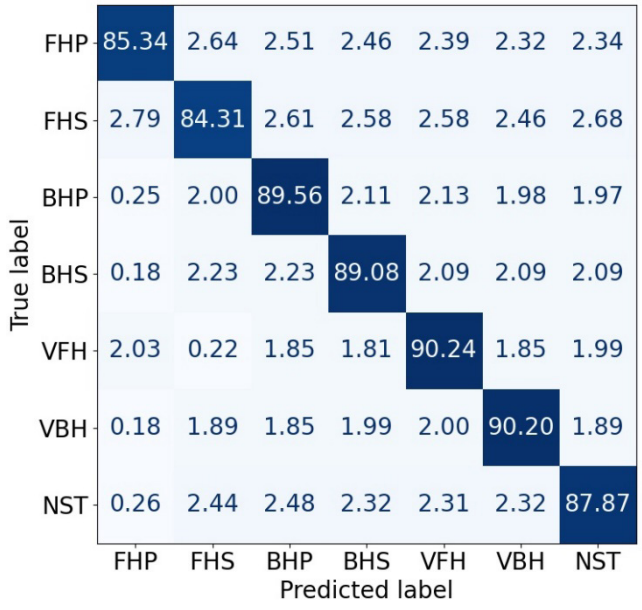


Fig. 6.2. Confusion (in%) for tennis movement **A3T-GCN** classification for image input data FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

The forehand preparation phase was confused with the next phase of this tennis move and also with the backhand preparation phase. The forehand stroke was misleading with regard to the preparation phase of this move and with no-stroke. Similar results were obtained for the backhand preparation phase that was confused with

the backhand stroke and the volley forehand. The backhand hit with racket swinging was equally misleading with the forehand stroke and the backhand preparation phase. The volley forehand was confused with the forehand preparation phase, while the volley backhand with the forehand type of these tennis moves. No-stroke was the most misclassified with both phases of backhand moves.

The learning parameters for the A3T-GCN model are presented in Fig. 6.3 and Fig. 6.4.

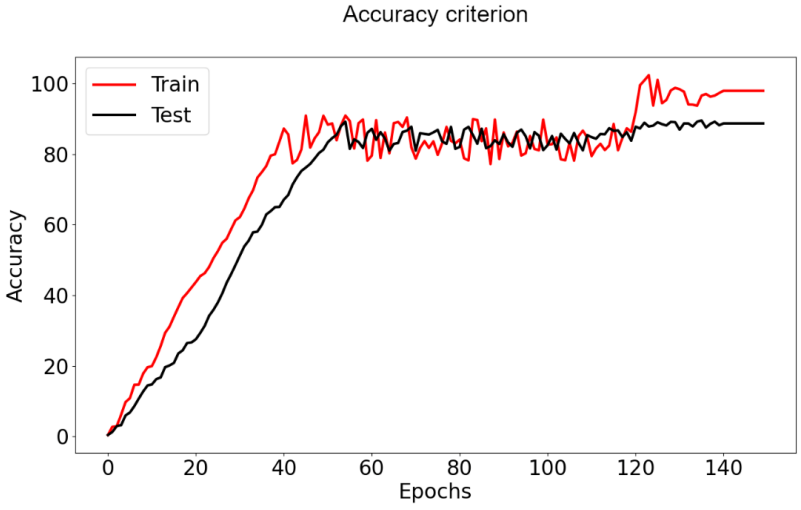


Fig. 6.3. Learning accuracy for the **A3T-GCN** classifier for image input data

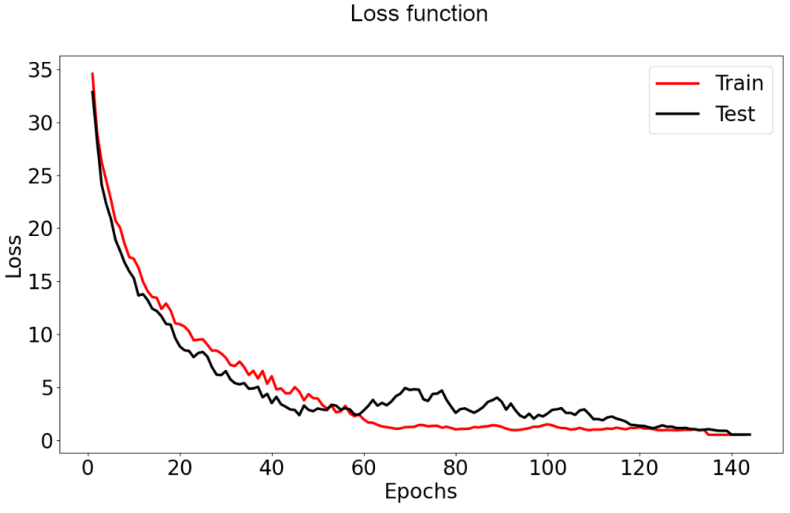


Fig. 6.4. Loss function for the **A3T-GCN** classifier for image input data

Learning accuracy is depicted in Fig. 6.3. As it was with ST-GCN, one can notice that reaching 140 epochs is enough to obtain satisfactory values. Testing values exceed 80%. Learning loss function is depicted in Fig. 6.4. Also, in this case the loss values are the lowest after reaching 140 epochs.

Obtained results for the A3T-GCN for image-based input prove that this classifier is also adequate for tennis movement recognition. Moreover, this model obtained higher results than the ST-GCN solution.

6.2. 3D-based classification

6.2.1. A3T-GCN classifier

The structure of the A3T-CGN classifier for three-dimensional data is depicted in Fig. 6.5. In comparison to Fig. 6.1, the type of input was changed from 2D to 3D. The input data structured is the same as it is described in 5.2.1. It must be emphasized that from the input data the graph was created based on all markers.

6.2.2. Tennis strokes recognition based on a tennis player model

The methodology of the study is the same as described in 5.2.3. The performance of the A3T-GCN was verified using accuracy, precision, recall, and F1 score measures (eq. 5.6–5.9).

Accuracy results for the A3T-GCN model are presented in Table 6.6. These values are 2.71% lower than for the same network for image-based input; however, they are 1.84% higher than for the ST-GCN classifier for the same data type.

Table 6.6. Obtained accuracy results for the **A3T-GCN** classifier for all classes for 3D tennis player model input

Mean	*Max*	*Min*	*±SD*
83.41%	89.93%	78.70%	7.74%

Detailed accuracy results for the A3T-GCN model for all classes are gathered in Table 6.7.

Table 6.7. Obtained accuracy results for the **A3T-GCN** classifier for all classes for 3D tennis player model input data

Class	Mean	Max	Min	±SD
FHP	81.52%	81.30%	70.91%	12.77%
FHS	80.40%	89.93%	70.29%	9.57%
BHP	86.87%	88.91%	78.23%	4.57%
BHS	86.18%	89.91%	80.11%	3.69%
VFH	88.21%	88.23%	82.34%	4.56%
VBH	87.79%	88.23%	83.23%	4.03%
NST	84.87%	87.34%	81.23%	3.73%

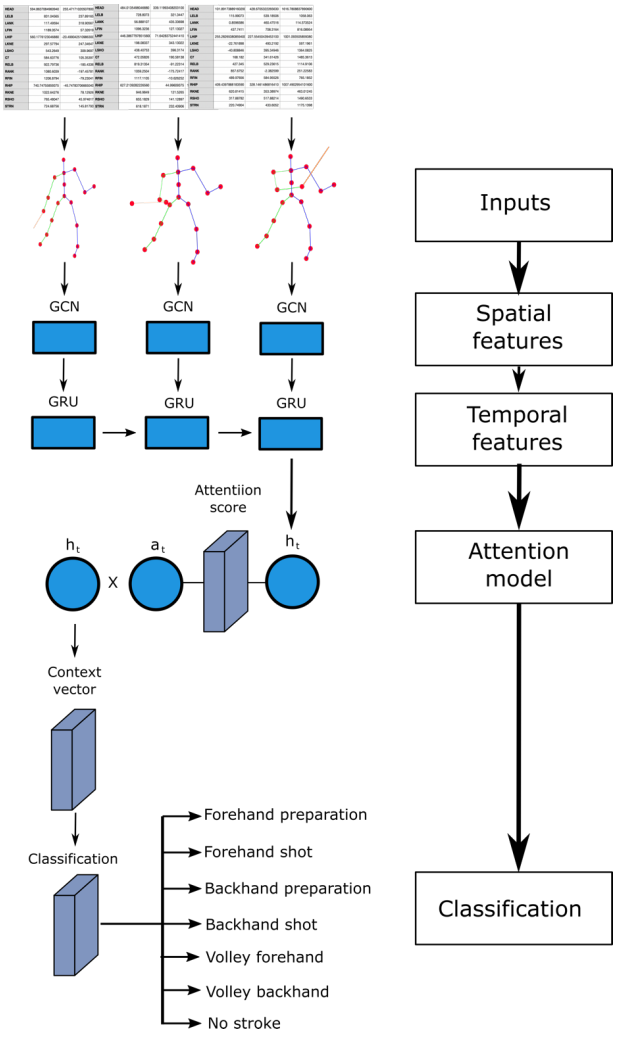


Fig. 6.5. **A3T-GCN** structure for 3D input data, based on [94]

Detailed results are also lower than those for the image-based A3T-GCN classifier; however, they are higher than those obtained for the ST-GCN model for the same type of input.

The average accuracy for all classes is considered in the range of 80.40% to 88.21%. The best classification was obtained for volley and backhand strokes. The lowest accuracy values were obtained for the forehand, followed by no-stroke. The standard deviation values are very low for all tennis moves. Only for the forehand groundstroke was it between 0.57% and 12.77%. The classifier is stable and adequate for tennis movement recognition.

Precision results for the A3T-GCN are gathered in Table 6.8. The highest ability of positive classification was obtained for volley moves, exceeding 89%. Slightly lower results were calculated for the backhand and for no-stroke moves. The lowest values are for forehand groundstrokes. One must notice that the lowest precision was gained at the level of 83%.

Precision results are lower than for the same classifier for image-based input, but higher than for the ST-GCN classifier for the same type of data.

Table 6.8. Obtained precision results for the **A3T-GCN** classifier for all classes for 3D tennis player model input data

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	83.96%	88.62%	79.06%	8.57%
FHS	83.04%	87.92%	78.25%	9.26%
BHP	88.61%	91.89%	85.11%	4.29%
BHS	88.06%	91.56%	84.40%	4.24%
VFH	89.52%	92.07%	86.95%	6.54%
VBH	89.33%	92.06%	86.15%	7.67%
NST	86.83%	90.55%	82.94%	6.76%

Recall results for the A3T-GCN are shown in Table 6.9. They range between 84.35%, for the volley forehand, and 93.37%, for the forehand preparation phase. The calculated values are slightly lower than for image-based input, but higher than for the ST-GCN classifier for 3D silhouette input data.

Table 6.9. Obtained recall results for the **A3T-GCN** classifier for all classes for 3D tennis player model input data

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	93.37%	95.40%	91.41%	7.02%
FHS	87.38%	90.99%	83.35%	7.28%
BHP	85.63%	89.57%	81.30%	5.21%
BHS	86.00%	90.08%	81.67%	7.43%
VFH	84.35%	88.18%	80.80%	6.26%
VBH	86.02%	89.57%	82.23%	6.95%
NST	86.47%	90.38%	82.47%	4.95%

Table 6.10. Obtained F1 score results for the **A3T-GCN** classifier for all classes for 3D tennis player model input data

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	88.41%	91.88%	84.79%	4.72%
FHS	85.15%	89.42%	80.72%	6.42%
BHP	87.10%	90.72%	83.16%	4.83%
BHS	87.01%	90.81%	83.01%	3.14%
VFH	86.86%	90.08%	83.76%	8.76%
VBH	87.64%	90.82%	84.14%	5.23%
NST	86.65%	90.46%	82.71%	5.35%

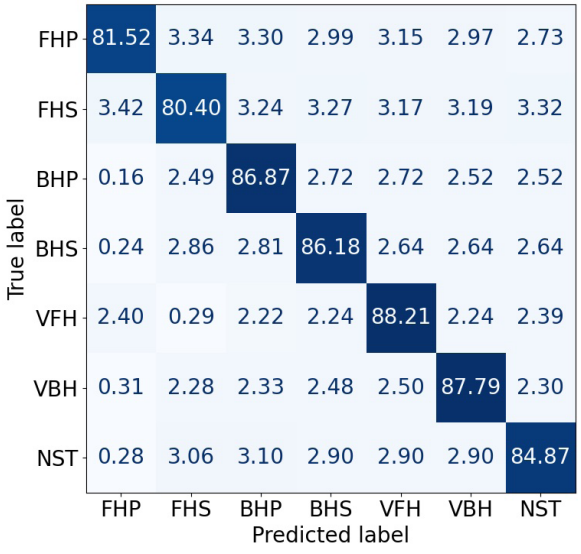


Fig. 6.6. Confusion matrix (in%) for the tennis movement **A3T-GCN** classification for 3D tennis player model input data: FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

The F1 score results are gathered in Table 6.10. They are comparable to the image-based A3T-GCN classifier; however, they are slightly lower than those obtained using the ST-GCN classifier for 3D tennis player model data.

The confusion matrix is depicted in Fig. 6.6. Tennis movements were rarely misclassified and only up to 3.34%. Similar classes were confused as was the case for the image-based input ST-GCN classifier.

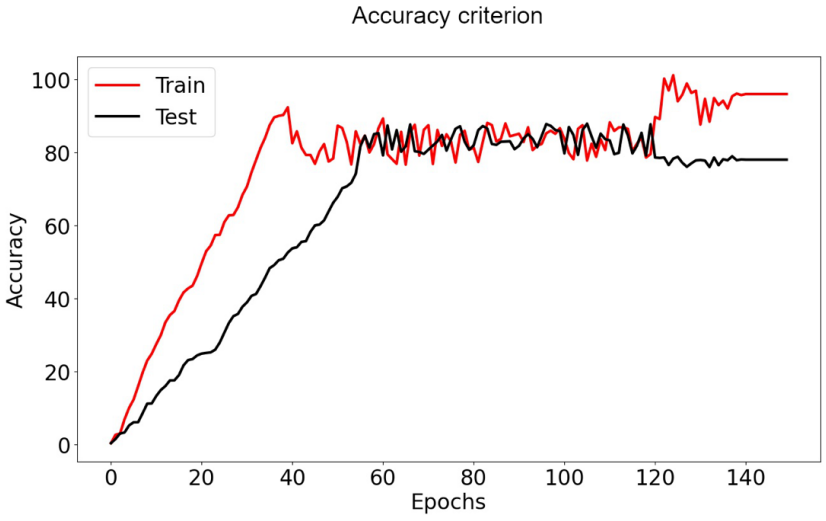


Fig. 6.7. Learning accuracy for the **A3T-GCN** classifier for 3D tennis player model input data

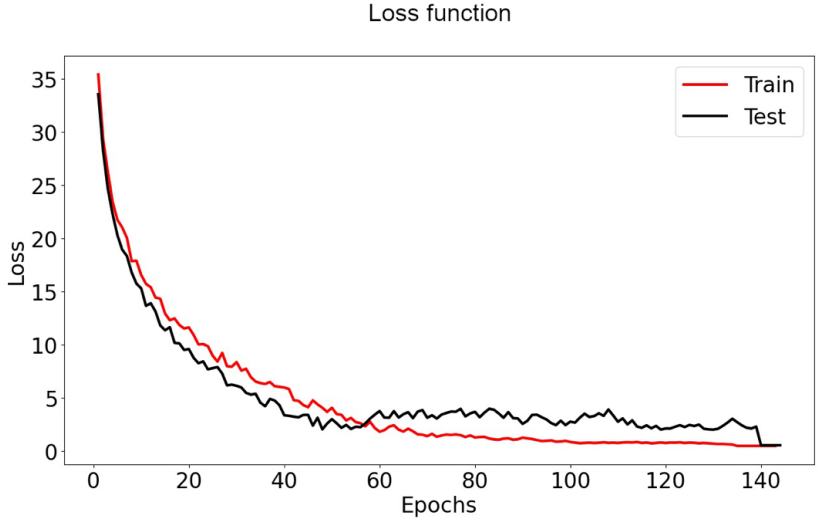


Fig. 6.8. Loss function for the **A3T-GCN** classifier for 3D tennis player model input data

Learning accuracy is depicted in Fig. 6.7. One can notice that reaching 140 epochs is enough to obtain satisfactory values. Testing values reach 80%. The learning loss function is depicted in Fig. 6.8. Also, in this case the loss values are the lowest after reaching 140 epochs.

6.2.3. Tennis strokes recognition based on a tennis racket model

The methodology of classification of three-dimensional data represented by a tennis racket model is similar to that described in 5.2.4 section. Accuracy results for the A3T-GCN based on a 3D tennis racket model as input is presented in Table 6.11. This input type classification quality is slightly lower than that based on 3D tennis player model input data and image-based recognition.

Table 6.11. Obtained accuracy results for the **A3T-GCN** classifier for all classes

Mean	*Max*	*Min*	*±SD*
81.06%	84.99%	77.19%	9.82%

Accuracy results for 3D racket-based classification is gathered in Table 6.12. These values are up to 9.9% lower than for image-based and up to 5.65% for 3D tennis player model classification. However, the 3D racket-based input A3T-GCN classifier obtained higher accuracy results than ST-GCN, up to 2.56%. The best classification was achieved for the volley and backhand strokes. Slightly worse results were obtained for no-stroke and forehand movements. The results range from 73.16% to 84.22%. The proposed model is also stable, which is confirmed by very low standard deviation values.

Table 6.12. Obtained accuracy results for the **A3T-GCN** classifier for all classes for 3D tennis racket model input data

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	74.55%	85.89%	77.42%	3.76%
FHS	73.16%	84.74%	77.25%	3.93%
BHP	81.69%	84.52%	77.19%	4.21%
BHS	80.91%	84.99%	77.53%	4.11%
VFH	84.22%	84.92%	83.23%	3.91%
VBH	82.96%	82.44%	78.77%	3.58%
NST	79.22%	81.76%	77.29%	3.59%

Precision results are gathered in Table 6.13. They indicate a slightly worse ability for positive classification than that based on images (up to 3.6%) and 3D tennis player model input (up to 2.6%). However, the A3T-GCN obtained up to 4.5% higher ability according to the ST-GCN for 3D racket-based input. The highest results were obtained for volley and backhand moves. The lowest ones are for the forehand groundstrokes, followed by no-stroke movements. The values are between 80.30% and 88.25%.

Recall results are presented in Table 6.14. They are slightly lower than those obtained for the A3T-GCN model, up to 2.82% for image-based input, and up to 2.34% for 3D tennis player model input, respectively. However, the results are higher up to 3.75% than for the ST-GCN for the same input type.

Table 6.13. Obtained precision results for the **A3T-GCN** classifier for all classes for 3D tennis racket model input data

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	81.35%	85.43%	77.75%	5.45%
FHS	80.30%	84.38%	76.58%	5.09%
BHP	86.72%	89.68%	84.17%	4.68%
BHS	86.00%	89.11%	83.42%	4.91%
VFH	88.15%	90.22%	86.37%	3.29%
VBH	87.68%	90.32%	85.31%	3.78%
NST	84.72%	88.01%	81.86%	4.67%

The highest results were for the forehand groundstroke and for no-stroke move. The lowest values were obtained for the volley forehand stroke. In the middle, both the backhand groundstroke and the volley backhand were gathered together.

Table 6.14. Obtained recall results for the **A3T-GCN** classifier for all classes for 3D racket input data

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	92.26%	94.20%	89.83%	5.29%
FHS	85.30%	88.43%	82.30%	4.41%
BHP	83.36%	86.93%	80.17%	5.24%
BHS	83.66%	87.26%	80.40%	3.79%
VFH	82.39%	85.36%	79.91%	4.24%
VBH	83.87%	87.31%	80.67%	4.16%
NST	84.22%	87.60%	81.29%	8.88%

The F1 scores for the A3T-GCN model are presented in Table 6.15. They are lower up to 2.83% for image–based input and up to 2.42% for 3D tennis player model input of the same classifier. They are, however, up to 4,01% better for the same input type according to the ST-GCN classifier. The highest results are calculated for the forehand preparation phase, then for volley strokes and backhand groundstrokes. The lowest results were gathered for the forehand shot with racket swinging.

Table 6.15. Obtained F1 score results for the **A3T-GCN** classifier for all classes for 3D tennis racket model input data

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	86.45%	89.59%	83.99%	5.35%
FHS	82.73%	86.35%	79.33%	4.74%
BHP	85.01%	88.29%	82.12%	4.94%
BHS	84.82%	88.17%	81.88%	4.34%
VFH	85.17%	87.73%	83.02%	6.75%
VBH	85.73%	88.79%	82.93%	3.91%
NST	84.47%	87.76%	81.57%	6.81%

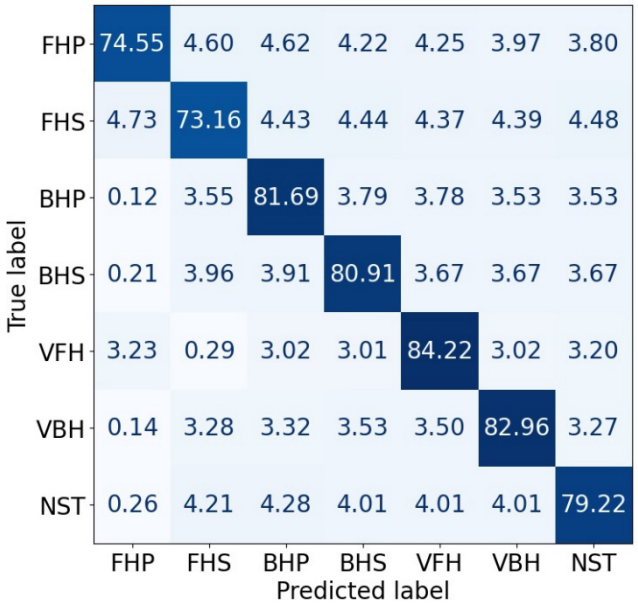


Fig. 6.9. Confusion matrix (in%) for the tennis movement **A3T-GCN** classification for 3D tennis racket model input data: FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

The confusion matrix for the A3T-GCN for the 3D tennis racket input is depicted in Fig. 6.9. The misclassifications are higher than for other types of input. However, these values do not exceed 4.73%. Similar classes were confused as they were for the 3D tennis racket model-based input ST-GCN classifier and the 3D tennis player model-based input A3T-GCN.

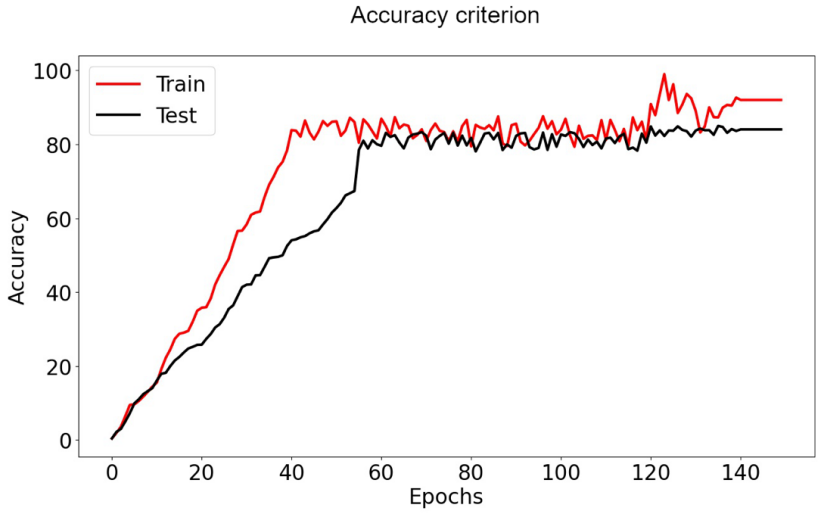


Fig. 6.10. Learning accuracy for the **A3T-GCN** classifier for 3D tennis racket model input data

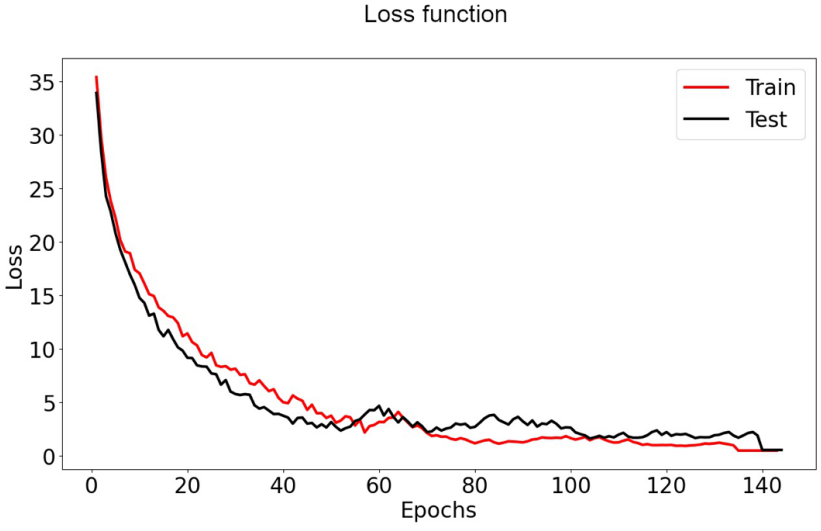


Fig. 6.11. Loss function for the **A3T-GCN** classifier for 3D tennis racker model input data

Learning accuracy is depicted in Fig. 6.10. One can see that 140 epochs are enough to reach satisfactory values. Testing values reach 80%. The learning loss function is depicted in Fig. 6.11. Also, in this case the loss values are the lowest after reaching 140 epochs.

