Classifying Karate Kumite Actions

A Novel Karate Kumite Dataset for Human Action Recognition

Bachelor Thesis

Faculty of Science, University of Bern

submitted by

Alessio Petrone

from Bellmund, Switzerland

Supervision:

PD Dr. Kaspar Riesen

Institute of Computer Science (INF)

University of Bern, Switzerland

Abstract

Human Action Recognition (HAR) has been explored extensively, yet no existing dataset addresses fine-grained human interactions specific to martial arts, particularly Karate Kumite. This research introduces a novel dataset tailored to this unique application, filling a critical gap in HAR research.

The dataset includes over 35,000 videos featuring six distinct action classes recorded from four perspectives, simulating the judging process used in World Karate Federation (WKF) competitions. Eight athletes perform in five different configurations, representing point-scoring attempts and competitive dynamics. Action classes cover official scoring gestures by judges (1, 2, 3 points), failed attempts ("no point"), inactivity ("nothing happening"), and competitors exiting the competition platform ("jogai").

Ground truth labels were created by evaluating videos from all perspectives simultaneously, enabling comprehensive judgment similar to human evaluation in real competitions. This multi-view assessment captures contextual nuances and action dynamics effectively.

To establish benchmarks, two models were tested. The RGB-based 3D-Xception model performed well, achieving up to 91% accuracy in binary tasks but struggling with six-class classification, reaching only 45% accuracy. The skeleton-based BRNN model demonstrated performance near random chance, indicating a failure to generalize effectively.

The study also explores a voting system of "x out of 4" perspectives, similar to the approach used in competitions for assigning scores by judges. When using this system with x=2, the accuracy for all binary classifiers improved by at least 3%, while for the multiclass classifier, the accuracy increased by 10%.

The research highlights the potential and limitations of HAR models in martial arts judging. The creation of the Karate Kumite dataset provides a valuable resource for advancing HAR research. Initial benchmarks indicate that existing models still require significant improvements to handle the complexities of multiclass action recognition effectively. The proposed voting system demonstrates promise for enhancing model robustness in real-world judging scenarios.

Acknowledgements

I am grateful to the University of Bern for supplying the Camera setup used to capture the novel dataset and for providing computation time on UBELIX, their high- performance computing cluster. Additionally, I extend my thanks to PD Dr. Kaspar Riesen for super- vising this thesis.

Contents

Chapter 1

Introduction

In this chapter the broad perspective of the topic is developed in section 1.1, the research questions are presented in section 1.2 and the road-map of this thesis is structured in section 1.3.

1.1 Overview of Human Action Recognition and Karate

Before delving into Karate-specific aspects, it is essential to provide an overview of Human Action Recognition within a broader context. This introduction will establish a foundational understanding of its general principles and applications. Following this, the focus will shift specifically to Karate, discussing its unique characteristics and how it fits into the broader HAR landscape.

1.1.1 Human Action Recognition (HAR)

In the last few years, with Artificial Intelligence being more present in our daily lives, there is a need for more sophisticated and automated processes. When a persons action is part of the process to be analyzed and recognized, the field is called Human Action Recognition or Human Activity Recognition (HAR). Health care, Surveillance, Sports, and others are all examples of topics of HAR.

As the reader can see in Figure 1.1, HAR is mainly part of the computer vision field, but it can also be related to adjacent fields in a select few topics. Computer Vision can be further split into Object Detection, Image Classification, Image Segmentation and HAR, just to name a few. A problem is part of HAR, when the human is the subject of the recognition task. This makes the boundaries between the fields a bit more hazy. One of the possible objects being detected in Object

Classification can be a person too and thus human. Also in Image Classification, Humans can be one of the Classes to be recognized and classified.

HAR is of great importance because it can be applied in various fields where the details of human actions are of utmost importance. In Surveillance systems, reviewing all the footage cameras have taken, to spot a particular detail for instance or in sports trying to compare a situation to various criteria in a split-second to make a decision and many more. The reason why HAR is applicable to such important moments, is largely because the details may be almost impossible to spot by the human eye. It can just as well be a matter of being too annoying and time consuming to do it manually for a person. However, it is just a matter of computation power for neural networks.

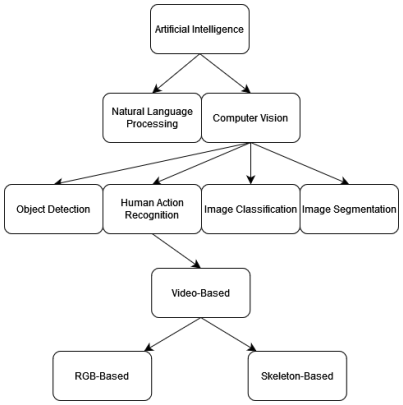


Figure 1.1: HAR into a broader context is mainly part of computer vision and can be named as HAR, as long as Humans or more precisely their actions are part of the classification task.

If given the chance and done well, HAR or artificial intelligence has the potential to simplify and improve Human tasks. An example for one such improvement in the last couple of years is the identification of possible diagnoses and treatment planning for cancer for instance. This is possible, because the AI looks at the data objectively or at least always by the same standard depending on the data it has been feed and trained with. For this reason, this standard is called the ground truth. Accordingly, we can see how essential the data we supply to the system is. It needs to be of good quality and capture exactly what needs to be classified or recognized for the model to actually learn, what we want it to. More details about this can be found in chapter 3.

So if HAR is applicable to various sports for split-second decisions from human perspective and also decides "objectively" when something corresponds to a pre-trained state or not, then theoretically it should be able to judge Karate techniques or give out points in a competition.

1.1.2 Karate Kumite

Karate is a unarmed martial arts for self-defense and can be practiced traditionally or competitively. It is used in various parts of the world and has an extensive

community.

When Karate is done as a competitive sport under the main organization of the World Karate Federation (WKF), it can be divided into two main disciplines: Kata (forms) and Kumite (sparring). We will focus on Kumite, but for the understanding of the reader a quick summary of Kata can be found in the appendix in section A.1

In Kumite the whole process works slightly different and is prone to less variations. But because sparring is done with two people together and every interaction happens fast, the evaluation in a fight might be more prone to errors and can be more difficult to judge. In general in a fight, there are approximately six to seven allowed techniques, such as straight punches and round, side, front and rear kicks as well as a variation of the roundkick and foot sweeps. Apart from all the attacking techniques, all blocking techniques are allowed. Furthermore, nowadays each competitor wears protective equipment such as gloves, boots, and more.

When a technique lands on one of the allowed spots on the body of the opponent without being blocked and while fulfilling some quality criteria, the athlete can achieve between one to three points. In a competition, a fights duration depends on the age category between one and a half to three minutes. When a point is being scored and a minimum of two of the four judges at the side of the fighting ring (tatami) indicate the same point, the referee stops the fight and the point is awarded by being indicated correspondingly. The referee may also interrupt a fight to hand out warnings in case he sees a foul. When the time has run out or one of the athletes has eight or more points difference to the other athlete or one of the athletes has received more than the allowed amount of warnings, the fight ends.

As the judges have a big responsibility and almost no pause in between fights, their judgment can vary strongly from fight to fight and may also be erroneous at times. Even though there may be some fail-safes for the athletes sport coaches to review some decisions, many errors still occur. An AI powered tool may be of help in such a case to assist the human judges to avoid unnecessary errors.

1.2 Specific Research Questions

Given that the computation power of normal devices in our daily life is getting more and more powerful, it stands to reason, that more and more complex problems of our daily life may be approached in an automatized and AI-powered approach. There already exist papers with many classical classification problems and many different datasets for various purposes. With this thesis the capabilities of two state of the art models in computer vision are tested on a new self created dataset with two athletes doing competitive Karate Kumite.

Therefore the leading questions of this thesis are the following:

- What are the key characteristics of a model capable of judging a fight, and what types of data are essential for its development?
- To what extent can Human Action Recognition serve as a substitute for human karate judges in prominent scenarios?
- What are the challenges and possibilities in using models to judge an entire fight?
- What are the technical requirements and limitations of using fight-judging models in real-time applications?
- How do human judges and AI-based judges compare in terms of accuracy, fairness, and consistency in fight evaluation?

Thus the goals are firstly to create a respresentative dataset for Karate Kumite and secondly to test this dataset on already existing models and therefore create some first benchmarks for this dataset.

1.3 Structural Road-map

To answer the before mentioned questions of the thesis, there is a need to go through a couple of topics first to understand the intricacies.

As such, this thesis will be guided by first starting with all the theoretical explanations and descriptions of the foundational concepts in chapter 2. Then going over the data acquisition and all the transformation for effectively using the acquired data, as well as defining the ground truth in chapter 3. At last the experiment on the chosen data and the evaluation of the dataset by using the selected few models in chapter 4. After the Evaluation a quick overview of the results with a second look at the research question is going to be made as well as a look at possible future work in chapter 5.

Chapter 2

Foundational Concepts

In this Chapter the description and development of important concepts like Human Action Recognition and Karate Kumite in section 2.1, as well as more technical concepts like Convolutional Neural Networks (CNN) in section 2.2 and Long Short Term Memory (LSTM) in section 2.3 in relation to Human Action Recognition is made. Lastly a closer look at existing datasets in regards to HAR and their limitations is made in section 2.4.

2.1 HAR and Karate Kumite

In this section, HAR is formally defined, and the fundamental Karate rules of WKF Kumite competitions are outlined to provide a foundational understanding.

2.1.1 HAR

As an important field in Computer Vision, Human Action Recognition can be further scaled into video-based, Sensor-based, multi-modal HAR and many more. The difference mainly lies in which sensors are being used for the Recordings. For instance, in the conference paper from Wenchao Jiang et al. [1] wearable sensors are used for activity recognition. In the paper from Jamie Shotton et al. [2] they use depth cameras for joint and therefore 3D pose estimation from still images. In the paper from Yangfan Sun et al.[3] they introduce a privacy-preserving fall detection from mmwave radar signals.

Even though many different sensors can be used, the most common are still RGB videos. Regardless of which sensor is used in the end, what they all have in common, is their goal to recognize Human Actions. As such they have many possible applications. Surveillance systems, Health care, Sports, and many more. In this thesis we are limiting ourselves to the application of RGB video-based Human

Action Recognition on Karate and further transformations on top for skeleton-based Human Action Recognition.

First of all, HAR should be defined more precisely. HAR can be considered as such, when a system can accurately describe human actions and their interactions through before mentioned sensors. This process typically involves several hierarchical steps according to Hieu H. Pham et al. [4]: Human Detection, Human Segmentation, Feature Extraction and Action Classification.

Though HAR is being applied in various fields and has gained on popularity, there are still challenges when trying to implement such a system. As Pawan Kamar Singh et al. remark in their paper [5], many of those challenges arise because of the unpredictability of Human motion. Inter-class similarity (similarity between different classes of actions), intra-class variations (variations of the same class executed by different individuals) are examples of such unpredictability. While illumination, camera angles and resolution are problems Pawan Kamar Singh et al. mention, they can be circumvented by good preparations and by controlling them to a certain extent.

2.1.2 Karate

Let us begin by getting to know some of the main terms used when dealing with Karate. The field where the athletes compete in is called a **tatami**. The karate athletes are formally called **Karatekas**. When two athletes compete together, they will each have either blue or red equipment for being distinguishable. The red athlete is called **Aka**, while the blue athlete is called **Ao**. A Karateka which takes the initiative to attack is called **Tori** while the one defending or countering is called **Uke**. For a more complete list of the Karate terms, consult the appendix section A.2.

Karate in general is not only a sport, but also an art. It has a fine balance between tradition, health and sport. Karate is often also referenced as a way of life and as self defense, which is apt, considering, that it has many philosophical and meditative components in it. Even when doing karate as a competitive sport, there are many traditional rules, like the bowing or the breathing which is part of a good form in techniques. Karate has two main disciplines and in the following subsection we are going into more details regarding Kumite and its competitive rules regarding the World Karate Federation. A complete list set of the Rules for 2024 can be found on World Karate Federation (WKF).

WKF Kumite Competition

The Setup in a WKF Kumite Competition follows a simple setup, which should allow the referee and judges the optimal angles to succeed their responsibilities as such. In total there are seven acting referees on one tatami at a time. Each of their responsibilities is slightly different but should when combined allow the optimal proceeding of the competition. We will briefly talk about the responsibilities of the main referee and the four judges. For more details about the helpers, the supervisor, the tatami manager, refer to the appendix section A.3

In a direct encounter of 1.5 - 3 minutes, the Karateka, which scores more points than the opponent and has done less fouls than is maximally allowed, wins. The maximally allowed fouls are five, but in exceptional cases two fouls may be awarded together. In case of a draw, the Karateka, which has scored the first point, wins. In case of losing the advantage of the first scorer because of a foul, the Karateka, which gets selected by a majority of the judges and the main referee, wins. To hand out points and fouls, as well as the the final decision of the winner in case of a draw, everything needs to follow some quality criteria. More details about the criteria will be discussed in section 2.1.2.

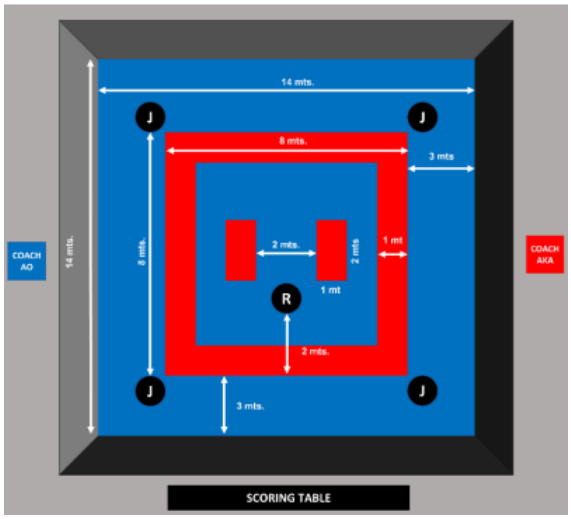


Figure 2.1: This is the tatami setup for Kumite. The fighting happens inside the red square. The J's around the corners of the red square stand for the judges and the R for the referee. When the referee starts, pauses or ends the fight the Karatekas always return to the red in the middle of the field.

Points

To understand point scoring events from the perspective of the judges, it is essential to know how different techniques are evaluated based on their execution and target area. There are three different types of points that can be scored. Every allowed punching technique to any allowed body part of the standing opponent is worth 1 point. Every allowed leg technique is inherently worth 2 points when executed at the torso and 3 points when executed at the head of the opponent. When doing a sweeping technique, where the opponent falls to the ground with a directly following punching technique, it is also worth 3 points.