Presented results for the A3T-GCN classifier for tennis racket model input in the form of three-dimensional data prove that adding the attention mechanism improve the quality of tennis motion recognition in comparison to the ST-GCN model. Based on the results it can be concluded that the same racket is not enough for obtaining high classifier quality. The tennis player model seems to be more important.

6.2.4. Tennis strokes recognition based on a combined tennis player and racket model

The methodology of classification of three-dimensional data represented by a tennis player model and the tennis racket model is similar to what is described in 5.2.5 section.

Accuracy results for the A3T-GCN based on the 3D tennis player model with a tennis racket model as input is presented in Table 6.16. They obtained the highest results for all analyzed inputs for the A3T-GCN.

Table 6.16. Obtained Accuracy results for the **A3T-GCN** classifier for all classes

Mean	*Max*	*Min*	*±SD*
89.45%	93.23%	85.23%	4.12%

Detailed accuracy results for combined input as three-dimensional data represented by a tennis player model together with a tennis racket model are gathered in Table 6.17. These values are highest according to other input data types. They are up to 2.46%, 6.37%, 13.61% higher for image-based input, 3D tennis player model input, and 3D tennis racket model input, respectively. The A3T-GCN is up to 6.48% more effective than the ST-GCN for the same input type.

The best classification, exceeding 90%, was obtained for volley and backhand strokes; however, the lowest was for forehand groundstrokes, between 86.77% and 87.58%. The no-stroke move also has very high accuracy.

Table 6.17. Obtained accuracy results for the **A3T-GCN** classifier for all classes for 3D tennis player model with tennis racket model input data

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	87.58%	95.34%	84.34%	6.02%
FHS	86.77%	94.84%	85.96%	6.29%
BHP	91.19%	95.85%	85.39%	3.16%
BHS	90.70%	95.01%	85.03%	5.06%
VFH	91.55%	94.95%	85.29%	3.35%
VBH	91.53%	94.99%	88.28%	3.68%
NST	89.76%	94.97%	85.13%	3.11%

Precision results are presented in Table 6.18. The A3T-GCN obtained the highest ability to achieve positive classification among other input data types, up to 5.88%, 3.73%, and 6.47% for image-based input, 3D tennis player model input, and 3D tennis racket model input, respectively. All tennis moves, except for the forehand stroke, reached very high results above 90%.

Table 6.18. Obtained precision results for the **A3T-GCN** classifier for all classes for 3D tennis player model with tennis racket model input data

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	90.10%	93.81%	85.81%	4.73%
FHS	89.47%	93.48%	84.83%	3.58%
BHP	92.91%	95.28%	89.96%	4.64%
BHS	92.62%	95.34%	89.32%	6.37%
VFH	93.01%	95.32%	90.55%	3.38%
VBH	93.01%	95.32%	90.49%	3.36%
NST	91.78%	94.81%	88.29%	7.93%

Recall results are presented in Table 6.19. They are the highest results for various data types of the A3T-GCN. They are up to 4.47%, 4.89%, and 6.90% higher for image-based input, 3D tennis player model input, and 3D tennis racket model input, respectively. The A3T-GCN obtained better results than the ST-GCN model of the same input type. All tennis moves, except the volley forehand, reached very high values, exceeding 90%. For the volley forehand a high mean result was also obtained.

Table 6.19. Obtained recall results for the **A3T-GCN** classifier for all classes for 3D tennis player model with tennis racket model input data

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	95.75%	96.92%	93.97%	8.03%
FHS	92.27%	95.28%	88.79%	7.87%
BHP	91.04%	94.18%	87.37%	8.22%
BHS	91.32%	94.59%	87.32%	9.87%
VFH	89.47%	92.92%	85.83%	4.15%
VBH	90.77%	93.86%	87.47%	7.98%
NST	91.71%	94.80%	88.18%	8.82%

The F1 score results, gathered in Table 6.20, obtained very high values for all tennis movements. They are up to 5.21%, 5.70%, and 8.12% for image-based input, 3D tennis player model input, and 3D tennis racket model input, respectively. The obtained results are also up to 5.63% higher for the same input of the ST-GCN classifier. Backhand strokes, volley moves, and no-stroke reached values exceeding 91%. The forehand preparation phase had higher result, 92.84%, than the forehand hit with racket swinging, namely 90.85%.

Table 6.20. Obtained F1 score results for the **A3T-GCN** classifier for all classes for 3D tennis player model with tennis racket model input data

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	92.84%	95.34%	89.70%	5.59%
FHS	90.85%	94.37%	86.77%	2.08%
BHP	91.96%	94.73%	88.64%	6.46%
BHS	91.96%	94.96%	88.31%	8.14%
VFH	91.21%	94.10%	88.12%	6.89%
VBH	91.88%	94.59%	88.96%	5.72%
NST	91.75%	94.80%	88.23%	8.36%

The confusion matrix for the A3T-GCN for the 3D combined input is depicted in Fig. 6.12. Misclassifications are lower than for other types of input of the same classifier. The minimal value was 2.32. Similar classes were confused as they were for the image-based input, 3D racket-based and the 3D silhouette-based input A3T-GCN.

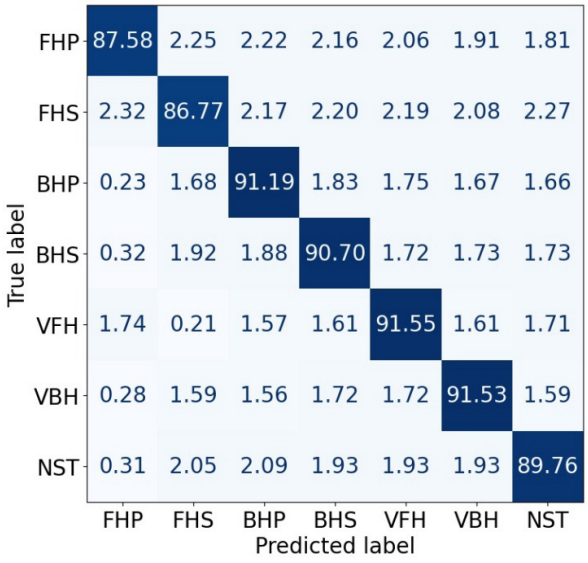


Fig. 6.12. Confusion matrix in%) for the tennis movement **A3T-GCN** classification for 3D tennis player model with tennis racket model input data: FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

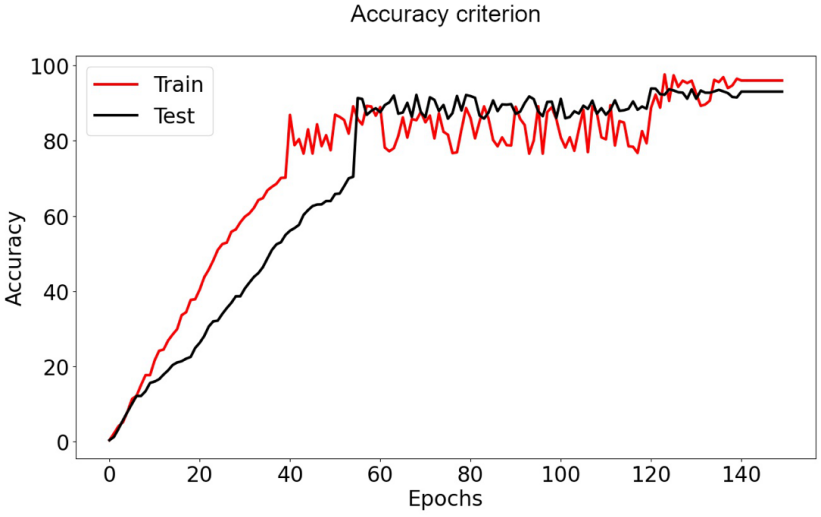


Fig. 6.13. Learning accuracy for the **A3T-GCN** classifier for 3D tennis player model with tennis racket model input data

Learning accuracy is depicted in Fig. 6.13. It can be seen that 140 epochs are enough to reach satisfactory values. Testing values are above 80%. The learning loss function is depicted in Fig. 6.14.

Also, in this case the loss values are the lowest after reaching 140 epochs.

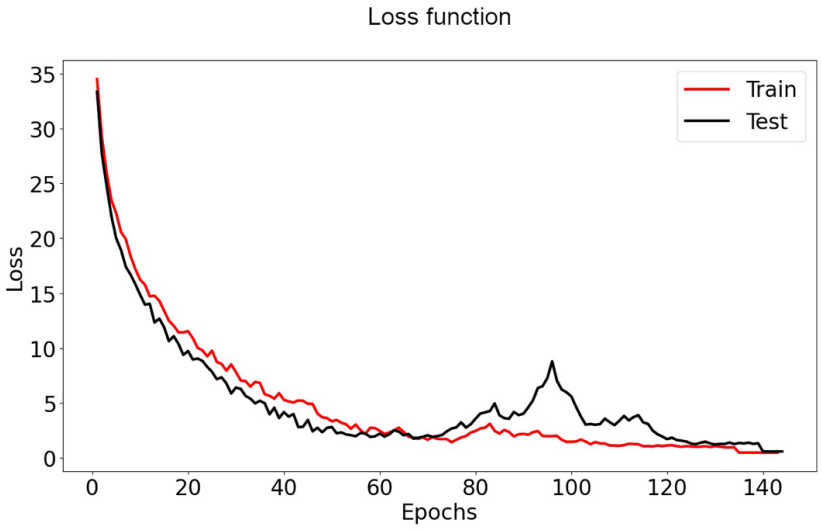


Fig. 6.14. Loss function for the **A3T-GCN** classifier for 3D tennis player model with tennis racket model input data

Results presented for the A3T-GCN clearly show that the best tennis movements recognition was obtained for the combined 3D data. Adding the attention method improved the performance of the proposed network model.

6.3. Classification utilizing fuzzification of input data

Fuzzification of input data was also applied for the A3T-GCN classifier. This process was the same as that described in 5.3 section.

Results for accuracy for all classes for the A3T-GCN with fuzzification of input are gathered in Table 6.21. It must be emphasized that adding fuzzification improves the mean accuracy of the A3T-GCN classifier, for image input by 1.35%, for 3D tennis player model by 1.76%, for 3D tennis racket model by 1.11%, and for 3D combined data by 2.77%. The highest values are obtained for input represented by the 3D tennis player model with a tennis racket model, while the lowest are for the 3D tennis racket model.

Table 6.21. Obtained accuracy results for **A3T-GCN** classifier for input fuzzification for successive classes

Input type	*Mean*	*Max*	*Min*	*±SD*
image	87.47%	91.94%	83.37%	4.61%
3D tennis player model	85.17%	92.49%	77.78%	5.94%
3D tennis racket model	82.17%	87.43%	77.18%	9.31%
3D tennis player model with a tennis racket model	92.22%	97.43%	87.03%	6.53%

Results for mean accuracy for the successive classes for the A3T-GCN with fuzzification of various input types are gathered in Fig. 6.15–6.18. Applying fuzzification improved the classification accuracy for all input types for all classes, except the forehand preparation stroke for image input.

In the case of image input, the accuracy classification was slightly improved by 0.1%. The highest improvement was reached for the forehand stroke. Slightly less accuracy was obtained for no-stroke. Both volley moves reached 0.03% improvements, while the backhand hit with racket swinging and the backhand preparation phase reached 0.02% and 0.01%, respectively.

Classification accuracy enhancement for the 3D tennis player model was from 0.14% to 0.48%. The highest accuracy was obtained for the forehand stroke. No-stroke, the backhand hit, and the forehand preparation phase were the three next in line, ranging 0.38%, 0.30%, and 0.22%, respectively. The backhand preparation phase gained 0.16%, while volley moves 0.17% and 0.14% for the forehand and backhand, respectively.

Regarding the single 3D tennis racket model as well as the 3D tennis player model with racket as inputs to the A3T-GCN classifier, the mean accuracy enhancement exceeded 2%. Improvements for the single racket model were obtained from 2.76%, for the volley forehand, to 4.69%, for the forehand stroke. The forehand preparation phase classification was also improved by 4.54%. Backhand movements recognition was improved by 3.47% and 3.40% for the hit and preparation phase, respectively. The lowest enhancement was reached for volley strokes: backhand by 3.17% and forehand by 2.76%. No-Stroke movement reached 3.66% improvement.

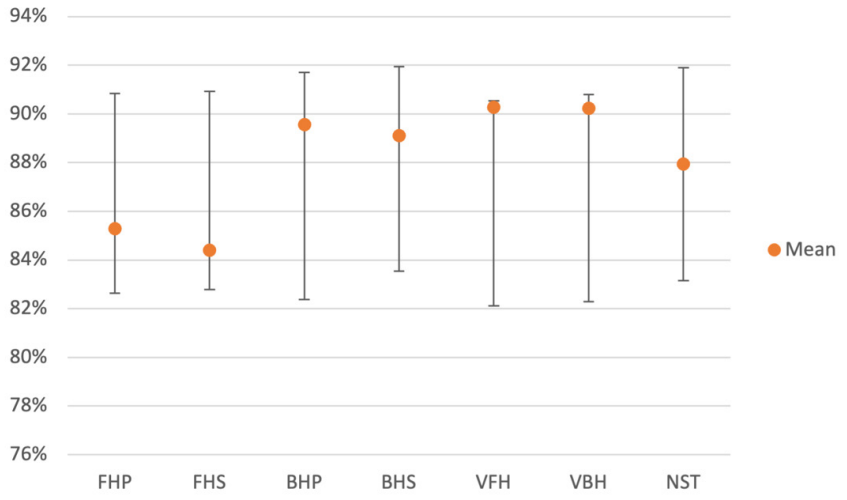


Fig. 6.15. Obtained accuracy results for the **A3T-GCN** classifier for image input fuzzification

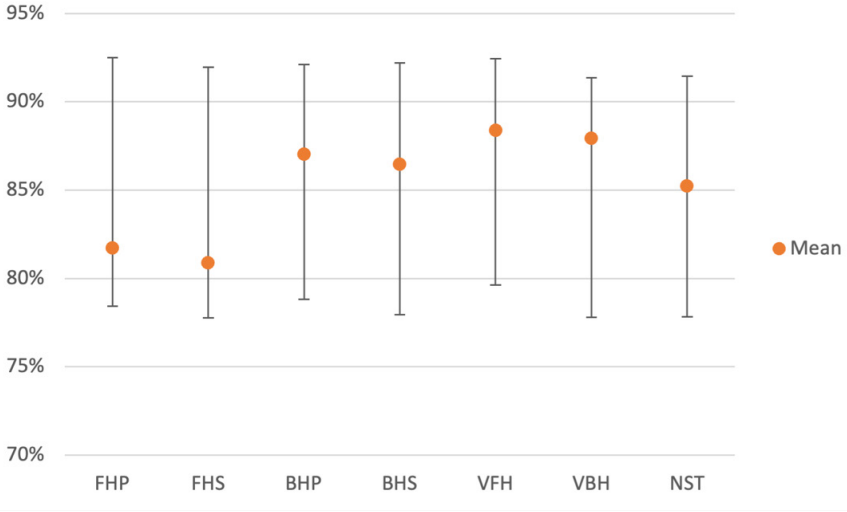


Fig. 6.16. Obtained accuracy results for the **A3T-GCN** classifier for 3D tennis player model input fuzzification

For 3D tennis player model input data forehand strokes were improved by 3.79% and 3.59% for the hit and preparation phase, respectively. Backhand moves obtained improvements by 2.73% and 2.45% for second and first phase, respectively. The least enhancements were gained for no-stroke (2.81%), the volley backhand (2.11%), and the volley forehand (2.07%).

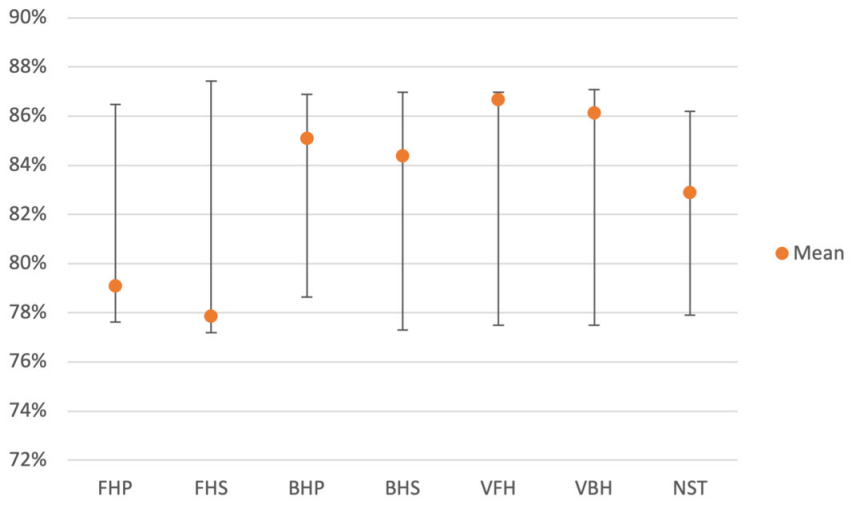


Fig. 6.17. Obtained accuracy results for the **A3T-GCN** classifier for 3D tennis racket model input fuzzification

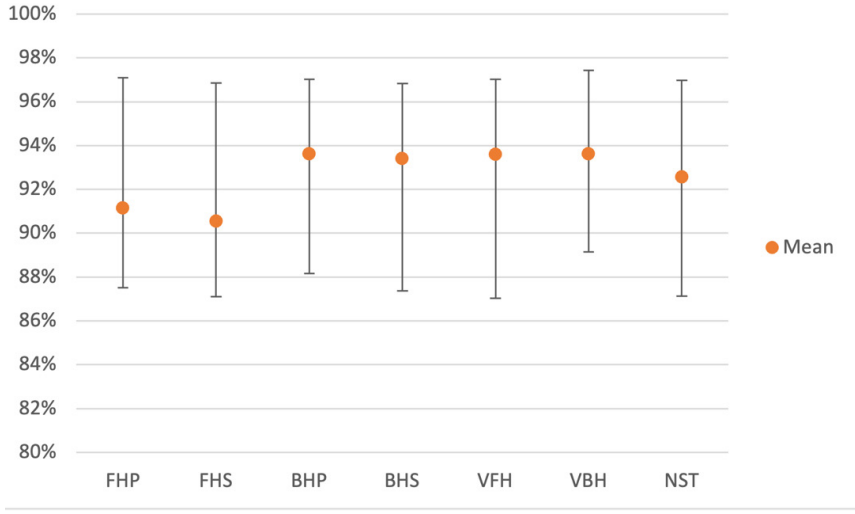


Fig. 6.18. Obtained accuracy results for the **A3T-GCN** classifier for 3D tennis player and racket model input fuzzification

In all cases the highest improvement was obtained for the forehand hit with a swinging racket.

Results for mean precision for the successive classes for the A3T-GCN with fuzzification for various input types are gathered in Fig. 6.19–6.22. Outcomes indicate that the fuzzification process improved the ability to achieve positive classification. For image-based input the highest

improvements for precision were obtained for the forehand hit and the preparation phase, ranging 3.77% and 3.63%, respectively. No-stroke obtained 2.89%. Backhand moves gained 2.66% and 2.56% for the hit and the preparation phase, respectively. The least enhancements were gained for the volley backhand and forehand, reaching 2.24% and 1.98%.

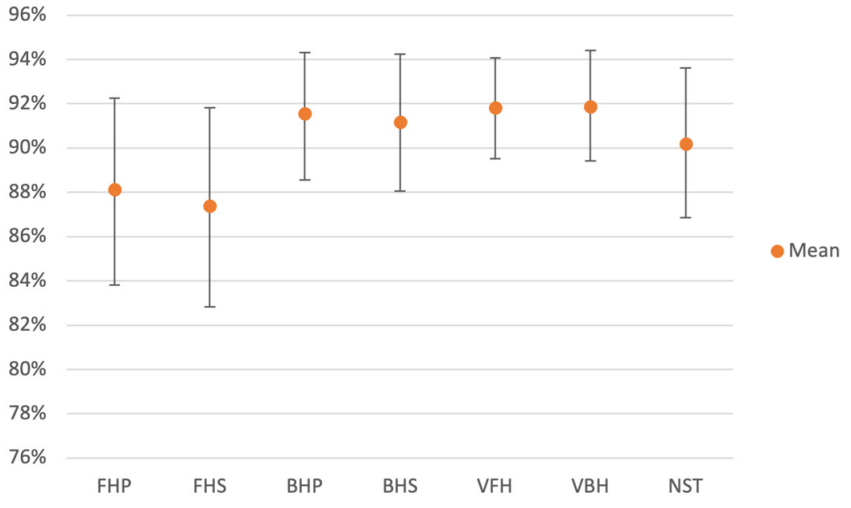


Fig. 6.19. Obtained Precision results for the **A3T-GCN** classifier for image input fuzzification

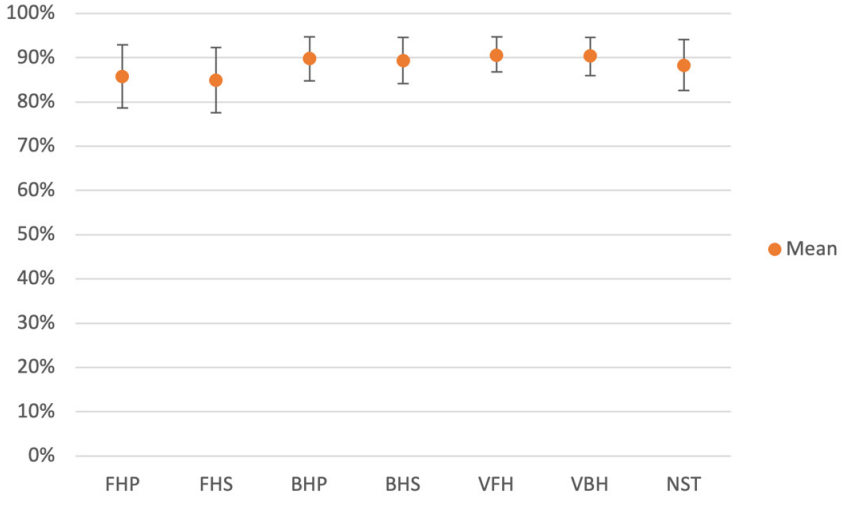


Fig. 6.20. Obtained Precision results for the **A3T-GCN** classifier for 3D tennis player model input fuzzification

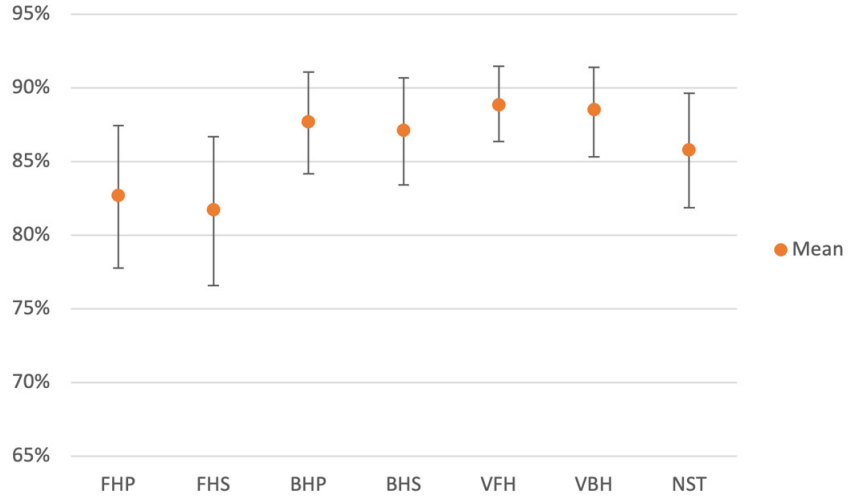


Fig. 6.21. Obtained Precision results for the **A3T-GCN** classifier for 3D tennis racket model input fuzzification

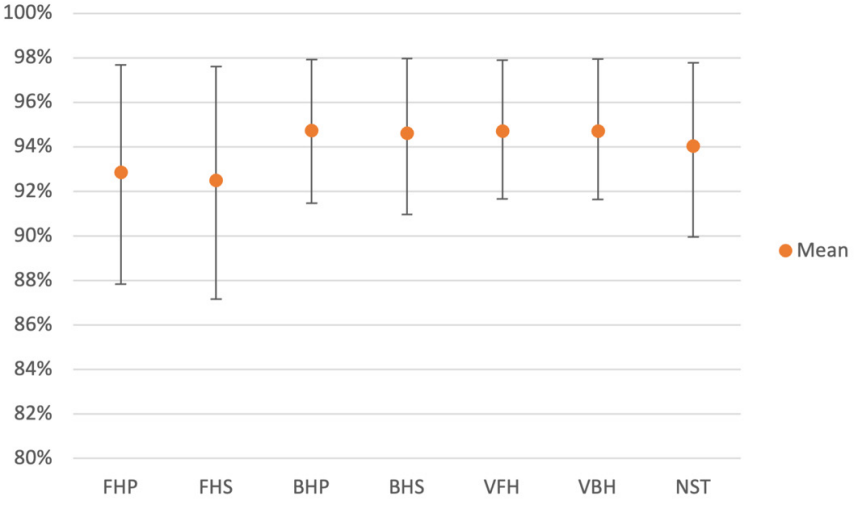


Fig. 6.22. Obtained Precision results for the **A3T-GCN** classifier for 3D tennis player and racket model input fuzzification

For 3D tennis player model input data, the highest improvements were obtained for the forehand hit (1.83%) and the preparation phase (1.78%). No-stroke reached 1.45% improvements, while backhand moves achieved 1.25% and 1.21% for the hit and the preparation phase, respectively. The least changes for better improvements were obtained for the volley forehand (1.06%) and the backhand (1.05%).