In order for the technique to be considered a score, the technique needs to have the potential to be effective, if it had not been controlled by the athlete. The technique also has to acquit to the following criteria:

1. Good form (Properly executed technique).
2. Sporting attitude (Delivered without intent to cause injury).
3. Vigorous application (Delivery with speed and power).
4. Maintaining awareness of the opponent both during and after the execution of the technique (Not turning away or falling down after completing a technique – unless the fall is caused by a foul by the opponent).
5. Good timing (Delivery of the technique at the correct moment).
6. Correct distance (Delivery at a distance where the technique would be effective).

All of these criteria are needed for a technique to be counted as a valid point. After discussing the different types of points and the criteria for scoring, the conditions under which a point can be awarded from the perspective of the main referee are examined.

- A point will only be awarded, if two or more judges indicate the same point potential.
- In case both Karatekas execute a hit in a situation, only the first point which matches all the point scoring criteria in the perspective of the judges will be indicated.
- In case of one of the two Karatekas executing techniques of different point potential, only the point will be awarded, which has more than two indications.
- In case of having one judge indicating a lower potential and another judge indicating a higher potential point for the same Karateka, the lower point will be awarded.
- In case of the Referee spoting a foul for a Karateka which has indicated but yet unawarded points, it supersedes the point awarding.

.

To understand when to award a point better, some examples are needed, you can find them in the appendix section A.6.

Fouls

To understand when a foul gets indicated, it is important to understand how fouls can happen. Even for a foul the criteria for scoring a point may be essential. It remains to mention, that the body parts a Karateka is allowed to hit, are the torso and the head only. Joints or anywhere below the belt, legs inclusive, are not allowed to be hit. The head and neck in particular may only be "skin touched". If one of the six criteria for scoring a point is not fulfilled but instead has an "excess", it will count as a foul.

For Example when Aka attacks with a punch to the head of the adversary and is too close (criterion correct distance not fulfilled), the hit will be much harder than a skin touch. Most foul types exist, in order to prevent harm to the Karatekas. An other example for such a foul is, when Aka hits Ao deliberately on a dangerous part of the body like a joint, or with a dangerous or prohibited technique. The above mentioned excess are just a few out of many. A complete list can be found in the appendix in the table Table A.3

Fouls which make a fight more strategic or more structured exist as well. An example for the former kind of foul is, when Aka runs away from Ao. An example for the latter is, when Aka exits the fighting ring. The latter foul is called **jogai**.

Both of these examples may also be used by athletes strategically to play for time. Jogai may for example be used by Aka when he is leading and only a few seconds are left. Aka may also exit the ring, when Ao executes a sidekick and pushes Aka out. In this case though, it will not be counted as a jogai, as it is not self inflicted.

As the reader may notice, not every foul is easy to identify and the responsibility for proposing the fouls has to come from the main referee. The foul will only be issued, after a minimum of two judges follow the proposal of the main referee. That also means, that not every foul the main referee sees, will be issued.

To underline how difficult the identification of certain situations may be and how easily they can vary, a few examples more can be found in the appendix section A.6 Also for a more complete overview of all the possible fouls, consult the appendix section A.5.

2.1.3 Shortcomings of Rules and error-handling

All the involved parties such as the referee and the judges are the main pillar of a fight, without them it is impossible to carry it out. If they just as much as blink in the wrong moment, they might miss an important detail. For example, if Ao is connecting a hit to the head of Aka with all criteria fulfilled and only Judge A and

Judge C can really capture the point criteria from their angle, but Judge A blinked in the moment when the point happened. Ao will get no point in this situation, because judge A did not see it. In the same situation but without the criteria being fulfilled entirely, Judge A is still indicating the last point, and judge C blinked in the wrong moment to see that the timing was wrong and indicates the new overlapping point. Ao will get the point even though the point should not have been given to him. Such cases may happen more often than not. This is a shortcoming which in most competitions cannot be bridged as errors and lapses in concentraition as well as the need to blink occasionally is human.

In case a point gets awarded wrongfully or not at all, even though it should have because the judges indicated it for instance, there is an option for the sportcoaches of the Karateaks to directly intervene and request a video-review. Let us take the last example from before, where judge A was still indicating the last hit and judge C wrongfully indicated a point. The coach from Aka sees the wrongdoing and requests a video review from a referee helper. If the review disproves the judges and referee, the mistake gets corrected, the fight proceeds and the coach of Aka is still allowed to request further reviews. If the coach was wrong, he can do no further video-review requests and the fight proceeds without correction. The same may be done for fouls too. This is a good approach and help for error-handling, but is not manageable for every competition because of time limitations and the resources needed and is thus not possible in most competitions.

In cases like this a real time AI tool might be helpful to mitigate the error and time constraints for such video review requests.

2.2 Convolutional Neural Networks in HAR

In this section the basic and advanced concepts of Convolutional Neural Networks (CNN) are explained as well as the used architectures for the experiments. The first subsection focuses more on the basic concepts of CNNs. We will explore and differentiate 2D-CNNs and 3D-CNNs and how 3D-CNN can be broken down to a similar complexity to 2D-CNN with the help of depth-wise and point-wise convolutions and residual connections. In the second subsection a few model architectures are presented and explained with regards to the final used model for the experiments.

2.2.1 Fundamentals of CNN Architecture

Convolutional Neural Networks are the most used model types for solving problems in computer vision. Problems such as classification, segmentation and so on. Com-

puter vision does not just include problems concerning images, but also videos, which are many images stacked in a little time frame. When processing images, the term 2D-CNN, or simply CNN, is commonly used. Incorporating the temporal dimension of videos introduces an additional dimension, leading to the use of 3D-CNN.

2D-CNN and 3D-CNN

Discussing CNNs inherently refers to a type of Artificial Neural Network (ANN). Like Keiron O'Shea and Ryan Nash said in their paper [6]: "Convolutional Neural Networks (CNNs) are analogous to traditional ANNs in that they are comprised of neurons that self-optimise through learning". That already tells us plenty of how CNNs work. A notable difference is that CNNs are primarily used in the field of pattern recognition for images. By encoding image-specific features directly into their architecture, CNNs significantly reduce the number of parameters needed, enhancing efficiency and performance in image-related applications.

Images mainly consist of three dimensions, two of them spatial (height and width of the image) and one in depth. The depth in images stands for the coloring and is often called channels. When using gray-scale images, the channel used is just one. When using colored images, the channels it consists of are three.

When using CNNs to process an image, the image is carried through layer after layer, where the number of neurons on the first layer consist of height by width by channels $h \times w \times c$ of the input image. The last layer holds the amount of neurons of classes in the used dataset. The layers in between the input layer and the last layer are called hidden layers and are mostly either convolutional layers, pooling layers or fully connected layers.

Let us break down the simple CNN in Figure 2.2 as an example. The first layer is the input layer and holds the pixel values of the image. The second layer here is a convolutional layer and determines the output of neurons by calculating the scalar product between the weights and the input and runs it through the activation function, which is often times the rectified linear unit (ReLu). The next layer is a pooling layer and reduces the size of the features of every dimension. The last layer is a

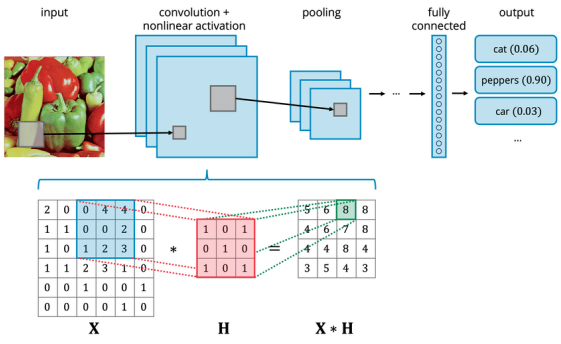


Figure 2.2: Simplification of a CNN architecture with an image of vegetables going through one convolutional layer, a pooling layer, followed by a fully connected layer and a classification at the end.[7]

fully connected layer and connects all the neurons together and runs them trough an activation function to get a score which is also the prediction. For a more detailed and mathematical approach the reader may take a look at the paper from Alex Krizhevsky et al.[8]

Depending on the kernel (or filter) size, stride size and the padding of a convolutional layer, as well as the defined output channels of the layer, the feature representation that results, may be completely different after an activation. Let us look at the example in Figure 2.2 The filter is a 3x3 with no padding, and stride of 1x1 and the output channels should be three. The image, in our case just random numbers which shall represent the pixels, is of size 6x6. When going through the first filter we may get 5, stride the 3x3 grid to once to right and get 6 and the next one, which is also the green colored one, results in 8. Using a 4x4 filter would produce a completely different result, as more pixels per stride are considered. Selecting additional output channels would increase feature representations and, consequently, the number of parameters. In the context of the image, the output channels correspond to the stacked blue squares. A larger stride would skip more pixels per step, covering fewer individual pixels. With a padding the size can be prevented from getting smaller by padding the outer rows and columns with zeros. The pooling layer also has a filter and stride, which work the same way, but here have a direct impact on the output size. Although this is not so for the output channels. The fully connected layers do not have to define all the different parameters. Fully connected layers just connect neurons with the adjacent layer of neurons. The output layer before the soft max, which gives the score representation, is generally a fully connected layer.

3D-CNN work exactly the same as 2D-CNN do, but with the additional dimension of time. Time is often also described as depth. It is not the same depth as channels though. Consider applying a convolutional layer to a 4-frame video. The process is similar to handling a single image, but the four frames are stacked along an additional dimension. Instead of using $h \times w \times c$, the depth is introduced, resulting in $h \times w \times d \times c$.

Separable Convolutions

Of course this additional dimension makes the computation and amount of parameters a lot higher. According to Francois Chollet et al. [9] there is a way to reduce the computation time and make it more efficient by introducing separable convolutions. Separable convolutions are convolutions, that separate the way the filter works. As you can see in Figure 2.3. That means instead of doing a 3x3x3 filter, a 1x3x3 is used first as a depthwise convolution, and then on the result a 1x1x1

pointwise convolution.

As Francois Chollet et al. describe in their paper, first a depth-wise convolution gets applied to reduce the dimensions separately and after that, a point-wise convolution (1x1) is applied to project the channels onto a new channel. This makes it much more efficient and faster, but at the cost of context which may go missing through the separation. Still, the author of the paper calls this concept likely to become a fundamental component of CNN design.

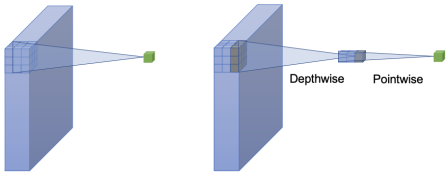


Figure 2.3: A Depthwise convolution of 1x3x3 followed by a pointwise convolution of 1x1x1 is computed instead of the traditional 3x3x3 convolution.

Residual Connections

When we start using different architectures, we notice, that accuracy of a model gets saturated and then degrades quickly, as the depth of the model increases. This is also called the degradation problem. This degradation is not due to overfitting, but rather due to the difficulty of the optimization in deeper networks. To solve this problem Kaiming He et al. [10] propose to use residual connections.

Residual connections, also known as shortcut connections, are a key concept in the design of Residual Networks (ResNets). They are a type of connection in a neural network that allows the output of a layer to bypass one or more subsequent layers and be added directly to the output of those layers. This is typically achieved by performing an identity mapping, where the output of a layer is added to the output of a layer further down the network, without any additional parameters or computational complexity.

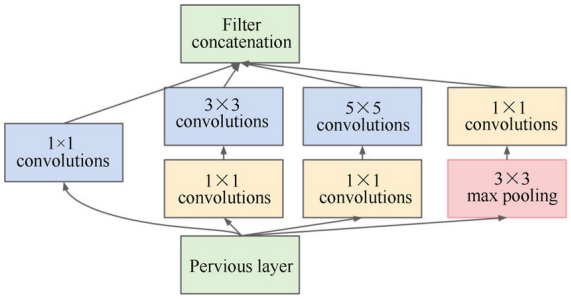


Figure 2.4: Illustration of the Inception Module, showcasing parallel operations on the input, including 1×1, 3×3, and 5×5 convolutions, as well as 3×3 max pooling. Each operation preserves spatial dimensions with 1×1 convolutions applied to reduce computational cost. Outputs from all branches are concatenated to form the final feature map.

The purpose of residual connections is to mitigate the degradation problem that occurs in very deep neural networks, where adding more layers can lead to higher training error. By using residual connections, the network can learn to perform identity mappings more easily, which can help in training deeper networks.

If the optimal function for a set of layers is close to the identity mapping, it is easier for the network to learn the necessary adjustments or residuals as they are called, with reference to the identity mapping, rather than learning the entire function from scratch.

In the context of the paper from Kaiming et al. [10], residual connections are used to create residual blocks, which are the building blocks of ResNets. These blocks are designed to learn residual functions with reference to the input and the output of the block is the sum of the input and the residual function learned by the block's layers. This approach has been shown to ease the training of very deep networks and has led to significant improvements in image recognition tasks. One such block can be seen in Figure 2.4. More about the inception module in the next subsection.

2.2.2 Review of CNN Models

The CNN architecture exists for many years now and there have been many papers, which have tried and succeeded at optimizing some short comings of the original by adding new kind of layers, making it deeper or applying different layers in parallel. In this subsection we will cover the Inception and the Xception model architecture.

Inception

Introduced the first time in the ImageNet Large-Scale Visual Recognition Challenge 2014 and later presented in a paper [11], the inception architectures main idea is to create a deep CNN, that can efficiently utilize computational resources while achieving high performance in image classification and detection tasks. This is achieved by a few key concepts: Inception Module, Dimension Reduction, Sparse Structure Approximation, Efficient Dense Computations, Balanced Computational Budget and Auxiliary Classifiers. To get the gist of it, let us take a brief closer look at the named key concepts.

The inception module is a building block, that allows the network to process visual information at various scales in parallel. As already mentioned in section 2.2.1, different kinds of filter size produce different kinds of results, because they focus on more or less context. The inception module introduces four different convolutions in parallel (1x1, 3x3, 5x5, max-pooling) and concatenates the results before handing it over as an input to the next layer. In Figure 2.4 the reader may look at the inception module.

To prevent an enormous computational complexity, the inception module uses 1x1 convolutions to reduce the dimensionality of the input before applying the more

expensive 3x3 and 5x5 convolutions. This reduces the number of input channels to these larger filters and thereby decreases the computational load, making it more efficient in the process.

Sparse Structure Approximation is what the authors of the Inception model call a clustering of neurons by correlated outputs and connecting them to the previous layer. This makes the network able to process the visual information more efficiently. Instead of using non-uniform sparse data structure, which is inefficient, the inception architecture uses dense components that can exploit optimized numerical libraries. This is what the authors call Efficient Dense Computations.

With Balanced Computational Budget, the design is guided by a computational budget, so that the network does not become too large and computationally expensive.

Upon considering all the facts metioned above, let us conclude with Auxiliary Classifiers. During training, the network includes additional or as they call it auxiliary classifiers, connected to intermediate layers, so that propagating gradients back is more efficient, as it does not need additional resources because of the better training of deeper layers.

The inception architecture is a significant advancement in the field of computer vision, offering a practical and efficient approach to constructing deep neural networks.

Xception

The Xception model, introduced by Francois Chollet in the paper "Xception: Deep Learning with depthwise Separable Convolutions" [9], is a CNN architecture that builds upon the inception model. The key innovation of Xception is the replacement of Inception modules with depthwise separable convolutions, which are more efficient and effective at utilizing model parameters. The Inception models are known for their cross-channel and spatial correlations separately. Xception takes this concept a step further by separating the dimensions completely. This is achieved through the use of depthwise separable convolutions as described in the last paragraph of section 2.2.1.

The Xception architecture is designed to be simple and modular, making it easy to define and modify. It has 14 modules with linear residual connections [10] around them, except the first and last modules. This design choice, along with the use of the mentioned separable convolutions [9], contributes to the models efficiency and performance.

In Summary, the model represents a significant advancement in the design of CNN, offering a more efficient architecture that simplifies the network while main-

taining or improving performance on image classification task. A closer look at the implemented Xception architecture can be taken in the appendix section A.7.

2.3 LSTM and Pose Estimation in HAR

In this section Recurrent Neural Networks and Long-Short-Term-Memory will be explained. Further, Pose Estimation in HAR is briefly introduced for feature extraction and at the end, the in the experiments used BRNN model is explained in regards to the extracted features.

2.3.1 Fundamentals of LSTM

Reccurrent Neural Networks

Recurrent Neural Networks (RNNs) are a class of artificial neural networks designed to handle sequential data, such as time series, text sequences, and other data where the order of the elements matters. Unlike traditional feedforward neural networks, which process input data independently, RNNs have connections that allow them to maintain a memory of previous inputs, which is crucial for understanding context in sequential data.

The key to this memory is the hidden state, which acts as a kind of scratchpad for the network. As the RNN processes each element in a sequence, the hidden state is updated based on the current input and the previous hidden state. In this way, the network is able to keep track of information it has seen before, which is essential for tasks where the order of the data is important.

RNNs achieve this through recurrent connections, which loop back from the output of the network to the input. This creates a feedback loop that allows the network to maintain information across time steps. When an RNN is unrolled in time, it can be seen, that it is essentially a chain of repeating modules, each passing information to the next.

Training an RNN involves a process called Backpropagation Through Time (BPTT), which is a variation of the standard backpropagation algorithm used in feedforward networks. BPTT works by unrolling the network through all time steps and then computing the gradients of the error with respect to the weights across all these steps.

However, RNNs can face challenges such as vanishing and exploding gradients. Vanishing gradients occur when the gradients become too small during backpropagation, making it difficult for the network to learn long-range dependencies. Contrarily, exploding gradients happen when the gradients grow too large, leading to

numerical instability.

To overcome these issues, more sophisticated RNN architectures have been developed, like LSTM and Gated Recurrent Units (GRU). These models include special mechanisms, such as gates, that control the flow of information and help to maintain long-term dependencies in the data.

In summary, RNNs are powerful tools to work with sequential data, capable of learning complex patterns and relationships over time. Their ability to maintain a hidden state and process sequences makes them a go-to choice for many applications in natural language processing, time series analysis, and beyond.

LSTM cells

In the paper "Deep Learning with Long Short-Term Memory for Time Series Prediction" [12] LSTM networks are introduced as a specialized type of RNN designed to overcome the limitations of standard RNNs in learning long-range dependencies in time series data.

LSTMs are equipped with a unique structure called a memory block, see Figure 2.5, which is the core component that enhances their ability to model long-term dependencies. This memory block contains three types of gates: input gates, output gates, and forget gates. These gates are composed of multiplicative units and are responsible for controlling the flow of information into and out of the memory cell.

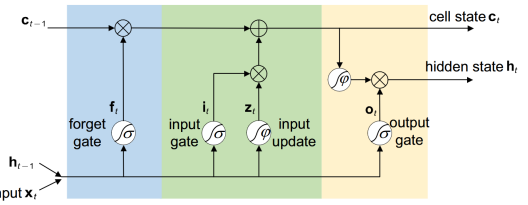


Figure 2.5: Structure of an LSTM memory block showing the flow of information through the forget gate (f_t), input gate (i_t), input update (z_t), and output gate (o_t). These gates control the updates to the cell state (c_t) and the hidden state (h_t), enabling the network to selectively remember or forget information over time. The sigmoid (σ) and tanh (ϕ) activations regulate the gating operations.

The input gates determine how much new information is allowed to flow into the memory cell, while the forget gates decide how much of the previous information should be retained. The output gates then control how much of the memory cell's information is used to compute the output activation of the memory block.

The paper explains that the memory cell within the LSTM is in charge of remembering the temporal state of the neural network. The gates work in conjunction with the memory cell to ensure that the LSTM can effectively handle long-term dependencies in time series data.

In summary, the paper presents LSTMs as a robust solution for time series

prediction tasks, capable of capturing complex patterns and dependencies over time due to their specialized architecture with memory cells and gates.

Bidirectional Recurrent Neural Network

A bidirectional RNN (BRNN) consists of two separate RNN layers: one that processes the input sequence in the normal, forward direction, and another that processes the input sequence in the reverse direction. Each layer scans the sequence from one end to the other, producing outputs at each time step. The outputs from both the forward and backward layers are then typically combined at each time step, either by concatenation or by summation, to form the final output of the bidirectional RNN [13].

The bidirectional approach is particularly useful in tasks, where the context from both the past and the future is important for understanding the current input. For example in natural language processing tasks like machine translation, knowing the words that come before and after a particular word can significantly improve the model's ability to understand the meaning of the sentence.

In case of the hierarchical recurrent neural network for skeleton-based action recognition, making the RNN bidirectional allows the network to better capture the temporal dynamics of human actions by considering the movement of skeleton joints both before and after the current frame. This leads to more accurate recognition of actions, as the network can leverage the context from the entire sequence, not just the preceding frames [13].

2.3.2 You Only Look Once

The You Only Look Once (YOLO) object detection system is a unified, real-time model that processes images to predict bounding boxes and class probabilities in one evaluation. Developed by Joseph Redmon et al.[14], YOLO is designed to be simple, fast, and accurate. It divides the input image into a grid and each cell in the grid predicts bounding boxes and class probabilities for objects whose centers fall within the cell.

YOLO uses a convolutional neural network with 24 convolutional layers followed by 2 fully connected layers, inspired by the GoogLeNet architecture. The network is trained to minimize a loss function that includes terms for bounding box coordinates and confidence, with adjustments to prioritize the localization of objects over the confidence of empty cells.

The system is trained on full images and directly optimizes detection performance, which allows it to maintain high average precision while being extremely

fast. YOLO can process images in real-time at 45 frames per second, with a smaller, faster version capable of processing 155 frames per second.

YOLO outperforms other real-time systems in terms of mean average precision and is less likely to produce false positives on background than other methods. However, it struggles with localization accuracy, especially for small objects. Despite this, YOLO's generalizability is high, making it robust for various applications and domains, including real-time video processing.

YOLO-Pose is an innovative extension of the YOLO object detection framework, specifically engineered to excel in the task of multi-person pose estimation. Unlike its predecessor, which is primarily focused on detecting and classifying a wide range of objects within an image, YOLO-Pose is tailored to not only identify individuals but also to meticulously map out their 2D poses by pinpointing key body joints. This specialized application sets YOLO-Pose apart, as it delves into the intricacies of human form and posture within visual data.[15]

The core distinction between YOLO-Pose and the standard YOLO model lies in their respective outputs and methodologies. YOLO object detection delivers bounding boxes encapsulating objects and their corresponding class labels, functioning as a versatile tool for general object recognition. YOLO-Pose, however, enhances this by providing not just the bounding boxes for humans but also a detailed skeleton map, indicating the precise locations of various body joints for each person detected. This advancement is achieved through a heatmap-free approach that directly optimizes the Object Keypoint Similarity (OKS) metric, a departure from conventional two-stage methods reliant on heatmaps.

Training and optimization strategies also diverge between the two models. YOLO object detection is honed to maximize accuracy in object localization and classification, whereas YOLO-Pose is trained end-to-end to simultaneously optimize for both bounding box precision and the OKS metric, a hallmark of pose estimation tasks.

Both models are celebrated for their computational efficiency, making them contenders for real-time processing. However, YOLO-Pose introduces the additional complexity of pose estimation while maintaining the framework's renowned speed and efficiency. This balance makes YOLO-Pose particularly well-suited for applications that demand a nuanced understanding of human figures, such as action recognition, augmented reality, and interactive systems.

In summary, YOLO-Pose builds upon the robust foundation of the YOLO framework, expanding its utility to cater to the specific needs of human pose estimation with its unique ability to provide detailed skeleton maps alongside object detection.

2.3.3 Skeleton Based Action Recognition with BRNN

Skeleton based action Recognition is straight forward in what it means. As already mentioned in subsection 2.3.2, the joints and thereby the skeleton from people can be extracted from images. When trying to analyse action pattern like in HAR based on the extracted skeletons, it is called Skeleton based action Recognition (SB-AR).

For SB-AR there are many approaches, from the using Graph Convolutional Networks to using RNNs and many others. In this thesis we have reconstructed a model from the paper "Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition" from Yong Du et al. [13]

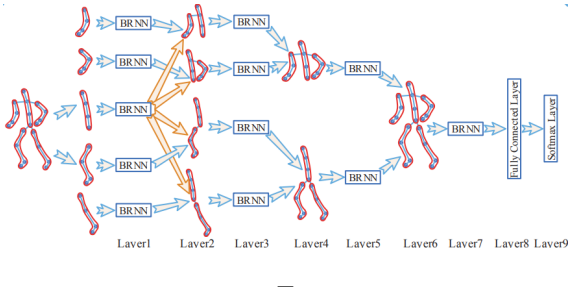


Figure 2.6: Novel method proposed by Yong Du et al. [13] to make a prediction about an action by analysing the skeletons. The skeleton is first separated into their limbs and layer after layer put back together.

The authors of the paper proposed hierarchical recurrent neural network (RNN) for SB-AR divide the human skeleton into five parts: two arms, two legs and a trunk, see Figure 2.6. This division is significant as it allows the network to model the movements of these individual parts and their combinations, which is crucial for effectively recognizing various human actions.

The significance of dividing the human skeleton into these parts lies in the fact that simple human actions are often performed by only one part of the body, such as punching with the arms or kicking with the legs. More complex actions , such as running, involve the coordination of multiple parts, which require the synchronized movement of arms, legs, and the trunk.

By separately processing the data from each skeleton part through its own bidirectional RNN (BRNN), the network can learn the spatial and temporal features specific to each body part. As the representations from these subnets are hierarchically fused in higher layers of the network, the model can capture the interactions between different body parts and recognize actions that are composed of movements from multiple parts of the body.

This part-based feature extraction and hierarchical fusion approach enhance the network's ability to model the complexity and variety of human actions. It also enables the network to focus on the most informative joints and movements for action recognition, potentially leading to improved accuracy and robustness in classification tasks.

2.4 Existing Datasets and Their Limitations

HAR is a large field with many possible applications. Sports is one of the fields that can profit the most from HAR. There are already many occurrences in sports, where some systems use AI to evaluate human situations. In the table **??** there are a few of the sport oriented datasets. As of now however, no dataset is focused on a competitive background in martial arts or simple bouts in a martial arts. This can stem from the complexity of the situation of bouts. Two competitors directly interact with each other in close proximity and most of the time with many other people present at the scene. There are already sophisticated classifiers for speech, emotions, body language, simple actions with distinct features. But there are not any, which try to classify complex movements, which may look similar to each other in martial arts. This legitimates the question and the demand for such a dataset. To address this gap, we propose a new dataset centered on competitive Karate Kumite. The next section discusses this dataset in detail, covering its creation process, structure, and features. This dataset is intended for evaluation using the two models referenced earlier: Xception and the BRNN model.

Dataset	**Description**	**#Videos**	**#Classes**
SportsSloMo[16]	8,000+ sports video clips across 22 categories	130,000	22
FineSports[17]	Multi-person sports videos with fine-grained annotations	10,000	12 / 52
3DYoga90[18]	RGB videos and 3D skeletons of yoga poses	6,177/5,526	90 poses
SportsHHI[19]	Human-human interactions in basketball and volleyball	11,398 keyframes	34
MultiSports[20]	Multi-person videos with spatio-temporal sports actions	3,200	4
KaKumite4P (ours)	Basic karate kumite interactions for RGB videos and 2D-skeletons from four perspectives	8952/35808	6

Table 2.1: Sports-related datasets with human interaction. Comparing number of videos and number of classes to our proposed dataset

Chapter 3

Dataset Acquisition

In this chapter an overview of the data acquisition process is shown with the setup in section 3.1, followed by the instructions given to the Karatekas, the organization of the data in section 3.2, the ground truth, data cleaning process and augmentation of the data in section 3.3 and at last the datasplits done for the experiments in section 3.4 and the relevance of this new dataset.

3.1 Acquisition Method and Setup

Depending on the kind of data needed the aquisition method may vary. For our Dataset only rgb video cameras are used. To be more precise, four GOPRO Hero 10 cameras are used. Because the Karateka which needs to be recorded has to move as fast and precise as in real competitions, the actual camera settings were as follows. The resolution was set on 3840x2160, while the recorded frames per second are set to 30. The cameras are used with the wide angle setting and the stabilization is set to standard.

(a) Camera A (b) Camera B (c) Camera C (d) Camera D

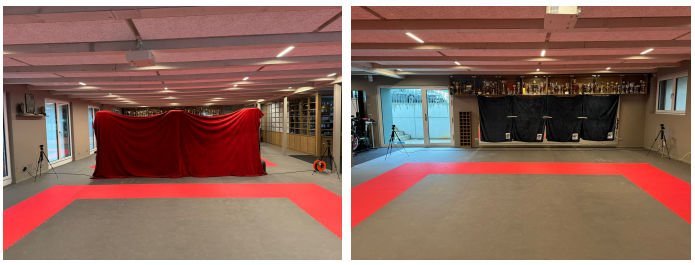
Figure 3.1: Individual camera views of the Tatami setup.

As already shown in Figure 2.1 and in Figure 3.1 the judges and the cameras respectively are situated in all four angles of a tatami. It is concluded to be the best camera positions in comparison to having them set to the middle of a side. The second position method would have made recording the fight situations in the angles down right impossible. The tatami used for the dataset is ordered into 6x6 meter instead of the standard required 8x8. The cameras were positioned one meter in the diagonal away from the tatami, exactly one meter high and with approximately eleven degrees, so that the middle of the tatami is focused directly, see Figure 3.2. Because the recordings are done on various dates and the cameras need to be removed and re-positioned each time, the positioning may vary to a small degree.
To make the recordings as clutter free as possible from variables, the mirrors in the practice facility have been occluded with covers and at the time of recording only the two confronting Karatekas are effectively observable by the four cameras, see Figure 3.3. The lighting is one of the variables which are not controlled, as there are too many windows in the room.