Enhancements for precision results were the least for 3D racket input data for the A3T-GCN classifier. Within these data, the improvements above 1% were obtained for the forehand hit (1.41%), the forehand preparation phase (1.36%), the backhand hit (1.12%), and no-stroke (1.07%). Refinement below 1% was gained for the backhand preparation phase (0.97%), the volley backhand (0.85%), and the volley forehand (0.70%).

In the case of 3D combined data, the highest improvements were obtained for the forehand hit (3.00%), the forehand preparation phase (2.74%), and no-stroke (2.23%). Backhand moves reached enhancement reaching almost 2%, 1.98% and 1.81% for the hit and preparation phases, respectively. Both types of volley moves obtained a lower refinement at the level of 1.68%.

Recall results for the successive classes for the A3T-GCN with fuzzification for various input types are gathered in Fig. 6.23–6.26. Also in this case, this process made the results higher. For image-based input to the classifier the biggest improvements were obtained for the forehand move, 6.27%, and for the preparation phase and hit 5.02%. No-stroke gained 2.85 enhancement. At a similar level the backhand groundstroke reached 2.08% and 1.78% for the hit and the preparation phase, respectively. The least improvements were received as 1.29% for the volley backhand and 0.5% for the volley forehand.

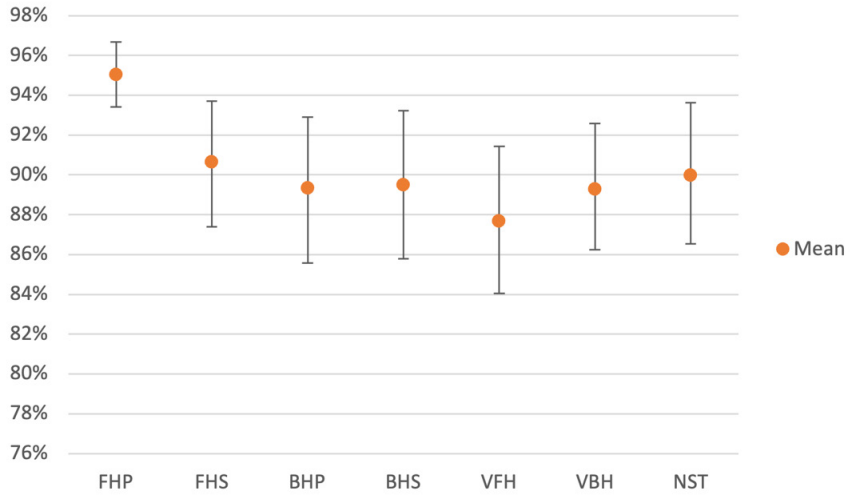


Fig. 6.23. Obtained recall results for the **A3T-GCN** classifier for image input fuzzification

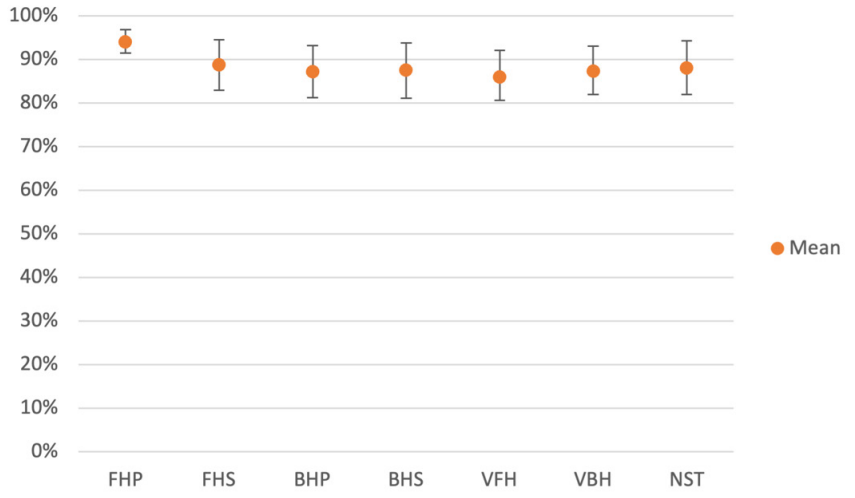


Fig. 6.24. Obtained recall results for the **A3T-GCN** classifier for 3D tennis player model input fuzzification

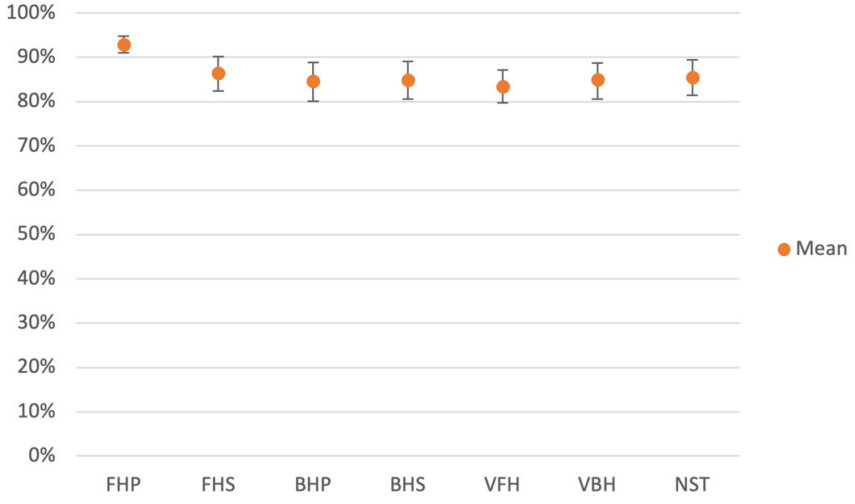


Fig. 6.25. Obtained recall results for the **A3T-GCN** classifier for 3D tennis racket model input fuzzification

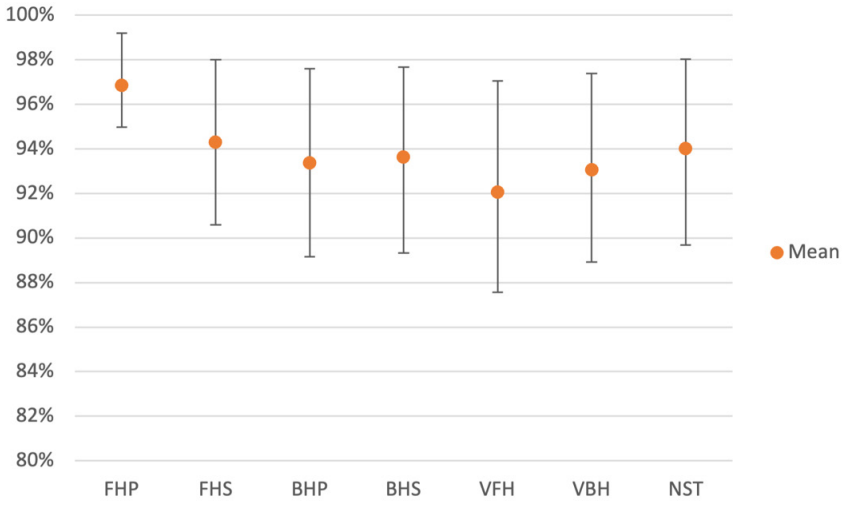


Fig. 6.26. Obtained recall results for the **A3T-GCN** classifier for 3D tennis player and racket model input fuzzification

Regarding the 3D tennis player model for volley moves no enhancement was reached. The highest improvements for this type of input data were obtained for the forehand preparation phase at 5.58% and for the forehand hit together with racket swinging at 3.64%. Backhand strokes reached improvements below 1%, while no-stroke gained enhancement at a level of 1.34%.

In the case of three-dimensional tennis racket data (the same as for previous input types) the highest improvements between classification with and without the fuzzification process were obtained for the backhand preparation phase and the backhand hit, reaching 4.95% and 4.84%, respectively. No-stroke was boosted by 4.76%, while the volley backhand by 4.71%. For the forehand hit and the volley forehand, the enhancements were by 4.61% and 4.30%, respectively. Least improvements were obtained for the forehand preparation phase at the level of 2.32%.

For three-dimensional input in the form of the tennis player model holding a tennis racket, classification results of all strokes were improved. Highest enhancement results were obtained for the forehand preparation phase and the hit, 4.01% and 3.47%, respectively. Almost twice a smaller boost was obtained for no-stroke. Improvements for the backhand preparation phase reached 1.41%, while for hitting they reached 1.68%. The least enhancements were obtained

for volley strokes, reaching 1.20% and 0.86% for the backhand and the forehand, respectively.

The F1 scores obtained after applying the fuzzification process are presented in Fig. 6.27–6.30. According to the A3T-GCN model the fuzzification improved the classification. In this case, for each input type, improvement results at a similar level were obtained.

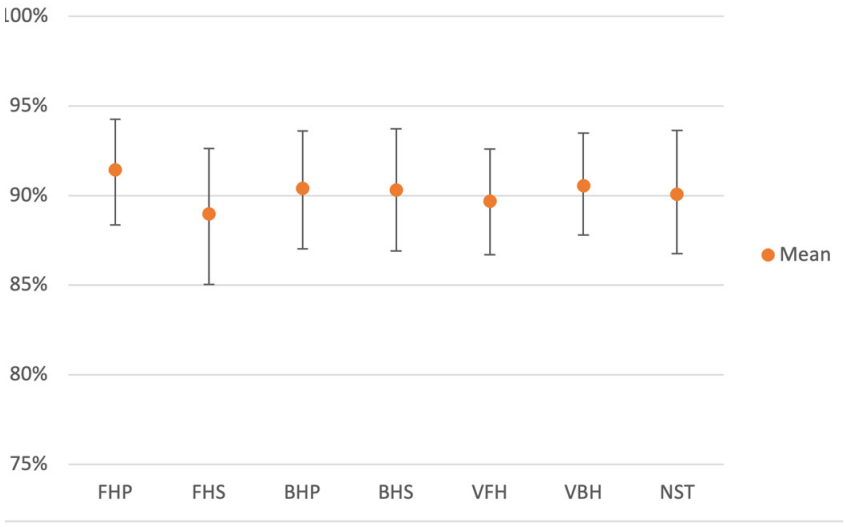


Fig. 6.27. Obtained F1 score results for the **A3T-GCN** classifier for image input fuzzification

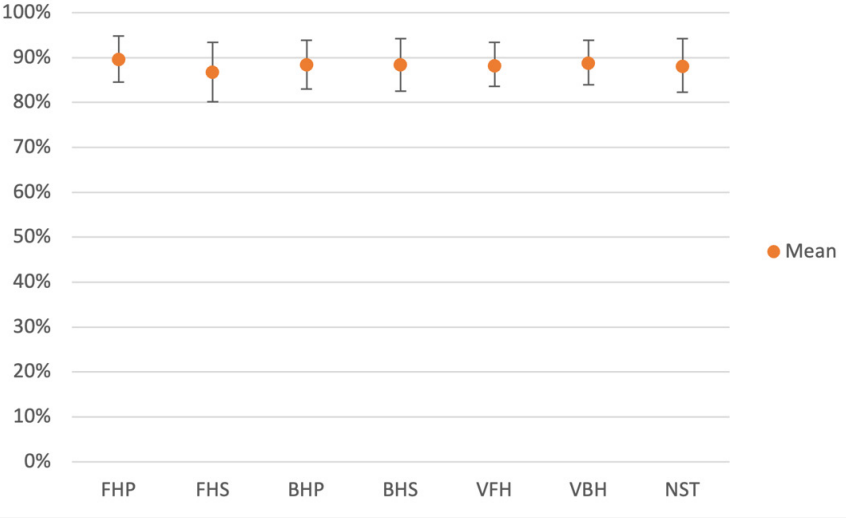


Fig. 6.28. Obtained F1 score results for the **A3T-GCN** classifier for 3D tennis player model input fuzzification

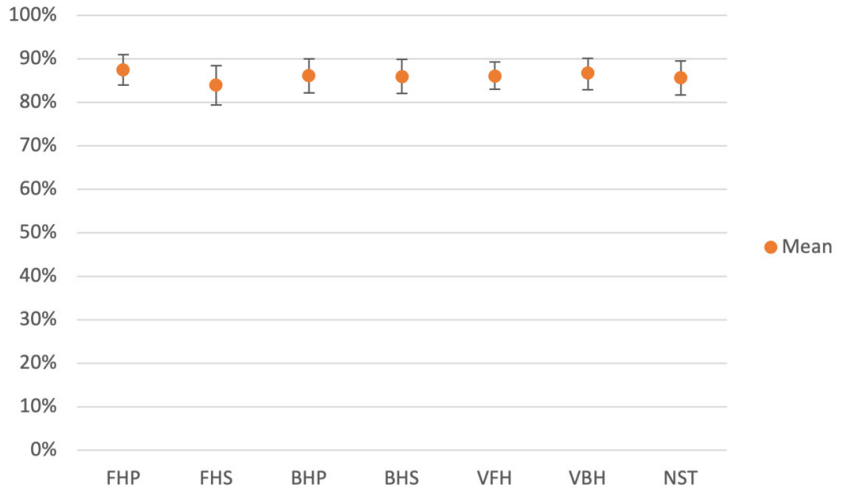


Fig. 6.29. Obtained F1 score results for the **A3T-GCN** classifier for 3D tennis racket model input fuzzification

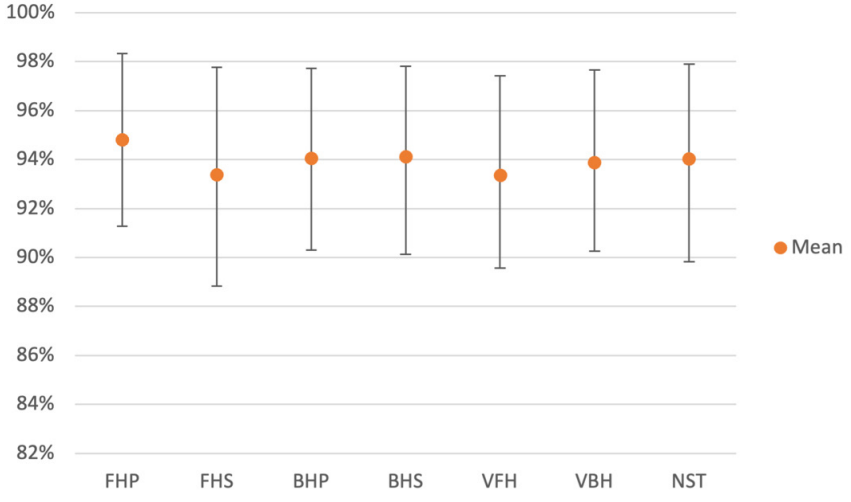


Fig. 6.30. Obtained F1 score results for the **A3T-GCN** classifier for 3D tennis player and racket model input fuzzification

Enhancements for image-based input were reached between 2.50% and 3.34%. The highest values were gained for hitting with the racket swinging for both the forehand and the backhand moves. Regarding the three-dimensional tennis player model, the improvements were obtained between 1.19% and 1.63%. As described above, the hitting part of strokes obtained the greatest boost. For three-dimensional input

data in the form of a tennis racket model, improvements were obtained around 1%, from 0.85% to 1.25%. For this input type, forehand hitting and no-stroke obtained the highest improvement values. In the case of data combining a three-dimensional model of player and tennis racket, the increase achieved reached between 1.96% and 2.56%.

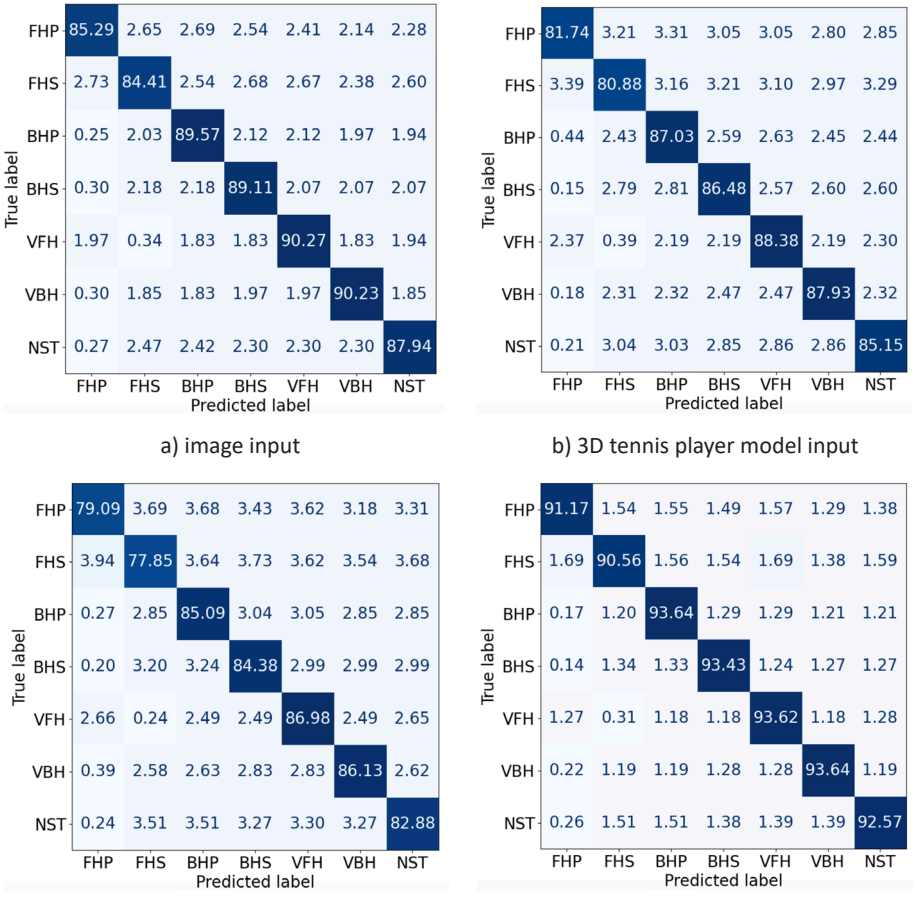


Fig. 6.31. Confusion matrices (in%) for the tennis movement **A3T-GCN** classification with fuzzification: FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

The confusion matrices for the A3T-GCN classifier with fuzzification for various input data are presented in Fig. 6.31. Similar to

results without fuzzification, the same moves are misclassified. However, the values of the confused classification slightly decreased, which proves that the input fuzzification process improves the effectiveness of tennis movement recognition.

The learning accuracy charts for the A3T-GCN with fuzzification process are depicted in Fig. 6.32, Fig. 6.34, Fig. 6.36, and Fig. 6.38. In the case of image-base data and 3D tennis player model with a tennis racket, accuracy exceeded 80%. For the input data of a tennis player model or a tennis racket model, accuracy values were below 80%. In each case 140 epochs were enough to learn the model. The loss functions for the A3T-GCN classifier with fuzzification process are presented in Fig. 6.33, Fig. 6.35, Fig. 6.37, and Fig. 6.39. For each input type the minimum loss value was reached at 140 epochs.

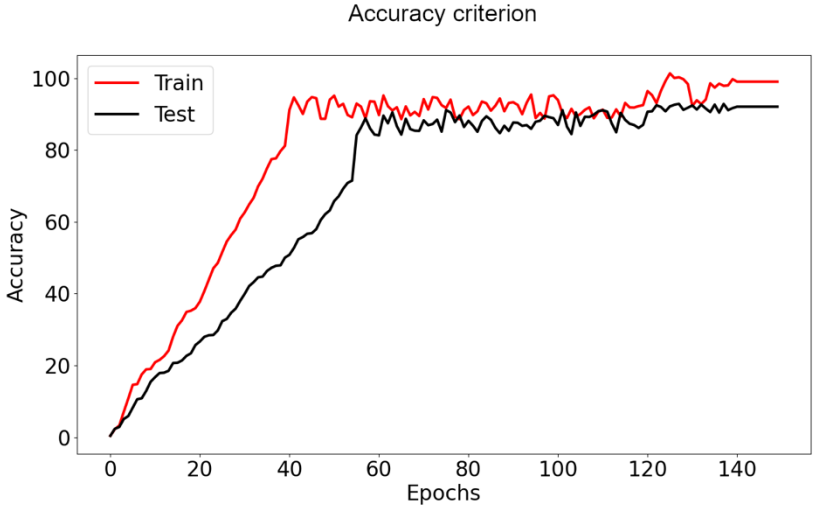


Fig. 6.32. Learning accuracy for the **A3T-GCN** classifier with fuzzification on image input data

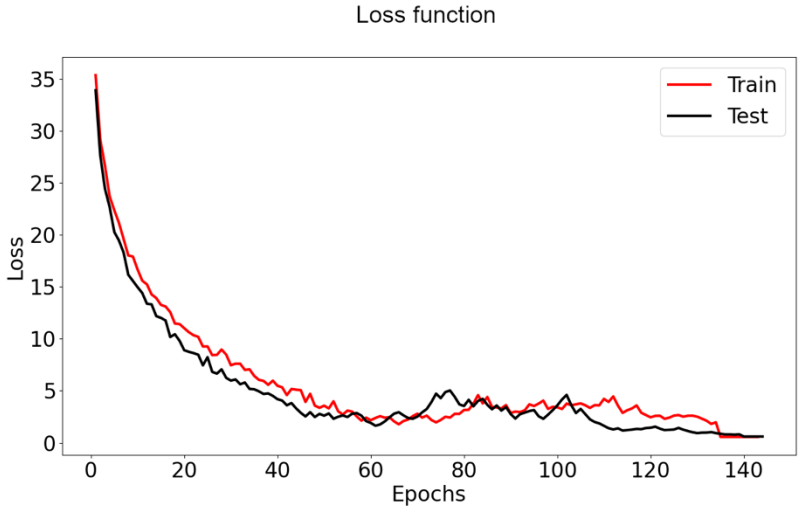


Fig. 6.33. Loss function for the **A3T-GCN** classifier with fuzzification on image input data

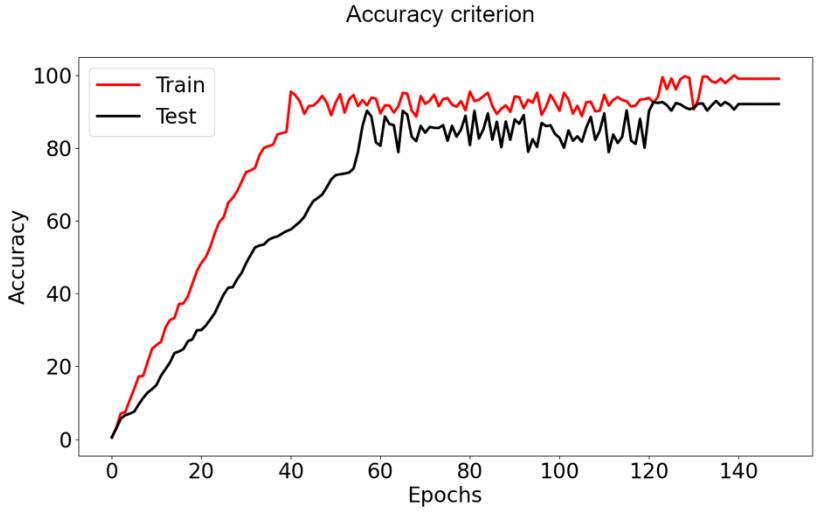


Fig. 6.34. Learning accuracy for the **A3T-GCN** classifier with fuzzification on 3D tennis player model input data

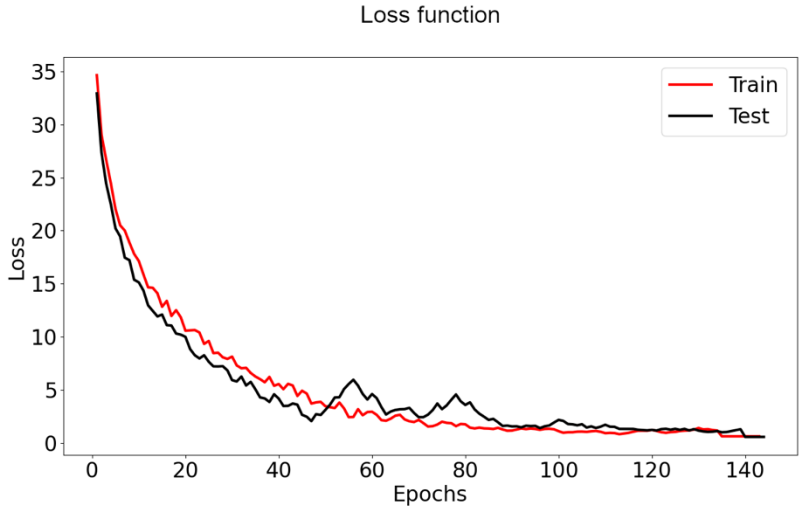


Fig. 6.35. Loss function for the **A3T-GCN** classifier with fuzzification on 3D tennis player model input data

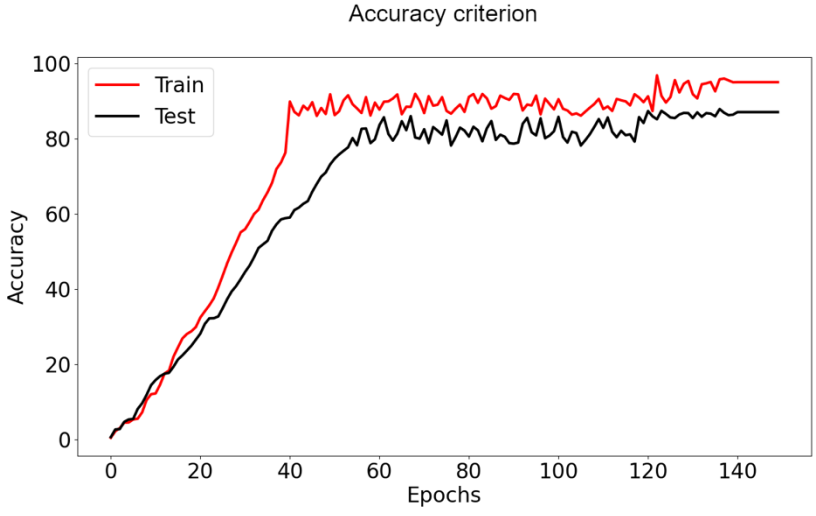


Fig. 6.36. Learning accuracy for the **A3T-GCN** classifier with fuzzification on 3D tennis racket model input data