Figure 3.2: The camera is set up exactly one meter from the tatami corner away on the diagonal. Denoted by a is the angle between camera and the middle axis, and by b the distance from the ground to the camera lens exactly one meter up from the ground.

3.2 Instructions and Data Organisation

What always may vary, are the technique combos each athlete has in their arsenal. To make this somewhat effective and representative, the Karatekas receive some instructions before a fight. Afterwards they have the opportunity to have a fight



(a) Covering the first half of the dojo.

(b) Covering the mirrors in the second half of the dojo.

Figure 3.3: Covering the mirrors for the Camera setup is essentital, so that the athletes are not represented twice from some perspectives through the mirrors.

of 1.5 minutes and try to accomplish their objective. The objective they generally have to fulfill is, to score points. Depending on the instructions, the points they need to score may be a bit easier to achieve or more difficult to reach. In the table Table 3.1 are the five main scenarios used for the goals followed in the next paragraph. As the reader is able to see, the scenarios have different instructions for Tori and Uke, respectively for Aka and Ao.

Scenario	**Description**
1	Aka provokes Ao, Ao attacks, and Ao scores a point.
2	Aka provokes Ao, Ao attacks, and Aka absorbs the hit and counterattacks to score a point.
3	Aka provokes Ao, Ao attacks, and Aka reacts to score a point at the same time.
4	Aka and Ao fight freely with a focus on scoring a specific point combination
5	Try to press the opponent into making a foul (e.g., jogai) or dilute a point by pushing the opponent outside the ring while scoring or being scored on.

Table 3.1: Overview of training instructions and goals.

The first scenario in Table 3.1 is mainly to gather some simple points and to achieve a near optimal timing. In the second and third scenarios, the main goal is to have disturbances, so the opposing athlete tries to defend, while either absorbing and countering, or directly countering. In the fourth and fifth scenarios the goals are to make the scenes as near to the competition situations as possible. As such the first to third scenarios are used by focusing on hand techniques and once more for leg techniques. But as the hand techniques are not always just used for scoring, but also for connecting various parts of combinations together, the athletes are allowed to still use hand techniques when using leg techniques to score the points. In the seventh instructions the athletes received scenario four and fight as they normally would, entirely free. In the eighth instruction they should try to press the opponent to make a foul like jogai, or sometime even dilute a point by pushing the opponent outside of the ring while scoring or being scored on. After having done one round the athletes are instructed to change their protectors to the other color and all the instructions are repeated. In this way eight trained athletes are paired up in five different combinations for a as good a represenation sample as possible.

3.3 Ground Truth, Cleaning and Transforming

After recording and making the data usable for models to train, it is necessary to cut the videos in exactly the same amount of frames after synchronizing the perspective of the videos. After careful deliberation, the number of frames per datapoint have been chosen to be 120 frames or four seconds long. Firstly, because most situations are longer than two second and shorter than five and therefore may be better representations of fights at competitions. And Secondly, because this length is still from a computational standpoint doable in a realistic timeframe.

After cutting the Data, it is organized by recording date as a first letter, instruction change as a number (which is incremented every time the instructions are changed), Camera perspective, and lastly the number of the cut clip, like so: A0A_0001.

After cutting the 29 hours of video material and organizing the videos accordingly into the four second fights, it needs to be further organized into categories. For an easier setup and because in karate mainly only one score per athlete is given, a single-label setup is preferred.

Code	**Description**
nh	Nothing happening
np	No points
1pK	1 point - Aka
1pO	1 point - Ao
2pK	2 points - Aka
2pO	2 points - Ao
3pK	3 points - Aka
3pO	3 points - Ao
fK	Foul - Aka
fO	Foul - Ao
pE	Point each
fE	Foul each
fp	Foul and point
D	Delete

Table 3.2: Categories of outcomes and their codes.

As one of the main goals of the thesis is to have a judge substitute, it is paramount to be able to judge at least the points. So it appears meaningful to make a label category for each point type as well as the no point type. To enlargen the details, the categories are further split into who marked the first point in the four second sequence. As everything else would be just put into no points the category has been further split. Firstly into the nothing happening category, added so, that "no points" can be further differentiated into when the athletes tried to score a point but did not fulfill all the criteria (no point) or did not try to score a point (nothing happening). Secondly, for when a foul happened, four additional categories are added, a foul for each athlete color, as well as when both athletes do a foul at the same time and lastly for the case when a foul is scored but at the same time a point, the foul and point category is added.

(a) Camera perspective A

(b) Camera perspective B

(c) Camera perspective C

(d) Camera perspective D

Figure 3.4: Frame of a original video form all four perspectives. The chosen frame is one of the most important ones in an action sequence, for there are many criteria, which may fail here. The athlete in red (Aka), is trying to score 1 point with a punch, while blue (Ao) is too slow while tying to get away.

At the end, the categories are as shown in the table Table 3.2: nothing happening (nh); no points (np); 1 point - Aka (1pK); 1 point - Ao (1pO); 2 points - Aka (2pK); 2 points - Ao (2pO); 3 points - Aka (3pK); 3 points - Ao (3pO); Foul - Aka (fK); Foul - Ao (fO); Point each (pE); Foul each (fE); Foul and point (fp); Delete.

Important to note is however, that while establishing the ground truth with these categories, the videos are not judged as each one video on their own, but always four videos of the same situation together. As the main goal of the whole judge setup around the tatami is to see and indicate the point and the "at-least-two-judges"-rule (see Article 8.1 in the WKF Kumite Competition Rules) exists, so that the point not only appears as a point, but in and of itself achieves all the criteria needed, it makes sense to judge the situation as a whole from all four perspectives together.

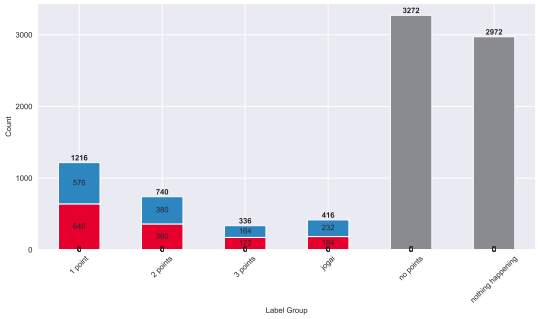
In the picture Figure 3.4 the point would not be indicated from the Figure 3.4b perspective, because it is clearly visible that the punch is missing the target. From the perspectives Figure 3.4a and Figure 3.4c we could argue, that it does not look wrong or even good. In the picture Figure 3.4d, the perspective may not give enough information as to if the point would be right or not. If we combine the perspectives, however, the picture changes. When combining all the perspectives, the result is

what in competitions is desired, but cannot be achieved, because four people would need to communicate in the split second it happens, which clearly is impossible.

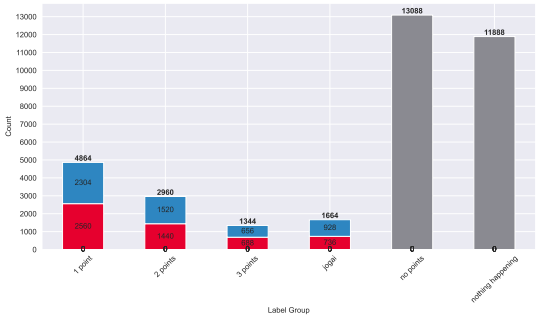
After establishing the ground truth, the data, which was deemed not usable has been discarded. Possible reasons for the cleaning are additional people or more than the athletes appearing in the video, someone being injured, one of the cameras malfunctioning or other similar reasons.

Also part of the cleaning, is going trough all the fouls again and separate the jogais from the rest, into two new categories, as the rest of the fouls are not enough to be actually trained on.

At the end of the cleaning the videodata, 8952 video files are left. Without any augmentation and transformation, the label distribution is as shown in the picture Figure 3.5a. To intensify the videodata and to have more of it for the training stage, all the videodata is augmented by mirroring the data horizontally, which already doubles the amount, and then cropped it into two big squares to double it again. The number of all the video files after this augmentation is 35808, the reader can see the distribution of the classes on Figure 3.5b. More precisely, the videos are first cropped from 3840x2160 to 3340x2160 pixels and then further into squares of the size of 2160x2160 pixels into a left-crop and a right-crop. Furthermore having squares as images or videos makes it easier for many models to be used out of the box. Additionally the pixelcolor is normalized for every pixel so that the computation of the Xception model is less resource intense.



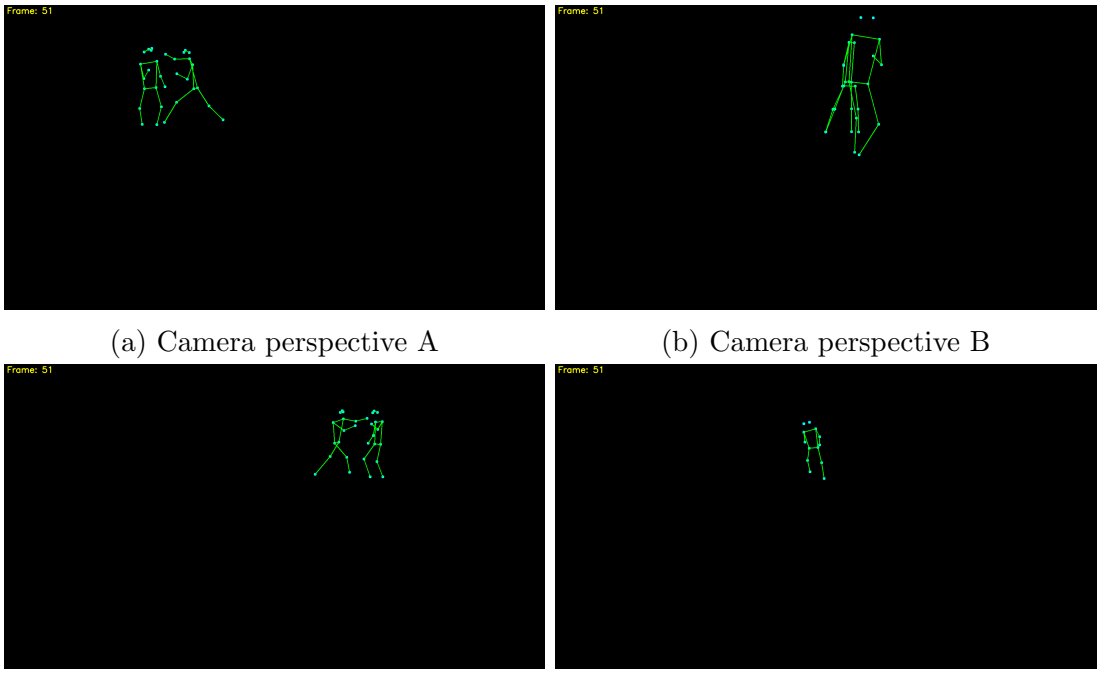
(a) Label distribution of all the labels after the cleaning and before the data transformation.



(b) Label distribution of all the labels after the data transformation.

Figure 3.5: Label distribution of all the labels before and after the data transformation. In total there are 8952 videos before and 35808 videos after the mirroring and cropping. In blue and red are separately depicted how many of the points annotated in the classes are scored by Aka or Ao respectively.

As a last transformation for the Xception model, the videos are either scaled down in resolution from 2160x2160 to 244x244 or in case of the BRNN model, the coordinates for every joint is extracted by YOLOv8 and combined with the



(a) Camera perspective A (b) Camera perspective B

(c) Camera perspective C (d) Camera perspective D

Figure 3.6: Frame of a original video, after extracting the joints with YOLO and mapping them on a black background for visualization. In the dataset made for the BRNN, the joints are not mapped on a background, but have an additional confidence score for the coordinates of each joint. In this frame one athlete, is trying to score 1 point with a punch, while the other one is trying to evade.

confidence score of each joint. As such, at the end there are two datasets. One with the videodata with a 244x244 pixel resolution and one with the extracted joints and their confidence scores as shown in a mapping of these joints directly onto a black background in Figure 3.6.

3.4 Datasplits and classes

After only the six classes remaine, they are combined into different datasplits. As shown in the table Table 3.3, the subsets are binary at first and for the last few sets more situations are combined. The differentiation between Ao and Aka is not used in our datasplits. The datasplits are consistently chosen in sets of four to ensure, that all camera perspectives are well represented. The subsequent division into training, validation, and test sets was carried out with the same level of attention to detail, maintaining a balanced representation of perspectives across these subsets. Additionally, to ensure class balance, the size of each class was limited to match the class with the fewest videos within the intended split.

Datasplit	classes	number of videos
npVS1p	no points, 1 point	9728
npVS2p	no points, 2 points	5920
npVS3p	no points, 3 points	2688
npVSj	no points, jogai	3328
npVSall_a	no points, 1 point, 2 points, 3 points, jogai, nothing happening	8064
npVSall_b	no points, 1 point, 2 points, 3 points, jogai, nothing happening	35808

Table 3.3: Classes and number of videos for all the datasplits. Both npVSall are the same, with the only difference, that the split denoted by a is balanced and the split denoted by b is unbalanced.

Chapter 4

Experimental Evaluation

4.1 Search and Evaluation metrics

In This section we discuss how the choosen hyperparameters are choosen and what the experimental setup is. Further the evaluation metrics for the models and the search are also part of this section.

4.1.1 Hyperparameter Search

To find the best possible hyperparameters for the chosen datasplits, a hyperparameter search was conducted. As such, the framework Optuna has been used for a grid search to make it more systematic and faster. The limitations of each hyperparameter and what kind of parameter to look for, was decided beforehand and the search was conducted for every datasplit separately.

As seen in the table above in Table 4.1 the parameters to vary are the learning rate with a range of 0.01 to 0.00001, accumulation step as a substitute for the batch size (because of memory limitations) in the range of 2 to 64, the optimizer between Adams and SGD (without varying the momentum), the learning rate

Hyperparameter	Start of Range	End of Range
Learning Rate	0.01	0.00001
Accumulation Step	2	64
Scheduler	StepLR	ReduceLROnPlateau
Optimizer	Adams	SGD
Patience	1	5
Factor	0.1	0.001
Stopper Patience	2	25
K-Fold	4	10

Table 4.1: Hyperparameter search ranges

scheduler between StepLR and ReduceLROnPlateau, the patience and the factor for the scheduler. More details can be gleaned from Table 4.1. Not in the referenced table, but important to note is, that the maximum number of epochs being used for training is not more than 50, because of time and resource constraints.

To choose the most robust parameters for the comparatively small datasplits, k-fold-validation is being used to evaluate them. The number of folds used for this, has also been varied to a certain extend (between 4 and 10) but has ultimately been chosen to be eight after a few preliminary tests.

Using this methodology and conducting approximately 30 to 60 experiments per datasplit on four Nvidia RTX 3090 by distributed training, a set of near-optimal hyperparameters are found. More about that in section 4.2.

Equally of interest are the metrics on which the evaluation of the entire search depends, as they play a crucial role in determining the overall effectiveness of the said search. In the following section, we will explore these metrics in more detail.

4.1.2 Model Metrics

Taking into account that we are looking for hyperparameters of a model by cross validation, we have three possible places to evaluate something on. On the training set and the evaluation set for every epoch in every fold and the test set at the end of every fold. Furthermore, the dataset is devided into 70% being used for the training-/validationset and 30% for the testset.

To see if the model is actually learning, any of the available metrics can be used, but the most common one is the loss. This is because the loss tends to decrease, if the model is picking up on what it is learning in each epoch. However, just looking at the loss on the training set, does not tell much about how effective the model is. This is, because the model might overfitt the training data, capturing specific patterns without developing the ability to generalize to unseen data. Therefore, additional evaluation on the validation or test data is necessary to determine its true performance.

Considering that some of the classes in the datasplits may not be evenly distributed, we may be dealing with imbalanced datasets. A simple but still pretty effective metric for handling this is the F1-Score. The F1-Score combines precision and recall into a single number, which helps avoid bias toward either one. This way, it balances false positives and false negatives equally, making it a solid choice for imbalanced data. To get a better feel for how learning is going in each epoch, we thus calculate the F1-Score to observe and to use as a means of evaluation.

After the search, the model with the chosen hyperparameters is trained by split-

ting the dataset into 70% for the training and validationset and 30% for the testset. For the training- and validationset ratio it is further split into a 70/30 split. The evaluation for these sets is being held with F1-Score, accuracy, precision, recall and the confusion matrix as well as the loss for seeing how good the model generalizes. But given, that the setup in a Karate fight is for four referees sitting around the tatami and the final evaluation is only meaningful, when the action on its own is according to the ground truth and not just from one perspective, it is worth considering using a majority voting or a x out of 4 perspectives voting on the testset. This voting is also useful for the transferability to real life situations, as a minimum of (x=) two out of four referees have to evaluate a situation similar to be able to allocate a point.

Besides the beforementioned metrics, as a way to see, how good the model can distinguish the classes, Matthew D. Zeiler and Rob Fergus introduced t-SNE in their paper [21]. We will make use of this mode of visualization as well. Additionally for seeing if the regions of interest and focus in the different datasplits are similar to each others for the model, saliency maps are used, as they are quite useful for visualizing focuspoints of the model over time according to the paper of Avlin G. Policar and Blaz Zupan [22].

4.2 Results and Discussion

In this section we will talk about the chosen hyperparameters from the grid search in detail and discuss the results through the received metrics.

Down below, in Table 4.2 the reader can see a quick overview of the results with F1-score and Accuracy, as well as the precision and the recall on the different datasplits for the two different models used. These are the results of the best hyperparameters found through the hyperparameter search and run with the whole dataset instead of the crossvalidation. In the table it can be seen, that the Xception model can generalize quite well in regards to the binary datasplits. Also when comparing all the different situations (six classes) together the model shows better than random accuracy. The same cannot be said about the BRNN model. As in the third and fourth rows of the table visible, the accuracy as well as the F1-score aligns with complete coincidence. For that reason, this section will be mainly about the Xception model results, unless otherwise stated.

As such in Table 4.3, the hyperparameters of the search for the Xception model are presented. Notably, the accumulation step, the scheduler and the patience for the scheduler are similar in most datasplits, however, not the learning rate and the factor for the scheduler. The "Number of Epochs" before stopping might stand out

Metric	npVS1p	npVS2p	npVS3p	npVSj	npVSall$_a$	npVSall$_b$
			Xception Model			
Xception F1-score	0.7800	0.8920	0.8638	0.9148	0.4288	0.4191
Xception Accuracy	0.7812	0.8925	0.8638	0.9160	0.4541	0.6041
Xception Recall	0.7799	0.8930	0.8669	0.9122	0.4540	0.4122
Xception Precision	0.7833	0.8994	0.8690	0.9214	0.4565	0.4768
			BRNN Model			
BRNN F1-score	—	—	—	—	—	—
BRNN Accuracy	—	—	—	—	—	—
BRNN Recall	—	—	—	—	—	—
BRNN Precision	—	—	—	—	—	—

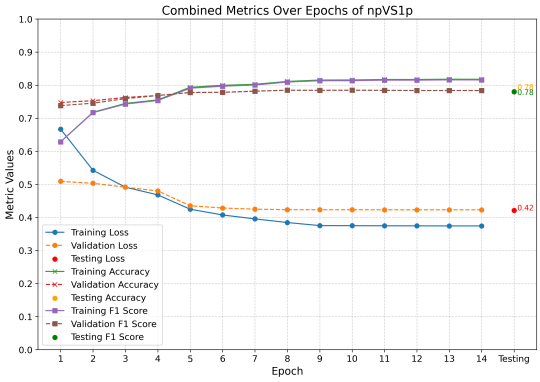
Table 4.2: F1-Score, Precision, Recall, and Accuracy for different data splits and both models. *Note: The BRNN model did not generalize, hence its metrics are not shown as they are completely coincidential.*

immediately because of the way it is portrayed in the table. The reason for the separation, is that this measurement does not directly apply to the hyperparameter search, but for the real training later on. What stands out about this value however is, that the Number of epochs are so little before the model starts overfitting. In the case of the datasplit npVS2p it only needs 3 epochs to achieve the best results. This could be an indication of the used model being too complex for the task, or the data being easily sepearble. When we look at the Accuracy and F1-Score respectively of the npVSall datasplits however, it can be seen that the the metrics are still at approximately 42%, after 10 epochs, which indicates, that the used model is either too simple to capture the features or does not have enough data or would need to be further fine tuned. A bit further down, we will take a closer look at this behaviour.

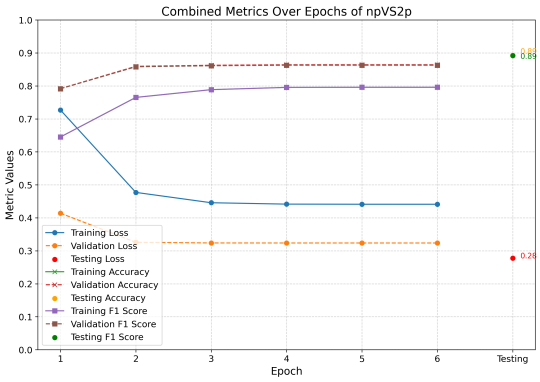
Taking a closer look at the linegraphs at Figure 4.1, we have the number of epochs on the x axis and the metric value on the y axis. We can see that the binary classifiers for the 1p and 2p classifiers are rather uneventful and the metrics

Parameter	npVS1p	npVS2p	npVS3p	npVSj	npVSall$_a$	npVSall$_b$
Learning Rate	0.0069	0.0076	0.0051	0.0099	0.0075	0.0024
Accumulation Step	15	12	13	13	15	14
Optimizer	SGD	SGD	SGD	SGD	SGD	SGD
Scheduler	StepLR	StepLR	StepLR	StepLR	StepLR	StepLR
Factor	0.0158	0.0921	0.0590	0.0157	0.0086	0.0142
Patience	4	1	5	5	5	5
Number of Epochs	9	3	6	14	10	9

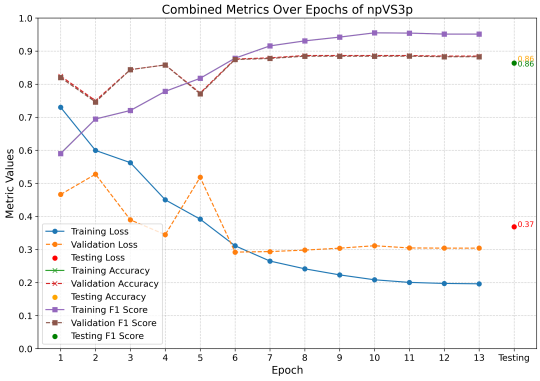
Table 4.3: Parameter Configuration for Different Data Splits on the Xception model



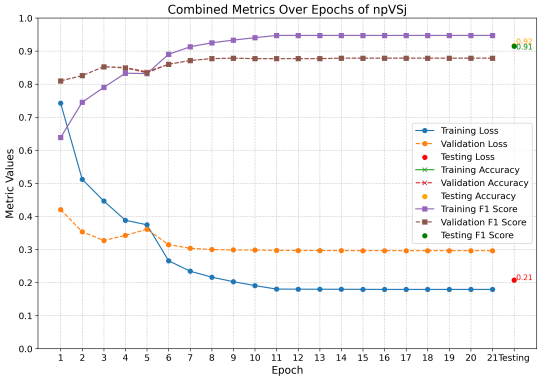
(a) Loss, Accuracy and F1-score for the npVS1p datasplit. Earlystopping done after 9 epochs.



(b) Loss, Accuracy and F1-score for the npVS2p datasplit. Earlystopping done after 3 epochs.



(c) Loss, Accuracy and F1-score for the npVS3p datasplit. Earlystopping done at 6 epochs.



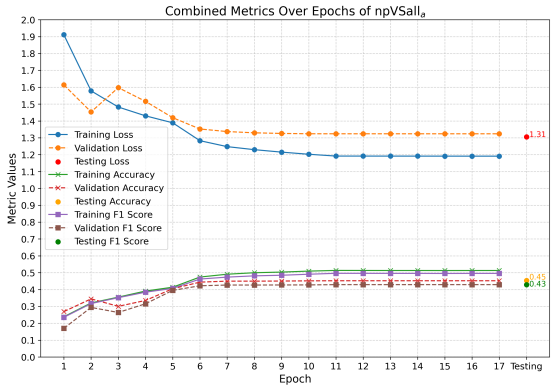
(d) Loss, Accuracy and F1-score for the npVSj datasplit. Earlystopping done at 14 epochs.

Figure 4.1: The Loss, Accuracy and F1-scores for the binary datasplits. X-Axis depicts the amount of Epochs with the last one being the Testing. The Y-Axis depicts the metrics value between 0 and 1.

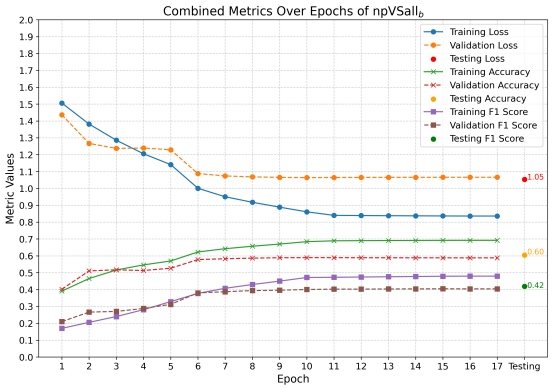
stop improving quite fast, after 9 epochs or 3 respectively. The jogai classifier is the same, though it needs a bit longer to find its optimum. For the 3p classifier however, it looks like a rather more eventful ride, seeing as it has more up and downs and crossing between the validation and training metrics.

The linegraphs for the multiclass classifiers in Figure 4.2, are rather uneventfull as well, though they keep a high loss until the end and lower metrics in general. In Figure 4.2b, we can see, that it is an unbalanced dataset, as the Accuracy and the F1-Score are not overlapping.

When looking at the confusion matrices of the different datasplits, we can observe a good balance of true positives and true negatives for all the binary datasplits in Figure 4.3, this is shown by having a deeper color in the left upper and right lower corners. In regards to False positives and false negatives in all the datasplits the false positives outweigh the false negatives slightly, but not excessively as seen



(a) The Loss, Accuracy and F1-scores for thenpVSall$_a$ datasplit. Earlystopping done at 10 epochs.



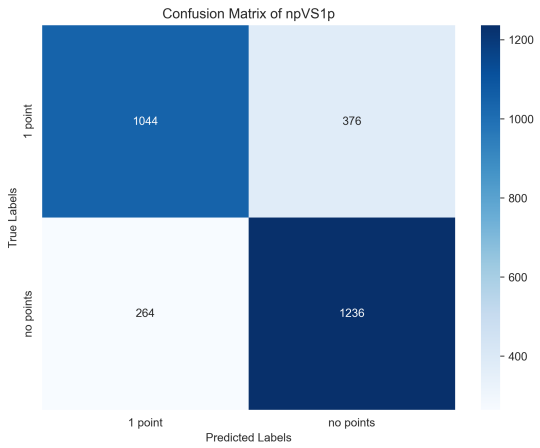
(b) The Loss, Accuracy and F1-scores for the npVSall$_b$ datasplit. Earlystopping done at 9 epochs.

Figure 4.2: The Loss, Accuracy and F1-scores for the multiclass datasplits. X-Axis depicts the amount of Epochs with the last one being the Testing. The Y-Axis depicts the metrics value between 0 and 2.

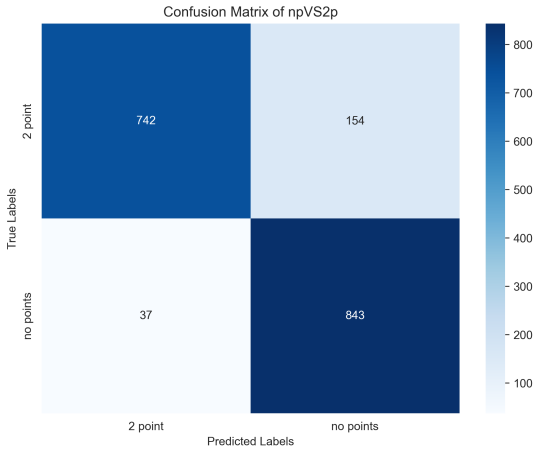
with a lighter blue color in the right upper and left lower corners and as it is also represented in Table 4.2.

When looking at the Figure 4.4, the distinction is less clear in regards to what the model has to do. For the balanced one Figure 4.4a it is still alright, though there is seems to be some slight confusion, when needing to classify no points (np) situations. Remembering the labeling stage, the np situations are labeled as such, when the athletes tried to score a point, but were not fulfilling all the criteria. This makes this class inherently a blend of all the classes, with only some small distinctions between all of them. We will examine this further, when looking at the Figure 4.6 further down. What stands out however, is that in the binary classifications the false positives were more common, while here the false negatives are more common. The confusion between the 1p and 2p, the 2p and 3p, the 3p and jogai are also happening a few times.