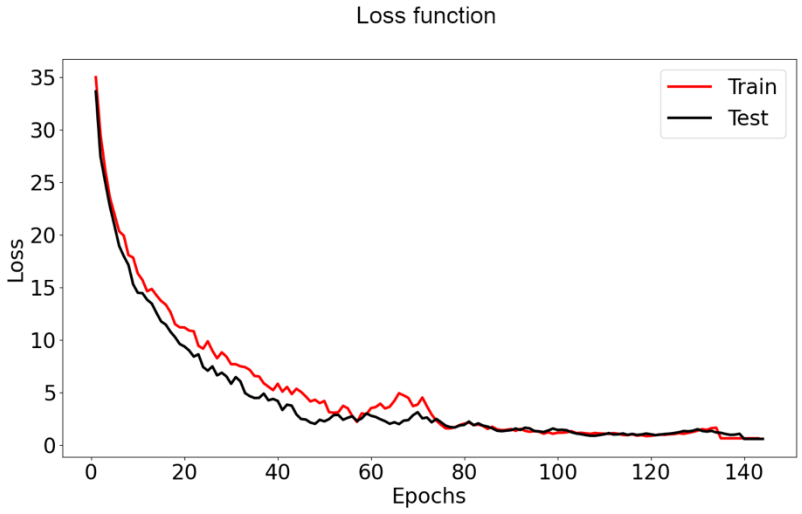


Fig. 6.37. Loss function for the **A3T-GCN** classifier with fuzzification on 3D tennis racket model input data

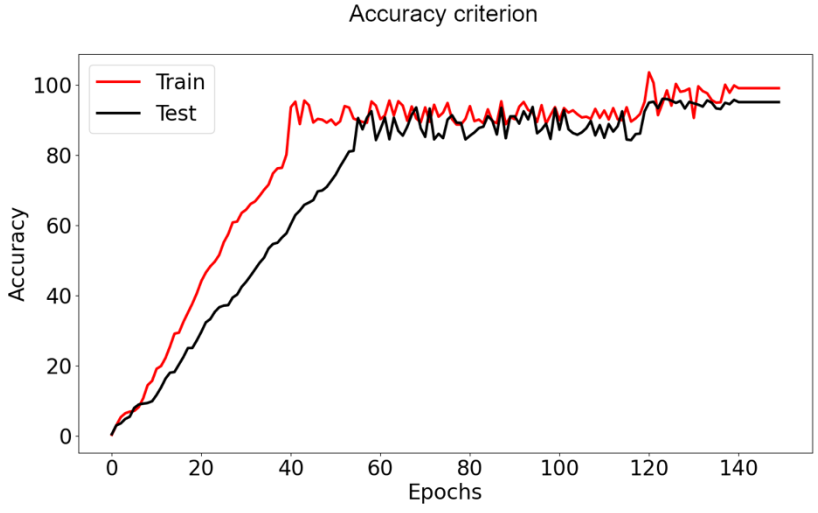


Fig. 6.38. Learning accuracy for the **A3T-GCN** classifier with fuzzification on 3D tennis player model with a tennis racket input data

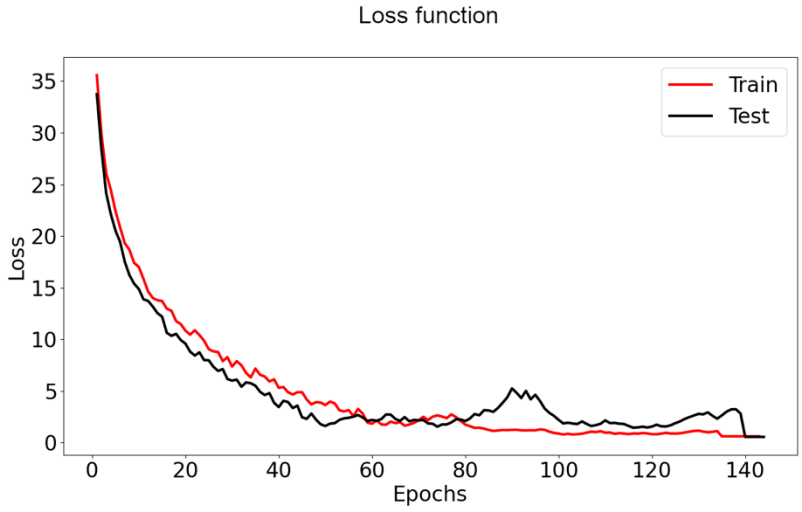


Fig. 6.39. Loss function for the **A3T-GCN** classifier with fuzzification on 3D tennis player model with a tennis racket input data

In comparison to the results obtained utilizing fuzzification of the ST-GCN classifier, one can see that the input data fuzzification process in the case of the A3T-GCN classifier achieved better improvement in the accuracy of the classifier. In the case of the ST-GCN classifier, the highest enhancement occurred for 3D tennis player model data (mean 2.72%). However, the fuzzification process for the A3T-GCN classifier improved its accuracy to the highest extent for the 3D tennis racket model and for the combined 3D tennis player model with a racket. The obtained mean was 3.72% for a racket model and 2.79% for the combined three-dimensional data. Moreover, it should be emphasized that the data fuzzification process improves the quality of classification for both applied classifiers.

Dual Attention Temporal Graph Convolutional Networks for tennis movements recognition

CHAPTER 7

In this chapter, the third new classifier for tennis movement recognition, proposed in this study, is combined with two separate soft attention modules dedicated to measuring the tennis player's body and the tennis racket alignment. The Dual Attention Graph Convolutional Network (DA-GCN) takes into account both spatial and temporal features [97]. It integrates Graph Convolutional Neural Network as well as the Long Short-Term Memory (LSTM) solution. The LSTM network was chosen because of its ability to learn long-term dependencies. The proposed DA-GCN was verified using both image and three-dimensional data as. The data are prepared as it was described in 5.1.1 and 5.2.1 for image and three-dimensional data, respectively.

7.1. Image-based classification

7.1.1. DA-GCN classifier

The architecture of the proposed classier is presented in Fig. 7.1. It consists of four main parts. The first two extract spatial and temporal features, respectively. The third one is formed with two attention modules. The fourth one is the fully connected layers for the classification purposes.

Phases of the tennis stroke in image-based form are the input for the classifier. Based on the tennis player model together with a tennis racket the graph is combined, as described in 5.1.1. Spatial and temporal features are extracted in order to obtain the most accurate prediction model for tennis movement. Spatial features are calculated utilizing the 3-layer GCN (eq. 6.1). Temporal features are extracted using the LSTM network based on three strokes at times $t,\ t_{-1}\ and$

t_{-2} They are extremely important to indicate time dependences for the tennis player model. The input (i_t), output (f_o), and forget (f_t) gates for the LSTM model are defined as [97]:

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \tag{7.1}$$

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \tag{7.2}$$

$$f_o = \sigma(w_o[h_{t-1}, x_t] + b_o) \tag{7.3}$$

where w_i , w_f *and* w_o denotes weights for input, output, and forget gates, respectively. The biases for these three gates are represented by b_i, b_f *and* b_o. The element h_{t-1} stands for the result of the earlier LSTM block at time t–1, while x_t for input at the present time. σ denotes the sigmoidal function.

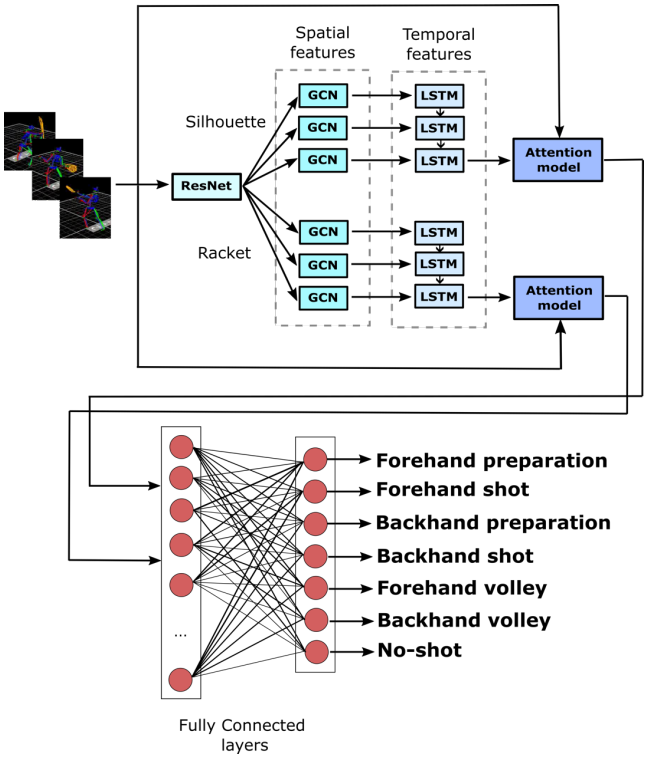


Fig. 7.1. The **DA-GCN** architecture for image-based input [97]

The sigmoidal function decides whether or how much information should be passed through. A zero value means that no information is passed through, while the number one value means that all information is important and passed through.

The gate states are calculated according to eq. 7.4–7.6 [97]:

$$\widetilde{c_t} = tanh(w_c[h_{t-1}, x_c] + b_c) \tag{7.4}$$

$$c_t = f_t * c_{t-1} + i_t * \widetilde{c_t} \tag{7.5}$$

$$h_t = f_o * tanh(c_t) \tag{7.6}$$

where c_t is the gate state at time t, $\widetilde{c_t}$ denotes the pretender for the gate state at time t and h_t is the output.

In the DA-GCN solution two attention modules are implemented for extracting features in order to obtain contextual information. They compute the feature attention matrices based on a tennis racket and a tennis player model. These data are necessary to indicate and predict their future positions. As it is shown in Fig. 7.2, the input to the module has the form of three-dimensional data.

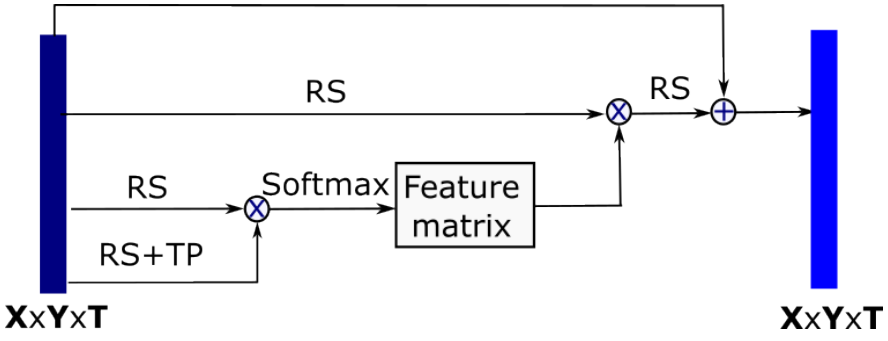


Fig. 7.2. Attention module schema. RS – matrix reshaping, TP – matrix transposition [97]

They are further transformed using a reshaping matrix (RS) and a transposition (TP). The feature matrix is calculated utilizing the Softmax function. Obtained values are multiplied by input parameters, and furthermore, the results are multiplied by a scale parameter.

Output values are calculated using a sum operation. The feature map is calculated using eq. 7.7.

$$O_n = \beta \sum_{k=1}^{X} (i_{nk} I_k) + I_n \tag{7.7}$$

where O denotes the output value, i_{nk} is the impact of the k^{th} feature, I_{n} represents input parameters, and β is a weight parameter.

7.1.2. Tennis strokes recognition

The methodology of the study is the same as described in 5.1.3. The performance of the DA-GCN was verified using accuracy, precision, recall, and F1 score measures (eq. 5.6–5.9).

The mean accuracy for the DA-GCN classifier for all classes is presented in Table 7.1.

Table 7.1. Obtained accuracy results for the **DA-GCN** classifier for image-based for all classes

Mean	*Max*	*Min*	*±SD*
87.09%	92.00%	82.30%	6.39%

The DA-GCN classifier with image-based input reached the highest accuracy values for tennis movements data among all proposed neural network solutions. This neural network model improved the classification process for mean accuracy by 5.52% and 0.97% according to the ST-GCN and A3T-GCN, respectively. It should be emphasized that this classifier, as the only one from all analyzed solutions, that reached a maximum accuracy exceeding 90%.

The detailed accuracy results, for all classes, for image-based input data represented by a tennis player model together with a tennis racket are presented in Table 7.2. The highest accuracy values were obtained for both types of volley strokes, reaching 90.55% and 90.44%, respectively. Slightly lower results, exceeding 89% were reached for the forehand and backhand hit together with racket swinging. Accuracy for no-stroke was 88.27%. The lowest values, however, were higher than 84% and were obtained for the forehand and backhand preparation phases. Obtained maximal accuracy values were higher than 90%.

Table 7.2. Obtained accuracy results for the **DA-GCN** classifier for image-based input

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	85.77%	90.90%	82.78%	6.48%
FHS	89.90%	92.00%	83.32%	6.55%
BHP	84.91%	91.90%	82.60%	9.54%
BHS	89.44%	91.20%	83.00%	6.67%
VFH	90.55%	91.70%	82.41%	6.28%
VBH	90.44%	91.79%	82.30%	8.87%
NST	88.27%	91.01%	83.82%	5.23%

Precision results for the image-based DA-GCN classifier were gathered in Table 7.3. Obtained values indicated that the proposed model is highly reliable in classifying samples as positive. The highest values, exceeding 90%, were obtained for the backhand strokes and for both types of volleys. The precision for no-stroke reached 88.98%. The lowest value, however, although it was higher than 85%, was calculated for the forehand moves.

Table 7.3. Obtained precision results for the **DA-GCN** classifier for image-based input

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	86.63%	92.50%	82.59%	9.42%
FHS	85.80%	91.94%	81.55%	8.26%
BHP	90.47%	94.44%	87.58%	7.71%
BHS	90.05%	94.37%	86.97%	7.15%
VFH	90.95%	94.37%	88.77%	6.55%
VBH	90.97%	94.47%	88.48%	7.68%
NST	88.98%	93.69%	85.63%	7.17%

Precision results are up to 8.58% higher in comparison to the ST-GCN model and up to 2.16% compared to those in the A3T-GCN model.

Recall results for the image-based DA-GCN classifier are shown in Table 7.4. The ability to detect the positive samples ranges from 86.53% to 94.44%. The highest values were obtained for the forehand moves. Slightly lower results were gathered for all other moves. The lowest results were for the volley forehand stroke, reaching 86.09%.

Recall results were up to 3.84% higher in comparison to the ST-GCN classifier and up to 1.85% in comparison to the A3T-GCN model.

Table 7.4. Obtained recall results for the **DA-GCN** classifier for image-based input

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	94.44%	96.77%	92.64%	8.78%
FHS	89.45%	93.92%	86.40%	5.75%
BHP	88.05%	92.95%	84.48%	9.98%
BHS	88.26%	93.46%	84.72%	9.36%
VFH	86.53%	91.54%	83.06%	9.85%
VBH	88.09%	92.72%	85.10%	8.75%
NST	88.71%	93.69%	85.14%	9.75%

Table 7.5. Obtained F1 scores for the **DA-GCN** classifier for image-based input

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	90.35%	94.59%	87.33%	6.61%
FHS	87.59%	92.92%	83.91%	6.48%
BHP	89.24%	93.69%	86.01%	6.27%
BHS	89.14%	93.91%	85.83%	7.24%
VFH	88.68%	92.93%	85.82%	7.69%
VBH	89.51%	93.58%	86.76%	7.68%
NST	88.84%	93.69%	85.38%	9.01%

The F1 score results for the image-based DA-GCN classifier are presented in Table 7.5. Values were in the range between 88.68% for no-stroke and 90.35% for the forehand preparation phase.

F1 scores were up to 3.95% higher in comparison to the ST-GCN model and up to 1.95% higher in comparison to the A3T-GCN classifier.

The confusion matrix for the DA-GCN for the image-based input is depicted in Fig. 7.3. Maximum misclassification was 2.69% for the forehand hit phase. The forehand preparation phase was confused with the second phase of this move and the backhand preparation phase. The forehand hit with swinging was confused with the forehand preparation phase. Both phases of backhands phases were also confused. The volley forehand was mistaken for no-stroke. The volley backhand was misclassified with both phases of the forehand and backhand strokes. No-stroke was confused with the forehand hitting and the backhand preparation phase.

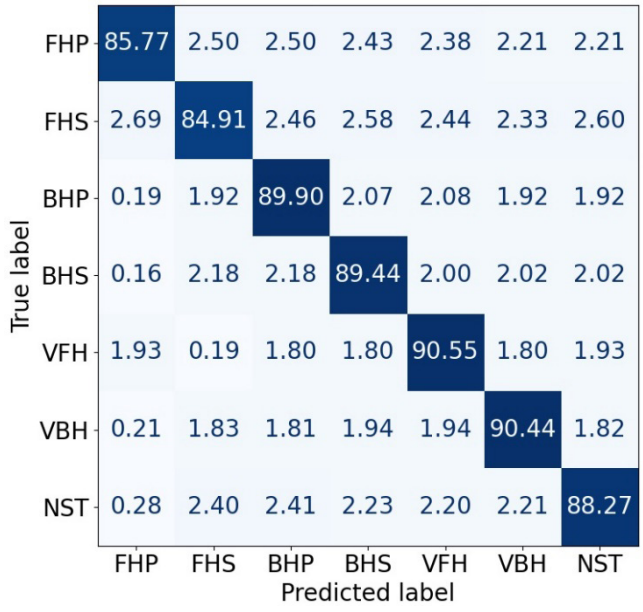


Fig. 7.3. Confusion matrix in (%) for the tennis movement **DA-GCN** classification for image-based input data: FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

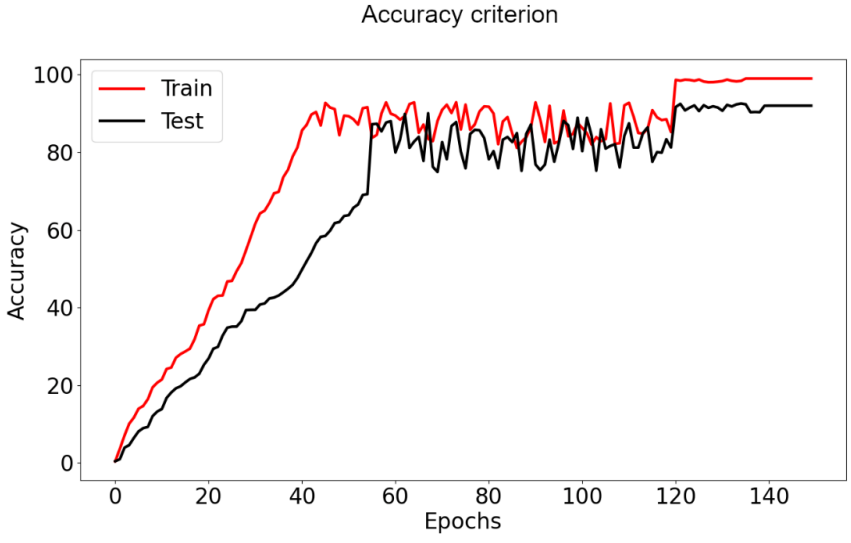


Fig. 7.4. Learning accuracy for the **DA-GCN** classifier for image-based input data

Learning accuracy and loss function are presented in Fig. 7.4 and Fig. 7.5. As in the case of the ST-GCN and A3T-GCN 140 epochs are enough to obtain a stable model.

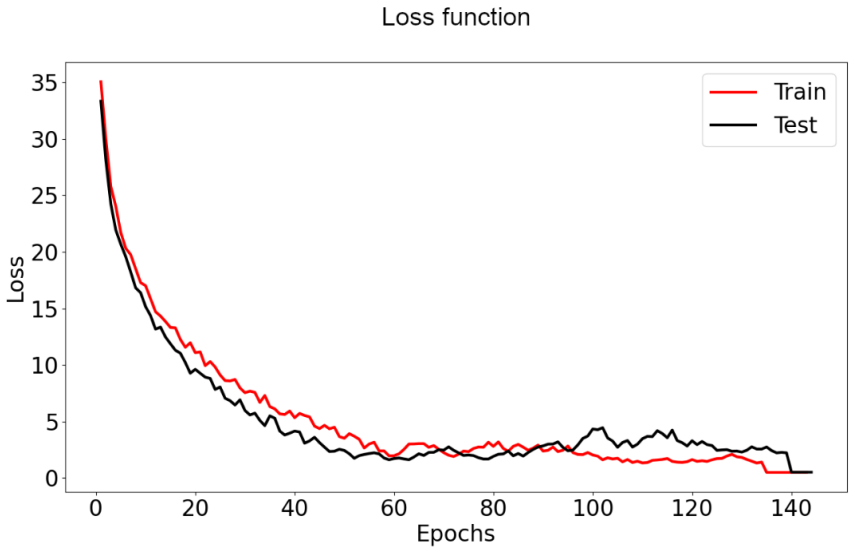


Fig. 7.5. Loss function for the **DA-GCN** classifier for image-based input data

7.2. 3D-based classification

7.2.1. DA-GCN classifier

The structure of the DA-GCN classifier for three-dimensional data is depicted in Fig. 6.5. In comparison to Fig. 7.1, the type of input was changed from 2D to 3D. The input data so structured is the same as that described in 5.2.1. It must be emphasized that from the input data the graph was created based on markers.

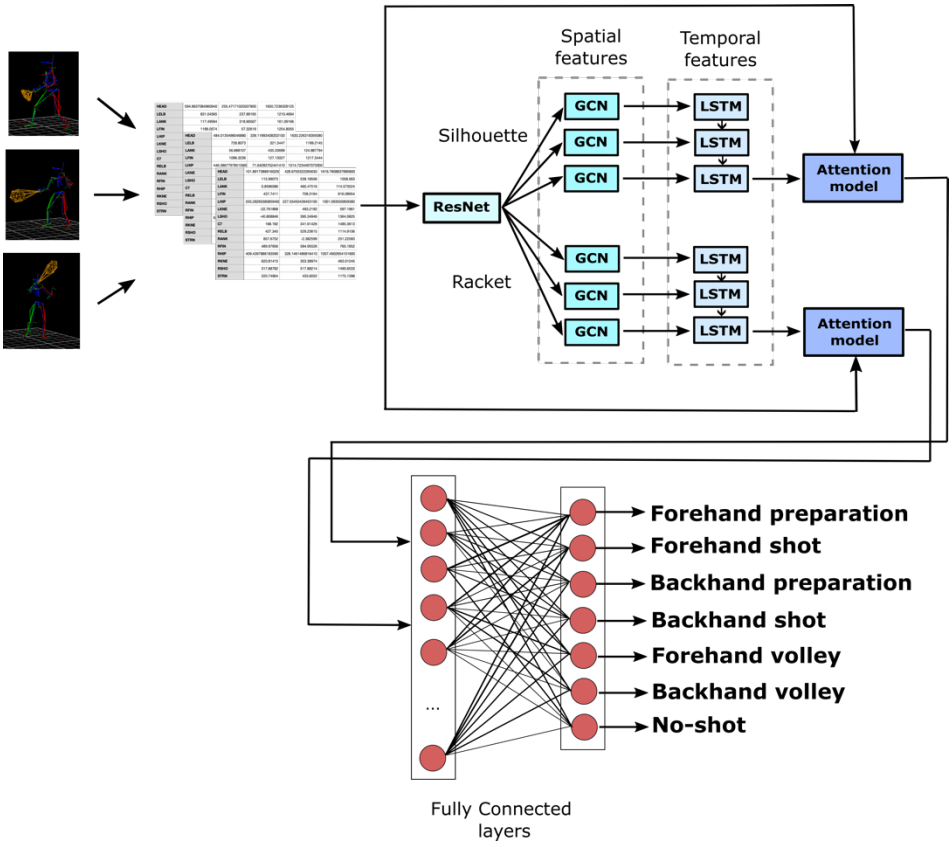


Fig. 7.6. The **DA-GCN** architecture for 3D-based input [97]

7.2.2. Tennis strokes recognition based on combined tennis player with a racket model

Due to the specific nature of the network architecture, namely involving two attention modules, studies were carried out exclusively with data containing both the tennis player model and a tennis racket model. This method was the same as that described in 5.1.3. Each data consists of 39 three-dimensional points corresponding to the tennis player and 7 three-dimensional points corresponding to a tennis racket. The performance of the DA-GCN was verified using accuracy, precision, recall, and F1 score measures (eq. 5.6–5.9).

Accuracy results for the DA-GCN model are presented in Table 7.6. The model reached very high performance results, exceeding 6.24% mean accuracy as for the same model dedicated to image-based input.

The DA-GCN for three-dimensional input data obtained the highest accuracy in comparison to other analyzed neural networks. Obtained average results are 9.81% and 3.88% higher than the ST-GCN and A3T-GCN values, respectively.

Table 7.6. Obtained accuracy results for the **DA-GCN** classifier for 3D tennis player model and tennis racket model for all classes

Mean	*Max*	*Min*	*±SD*
93.33%	97.87%	89.37%	2.76%

Detailed accuracy results, for all classes, for 3D-based input data represented by a tennis player model together with a tennis racket are presented in Table 7.7. Highest accuracy values were obtained for backhand moves and both types of volley strokes, exceeding 94%. Slightly lower results, however exceeding 93%, were reached for no-stroke. The lowest values, however higher than 91%, were obtained for the two phases of the forehand moves. Obtained maximal accuracy values exceeded 97%.

Accuracy results are up to 11.36% and 4.88% higher compared to the same input data for the ST-GCN and A3T-GCN, respectively. The DA-GCN with three-dimensional input data also improved the accuracy up to 9.30% compared with the same classifier for the same input data.

Table 7.7. Obtained accuracy results for the **DA-GCN** classifier for 3D tennis player model and a tennis racket model

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	92.11%	97.00%	89.39%	3.75%
FHS	91.65%	96.75%	89.42%	3.81%
BHP	94.21%	96.88%	89.70%	2.46%
BHS	94.13%	96.91%	89.48%	2.63%
VFH	94.01%	96.53%	89.37%	3.71%
VBH	94.28%	96.56%	89.78%	2.85%
NST	93.51%	96.95%	89.81%	3.81%

Precision results for 3D-based input DA-GCN classifier were gathered in Table 7.8. Obtained values indicated that the proposed model is highly reliable in classifying samples as positive. The highest values,

exceeding 95% were obtained for backhand strokes and for both types of volleys. The precision for no-stroke reached 94.90%. The lowest value, however, higher than 93%, was calculated for the forehand moves.

Precision results were up to 10.51% and 4.15% higher, compared to with the ST-GCN and A3T-GCN for the same input data, respectively. The analyzed DA-GCN with 3D input data improved precision up to 7.82% in comparison to the image-based input data.

Recall results for three-dimensional input data of the DA-GCN classifier are shown in Table 7.9. The ability to detect the positive samples is very high, ranging from 93.11% to 97.28%. Highest values were obtained for the forehand moves. Slightly lower results were gathered for backhand strokes, both types of volleys, and no-stroke.

Table 7.8. Obtained precision results for the **DA-GCN** classifier for 3D tennis player model and a tennis racket model

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	93.96%	97.50%	89.79%	2.43%
FHS	93.62%	97.61%	89.37%	4.06%
BHP	95.45%	97.93%	92.91%	4.82%
BHS	95.39%	97.85%	92.50%	5.48%
VFH	95.39%	97.91%	92.89%	5.26%
VBH	95.34%	97.81%	92.86%	4.15%
NST	94.90%	97.72%	91.66%	3.05%

Obtained recall results were up to 8.66% and 3.64% higher in comparison to the ST-GCN classifier and A3T-GCN models, respectively. For the three-dimensional input data, the DA-GCN improved the recall values up to 6.58% compared to image-based input.

Table 7.9. Obtained recall results for the **DA-GCN** classifier for 3D tennis player model and a tennis racket model

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	97.28%	99.19%	95.44%	3.13%
FHS	95.10%	97.84%	92.14%	7.35%
BHP	94.29%	97.35%	91.02%	4.87%
BHS	94.62%	97.54%	91.19%	3.34%
VFH	93.11%	96.81%	89.18%	2.78%
VBH	94.04%	97.17%	90.63%	4.74%
NST	94.96%	97.89%	91.60%	3.55%

Obtained F1 score results for three-dimensional input data of the DA-GCN are presented in Table 7.10. The classifier reached a very high level of performance. All the analyzed tennis moves gained values between 94.23% and 95.58%. These data are up to 9.13% and 3.5% higher than in the ST-GCN classifier and A3T-GCN models, respectively. For the three-dimensional input data, the DA-GCN improved the F1 scores up to 6.76% compared with image-based input.

Table 7.10. Obtained F1 results for the **DA-GCN** classifier for 3D tennis player model and a tennis racket model

Class	*Mean*	*Max*	*Min*	*±SD*
FHP	95.58%	98.34%	92.55%	3.37%
FHS	94.35%	97.73%	90.80%	2.73%
BHP	94.86%	97.64%	91.96%	2.68%
BHS	95.00%	97.69%	91.84%	3.92%
VFH	94.23%	97.36%	91.03%	4.07%
VBH	94.68%	97.49%	91.73%	3.41%
NST	94.93%	97.80%	91.63%	3.83%

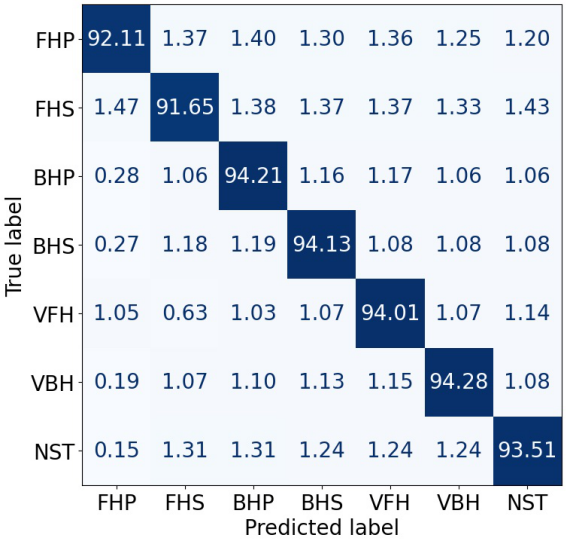


Fig. 7.7. Confusion matrix in (%) for the tennis movement **DA-GCN** classification for 3D tennis player model and a tennis racket model: FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

The confusion matrix for the DA-GCN for the 3D-based input is presented in Table 7.10. Maximum misclassification is very low and amounts to 2.32%. The most often forehand preparation phase was confused with the second phase of this move and the backhand preparation phase. The next move, the forehand hit with racket swinging was misclassified with the forehand preparation phase. Both phases of the backhands were also confused. Additionally, the backhand hit was mainly confused with the forehand hit. The volley forehand was misclassified with the forehand preparation phase and no-stroke was the most often confused. The volley backhand was confused at the same level as the other type of volley and backhand hit. Finally, no-stroke was misclassified with the forehand hit together with racket swinging.

Learning accuracy and loss function are presented in Fig. 7.8 and Fig. 7.9. As in the case of the ST-GCN and A3T-GCN 140 epochs were enough to obtain a stable model. The same situation occurred in the case of the DA-GCN for image-based input data.

The proposed DA-GCN model appeared to be the most suitable for tennis motion data. Obtained results clearly indicated that the highest network performance was gained for 3D-based input data that were represented as three-dimensional points of a tennis player model together with a tennis racket model.

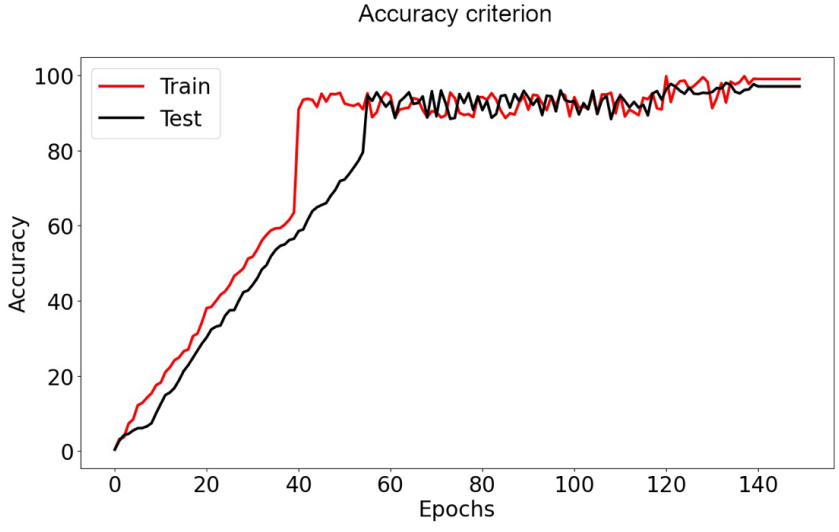


Fig. 7.8. Learning accuracy for the **DA-GCN** classifier for 3D tennis player model and a tennis racket model

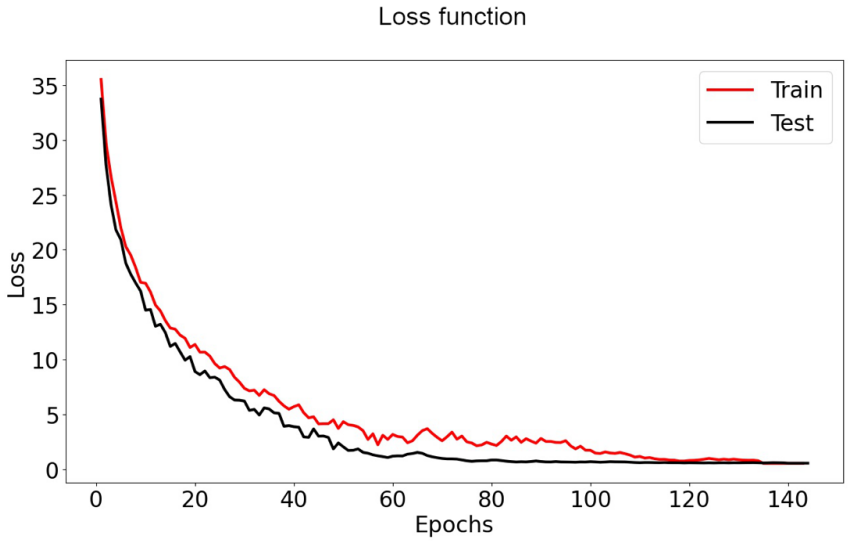


Fig. 7.9. Loss function for the **DA-GCN** classifier for 3D tennis player model and a tennis racket model

7.3. Classification utilizing fuzzification of input data

Fuzzification of input data was also applied to the DA-GCN classifier. This process was the same as that described in 5.3 section.

Results for accuracy for all classes for the DA-GCN with fuzzification of input are presented in Table 7.11. The fuzzification process improved the DA-GCN accuracy by 5.55% and by 4.71% for image-based input and 3D-based input, respectively. It must be stated that accuracy for 3D-based input data ranged between 91.62% and 95.47%, which means that these results exceeded the 90% level.

Table 7.11. Obtained accuracy results for the **DA-GCN** classifier for input fuzzification for successive classes

Input type	*Mean*	*Max*	*Min*	*±SD*
image	92.64%	94.99%	87.81%	4.51%
3D tennis player model with a tennis racket	95.47%	99.01%	91.62%	3.81%

Results for accuracy for the successive classes for the DA-GCN with fuzzification of various input types are presented in Fig. 7.10 and

Fig. 7.11. Applying the fuzzification process improved classification accuracy for all input types for all classes.

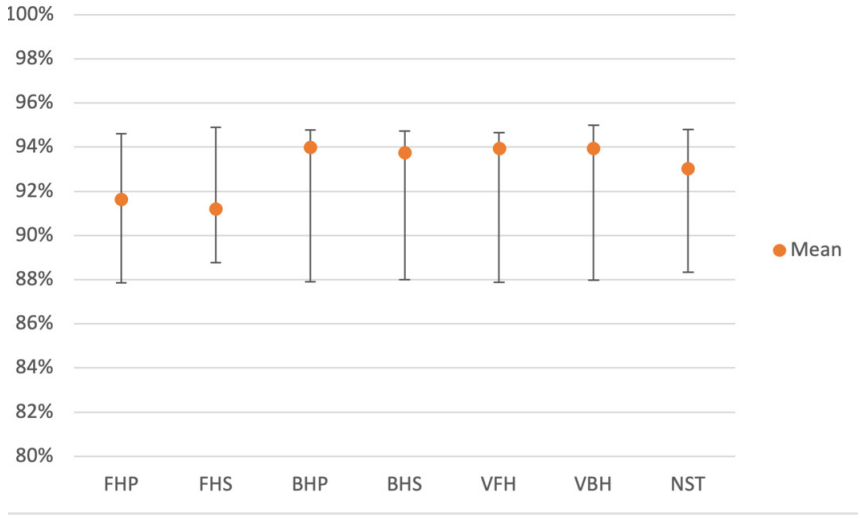


Fig. 7.10. Obtained accuracy results for the **DA-GCN** classifier for image input fuzzification

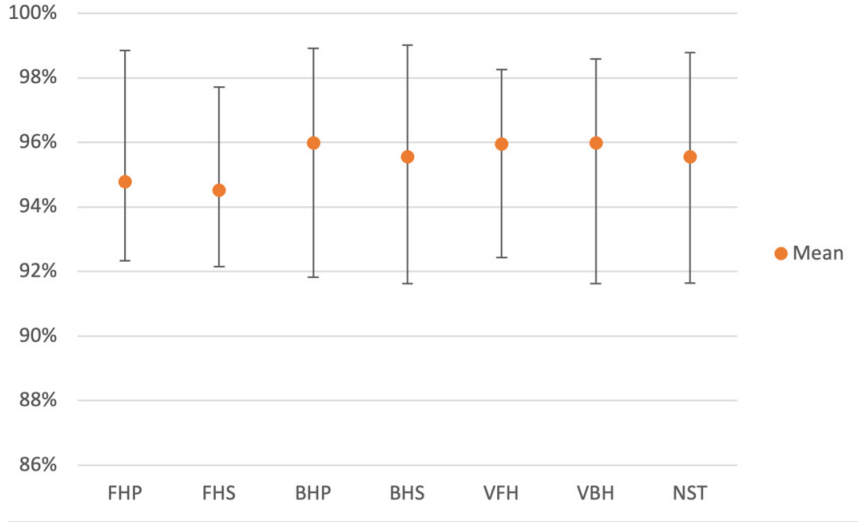


Fig. 7.11. Obtained accuracy results for the **DA-GCN** classifier for 3D tennis player and racket model input fuzzification

The mean accuracy results for 3D-based input were higher than for image-based input. Moreover, the fuzzification process improved

the accuracy results in comparison to the DA-GCN without applying fuzzification to the input.

In the case of image-based input, the mean accuracy was improved for all classes, up to 9.07%. The highest increase was obtained for the backhand preparation phase. Slightly lower results were for the forehand preparation phase. Accuracy improvements at similar level, between 3.39% and 4.75%, were achieved for no-stroke, the backhand hit, two types of volleys, starting from the highest values. The lowest improvement, 1.29%, was noticed for the forehand hit with the racket swinging.

For 3D-based input, represented as a tennis player model together with a tennis racket model, the lower enhancement was observed, up to 2.87%. Highest improvements were obtained for both phases of the forehand stroke. Improved accuracy exceeding 2% was also noticed for the no-stroke move. The backhand and volleys strokes were improved between 1.42% and 1.94%.

It should be emphasized that for the DA-GCN the highest enhancement was achieved in comparison to the ST-GCN and A3T-GCN models.

Precision results for the successive classes for the DA-GCN with fuzzification of various input types are presented in Fig. 7.12 and Fig. 7.13. Their values are very high, exceeding 91%.

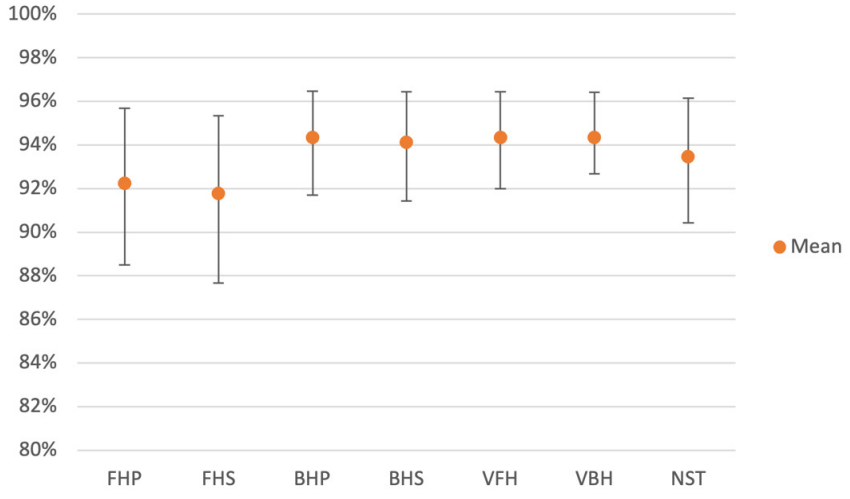


Fig. 7.12. Obtained precision results for the **DA-GCN** classifier for image input fuzzification

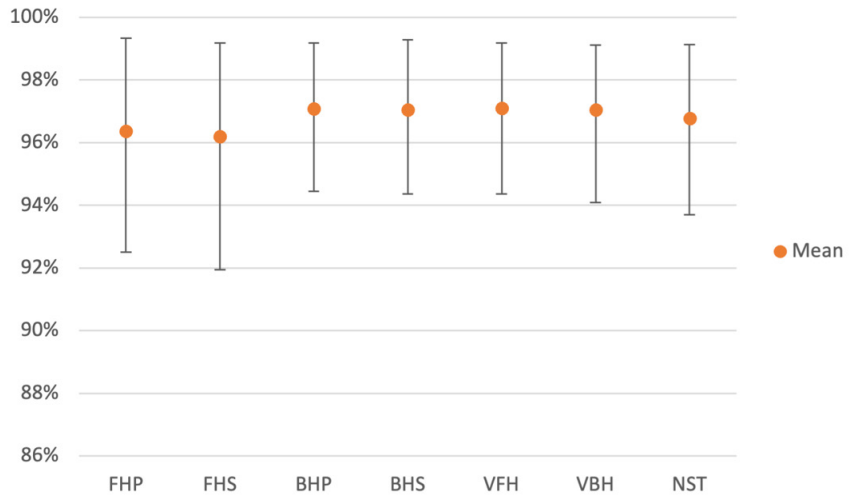


Fig. 7.13. Obtained precision results for the **DA-GCN** classifier for 3D tennis player and racket model input fuzzification

In the case of image-based input the mean precision improvement after applying the fuzzification process for the DA-GCN was reached between 3.38% and 5.97%. The highest enhancement was achieved for the forehand preparation and hit strokes, reaching 5.62% and 5.97%, respectively. No-stroke obtained refinement at the level of 4.48%. The improvements for the backhand hit and the preparation phase were slightly lower, reaching 4.08% and 3.86%, respectively. Both types of volley moves achieved comparable results, gaining 3.38% and 3.39% improvements.

Regarding 3D-based input data for the DA-GCN, the enhancement was twice lower compared to the image-based input data. However, the highest improvement was reached for the forehand hit and the preparation phase, gaining 2.57% and 2.39%, respectively. No-stroke achieved a refinement at the level of 1.86%, while both types of volleys obtained 1.69%. The backhand hit and the preparation phase reached 1.69% and 1.64%, respectively.

Precision improvements were the highest comparing the ST-GCN and A3T-GCN models.

Recall results for the successive classes for the DA-GCN with fuzzification of various input types are presented in Fig. 7.14 and Fig. 7.15. The values are also very high, exceeding 91% and 95% for image-based and 3D-based input data, respectively. It can be observed

that the DA-GCN classifier for 3D-based input has a higher ability to classify the positive samples than does the DA-GCN network for image-based input.

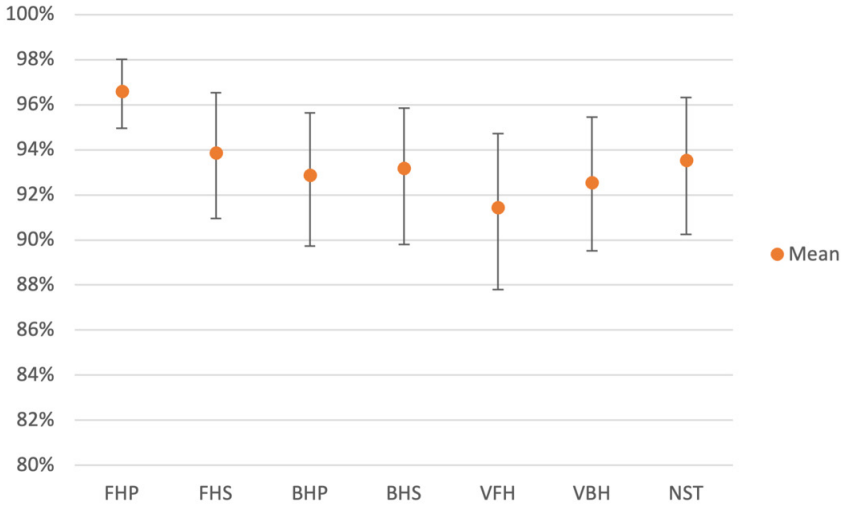


Fig. 7.14. Obtained recall results for the **DA-GCN** classifier for image input fuzzification

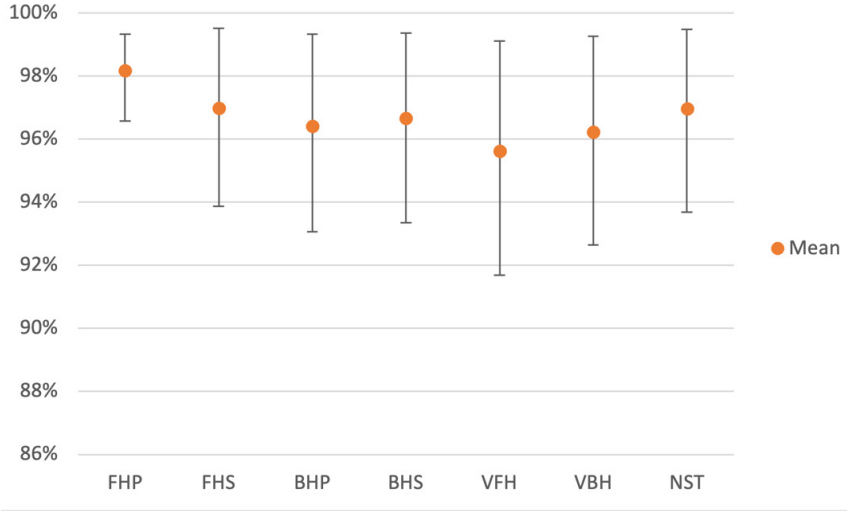


Fig. 7.15. Obtained recall results for the **DA-GCN** classifier for 3D tennis player and racket model input fuzzification

The fuzzification process, as in the case for the ST-GCN and A3T-GCN, improved the recall results. Regarding the DA-GCN image-based input, the enhancement reached up to 4.90%. The lowest

improvement, equal to 2.15%, was obtained for the forehand preparation phase. The second phase, the forehand hit together with racket swinging, reached a 4.40% refinement. A slightly higher enhancement, 4.43%, was gained for the volley backhand stroke. The remaining moves had comparable improvements between 4.80% and 4.90%. Highest results were gathered for the backhand hit with racket swinging and for the volley forehand.

In the case of the DA-GCN 3D-based input, the enhancement was almost twice lower, compared to the image-based input, reaching up to 2.51%. Least improvements were obtained for the forehand stroke at a level of 0.89% and 1.87% for the preparation phase and hit together racket swinging, respectively. The remaining tennis moves had similar improvements, ranging between 2.00% and 2.51%. Highest results were obtained for the volley forehand.

Even though improvements after applying fuzzification were lower than those obtained after the process for the ST-GCN and DA-GCN, they were less variable than in other classifiers. However, final recall results for the DA-GCN were the highest compared with those in other proposed neural networks.

F1 score results for the successive classes for the DA-GCN classifier after applying the fuzzification process are gathered in Fig. 7.16 and Fig. 7.17. For image-based input the values ranged between 92.80% and 94.36%. For three-dimensional data they exceeded 96.30%, which proves that this type of input ensures higher classifier accuracy in comparison to image-based input.

The fuzzification process clearly improved the DA-GCN model accuracy for two types of input. Regarding image-based input, F1 improvements were reached between 3.91% and 5.21%. The highest was obtained for the forehand hit with racket swinging. Slightly lower, equal to 4.64% was for no-stroke. The backhand hit with racket swinging and the preparation phase of this tennis move reached 4.50% and 4.35% enhancement, respectively. The volley forehand and forehand preparation phase gained 4.28% and 4.01% improvements, respectively. The volley backhand was raised to 3.91%.

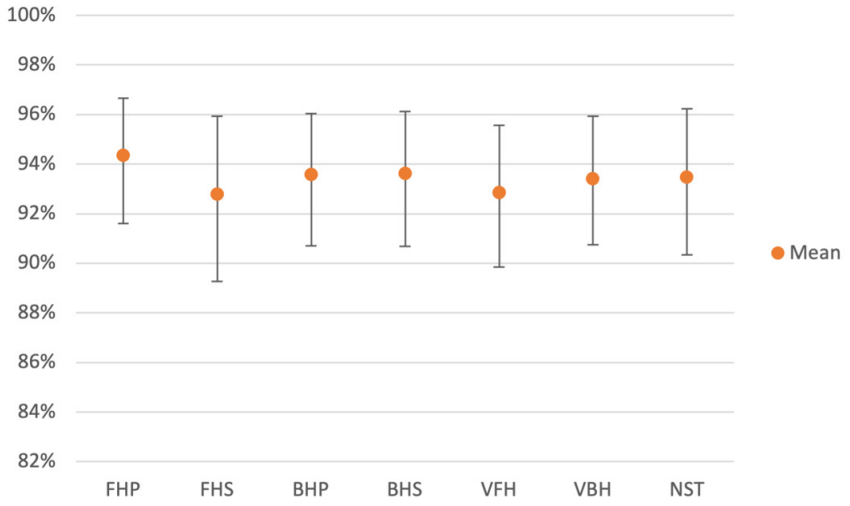


Fig. 7.16. Obtained F1 results for the **DA-GCN** classifier for image input fuzzification

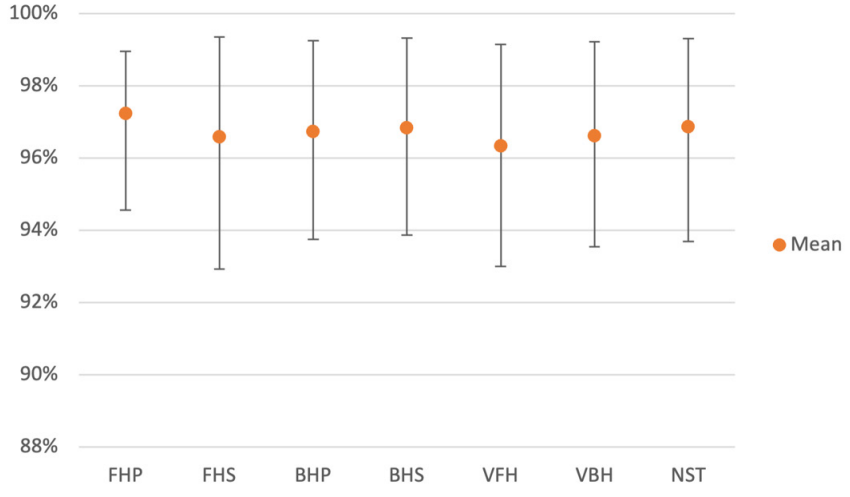


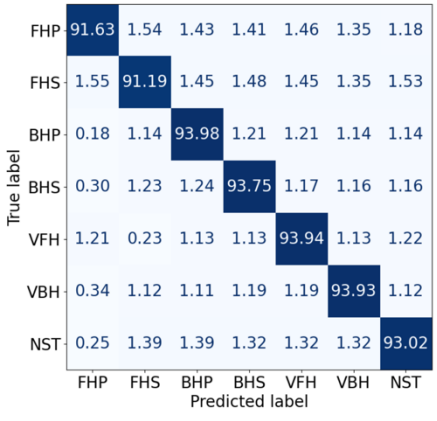
Fig. 7.17. Obtained F1 results for the **DA-GCN** classifier for 3D tennis player and racket model input fuzzification

In the case of the DA-GCN 3D-based input data, the fuzzification process improved F1 results from 1.66% to 2.23%. The highest enhancements, equal to 2.23% and 2.11%, were reached for the forehand hit with racket swinging and the volley forehand, respectively. The volley backhand gained 1.94%, while no-stroke obtained 1.93%. The backhand move, both the preparation phase and the hit, was boosted almost equally, reaching 1.87% and 1.84%, respectively.