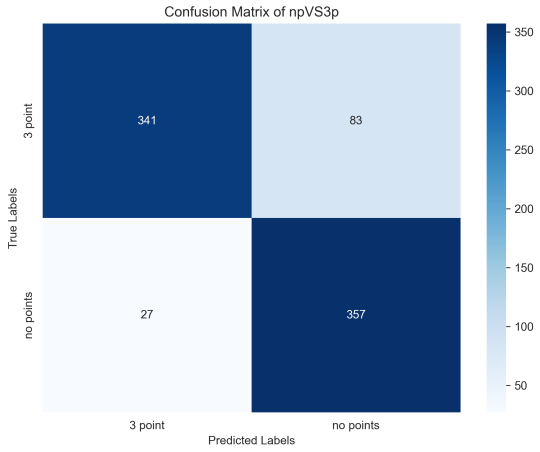
When looking at the Figure 4.4b, however it is difficult to make sense of it at first glance, because the np and nh situations are clearly more pronounced. The other classes are in comparison underrepresented. There are still a few things standing out. The 1p and np situations being confused more often for example, even though this time, there are again more false negatives for the 1p than in the balanced datasplit in Figure 4.4a or the binary confusion matrices in Figure 4.3. Other then that, there is almost no rightly classified occurrence of the 3p situation, though the 1p situation and the 2p situation are classified right more often than in the balanced datasplit.



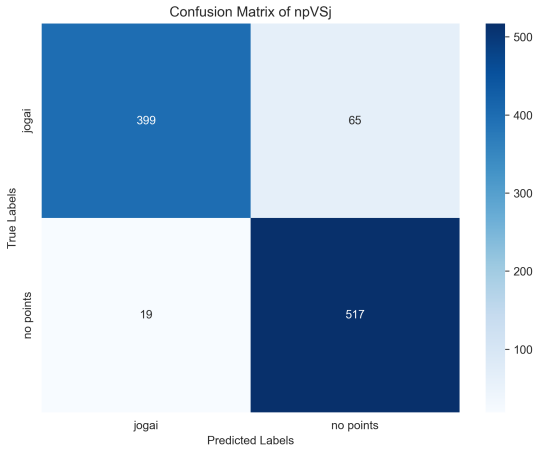
(a) Confusion Matrix of the npVS1p datasplit.



(b) Confusion Matrix of the npVS2p datasplit.

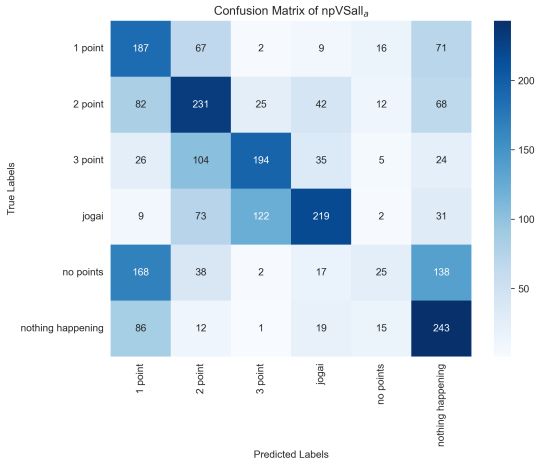


(c) Confusion Matrix of the npVS3p datasplit.

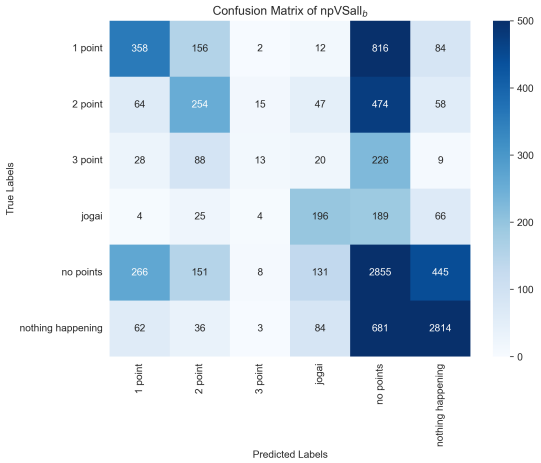


(d) Confusion Matrix of the npVSj datasplit.

Figure 4.3: The Confusion Matrices for the binary datasplits. Darker shade of blue is better on the top left to the bottom right in the diagonal. For the other fields it is better to be of a lighter shade of blue.



(a) Confusion Matrix of the npVSall$_a$ datasplit.



(b) Confusion Matrix of the npVSall$_b$ datasplit.

Figure 4.4: The Confusion Matrices for the multiclass datasplits. Darker shade of blue is better on the top left to the bottom right in the diagonal. For the other fields it is better to be of a lighter shade of blue.

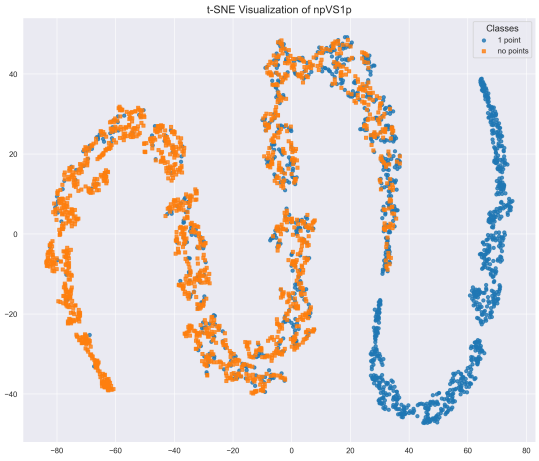
Having gone through that, it is time to analyze the decision boundaries. If we take a look at the t-SNE illustrations, we get an indication as to why it might be possible for the confusion matrices in the multiclass classifiers in Figure 4.4 to be rather confused in comparison to the binary classifiers in Figure 4.3.

First and foremost, it is important to emphasize that the pattern in the graph is not significant and does not hold any particular meaning. What matters in these graphs is the clustering. This aspect allows us to evaluate how effectively the classifier groups the data and how distinct the decision boundaries are, even when the data points may not be clearly separable on the given measurement scale.

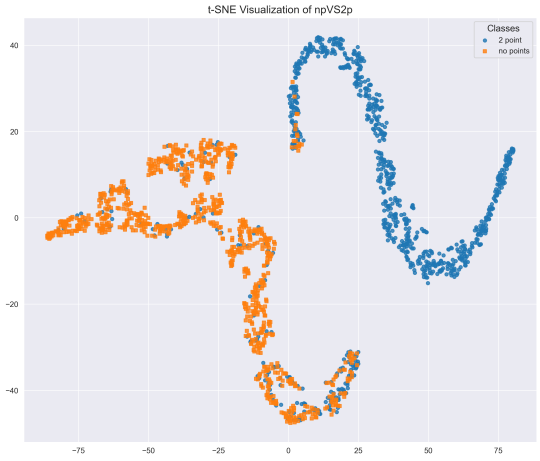
That said, when looking at the graphs in Figure 4.5, most groups are clustering visibly and densely together. Particular is, that the 1p (Figure 4.5a), 2p (Figure 4.5b) and 3p (Figure 4.5c) classifiers have distinct groups which are separated from the rest, but still have a lot of overlapping datapoints with the other classes. The jogai classifier (Figure 4.5d) however, does not have as many overlapping datapoints, the clustering is more clear, even though it does not have a clear separation.

Looking at the Figure 4.6, the only clustering which almost directly stands out, is in the balanced dataset in Figure 4.6a on the right upper corner. This one consists of the jogai and 3p classes. There are more of those datapoints also scattered in the rest of the graph, but not as many as in the said cluster. Further standing is the 1p class, which is more pronounced on the left side in blue circles, as well as the nh class in the upper middle with the brown triangles and the 2p class with the orange squares on the lower middle.

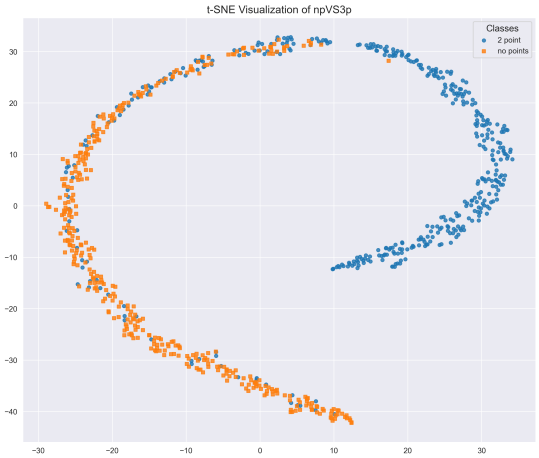
The unbalanced datasplit Figure 4.6b, however looks different. Signaled by the same marks and color combination for the classes, the overwhelming amount of brown triangles (nh) and violet stars (np) can clearly be spotted. The blue circles (1p) are mostly mixed in where the np class is and the red crosses (jogai) where the nh class is. The orange squares (2p) and green diamonds (3p) are mostly in the upper middle, but also scattered well into the np class. This representation gives an intuition how close many of the datapoints are to each other. They are not clustering well, because they are quite similar to each others in most cases.



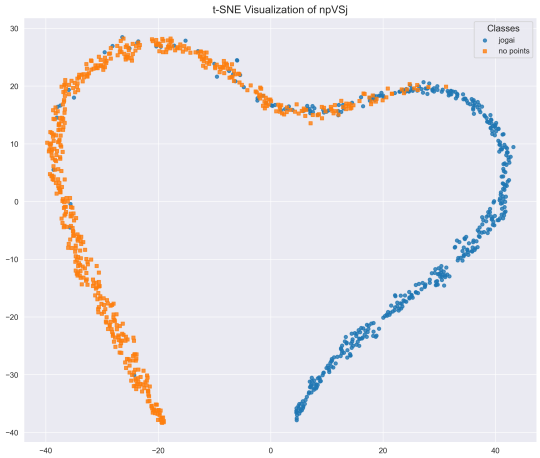
(a) t-SNE for the npVS1p datasplit.



(b) t-SNE for the npVS2p datasplit.

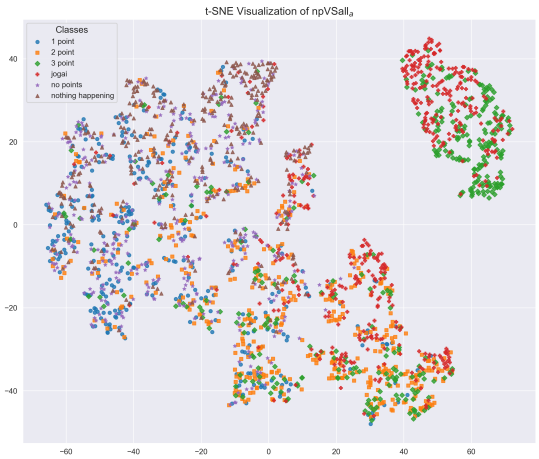


(c) t-SNE for the npVS3p datasplit.

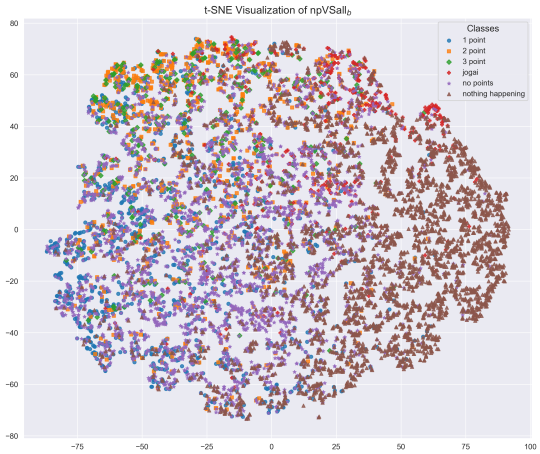


(d) t-SNE for the npVSj datasplit.

Figure 4.5: t-SNE for the binary datasplits. The Clustering is more important than the shape.



(a) t-SNE for the npVSall$_a$ datasplit.



(b) t-SNE for the npVSall$_b$ datasplit.

Figure 4.6: t-SNE for the multiclass datasplits. The Clustering is more important than the shape.

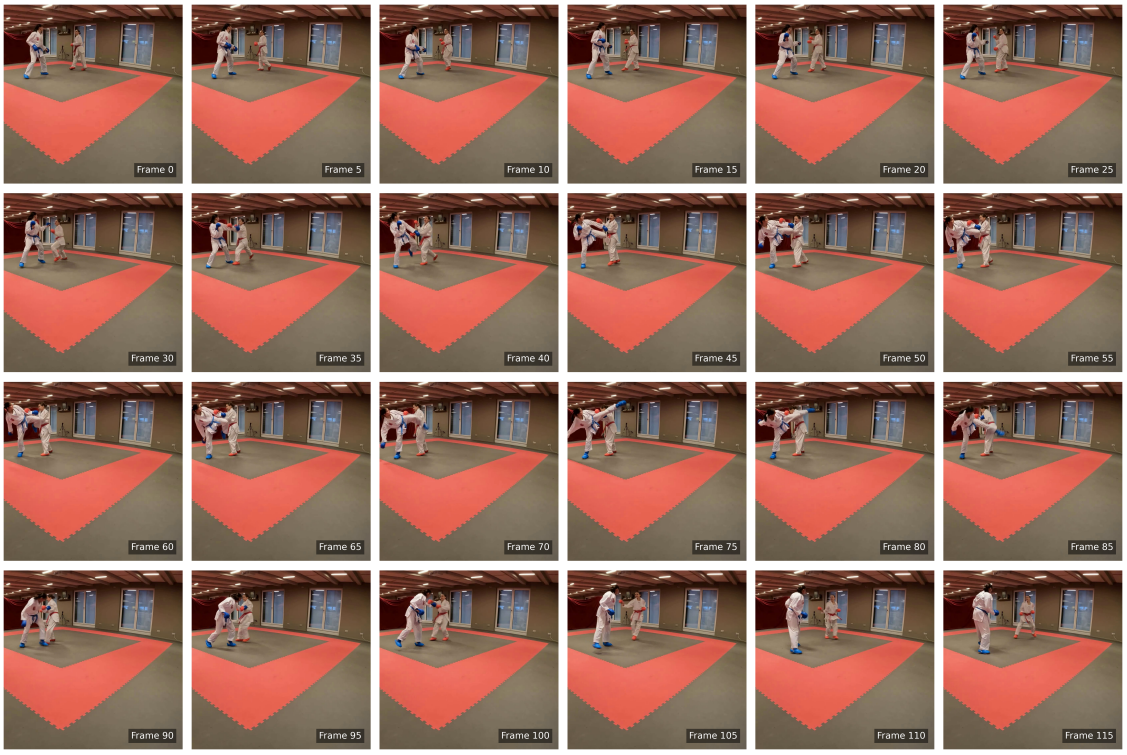
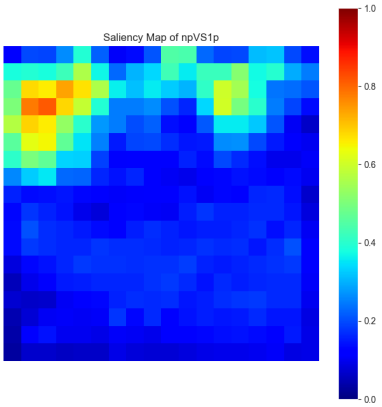


Figure 4.7: A 24 frames slice of the video crop_1_mir_V1−0016_D9D. It portraits perspective D, mirrored, the second crop. This video depicts a np situation.