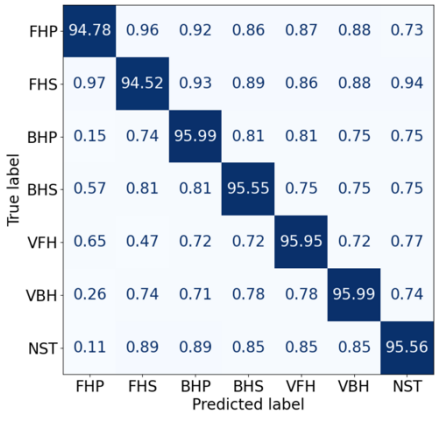
The lowest improvement, 1.66%, was obtained for the forehand preparation phase.

Comparing the obtained F1 results with the fuzzification process for the DA-GCN image-based input with the ST-GCN and A3T-GCN, the highest improvements were clearly reached for the classifier with two attention modules. They were almost twice higher than those obtained in other classifiers. The situation for the fuzzification process is comparable for three-dimensional data but is different. The improvements for all proposed classifiers reached comparable results, slightly exceeding 2%. However, it must be emphasized that F1 results were the highest for the DA-GCN neural network.

Confusion matrices for the DA-GCN classifier with fuzzification for various input data are presented in Fig. 7.18. Similar to results without fuzzification, the same tennis moves are misclassified. However, values of the confused classification slightly decreased for image-based input, which proves that the input fuzzification process improves the effectiveness of tennis movement recognition. For 3D-based input the values of misclassification are similar to the classifier without the fuzzification of the input data.



a) image input



b) 3D tennis player model with a tennis racket model

Fig. 7.18. Confusion matrices (in%) for the tennis movement **DA-GCN** classification with fuzzification: FHP – Forehand Preparation Phase, FHS – Forehand Shot, BHP – Backhand Preparation Phase, BHS – Backhand Shot, VFH – Volley Forehand, VBH – Volley Backhand, NST – No Stroke

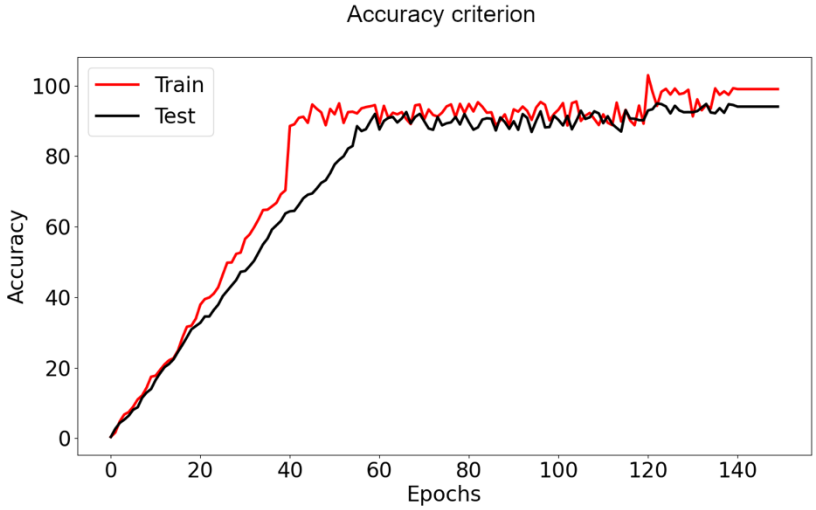


Fig. 7.19. Learning accuracy for the **DA-GCN** classifier with fuzzification for image-based input data

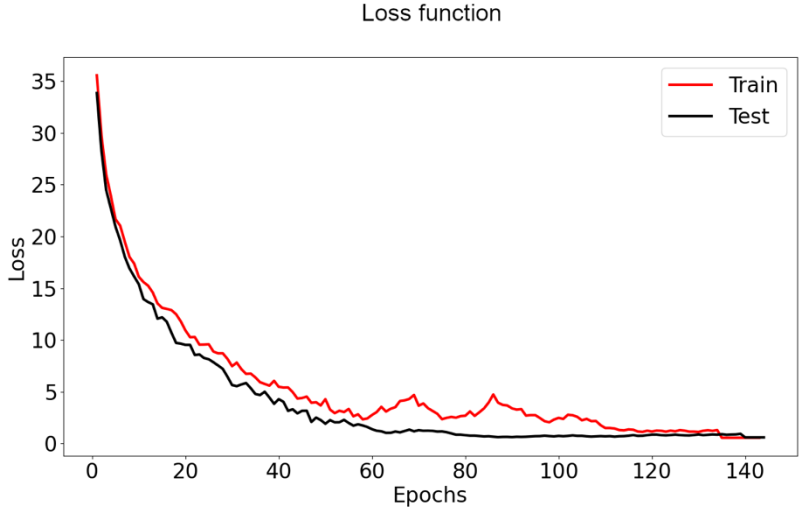


Fig. 7.20. Loss function for the **DA-GCN** classifier with fuzzification for image-based input data

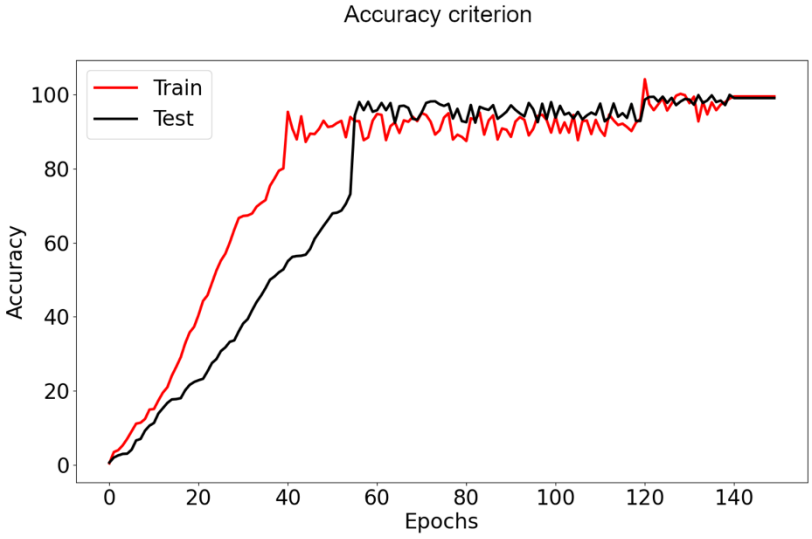


Fig. 7.21. Learning accuracy for the **DA-GCN** classifier with fuzzification for 3D-based input data

Learning accuracy charts for the DA-GCN with fuzzification process for image-based and 3D-based data are presented in Fig. 7.19 and Fig. 7.21, respectively. Both accuracies exceeded 90%. After reaching 140 epochs a stable model was obtained. In the case of 3D-base input data, the training and testing accuracies are similar. However, for image-based input, testing accuracy was lower than the training one.

The loss functions for the DA-GCN for image-based and 3D-based input data are depicted in Fig. 7.20 and Fig. 7.22, respectively. For both types the analyzed inputs, after reaching 140 epochs, minimal values of loss functions were obtained.

The ST-GCN, A3T-GCN and DA-GCN neural network solutions needed 140 epochs to reach minimal loss value and to obtain a stable model.

The fuzzification process of input data improved the effectiveness of all of the proposed classifiers compared to not using fuzzification. Highest improvements were obtained for image-based input. However, highest accuracy, precision, recall, and F1 scores, were gained for 3D-based input data.

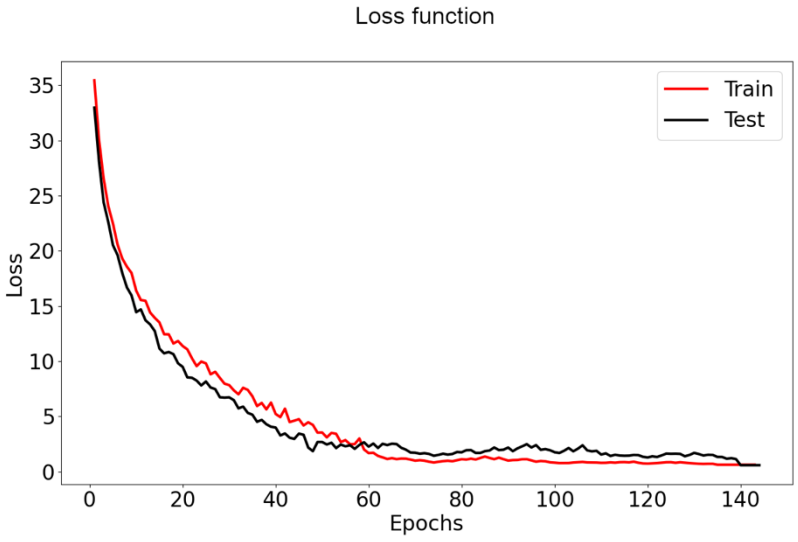


Fig. 7.22. Loss function for the **DA-GCN** classifier with fuzzification for 3D-based input data

CHAPTER

8

Discussion

In this monograph, the challenging problem of tennis movements recognition is tackled. Four basic tennis strokes are included: forehand, backhand, volley forehand, and volley backhand. In order to distinguish strokes from the player's movement on the court, another class is added to the classification reflecting the no-stroke move.

The proper technique in sport is an extremely important aspect. Athletes try to improve their skills so as to perform tennis strokes as much as possible. Their performance affects the ball's trajectory, which allows them to gain points in matches and, consequently, to achieve further successes in their careers [22][98]. The proper execution of the individual phases of the strokes plays a huge role in performance. Motion analysis is widely used both for athletes' performance and to verify how injuries affect their movement [99]. Detection of tennis strokes and their phases allows the coach both to evaluate the technique of the strokes performed and also to optimize training in order to improve the overall performance [26]. Moreover, after injuries it is worth checking whether the strokes performed include all the necessary phases. The performance of strokes phases is a crucial part of monitoring training [22][100]. Motion capture analysis can include an assessment of the movements of the entire tennis player, the tennis player together with the tennis racket, and the racket itself. This allows for analysis of individual phases of the shots taking into account the movement of the body, as well as the arrangement of a tennis racket.

One of the applications of tennis stroke classification is their integration into systems which support learning the game. Tennis stroke detection is a built-in part of these systems. After stroke recognition, a message can be sent to the user, as well as guidance on how to improve the stroke. This solution can be an interesting complement to in-person trainings, which are quite expensive.

Tennis stroke detection is also used in video statistics. For example, the types of strokes performed are counted [13]. The systems also allow for improving the user's reaction to a tennis ball approaching at various speeds [101]. An element of tennis learning systems can also be the animation of the avatar's movement, after the appropriate classification of the stroke or its phase.

This ambitious recognition task utilized three unique approaches, the ST-GCN, A3T-GCN, and DA-GCN, proposed for both image-based and 3D-based input data. All three proposed solutions were based on the Graph Convolutional Networks. Due to the fact that for tennis movements should be recognized based on several frames of the recordings, the temporal features and spatial ones are extracted. It should be emphasized that these approaches are the first applications to tennis movement recognition.

In the literature, there have been several attempts recently to classify tennis movements based on video data, motion capture data and sensor data. Video-based classification has usually been performed using broadcast films and videos from Kinect, and then gathered into the THETIS dataset. Various types of feature extractions and classification methods were applied. In the case of broadcast videos, racket swinging was analyzed, also divided into left and right sides, from the moment the ball was hit. In addition, basic tennis movements were analyzed such as forehand, backhand, volleys, smash and serve, as well as all their varieties, such as flat or slice. Studies also concerned the detection of tennis movements that were not tennis strokes, but which the player performed during a match or during exercises, while moving on a court. Regarding the THETIS dataset, the classification studies concerned up to twelve recorded tennis movements. In video data, a tennis player's body together with a tennis racket were usually taken into consideration. SVM was the most applied classifier, reaching accuracy in the range from 46.48% to 90.21%, depending on the used feature extraction methods [9][10][15][16]. Classification utilizing a generative approach reached 58.72% [13], FDA – 84.50% [12], and CRF – 86.44% [16]. The LSTM approach for THETIS data reached between 43.19% and 62% [17][4].

In the case of data recorded using passive motion capture systems, only basic tennis strokes were analyzed, such as forehand, backhand, and serve, without their varieties. The classification accuracy results were higher in comparison to video data, reaching 82.14%. For the HMM and

the RBF methods the recognition accuracy was obtained between 84,50% and 94.60% [20][19]. The recognition of forehand, backhand as well as no-shot obtained the mean accuracy between 92.57% and 95.88% for Inception V-3 and in range 95.11% and 95.89% for MobileNet [102].

A great number of studies concerning tennis movements classification use sensor data, such as the SensorTile, IMU and PIQ Robot. Sensors are attached to a tennis racket, to the hand in which the tennis racket was held and to the player's body. In studies which applied one SensorTile, the following classification methods were applied: FCN, ResNet, NN, DT, RF, *k*-NN, and SVM. Accuracy results were the highest in comparison to other data types, reaching between 93.89% and 99.72%, depending on the chosen classifier [22][26]. In tennis movement classification based on IMU sensors, SVM, cubic kernel SVM, *k*-NN, and the longest common subsequence algorithm were the most often applied. For this type of data the accuracies were obtained between 82.43% and 97.40% [23][24][28]. Regarding the JY-6 sensor and the PIQ Robot, acceleration fluctuations as well as LVQ were implemented, reaching 90.94% and up to 90%, respectively [25][27].

Unlike many studies, where wearable sensors were attached to a racket or to a hitting arm, in [103], a GPS unit was placed on a player's body. This provided a record of movement patterns as players moved during matches. Various types of forehand and backhand strokes were analyzed [100]. The results showed that the highest accuracy was reached for forehand drive (95%) and forehand block (75%). Similar results were obtained for the second groundstroke, where the backhand drive and slice reached 96% and 74%, respectively. The study showed that wearing a device attached to a player's body is not sufficient for detecting volley strokes. The accuracy results were lower than 45%. On the contrary, this sensor data were enough to recognize the following player movements: alert, dynamic, running, and low Intensity.

In this monograph, it has been shown how the type of data given as input to the classifier affects its quality. Four input types were taken into consideration: image-based obtained from video data, 3D-based of a tennis racket model, 3D-based of a tennis player model, and finally 3D-based combined data of a tennis player model holding a tennis racket. All data were obtained from passive motion capture system recordings. Three various types of GCN approaches were applied for the forehand, backhand, volley forehand, volley backhand moves

and no-stroke. Additionally, the forehand and backhand strokes were divided into two phases, the preparation one, and the hitting together with racket swinging, both of which were also classified. This aspect distinguishes the presented results from studies conducted on the basis of motion capture data. In the case of the ST-GCN, accuracy results for image-based input ranged between 77.11% and 84.70%, while for 3D-based data between 70.21% and 88.91%. For the A3T-GCN, accuracy values ranged between 80.10% and 88.70% for image-based data between 77.19% and 93.23% for 3D-based data, respectively. Regarding the DA-GCN, the results were the highest, between 82.30% and 92% for image-based data and between 89.37% and 97.87% for 3D-based combined data. Obtained results are much higher than for THETIS data and broadcast videos, slightly higher for motion data, and comparable to classifications utilizing sensor data. Taking into consideration the obtained results for the ST-GCN, A3T-GCN, and DA-GCN, it can be definitely stated that Graph Convolutional Neural networks are suitable tools for tennis movement recognition based on motion data. These types of neural networks are adequate both for the tennis player model and for the tennis racket model obtained from the Vicon motion capture system. The player model consisted of 39 specific, interconnected points that determine the shape and structure of the player's body [104]. On the other hand, on the tennis racket the characteristic points, also connected to each other, were selected in order to represent its shape and arrangement [105].

Another important issue is to study in detail how the input data types influence recognition performance. In this monograph only motion capture data are analyzed. They are represented as 2D images, presenting a tennis player and a racket models obtained from the Vicon Nexus software, and presented as 3D data. They are specified as points represented only on the tennis player model or on the tennis racket model and finally on the combined player and racket model.

Based on the accuracy results obtained for four various types of input data, the following similarity can be noticed for the ST-GCN and A3T-GCN. The lowest accuracy values are obtained for data represented by a tennis racket model (70.21–81.91% and 77.19–84.99% for the ST-GCN and the A3T-GCN, respectively), while the highest results are gathered for data represented by a tennis player model holding

a tennis racket, represented as three-dimensional points (79.10–88.91% and 85.23–93.23% for the ST-GCN and the A3T-GCN, respectively). For the ST-GCN classifier accuracy is higher for 3D-based data of a tennis player model (78.10–88.70%) than for image-based data (77.11–84.70%). On the other hand, the following situation occurs for the A3T-GCN, where image-based data obtain higher results (80.10–88.70%) compared to the 3D-based tennis player model (78.70–89.93%). In the case of the DA-GCN for image-based input accuracies (82.30–92.00%) are lower than for 3D-based input (89.37–97.87%). These results clearly indicate that tennis strokes recognition based on data containing a full body tennis player model with a tennis racket model allows for higher classification performance when the data are represented as three-dimensional coordinates. This type of data is more accurate than the same data represented as an image and thus provides better performance [102].

Another interesting issue that is raised in this monograph concerns verifying whether the use of attention modules for the GCN allows one to improve the network performance. For the purpose of this study two out of three proposed GCN methods are implemented utilizing soft attention modules for feature extractions. In the A3T-GCN approach one attention module is proposed, while in the DA-GCN – two are proposed, one for the tennis player model and the second for the tennis racket model, respectively. Applying one attention module improves the classification mean accuracy up to 5.93% for 3D combined data. Implementation of two separate soft modules allow one to obtained improvements up to 9.81% and up to 3.88%, compared to a neural network without an attention module and a network with one attention module, respectively. These results confirm that the use of attention modules allows one to improve the performance of the tennis stroke classification.

This monograph also attempts to verify whether the fuzzification process of the input data affects the quality of tennis stroke classification. In the case of forehand and backhand groundstrokes, the analyzed phases of tennis moves can often be confused, i.e., the preparation phase and the actual hit, which begins with the contact of the racket and ball and the racket swinging to complete the stroke. The boundary between these phases is not sharp. That is why in this study fuzzy logic was implemented into the input of the proposed GCN approaches

utilizing trapezoidal functions. This process allowed one to model the uncertain items. For all proposed neural methods, the fuzzification process improved the classification performance. The mean accuracy enhancements after applying the fuzzification process were obtained at the level of 1.33%, 1.69%, and 3.33% for the ST-GCN, A3T-GCN, and DA-GCN, respectively. It should be stated that the highest ability to classify tennis movements is gained by the DA-GCN approach. The obtained results for all graph networks clearly indicated that the fuzzification of input improves performance of tennis movements classification.

The average computation times for training the proposed networks are presented in Fig. 8.1–8.3. The lowest training times were recorded for the ST-GCN classifier, and the highest for the DA-GCN. It should be noted that adding the attention module causes an increase in the network training time. In the case of ST-GCN and A3T-GCN solutions, image input data required the most time. The least time was needed for 3D data in the form of a racket model, while the most time was required for a tennis player model with a racket. This depends on the number of markers from which a given input data model was created. The network training computation times with and without input fuzzification are comparable.

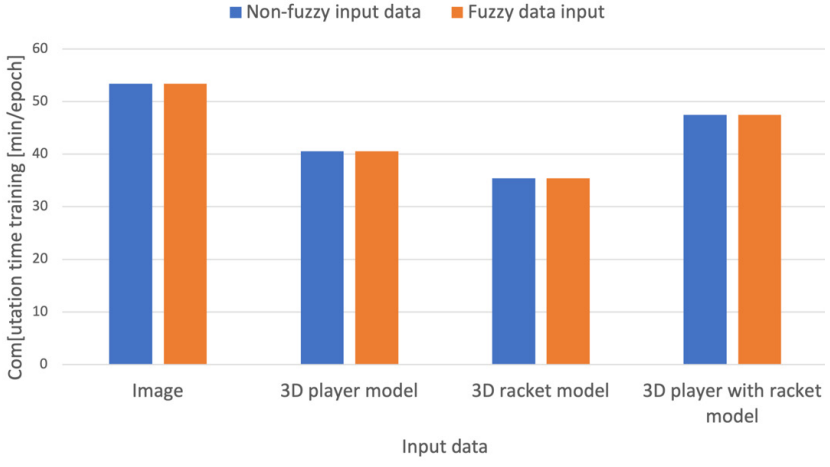


Fig. 8.1. Average computation time for training the **ST-GCN** classifier

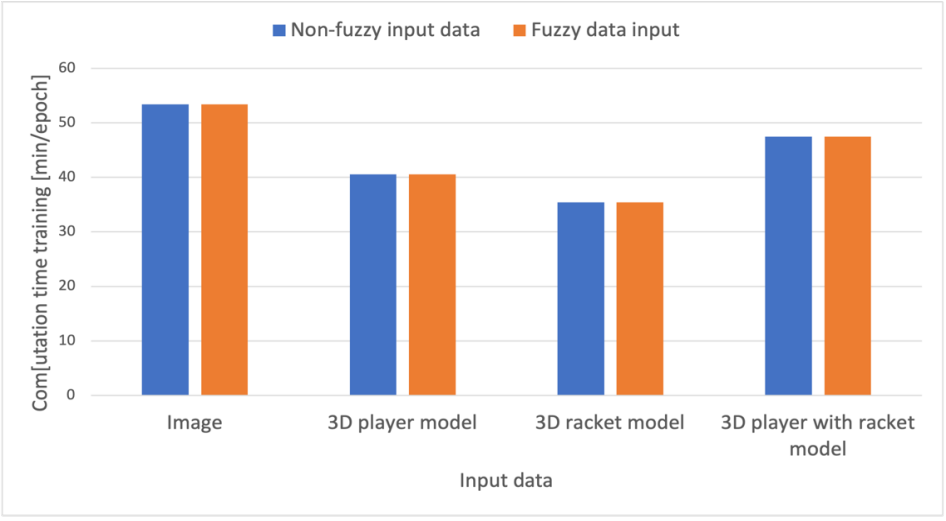


Fig. 8.2. Average computation time for training the **A3T-GCN** classifier

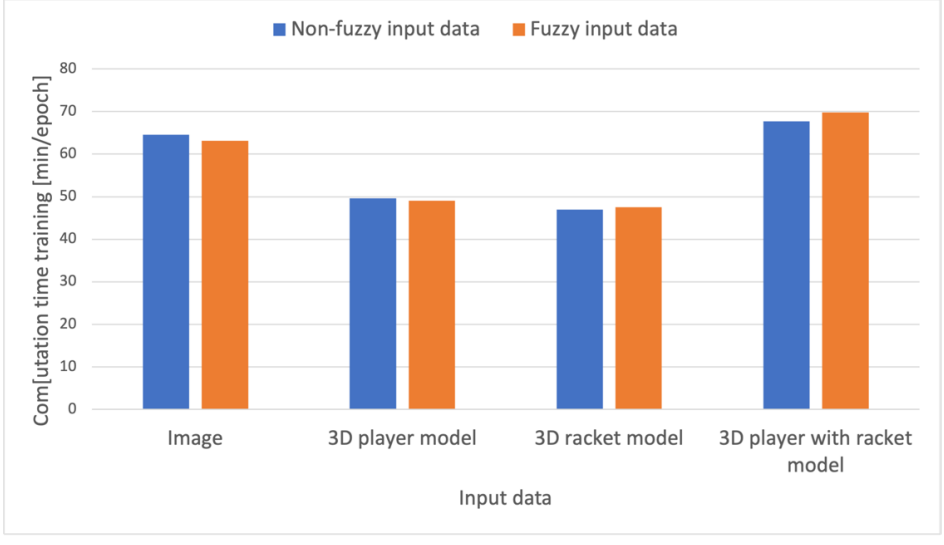


Fig. 8.3. Average computation time for training the **DA-GCN** classifier

The performance of all proposed models was verified using Leave-One-Out Cross Validation (LOOCV). The evaluation of the model is calculated based on training and testing sets. The testing set contains one element. The idea is to obtain performance for one element (observation) from the testing set. This method is stated to be appropriate

for the gathered tennis dataset. Although it is very expensive computationally, it allowed one to obtain unbiased information about the models, and to evaluate how each data represents the whole dataset. Applying the LOOCV the root mean squared error (RMSE) was utilized:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{y_i}\right)^2 \tag{8.1}$$

where n states the number of dataset elements, $y_1$is true value, while $\widehat{y_i}$ the predicted one.

$$RMSE = \sqrt{MSE} \tag{8.2}$$

The obtained results for all presented models, gathered in Table 8.1, indicate that all proposed networks obtained high-performance. They perform well on the new data that were not used for training these models.

CNN has gained impressive results in image classification and for recognition purposes. Their success was mainly involved with convolutional operations performed utilizing kernels or filters. A very interesting approach is understanding how the images are interpreted by neural networks and what features are extracted. Before final classification, the neural networks reveal relevant information from an input image. These data are further used in the recognition. The classification performance relies on the obtained features. The more adequately the information is revealed, the more relevant features are recognized and thus a higher classification performance may be obtained. Moreover, the visualization may provide information about what features are extracted between network layers.

In this monograph, the effort was put on an interpretability aspect, which reveals the vital information from a DL model and indicates which part of input data are the most relevant. It is the level to which a human may understand how the classification is performed [106]. The concept of explainability is related to interpretability. The explainability reveals the internal mechanics of the ML approach. There are four main types of techniques applied in terms of interpretability: global and model-agnostic, global and model-specific, local

and model-agnostic and finally local and model-specific [106][107]. The third one was used in this monograph. It is often applied in sophisticated DL models by providing various mechanism indicating on which part the approaches pay attention. The attention maps as well as heat maps revel local surroundings that are crucial for prediction and detection [108].

Table 8.1. Obtained LOOCV values for the proposed models

Model	Input type	RMSE	±SD
ST-GCN	image	8.34	5.23
	3D tennis player model	8.94	5.89
	3D tennis racket model	9.11	5.47
	3D tennis player model with a tennis racket model	7.92	5.05
Fuzz ST-GCN	image	8.11	6.01
	3D tennis player model	8.87	5.77
	3D tennis racket model	8.91	5.42
	3D tennis player model with a tennis racket model	7.89	4.96
A3T-GCN	image	8.89	5.77
	3D tennis player model	8.64	5.13
	3D tennis racket model	9.57	5.73
	3D tennis player model with a tennis racket model	6.83	5.01
Fuzz A3T-GCN	image	7.53	5.23
	3D tennis player model	7.77	4.98
	3D tennis racket model	8.04	5.55
	3D tennis player model with a tennis racket model	7.11	5.01
DA-GCN	image	6.69	3.45
	3D tennis player model	6.58	4.82
	3D tennis racket model	6.93	4.17
	3D tennis player model with a tennis racket model	6.13	4.07
Fuzz DA-GCN	image	6.28	4.43
	3D tennis player model	5.99	4.87
	3D tennis racket model	6.03	4.23
	3D tennis player model with a tennis racket model	5.11	3.35

Class activation maps (CAM) have the ability to reveal a part of an image that is extremely important for specific classification purposes. After each layer a global average pooling is applied. The process of extracting features from images presenting a Vicon tennis model with a tennis racket for the forehand tennis move is depicted in Fig. 8.4 and Fig. 8.5. The brighter the color obtained, the more important part of the image is revealed. As it can be easily seen, the whole tennis player model together with a tennis racket is extracted as relevant information for tennis strokes recognition.

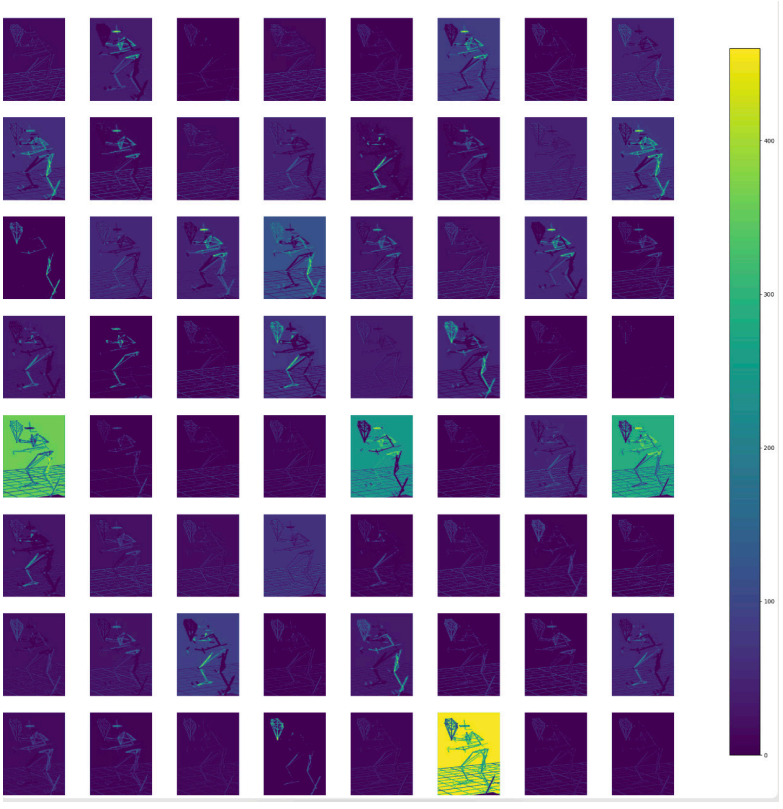


Fig. 8.4. Feature maps in the first convolution layer in the first convolution block

However, it should be noticed that various filters in convolutional operations indicated different parts of the tennis player model, a whole tennis racket, or both. Apart from a tennis racket, the left side of the player model seemed to be more relevant in tennis movements recognition than the right one. Positioning of the player's body, both for

upper and lower parts, is crucial for the correct stroke performance. In the analyzed images (Fig. 8.4), the tennis player is positioned in such a way as to depict the entire player model holding a racket in preparation for the forehand stroke. Even though the racket model is behind the player, its position is crucial for classification of this phase. Not only is the arm holding the tennis racket relevant; the other arm's location is also relevant. Also important are: positioning and twisting the upper body during impact as well as preparing for impact. Additionally, floor and a vector representing ground reaction forces from biomechanical platforms are also indicated as significant.

Analyzing the extracted features after the first and third convolution, it can be noticed that in both cases, characteristic data necessary to recognize the forehand preparation move were obtained. It is clearly visible that various filters applied reveal various pieces of vital information.

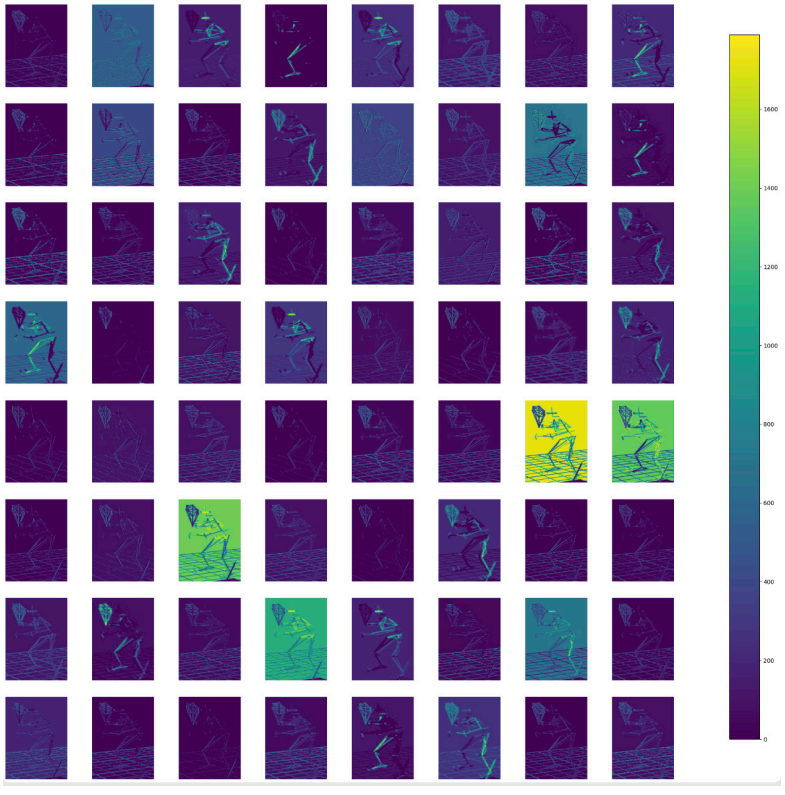


Fig. 8.5. Feature maps in the second convolution layer in the first convolution block

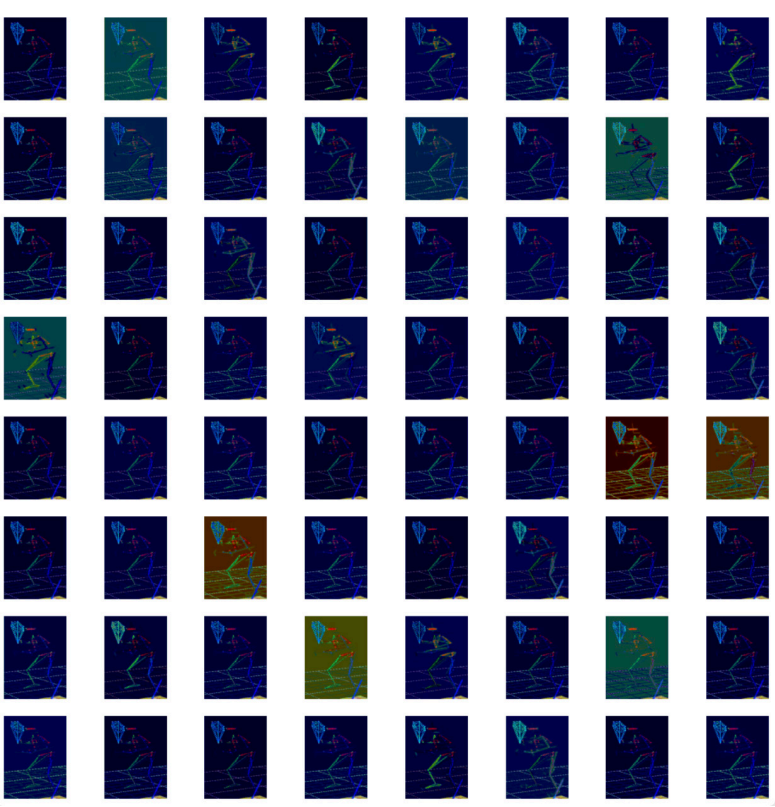


Fig. 8.6. Heat maps obtained in the first pooling layer in the first convolution block

In the case of heatmaps, the most relevant features taken for classification purposes combined are shown in a set of images. As it is depicted in Fig. 8.6, the tennis player model together with the racket model are extracted.

Gradient-weighted Class Activation Mapping (Grad-CAM) is another method for neural network interpretability that utilizes gradients to obtain a localization map of important image regions from the final convolutional layer [109]. It can be observed which parts of images and patterns are relevant for classification purposes. The obtained maps are presented in Fig. 8.7–8.12.

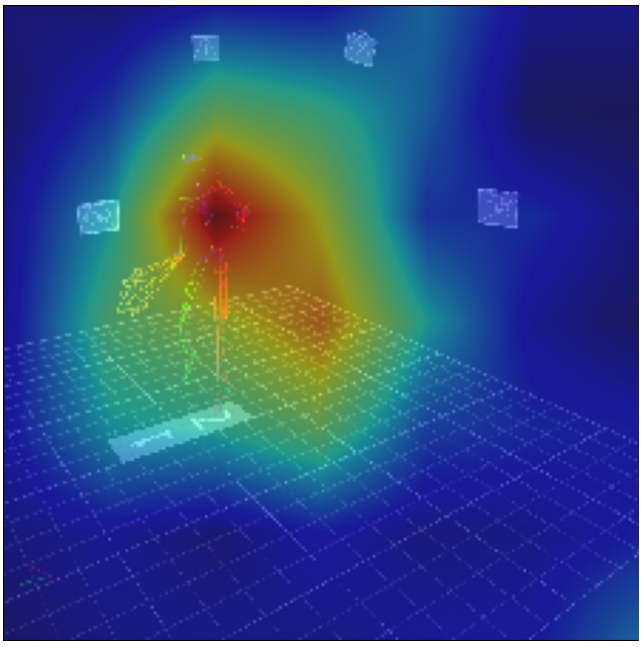


Fig. 8.7. Grad-CAM for the **ST-GCN** classifier for tennis forehand

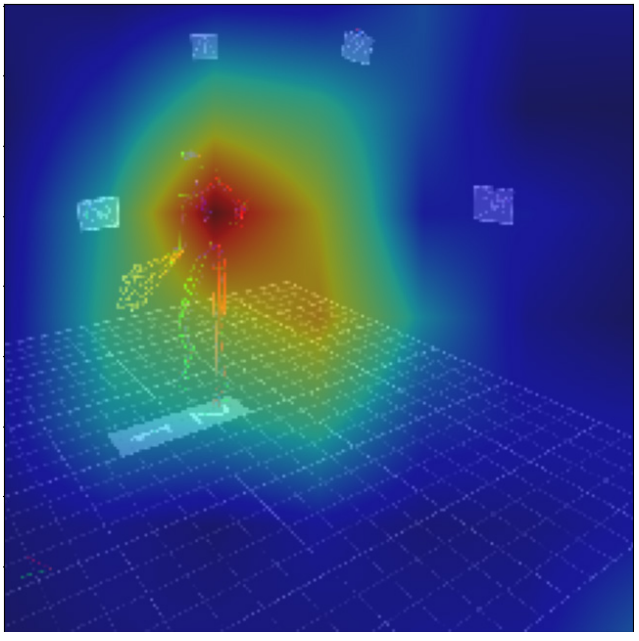


Fig. 8.8. Grad-CAM for the **ST-GCN** classifier with fuzzification for tennis forehand

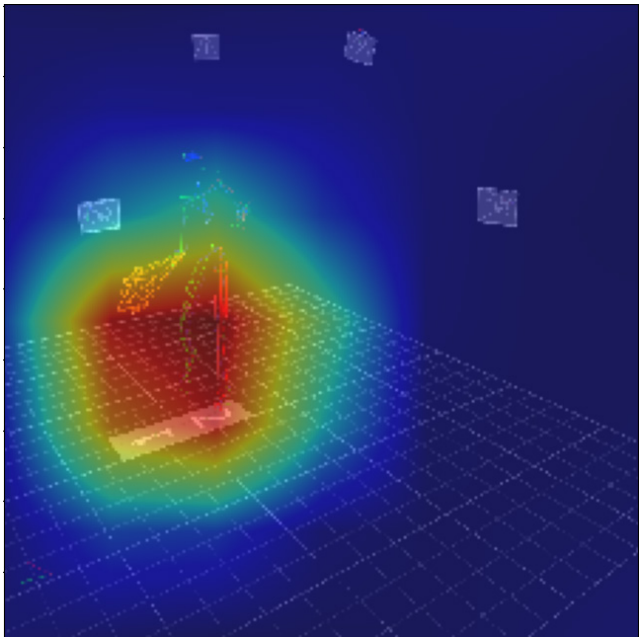


Fig. 8.9. Grad-CAM for the **A3T-GCN** classifier for tennis forehand

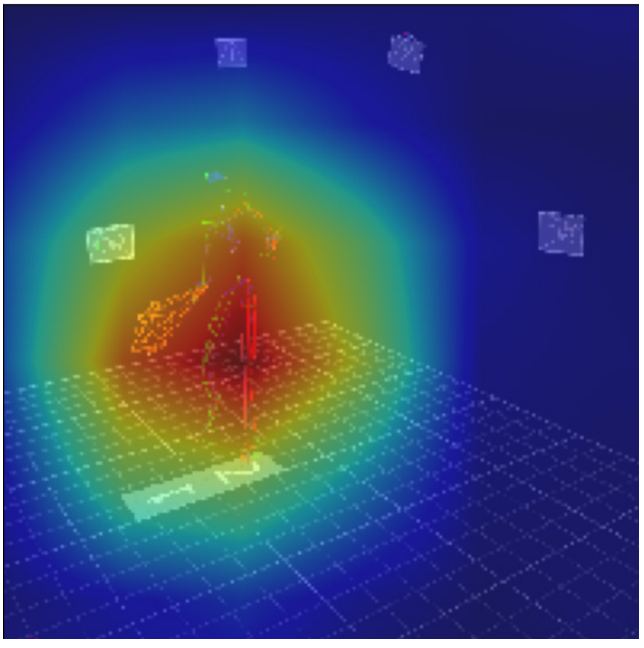


Fig. 8.10. Grad-CAM for the **A3T-GCN** classifier with fuzzification for tennis forehand

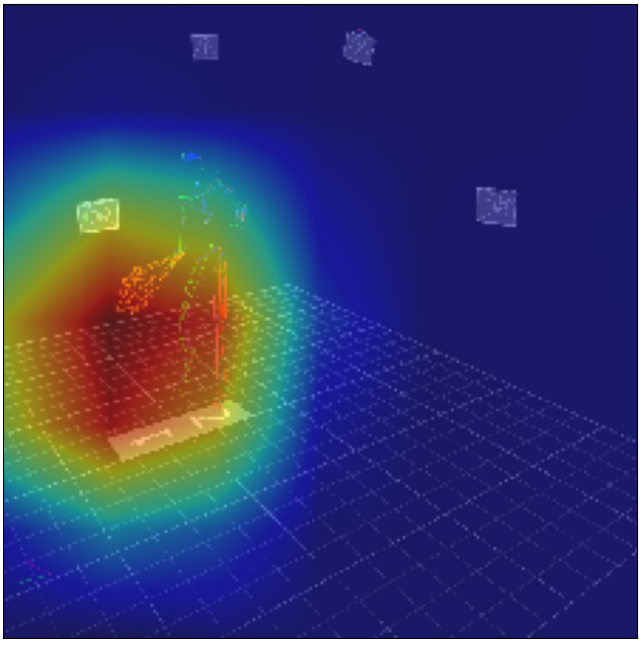


Fig. 8.11. Grad-CAM for the **DA-GCN** classifier for tennis forehand

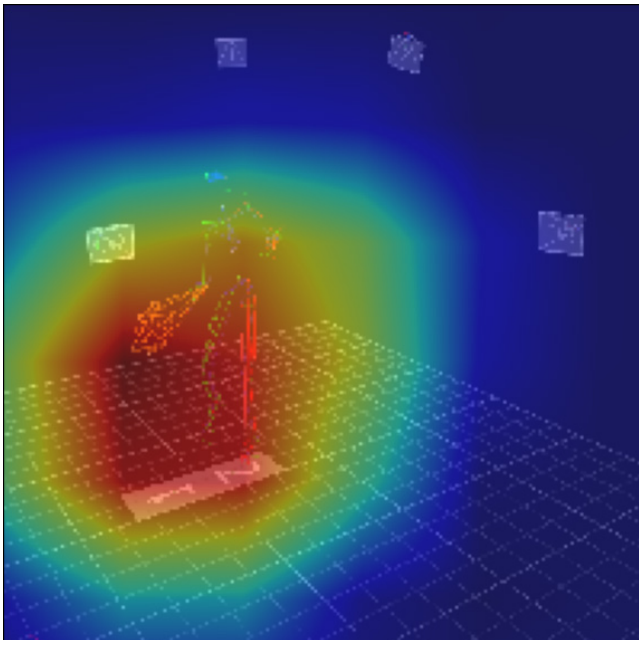


Fig. 8.12. Grad-CAM for the **DA-GCN** classifier with fuzzification for tennis forehand

In case of the ST-GCN model with and without fuzzification (Fig. 8.7, 8.8), the classifier gives the highest attention to the upper body part. Its location is relevant to distinguish tennis movements. Less attention was paid to important elements such as the racket and lower body of a tennis player. For the fuzzification process the player tennis model was most relevant. In case of the A3T-GCN classifier (Fig. 8.9), the network focused most of its attention on the lower body of the tennis player and the racket. The upper body was also considered as important information. Fuzzification resulted in the most significant elements being the whole figure of the tennis player and the racket (Fig. 8.10). For the DA-GCN classification, the network focused most of its attention on the tennis racket and the lower body of the tennis player (Fig. 8.11). Then, the upper body was also considered as important information. The application of fuzzification made the tennis racket and the whole body of the tennis player the most significant elements (Fig. 8.12).

Grad-CAM visualizes how the application of fuzzification process improves the classification results. In each case, the proposed neural network models determine strokes based on the silhouette of a tennis player and a tennis racket, taking into account its different parts. The fuzzification process causes the network to verify the greater part of the silhouette and the tennis racket as necessary features.

Due to limited access to datasets containing accurate tennis data, a professional dataset was created as the first step in this study. The 3DTennisDS contains the basic movements of tennis strokes, such as forehand, backhand, volley forehand, and volley backhand. They were recorded using an optical motion capture system utilizing retro-reflective markers. Both a tennis player's and a tennis racket's moves were captured. Because of the dimensions of the motion capture laboratory, it was not no possible to record tennis moves such as serve or smash. The obtained data were accurate, and thus are suitable for further studies. Moreover, forehand and backhand groundstrokes were divided into two phases, each. The first one is the preparation phase, while the second includes movements from hitting the ball with a tennis racket together with the racket swinging. All tennis strokes were supervised by the professional tennis coaches. They also approved the recordings and specified their division into separate phases. Among available tennis datasets, like THETIS or Mimetics, recorded using a Kinect device, or based on sensor data, and finally created based on

the broadcast videos, there are not many motion capture datasets recorded using an optical motion capture system. The Tennis-Mocap is one example of this kind of data. However, they are provided using a bvh format. In the prepared and presented 3DTennisDS, the data are stored in c3d format. It should be emphasized that more points on the tennis player model was defined, 39 according to the Plug-in Gait model, compared to 34 in Tennis-Mocap dataset. It should be emphasized that, in 3DTennisDS the tennis racket model was also defined. These kinds of combined data allowed for sophisticated classification utilizing GCN and attention modules.

From the created 3DTennisDS two types of data were generated. First, images from the Vicon motion capture software were generated. They contained the model of a tennis player together with the model of a tennis racket. Second, from c3d recordings, three-dimensional data were generated, storing information from all markers corresponding to a player's body and a racket.

CHAPTER

9

Conclusions

This monograph deals with the challenging issue of recognizing tennis strokes. The main focus is placed on four basic movements, such as forehand, backhand, volley forehand and volley backhand. Furthermore, the player's movements between these moves are also analyzed and reflect a moving tennis player on a court. The forehand and backhand strokes are further divided into phases, which allow the detection of stroke stages. This allows one to analyze how a player prepares to execute the stroke, and then how well he or she performs it.

In this monograph, the main focus is put on classification utilizing graph convolutional networks. For the purpose of this study three GCN solutions have been developed. Comprehensive research has been carried out to select the most appropriate tool for recognizing tennis moves. Several issues have been studied in depth. It was stated that GCNs were adequate tools for tennis movements recognition. It has been verified that classification performance depends on the input data type. Recognition based on 3D-based data reflecting a tennis model with a racket model obtains higher performance than for the same models formed as images. Another issue that has been analyzed in detail consisted of applying attention modules. It has appeared that the highest performance was achieved for the DA-GCN with two soft attention modules. How input fuzzification affects the quality of classification has also been investigated. In this monograph, the use of this process with a trapezoidal function improved the classification performance of each classifier for each type of input data. As a result, the proposed DA-GCN approach turned out to be the best solution for tennis movement recognition using three-dimensional data.

In this monograph, the interpretability aspect of GCN was also presented. This issue of significant importance has been examined by

presenting feature maps and heatmaps obtained after selected convolutional operations. These results showed how individual filters in GCNs extract features that are relevant for recognizing tennis movements. In order to conduct a comprehensive analysis, a dataset of tennis strokes, titled 3DTennisDS, was developed. All the tennis movements stored in it were performed by ten professional tennis players and recorded using a passive optical motion capture system. This resulted in obtaining a dataset of very accurate motion data.

However, the presented studies also have limitations. First of all, the capturing of tennis strokes was performed in a laboratory, not on a tennis court. For this reason, the participant in the recording was constantly moving, making tennis strokes after running around a bollard. Second, the height of the laboratory prevented recording of the serve stroke.

While analyzing the validation and testing accuracy presented on graphs, oscillations can be seen before the model stabilizes. They can result from additional artifacts presented in the images (the presence of biomechanical platforms and the floor) apart from the tennis player and the racket models, adjusting the image to the classifier input, or creating a graph utilizing operations based on the image. Despite the initial instability of the model, it can be seen that in the final phase of training the network achieves stabilization at an acceptable level. The results show that the number of 140 epochs is sufficient for the model to function correctly. Moreover, based on confusion matrices for all proposed classifiers, it can be stated that a small percentage of tennis movements were incorrectly classified. What is more, the same strokes were often misclassified.

This monograph presents a wide perspective of tennis strokes recognition. The presented studies detailed a concern for the recognition of tennis movements based on accurate data and will allow in the future the creation of realistic 3D animations reflecting tennis movements. The described research may also be the basis for implementing applications that support learning basic tennis strokes. Beginning players and amateurs may thereby be able to enhance their enjoyment of the sport. In a future study, the author will try to achieve faster convergence of the model by changing hyperparameters such as: kernel size, stride, or activation function. Various methods of aggregations may be considered [110][111] in order to improve the model's performance and thus enhance tennis movements recognition.

References

[1] Host K., Ivašić-Kos M., *An Overview of Human Action Recognition in Sports Based on Computer Vision*, "Heliyon" 2022, vol. 8(6).