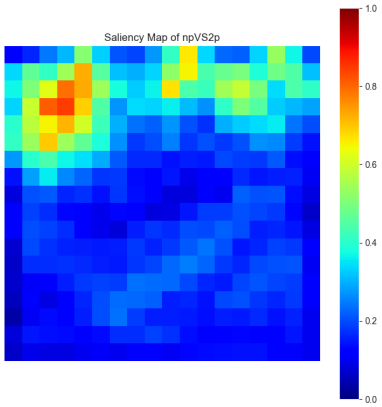
Additionally to see how similar the models on the different datasplits analyses a video sequence, we use saliency maps. In Figure 4.8 and Figure 4.9, we let the models run through a video, where the athlete tries to score a 3p but does not succeed and as such is a np situation. A slice of the video is portrayed in Figure 4.7.

The saliency map shows with a color gradient, which looks like a heat map, where the focus of the model is averaged over the video. In npVS1p classifier in Figure 4.8a we can see, that leg techniques are not as focused on as in the classifier of npVS3p in Figure 4.8c. For the npVS2p in Figure 4.8b the attention seems to be more pronounced but still not as strong as on the npVS3p. In the npVSj classifier in Figure 4.8d, the upper part of the picture is almost entirely dark, however the tatamicorners are more pronounced as than for instance in the npVS1p, which is exactly where a jogai does happen.

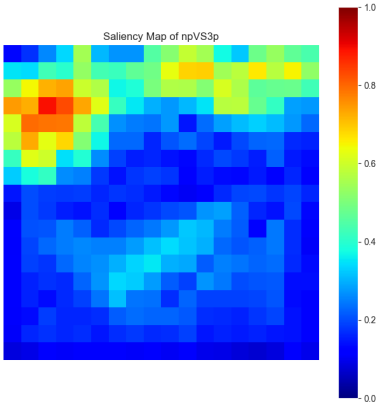
Interestingly enough the saliency map of the npVSall$_a$ classifier in Figure 4.9a and the npVS3p look quite similar, but with slightly more scattered focus points. the unbalanced variant of the npVSall split however, does not behave in the same way. It looks completely focused on one spot, where the acting athlete is situated.



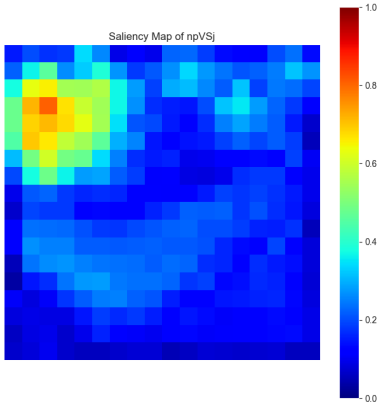
(a) Saliency map for npVS1p datasplit.



(b) Saliency map for npVS2p datasplit.

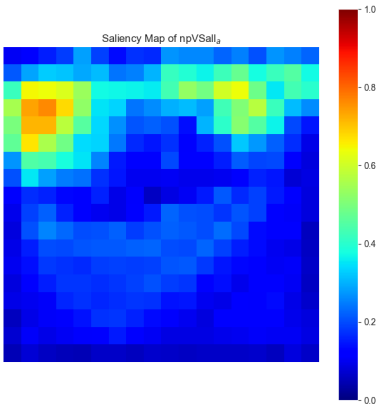


(c) Saliency map for npVS3p datasplit.

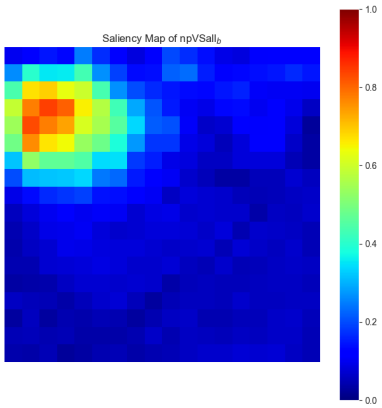


(d) Saliency map for npVSj datasplit.

Figure 4.8: Saliency map for the binary datasplits, run on the same video as in Figure 4.7. Darker red stands for paying more attention, while darker blue stands for being less important.



(a) Saliency map for npVSall$_a$ datasplit.



(b) Saliency map for npVSall$_b$ datasplit.

Figure 4.9: Saliency map for the multiclass datasplits, run on the same video as in Figure 4.7. Darker red stands for paying more attention, while darker blue stands for being less important.

x out of 4	npVS1p	npVS2p	npVS3p	npVSj	npVSall$_a$	npVSall$_b$
Base Accuracy	0.7812	0.8925	0.8638	0.916	0.4541	0.6040
At least 1 correct	0.8945	0.9505	0.9406	0.9680	0.6860	0.8276
At least 2 correct	0.8219	0.9189	0.9010	0.9480	0.5455	0.7029
At least 3 correct	0.7575	0.8761	0.8267	0.9120	0.3769	0.5525
All 4 correct	0.6493	0.8243	0.7871	0.8360	0.2083	0.3332

Table 4.4: Accuracy Metrics for Different Data Splits on the Xception model when using x out of a set of four as the amount of being correct in comparison to the base accuracy

To get back to a more reality near approach and with only the accuracy used as a metric, there has also been conducted a x out of 4 voting on the trained models. The results are portrait in Table 4.4. To make the results of the x out of Four voting more easily comparable, the accuracy from Table 4.2 for the Xception model has been integrated in the first row. Clearly visible in this table is that the "one out of four" and the "two out of four" accuracy is higher than the Base Accuracy. Let us compare just the npVS1p datasplit at first. The Base Accuracy tells us, that the every time the model looks at a fighting sequence where either a 1 pointer is scored or no points are scored, it can classify this correctly in 78% of the cases. As a reminder, for a real life application a referee can only give out a point to an athlete, if a minimum of two out of four judges indicate a point for the same athlete for the same action. Let us say we have four instances of the model which look at an action from four different perspectives and everyone decides on the action, if the point is scored. There is still a 22% probability, that either one instance classifies the action wrongly. Letting those four instances now make an educated classification from the four perspectives and then look at the results of all of them at the same time. For all four instances to classify this wrongly, is less probable, only 0.23%. In other words, the voting makes the decision more robust and less prone to wrong judgments as long as the ground truth holds. In general that means, the more x out of four the better. In the table, the first column indicates with the x, how many instances of the four available one voting do vote on the same video, from a different perspective, the same way.

Chapter 5

Conclusions and Future Work

This concluding chapter addresses the central questions of this thesis in section 5.1, beginning with a summary of the work conducted and the key results achieved. The main findings are highlighted, accompanied by a critical analysis of their implications and limitations. Following this, section 5.2 offers recommendations for future research, suggesting potential avenues to expand and build upon the outcomes of this research.

5.1 Key findings and Insights

Following the path through this thesis, the main objective is to gather a dataset, which maps complex real world scenarios, specifically from Karate Kumite, and see how it fares with modern HAR models. To do this, datapoints have been gathered, representing the most common situations and split into six different datasplits, four binary ones and two multiclass ones, whereas one of them was balanced and one was not. These datasplits have been run through and trained on two different models, a Convolutional Neural Network model called Xception and a Bidirectional Long-Short-Term-Memory model we call BRNN.

The Xception model consistently outperformed the BRNN model across all data splits and therefore has been chosen as the main model of evaluation for this thesis. Performance on the binary datasplits did particularly well in comparison to the multiclass. This has been measured by analyzing the Loss, F1-score and Accuracy, as well as the Confusion matrices and additional insights have been gathered by means of some visual analysis through Saliency maps and t-SNE visualizations.

Inspired by realworld Karate Kumite competitions the four perspective setup was used to do a x out of four voting mechanism to have more robustness and trust in the classification. This approach demonstrates, that leveraging multiple perspectives could significantly improve accuracy and reduce errors, especially for

binary classifiers.

To answer the research questions introduced in section 5.1, let us revisit them here:

1. What are the key characteristics of a model capable of judging a fight, and what types of data are essential for its development?

2. To what extent can Human Action Recognition serve as a substitute for human karate judges in prominent scenarios?

3. What are the challenges and possibilities in using models to judge an entire fight?

4. What are the technical requirements and limitations of using fight-judging models in real-time applications?

5. How do human judges and AI-based judges compare in terms of accuracy, fairness, and consistency in fight evaluation?

To answer the research question about how a model might be characterized, which judges a fight, in item 1, we need to get more in depth about the gathered insight through the results section.

First, consider the BRNN model. In general, it can be concluded that this model was either not well-suited for this specific dataset, or the feature extraction process failed to capture the underlying patterns required for effective learning. Several factors could contribute to these challenges. For instance, the dataset itself may lack sufficient discriminative features, resulting in poor class separability. Another possibility is that the preprocessing or feature engineering steps did not effectively preserve critical information or inadvertently introduced artifacts that misled the model.

In contrast, the Xception model seems to generalize particularly well when using it on the binary datasplits. When going over the line graphs, we can see how fast the model generalizes in regard to the accuracy and F1-score, which can indicate that the model is particularly well-suited to this type of problem. Further looking at the loss can give us a hint that the dataset has not been fully exploited yet. Combining this finding with the confusion matrices, we can still see enough examples of the test set being wrongly classified. This behavior can also be further underlined by the t-SNE. Although we can see the clustering being quite successful, there is still a lot of space for improvement. Comparing the different binary datasplits together, it is safe to say that the one which generalizes the best is the jogai. Curiously, it is also the one with the most epochs used for training and the second least amount of

datapoints. This may stem from the simplicity of what needs to be looked at when evaluating a jogai. If a body part crosses a line, it is indicative of having a jogai. In contrast, scoring a point takes much more consideration and is not as easy to differentiate. Interestingly enough though, the saliency map shows that the pattern the model focuses its attention on is not really the line, but both athletes, with a bit more attention on the athlete being near the line. This behavior can be confirmed by different videos. The same thing can be observed with different splits; the focus starts by being more on the athlete at first but transfers to the action itself over the course of the video.

When looking at the multiclass classifiers, the assumption that having such low results stems from having not enough datapoints to support the generalization. Having the dataset balanced might augment the distribution of the generalization, but at the cost of having fewer datapoints per class. This behavior is supported by looking at the t-SNE diagrams of these two datasplits and the loss as well as the F1-score in the line graphs. Not expected, however, is the difficulty for the balanced dataset to differentiate between the np class and the 1p class. It might come from using punching techniques as connectors between actual scoring techniques or as feints to confuse the opponent. As such, they are more commonly represented in the np class than not-succeeded leg techniques, which have a 2p or 3p potential.

Going further and analyzing Table 4.4, we can see that the methodology of not only using the base accuracy but using a "x out of 4" voting improves the results considerably. The multiclass classifiers gain the most by using this method but are also not really representative in using the "one out of four," as having one out of six classes represented correctly in one out of four perspectives does not make the result reliable. Having it right in two out of four makes it more believable already. For the binary classifiers, it makes them more robust, and they still profit, but not in the same amount.

Therefore, to promote this methodology more, special attention should be given in binary classifiers to the recall and precision metrics, as giving out points which have not really been scored can be much worse than the other way around. As it stands right now, both cases are equally represented.

With all this said, let us round back to the research questions. It is safe to say that HAR models are advanced enough to be used as classifiers for different actions, particularly in karate, addressing item 2. As it stands, there might be models better suited to the task than the ones used. Specifically for real-time use, it might be needed to use more lightweight models, like skeleton-based models such as the BRNNs, referring to item 4. Still, to be completely sure, the dataset would need to be extended to include more situations, in particular more datapoints for the

underrepresented classes. Even more so when the intention to have a more holistic approach exists, as certain fouls cannot be gathered consciously (like excessive contact). From the standpoint of karate competitions, a lot more would need to be introduced, for instance a way to judge a whole fight when post-analyzing the fight or a way to detect when something of importance might happen in case of real-time usage, addressing item 3.

In regard to the question if human and artificial judges are comparable (see item 5), the outcome of the experiment lets us make an educated guess, but no conclusive answer. As it stands, the multiclass classifier is by no means good enough to even remotely be considered close to the capabilities of a human judge. The binary classifiers, however, might come close to the actual human level, albeit not exactly.

5.2 Future Research Directions and Improvements

To enhance performance and context capture, future research could focus on incorporating advanced architectures such as transformers, specifically models like MViT, which are well-suited for processing video data. These models could provide improved attention mechanisms and contextual understanding. Additionally, exploring models that avoid separable convolutions may help retain finer-grained details within the data. Attention networks could also play a crucial role in capturing relationships between features across sequences.

An essential improvement lies in expanding the dataset. Gathering more diverse data, such as videos with spectators, longer fight sequences, and edge cases like fouls, will better align the dataset with real-world scenarios. Including videos of varying quality from different devices, mapped onto a time-graph for consistency across a set of four perspectives, can improve data representation. Furthermore, alternative labeling strategies and employing models specialized for detailed analysis could enrich the insights gained from the data.

A transformative direction for future work involves shifting from analyzing individual sequences to examining entire fight contexts. Developing a network trained to segment fights into sequences dynamically, while accounting for variables like interference from referees or spectators, could lead to a holistic understanding of the events. Additionally, creating an AI system capable of serving as a "referee" for various sports contexts could provide a generalized framework for action recognition and decision-making.

The responses to research questions have demonstrated the datasets utility, paving the way for practical applications in sports analytics and automated referee-

ing systems. These advancements highlight the potential for scaling the approach to other domains. By addressing the outlined future directions, this research can evolve to accommodate complex real-world scenarios, offering significant contributions to action recognition and decision-support systems.

Building upon the conclusions drawn in the previous section, future work should focus on expanding the dataset with additional classes and scenarios to improve model generalization. Developing more lightweight models optimized for real-time performance remains a key priority, as mentioned in the conclusion. Furthermore, incorporating advanced evaluation metrics such as interpretability and robustness would strengthen the system's applicability in diverse environments. These efforts will not only extend the current work, but also ensure broader adoption across similar sports and action recognition contexts.

Appendix A

Additional Material

A.1 Kata

Kata is a form or a choreography of techniques demonstrated by one athlete against invisible opponents. When done in a competition, two athletes are pitted against each other and each athlete demonstrates one Kata. These Katas are being evaluated with scores at the end of the demonstration and through that, compared. The athlete with the better score gets promoted to the next round. The Proceeding is like in many sports. There are various criteria by which the demonstrated Kata will be evaluated on. Many styles exist from which athletes can choose Katas from and usually every athlete performs Katas almost solely from its original style. Only the few more popular of the existing and practiced styles are allowed on WKF competitions.

A.2 Karate Terms

In Table A.1 is a short list of used Karate terms in the thesis.

Karate Term	Description
Karateka	Name for practitioners of karate.
Dojo	Training hall where Karatekas train in.
Tatami	The field where athlete compete on. Also the floor in a Dojo.
Aka	The Karateka in a bout which wears red protectors.
Ao	The Karateka in a bout which wears blue protectors.
Tori	The attacking party in a boutsequence
Uke	The defending party in a boutsequence.