[2] Host K., Ivašić-Kos M., Pobar M., *Action Recognition in Handball Scenes, [in:] Intelligent Computing. Lecture Notes in Networks and Systems*, Springer 2022, vol. 283, https://doi.org/10.1007/978-3-030-80119-9_41.

[3] Rahmad N.A., As'ari M.A., Ghazali N.F., Shahar N., Sufri N.A.J., *A Survey of Video Based Action Recognition in Sports,* "Indonesian Journal of Electrical Engineering and Computer Science" 2018, vol. 11, p. 987–993.

[4] Cai J., Hu J., Tang X., Hung T.Y., Tan Y.P., *Deep Historical Long Short-Term Memory Network for Action Recognition*, "Neurocomputing" 2020, vol. 407, p. 428–438.

[5] Salisu S., Ruhaiyem N.I.R., Eisa T.A.E., Nasser M., Saeed F., Younis H.A., *Motion Capture Technologies for Ergonomics: A Systematic Literature Review*, "Diagnostics" 2023, vol. 13(15), 2593.

[6] Yu, Z., Yan, W.Q., *Human Action Recognition Using Deep Learning Methods,* [in:] *2020 35th International Conference on Image and Vision Computing New Zealand (IVCNZ)*, IEEE, 2020, p. 1–6.

[7] Skublewska-Paszkowska M., Powroznik P., Lukasik E., *Learning Three Dimensional Tennis Shots Using Graph Convolutional Networks,* "Sensors" 2020, vol. 20, p. 1–12.

[8] *International Tennis Federation*, https://www.itftennis.com/en/growing-the-game/participation/ [access: 1.05.2024].

[9] Zhu G., Xu C., Gao W., Huang Q., *Action Recognition in Broadcast Tennis Video Using Optical Flow and Support Vector Machine,* [in:] *Proceedings of the European Conference on Computer Vision,* Springer, 2006, p. 89–98.

[10] Zhu G., Xu C., Huang Q., Gao W., *Action Recognition in Broadcast Tennis Video,* [in:] *Proceedings of the International Conference on Pattern Recognition,* IEEE, 2006, vol. 1, p. 251–254.

[11] Zhu G., Xu C., Huang Q., Gao W., Xing L., *Player Action Recognition in Broadcast Tennis Video with Applications to Semantic Analysis of Sports Game,* [in:] *Proceedings of the 14th ACM International Conference on Multimedia,* Association for Computing Machinery, 2006, p. 431–440.

[12] De Campos T., Barnard M., Mikołajczyk K., Kittler J., Yan F., Christmas W., Windridge D., *An Evaluation of Bags-of-Words and Spatio-Temporal Shapes for Action Recognition,* [in:] *2011 IEEE Workshop on Applications of Computer Vision,* IEEE, 2011, p. 344–351.

[13] Farajidavar N., De Campos T., Kittler J., Yan F., *Transductive Transfer Learning for Action Recognition in Tennis Games,* [in:] *Proceedings of the IEEE International Conference on Computer Vision Workshops*, IEEE, 2011, p. 1548–1553.

[14] Jiang K., Izadi M., Liu Z., Dong J.S., *Deep Learning Application in Broadcast Tennis Video Annotation,* [in:] *25th International Conference on Engineering of Complex Computer Systems*, IEEE, 2020, p. 53–62.

[15] Gourgari S., Goudelis G., Karpouzis K., Kollias S., *Thetis: Three Dimensional Tennis Shots a Human Action Dataset,* [in:] *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops,* IEEE, 2013, p. 676–681.

[16] Vainstein J., Manera J., Negri P., Delrieux C., Maguitman A., *Modeling Video Activity with Dynamic Phrases and its Application to Action Recognition in Tennis Videos,* [in:] *Proceedings of Iberoamerican Congress on Pattern Recognition,* Springer, 2014, p. 909–916.

[17] Mora S.V., Knottenbelt W.J., *Deep Learning for Domain-Specific Action Recognition in Tennis,* [in:] *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, (CVPRW),* IEEE, 2017, p. 170–178.

[18] Ullah M., Mudassar Yamin, M., Mohammed A., Daud Khan S., Ullah H., Alaya Cheikh F., *Attention-Based LSTM Network Action Recognition in Sports,* "Electronic Imaging" 2021, vol. 6, p. 302-1–302-5.

[19] Bačić, B., *Extracting Player's Stance Information from 3D Motion Data: A Case Study in Tennis Groundstrokes,* [in:] *Image and Video Technology – PSIVT 2015 Workshops,* Springer International Publishing, 2015, p. 307–318.

[20] Yamada K., Matsuura K., Hamagami K., Inui H., *Motor Skill Development Using Motion Recognition Based on an HMM,* "Procedia Computer Science" 2013, vol. 22, p. 1112–1120.

[21] Bacic B., *Towards the Next Generation of Exergames: Flexible and Personalised Assessment-Based Identification of Tennis Swings, arXiv 2018*, preprint arXiv:1804.06948.

[22] Ganser A., Hollaus B., Stabinger S., *Classification of Tennis Shots with a Neural Network Approach,* "Sensors" 2021, vol. 21(17), 5703.

[23] Whiteside D., Cant O., Connolly M., Reid M., *Monitoring Hitting Load in Tennis Using Inertial Sensors and Machine Learning,* "International Journal of Sports Physiology and Performance" 2017, vol. 12(9), p. 1212–1217.

[24] Büthe L., Blanke U., Capkevics H., Tröster, G., *A Wearable Sensing System for Timing Analysis in Tennis,* [in:] *2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks,* IEEE, 2016, p. 43–48.

[25] Pei W., Wang J., Xu X., Wu Z., Du X., *An Embedded 6-Axis Sensor Based Recognition for Tennis Stroke,* [in:] *Proceedings of IEEE International Conference on Consumer Electronics,* IEEE, 2017, p. 55–58.

[26] Ma K., *A Real Time Artificial Intelligent System for Tennis Swing Classification,* [in:] *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics,* IEEE, 2021, p. 000021–000026.

[27] Bezobrazov S., Sheleh A., Kislyuk S., Golovko V., Sachenko A., Komar M., Dorosh V., Turchenko V., *Artificial Intelligence for Sport Activitity Recognition* [in:] *2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, IEEE, 2019, vol. 2, p. 628– 632.

[28] Ó Conaire C., Connaghan D., Kelly P., O'Connor N.E., Gaffney M., Buckley J., *Combining Inertial and Visual Sensing for Human Action Recognition in Tennis,* [in:] *Proceedings of the First ACM International Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams,* Association for Computing Machinery, 2010, p. 51–56.

[29] Weinzaepfel P., Rogez G., *Mimetics: Towards Understanding Human Actions out of Context,* "International Journal of Computer Vision" 2021, vol. 129(5), p. 1675– 1690.
[30] Valencia-Marin C.K., Pulgarin-Giraldo J.D., Velasquez-Martinez L.F., Alvarez-Meza A.M., Castellanos-Dominguez G., *An Enhanced Joint Hilbert Embedding-Based Metric to Support Mocap Data Classification with Preserved Interpretability,* "Sensors" 2021, vol. 21(13), 4443.
[31] Skublewska-Paszkowska M., Powroznik P., Lukasik E., Smolka J., *Tennis Patterns Recognition Based on a Novel Tennis Dataset – 3DTennisDS,* "Advances in Science and Technology Research Journal" 2024, vol. 8(6), p. 159–176.
[32] Davis R.B., Ounpuu S., Tyburski D., Gage J.R., A Gait *Analysis Data Collection and Reduction Technique,* "Human Movement Science" 1991, vol. 10(5), p. 575–587.
[33] LeCun Y., Kavukcuoglu K., Farabet C., *Convolutional Networks and Applications in Vision,* [in:] *Proceedings of 2010 IEEE International Symposium on Circuits and Systems,* IEEE, 2010, p. 253–256.
[34] Li Z., Liu F., Yang W., Peng S., Zhou J., *A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects*, "IEEE Transactions on Neural Networks and Learning Systems" 2021, vol. 33(12), p. 6999–7019.
[35] LeCun Y., Boser B., Denker J.S., Henderson D., Howard R.E., Hubbard W., Jackel, L.D., *Backpropagation Applied to Handwritten Zip Code Recognition,* "Neural Computation" 1989, vol. 1(4), p. 541–551.
[36] Krizhevsky A., Sutskever I., Hinton G.E., *ImageNet Classification with Deep Convolutional Neural Networks,* "Advances in Neural Information Processing Systems" 2012, vol. 25, p. 84–90.
[37] Gu J., Wang Z., Kuen J., Ma L., Shahroudy A., Shuai B., Liu T., Wang X., Wang G., Cai J., Chen, T., *Recent Advances in Convolutional Neural Networks,* "Pattern Recognition" 2018, vol. 77, p. 354–377.
[38] Maas, A.L., Hannun, A.Y. Ng, A.Y., *Rectifier Nonlinearities Improve Neural Network Acoustic Models,* [in:] *Proceedings of the International Conference on Machine Learning,* ICML, 2013, vol. 30(1).

[39] He, K., Zhang, X., Ren, S., Sun, J., *Delving Deep Into Rectifiers: Surpassing Human-Level Performance on Imagenet Classification*, [in:] *Proceedings of the International Conference on Computer Vision,* IEEE, 2015, p. 1026–1034.

[40] Xu, B., Wang N., Chen T., Li M., *Empirical Evaluation of Rectified Activations in Convolutional Network, arXiv 2015*, preprint arXiv:1505.00853.

[41] Wang T., Wu D.J., Coates A., Ng, A.Y., *End-to-end Text Recognition with Convolutional Neural Networks,* [in:] *Proceedings of the 21st International Conference on Pattern Recognition*, IEEE, 2012, p. 3304–3308.

[42] Boureau Y. L., Ponce J., LeCun Y., *A Theoretical Analysis of Feature Pooling in Visual Recognition,* [in:] *Proceedings of the 27th International Conference on Machine Learning*, Omnipress, 2010, p. 111–118.

[43] Hyvärinen A., Köster U., *Complex Cell Pooling and the Statistics of Natural Images,* "Network: Computation in Neural Systems" 2007, vol. 18(2), p. 81–100.

[44] Estrach J B., Szlam A., LeCun Y., *Signal Recovery From Pooling Representations,* [in:] *International Conference on Machine Learning*, IEEE, 2014, p. 307–315.

[45] Yu D., Wang H., Chen P., Wei, Z., *Mixed Pooling for Convolutional Neural Networks,* [in:] *Rough Sets and Knowledge Technology,* Springer, 2014, p. 364–375.

[46] Simonyan K., Zisserman A., *Very Deep Convolutional Networks for Large-Scale Image Recognition, arXiv 2014,* preprint arXiv:1409.1556.

[47] Zeiler M.D., Fergus R., *Visualizing and Understanding Convolutional Networks,* [in:] *Computer Vision – ECCV 2014: 13th European Conference, Proceedings,* Part I 13*,* Springer, 2014, p. 818–833.

[48] Hinton G. ., Srivastava N., Krizhevsky A., Sutskever I., Salakhutdinov R.R. *Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors, arXiv 2012,* preprint arXiv:1207.0580.

[49] Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich, A., G*oing Deeper with Convolutions,* [in:] *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* IEEE, 2015, p. 1–9.

[50] He K., Zhang X., Ren S., Sun J., *Deep Residual Learning for Image Recognition,* [in:] *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* IEEE, 2016, p. 770–778.
[51] Yang T.J., Howard A., Chen B., Zhang X., Go A., Sandler M., Sze V., Adam, H., *NetAdapt: Platform-aware Neural Network Adaptation for Mobile Applications,* [in:] *Proceedings of the European Conference on Computer Vision,* 2018, p. 285–300.
[52] Jin X., Xie Y., Wei X.S., Zhao B.R., Chen Z.M., Tan X., *Delving Deep into Spatial Pooling for Squeeze-and-excitation Networks,* "Pattern Recognition" 2022, vol. 121, 108159.
[53] Zhang X., Zhou X., Lin M., Sun J., *Shufflenet: an Extremely Efficient Convolutional Neural Network for Mobile Devices,* [in:] *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2018, p. 6848–6856.
[54] Ma N., Zhang X., Zheng H.T., Sun J., *Shufflenet v2: Practical Guidelines for Efficient CNN Architecture Design*, [in:] *Proceedings of the European Conference on Computer Vision,* Springer, 2018, p. 116–131.
[55] Xu X., Zhao X., Wei M., Li Z., *A Comprehensive Review of Graph Convolutional Networks: Approaches and Applications,* "Electronic Research Archive" 2023, vol. 31(7), p. 4185–4215.
[56] Wei F., Mei, K., *Towards Self-Explainable Graph Convolutional Neural Network with Frequency Adaptive Inception,* "Pattern Recognition" 2024, vol. 146, 109991.
[57] Zhang S., Tong H., Xu J., Maciejewski R., *Graph Convolutional Networks: a Comprehensive Review*, "Computational Social Networks" 2019, vol. 6(1), p. 1–23.
[58] Kipf T.N., Welling, M., *Variational Graph Auto-Encoders", arXiv 2016,* preprint arXiv:1611.07308.
[59] Ding Y., Tian L. P., Lei X., Liao B., Wu F.X., *Variational Graph Auto-Encoders for miRNA-Disease Association Prediction,* "Methods" 2021, vol. 192, p. 25–34.
[60] Ruiz L., Gama F., Ribeiro A., *Gated Graph Recurrent Neural Networks,* "IEEE Transactions on Signal Processing" 2020, vol. 68, p. 6303–6318.
[61] Beck D., Haffari G., Cohn T., *Graph-To-Sequence Learning Using Gated Graph Neural Networks, arXiv 2018,* preprint arXiv:1806.09835.

[62] Veličković P., Cucurull G., Casanova A., Romero A., Lio P., Bengio Y., *Graph Attention Networks, arXiv 2017*, preprint arXiv:1710.10903.
[63] Lee J. B., Rossi R., Kong X., *Graph Classification using Structural Attention,* [in:] *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining,* Association for Computing Machinery, 2018, p. 1666–1674.
[64] Wang H., Wang J., Wang J., Zhao M., Zhang W., Zhang F., Xie X., Guo M., *GraphGAN: Graph Representation Learning with Generative Adversarial Nets,* [in:] *Proceedings of the AAAI Conference on Artificial Intelligence,* AAAI Press, 2018, vol. 32(1), p. 2508–2515.
[65] You J., Ying R., Ren X., Hamilton W., Leskovec J., GraphRNN: *Generating Realistic Graphs with Deep Auto-regressive Models,* [in:] *International Conference on Machine Learning,* Curran Associates, Inc., 2018, p. 5708–5717.
[66] Dai H., Nazi A., Li Y., Dai B., Schuurmans D., *Scalable Deep Generative Modeling for Sparse Graphs,* [in:] *International Conference on Machine Learning,* Curran Associates, Inc., 2020, p. 2302–2312.
[67] Duvenaud D.K., Maclaurin D., Iparraguirre J., Bombarell R., Hirzel T., Aspuru-Guzik A., Adams R.P., *Convolutional Networks on Graphs for Learning Molecular Fingerprints,* [in:] *Advances in Neural Information Processing Systems,* The MIT Press, 2015, vol. 28, p. 2224–2232.
[68] Simonovsky M., Komodakis N., *Dynamic Edge-conditioned Filters in Convolutional Neural Networks on Graphs,* [in:] *Proceedings of the IEEE Conference On Computer Vision and Pattern Recognition*, 2017, p. 3693–3702.
[69] Wu Z., Pan S., Chen F., Long G., Zhang C., Philip S.Y., *A Comprehensive Survey on Graph* Neural Networks, "IEEE Transactions on Neural Networks and Learning Systems" *2020,* vol. 32(1), p. 4–24.
[70] Yan S., Xiong Y., Lin D., *Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition,* [in:] *Proceedings of the AAAI Conference on Artificial Intelligence,* AAAI Press, 2018, vol. 32(1).
[71] Yu B., Yin H., Zhu Z., *Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting, arXiv 2017,* preprint arXiv:1709.04875.

[72] Bengio, Y. LeCun, Y., eds., *3rd International Conference on Learning Representations,* ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings, https://arxiv.org/abs/1409.1556

[73] Itti L., Koch C., Niebur E., *A Model of Saliency-Based Visual Attention for Rapid Scene Analysis,* "IEEE Transactions on Pattern Analysis and Machine Intelligence" 1998, vol. 20(11), p. 1254–1259.

[74] Rensink R.A., *The Dynamic Representation of Scenes*, "Visual Cognition" 2000, vol. 7, p. 17–42.

[75] Corbetta M., Shulman G.L., *Control of Goal-Directed and Stimulus-Driven Attention in the Brain,* "Nature Reviews Neuroscience" 2002, vol. 3, p. 201–215.

[76] Yang X., *An Overview of the Attention Mechanisms in Computer Vision,* "Journal of Physics: Conference Series", vol. 1693(1), 012173.

[77] Wang F., Jiang M., Qian C., Yang S., Li C., Zhang H., Wang X., Tang X., *Residual Attention Network for Image Classification,* [in:] *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2017, p. 3156–3164.

[78] Hu J., Shen L., Sun G., *Squeeze-and-excitation Networks,* [in:] *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* IEEE, 2018, p. 7132–7141.

[79] Tang Y., Nguyen D., Ha D., *Neuroevolution of Self-Interpretable Agents,* [in:] *Proceedings of the 2020 Genetic and Evolutionary Computation Conference,* Association for Computing Machinery, 2020, p. 414–424.

[80] Matsuda T., Uchida K., Saito S., Shirakawa S., HACNet: *End-to-end Learning of Interpretable Table-to-image Converter and Convolutional Neural Network,* "Knowledge-Based Systems" 2024, 284, 111293.

[81] Yan S., Smith J. S., Lu W., Zhang B., *Hierarchical Multi-Scale Attention Networks for Action Recognition,* "Signal Processing: Image Communication", 2018, vol. 61, p. 73–84.

[82] Woo S., Park J., Lee J.Y., Kweon I.S., CBAM: *Convolutional Block Attention Module,* [in:] *Proceedings of the European Conference on Computer Vision,* 2018, p. 3–19.

[83] Liu J., Shahroudy A., Xu D.A., Wang G., *Spatio-Temporal LSTM with Trust Gates for 3D Human Action Recognition,* [in:] *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* IEEE 2016, p. 816–833.

[84] Zhang S., Liu X., Xiao J., *On Geometric Features for Skeleton-Based Action Recognition Using Multilayer LSTM Networks,* [in:] *Proceedings of the IEEE Winter Conference on Applications of Computer Vision,* IEEE, 2017, p. 148–157.

[85] Duddu V., Samanta D., Rao V.D., *Fuzzy Graph Modelling of Anonymous Networks, arXiv 2018,* preprint arXiv:1803.11377.

[86] Xiao Z., Xia S., Gong K., Li, D. *The Trapezoidal Fuzzy Soft Set and its Application in MCDM,* "Applied Mathematical Modelling" 2012, vol. 36(12), p. 5844–5855.

[87] Song W., Hagras H., *A Big-Bang Big-Crunch Fuzzy Logic Based System for Sports Video Scene Classification,* [in:] *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE),* Vancouver, BC, Canada, IEEE 2016, p. 642–649. doi: 10.1109/FUZZ-IEEE.2016.7737747.

[88] Trawinski K., *A Fuzzy Classification System for Prediction of the Results of the Basketball Games,* [in:] *International Conference on Fuzzy Systems*, Barcelona, Spain, IEEE, 2010, p. 1–7. doi: 10.1109/FUZZY.2010.5584399.

[89] Krleza D., Fertalj K., *Graph Matching Using Hierarchical Fuzzy Graph Neural Networks,* "IEEE Transactions on Fuzzy Systems" 2017, vol. 25, p. 892–904.

[90] Shekarian E., Kazemi N., Abdul-Rashid S.H., Olugu E.U., *Fuzzy Inventory Models: A Comprehensive Review,* "Applied Soft Computing" 2017, vol. 55, p. 588–621.

[91] Sadollah, A., *Introductory Chapter: which Membership Function is Appropriate in Fuzzy System?,* [w:] *Fuzzy Logic Based in Optimization Methods and Control Systems and its Applications,* IntechOpen 2018.

[92] Princy S., Dhenakaran S.S., *Comparison of Triangular and Trapezoidal Fuzzy Membership Function,* "Journal of Computing Science and Engineering" 2016, vol. 2(8), p. 46–51.

[93] Derbel I., Hachani N., Ounelli H., *Membership Functions Generation Based on Density Function,* [in:] *2008 International Conference on Computational Intelligence and Security,* IEEE, 2008, vol. 1, p. 96–101.

[94] Skublewska-Paszkowska M., Powroznik P., Lukasik E., *Attention for the successive classes Graph Convolutional Network for Tennis Groundstrokes Phases Classification,* [in:] *2022 IEEE International Conference on Fuzzy Systems,* IEEE, 2022, p. 1–8.
[95] Ling Z., Yujiao S., Yu L., Pu W., Tao L., Min D., Heifeng L., T-GCN: *A Temporal Graph Convolutional Network for Traffic Prediction,* "IEEE Transactions on Intelligent Transportation Systems" 2020, Vol. 21, No. 9, p. 3848–3858.
[96] Bai J., Zhu J., Song, Y. Zhao L., Hou Z., Du R., Li H., A3T-GCN: *Attention Temporal Graph Convolutional Network for Traffic Forecasting,* "ISPRS International Journal of Geo-Information" 2021, vol. 10(7), 485.
[97] Skublewska-Paszkowska M., Powroznik P., Barszcz M., Dziedzic K., *Dual Attention Graph Convolutional Neural Network to Support Mocap Data Animation,* "Advances in Science and Technology Research Journal" 2023, vol. 17(5), p. 313–325.
[98] Ahmadi A., Destelle F., Monaghan D., O'Connor N.E., Richter C., Moran K. *A Framework for Comprehensive Analysis of a Swing in Sports Using Low-Cost Inertial Sensors,* Sensors IEEE 2014, p. 2211–2214.
[99] Ahmadi A., Mitchell E., Richter C., Destelle F., Gowing M., O'Connor N.E., Moran, K., *Toward Automatic Activity Classification and Movement Assessment During a Sports Training Session,* "IEEE Internet of Things Journal" 2014, vol. 2(1), p. 23–32.
[100] Perri T., Reid M., Murphy A., Howle K., Duffield R., *Prototype Machine Learning Algorithms from Wearable Technology to Detect Tennis Stroke and Movement Actions*, "Sensors" 2022, vol. 22(22), 8868.
[101] Jiang S., Rekimoto, J., *Mediated-Timescale Learning: Manipulating Timescales in Virtual Reality to Improve Real-World Tennis Forehand Volley,* [in:] *Proceedings of the 26th ACM Symposium on Virtual Reality Software and Technology,* Association for Computing Machinery, 2020, p. 1–2.
[102] Skublewska-Paszkowska M., Lukasik E., Szydlowski B., Smolka J., Powroznik P., *Recognition of Tennis Shots Using Convolutional Neural Networks Based on Three-Dimensional Data,* [in:] *Man-Machine Interactions 6: 6th International Conference on Man-Machine Interactions,* Springer International Publishing, 2020, p. 146–155.

[103] Perri T., Reid M., Murphy A.P., Howle K., Duffield R., *Validating an Algorithm from a Trunk-Mounted Wearable Sensor for Detecting Stroke Events In Tennis,* "Journal of Sports Sciences" 2020, vol. 40(10), p. 1168–1174.

[104] Skublewska-Paszkowska M., Dekundy M., Lukasik E., Smolka J., *Analysis of Two Methods Indicating the Shoulder Joint Angles Using Three Dimensional Data,* "Physical Activity Review" 2018, vol. 6, p. 37–44.

[105] Skublewska-Paszkowska M., Lukasik E., Smolka J., *Algorithms for Tennis Racket Analysis Based on Motion Data,* "Advances in Science and Technology Research Journal" 2016, vol. 10(31), p. 255–262.

[106] Hakkoum H., Abnane I., Idri A., *Interpretability in the Medical Field: A Systematic Mapping and Review Study,* "Applied Soft Computing" 2022, vol. 117, 108391.

[107] Stiglic G., Kocbek P., Fijacko N., Zitnik M., Verbert K., Cilar L., *Interpretability of Machine Learning Based Prediction Models in Healthcare,* "WIREs Data Mining and Knowledge Discovery" 2020, vol. 10. doi: http://dx.doi.org/10.1002/widm.1379.

[108] Won M., Chun S., Serra X., V*isualizing and Understanding Self-Attention Based Music Tagging, arXiv 2019,* preprint arXiv:1911.04385.

[109] Selvaraju R.R., Cogswell M., Das A., Vedantam R., Parikh D., Batra, D., Grad-CAM: *Visual Explanations from Deep Networks Via Gradient-Based Localization,* "International Journal of Computer Vision" 2020, 128, p. 336–359.

[110] Skublewska-Paszkowska, M., Powroznik, P., Karczmarek, P., Lukasik, E., *Aggregation of Tennis Groundstrokes on the Basis of the Choquet Integral and Its Generalizations*, [in:] *2022 IEEE International Conference on Fuzzy Systems (FUZZIEEE)*, IEEE 2022, p. 1–8. doi: 10.1109/FUZZ-IEEE55066.2022.9882592.

[111] Skublewska-Paszkowska, M., Karczmarek, P., Powroznik, P., Lukasik E., Smolka J., Dolecki M., *Aggregation of Tennis Multivariate Time-Series Using the Choquet Integral and Its Generalizations* [in:] *2023 IEEE International Conference on Fuzzy Systems (FUZZ)*, IEEE 2023, p. 1–6. doi:10.1109/FUZZ52849.2023.10309746.