Table A.1: Table with all the Karate terms used in the the thesis and a short explaination of the term.

A.3 Karate Referee setup

In Table A.2 there is a Referee setup as used in a WKF competition in case of a full setup. The Main Referee, the four Judges, the Match supervisor and the two Video Review Supervisors are the directly involved referees in a fight, while the rest serve as supporting officials ensuring the smooth proceeding operation of the competition.

Refereeing Panel	Description
1x Main Referee	Main referee, walking around the tatami, starting, stopping the bout and interacting with the athletes.
4x Judges	One Judge sitting on each corner in case of Kumite, indicating points during a bout.
1x Match Supervisor (Kansa)	Responsible for the correct proceeding during a bout. Indicates when the main referee does oversee two or more judges indicating a point or the time not being stopped when the main referee stops the fight.
Score Supervisor	Supervises the score and keeps a separate count from the score/timekeeper.
2x Video Review Supervisors	Responsible for the video review when asked for.
1x Tatami Manager	Takes care of the whole tatami, including which referee is used for which role in every fight.
3x Tatami Manager Assistants	Assistants to the Tatami manager. Have the same responsibilites as the Tatami Manager.
1x Score/Timekeeper	Takes care of depicting the right score on the official screen and stopping and starting the bout when the main referees indicates it.
2x Kansa Assistants	Assistants to the Kansa and hold the same responsibilities.
2x Coach Supervisors	Takes video review requests and indicates them to the Video review Supervisors. Also responsible for the coaches in general during a bout.

Table A.2: Complete Refereeing Panel in case of a complete setup in a WKF competition.

A.4 Karate Points

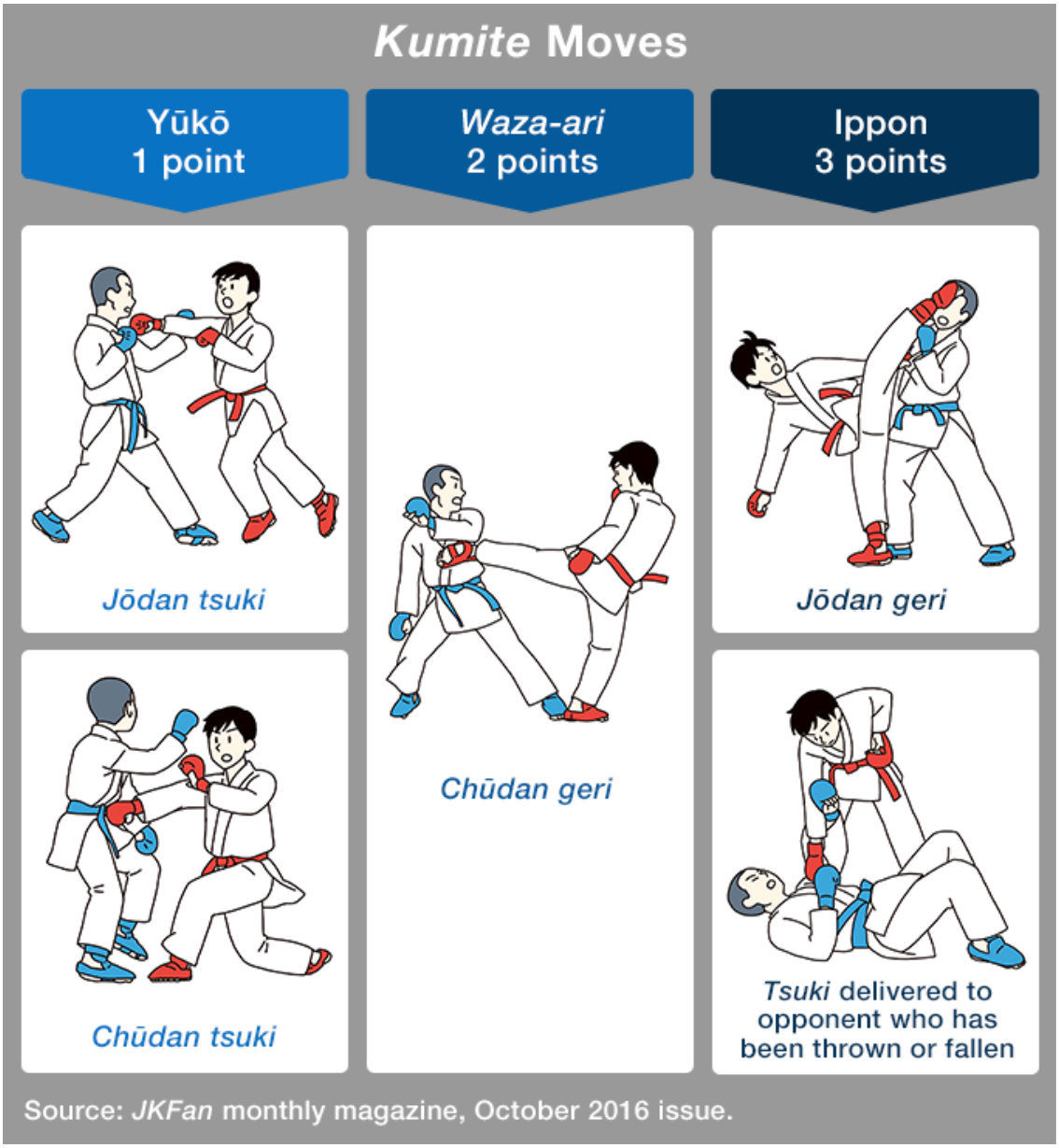


Figure A.1: Depiction of the different possibilities to score a point. One point by delivering a punch to the head or torso, two points by delivering a kick to the torso, three points by delivering a kick to the head or take the opponent to the ground and executing a punching technique.[23]

A.5 Karate Fouls

A complete list of fouls can be found in Table A.3. When talking about fouls, warnings are an important component, which cannot be left out. As such, in the table a few names are mentioned which stand for certain warnings. As a quick overview, there are three warnings which can happen when any of the fouls do happen in normal circumstances. We call them **CHUI**. If a competitor already has three CHUIs, and still commits a regular foul, a **HANSOKU CHUI** gets assigned, which is a warning for an incoming disqualification from the bout, if the competitor is not careful. In case the competitor still commits a warning, he gets disqualified from the bout by the main referee with a **HANSOKU**. If the competitor does something hardly acceptable like ignoring referee instructions, he can also get directly disqualified from the tournament by receiving a warning called **SHIKAKU**.

A.6 Examples

Scenario 1: Assume Ao and Aka score at the same time with the exact same technique and both hit the head of the opponent. Two Judges indicate Aka, while one judge indicates Ao, the point will officially be awarded to Aka by the main referee.

Scenario 2: The same situation as before happens again, but this time Ao is slightly faster and is seen by more than two judges. Ao gets more indications than Aka and will be awarded the point.

Scenario 3: The same situation as Scenario 2, but this time Ao does not fulfill all the criteria, because the timing was all wrong. But aka still does everything as it should. Aka will receive the point if seen by enough judges.

Scenario 4: Ao executes first a technique with the hand to the head of Aka and follows up with a leg technique. If a judge first indicates the punching point, he can not change the indication and cannot indicate the leg technique too. If one judge indicates the punching point and a second one indicates the leg technique as a point, then the higher of the two points gets awarded.

Scenario 5: If a third judge also sees the punching technique as a point in Scenario 4, the punching technique is the final counting technique, as it gets overruled by the majority.

In the example with the punch excess in section 2.1.2, the situation could also unfold completely different. If Aka hits the head and the contact is too strong, but in general would have been perfectly fine, but Ao instead of protecting and retreating walks straight into the punch without so much as his guard up, he endangered

himself. The foul in this case would be issued to Ao and not Aka. To change things up, let us say Aka still punches Ao, but Ao gets knocked down even tough Ao tried to defend himself. Ao fortunately is not knocked out and can still fight, he did not simulate or anything but he even bleeds now. The main referee may decide in such situations to propose two or even three fouls of the same kind directly for this one action or maybe even disqualify him directly if Ao had been knocked out for more than 10 seconds.

A.7 Xception Architecture

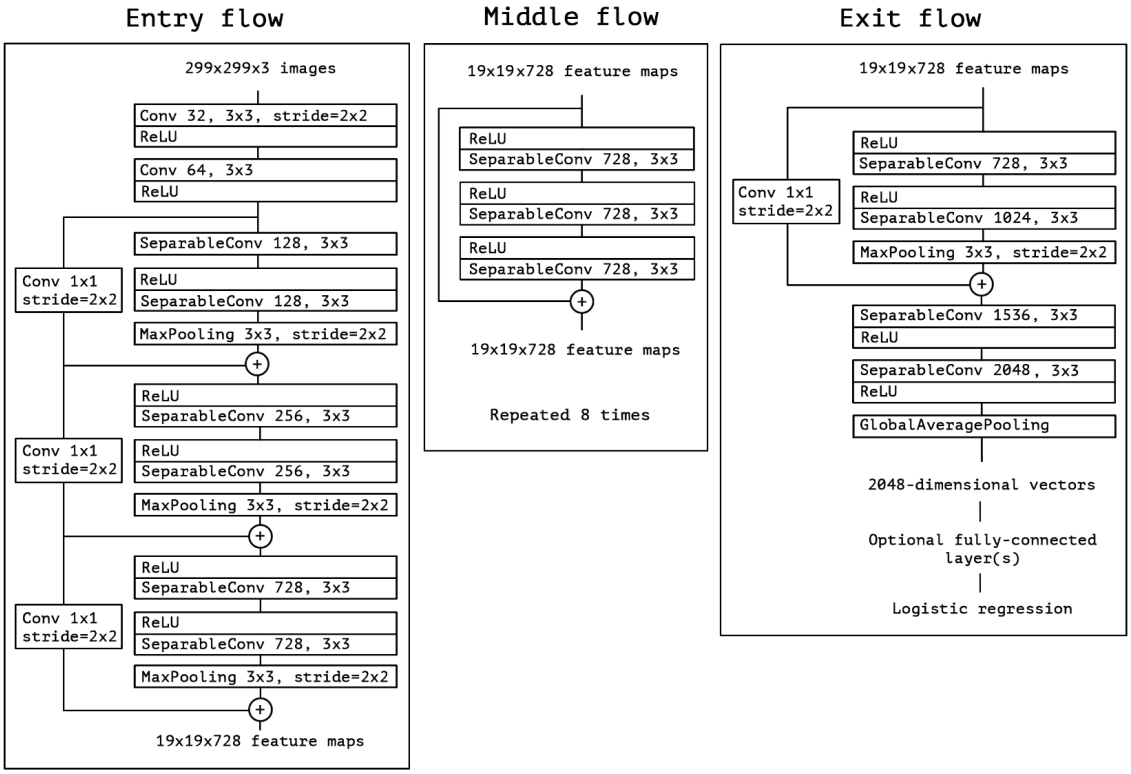


Figure A.2: Xcpetion Architecture as shown in the paper of Francois Chollet[9]. In has a Entry flow, a middle flow which is repeated eight times and and end flow. The Architecture used for the experiments is the same, but combined with an additional temporal dimension for videos.[24]

Foul	Description
Excessive contact	Where contact is considered by the Referee to be too strong, but does not diminish the Competitor's chances of winning, a warning (CHUI) may be given.
Contact causing injury	Any technique that results in injury, unless caused by the recipient, may result in a warning or penalty. Competitors must perform all techniques with control and good form.
Observation after contact	The Referee must observe the injured Competitor until the bout resumes, allowing adequate time for symptoms to develop or reveal tactical exaggeration attempts.
Overreaction to contact	A slight overreaction will receive CHUI; an obvious exaggeration will result in HANSOKU CHUI; severe exaggeration may lead to HANSOKU or SHIKKAKU.
Feigning an injury	Any feigning of injury results in a minimum CHUI; obvious exaggerations may lead to HANSOKU CHUI or SHIKKAKU, especially if feigning a valid scoring technique.
Contact to the throat	Any contact to the throat, unless caused by the recipient's fault, must result in a warning or penalty.
Illegal throwing techniques	Throws must not exceed hip level, and opponents must be held for safe landing. Over-the-shoulder and "sacrifice" throws are prohibited.
Catching a kick	Grabbing a kicking leg with both hands is permitted only for executing a takedown while controlling the fall.
Grabbing the legs	It is forbidden to grab or lift an opponent below the waist. Injuries caused by such throws may result in warnings or penalties.
One-hand grabbing	Grabbing the opponent's arm or Karategi with one hand is allowed for throws or scoring techniques but not for continuous holding.
Holding on to break a fall	Holding the opponent's Karategi with one hand to break a fall is permitted.
Exiting the competition area (JOGAI)	JOGAI occurs when a Competitor steps outside the competition area unless forced by the opponent or exiting after scoring.
Self-endangerment (MUBOBI)	A warning or penalty is issued when a Competitor is hit or injured due to their own negligence, such as turning their back or failing to block.
Passivity	Passivity occurs when neither Competitor attempts to score, or one avoids scoring despite being behind. It is not penalized in the first or last 15 seconds of a bout.
Avoiding combat	Avoiding combat, especially during the last 15 seconds, results in HANSOKU CHUI or loss of SENSHU. This includes time-wasting or constant retreating.
Not following instructions	Refusal to follow the Referee's instructions or losing temper results in SHIKKAKU, which can be imposed before, during, or after the bout.
Excessive celebration or demonstrations	Excessive celebration, political, or religious demonstrations during or after a bout are prohibited and may result in a fine.

Table A.3: Categories of fouls and their descriptions. Of relevance for this paper is only the jogai. In the table the words in bold letters are names of warnings a competitior can receive. The first to third warnings are called CHUI, the fourth is called HANSOKU CHUI and the disqualifiacation from the bout is called HANSOKU. Warnings in most of the time accumulate gradually. If it is not the case, the competitor can receive a SHIKKAKU, which is a disqualification from the tournament.

List of Figures

Bibliography

[1] Wenchao Jiang and Zhaozheng Yin. Human activity recognition using wearable sensors by deep convolutional neural networks. In Xiaofang Zhou, Alan F. Smeaton, Qi Tian, Dick C. A. Bulterman, Heng Tao Shen, Ketan Mayer-Patel, and Shuicheng Yan, editors, *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference, MM '15, Brisbane, Australia, October 26 - 30, 2015*, pages 1307–1310. ACM, 2015.

[2] Jamie Shotton, Andrew W. Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. pages 1297–1304, 2011.

[3] Yangfan Sun, Renlong Hang, Zhu Li, Mouqing Jin, and Kelvin Xu. Privacy-preserving fall detection with deep learning on mmwave radar signal. In *2019 IEEE Visual Communications and Image Processing, VCIP 2019, Sydney, Australia, December 1-4, 2019*, pages 1–4. IEEE, 2019.

[4] Hieu H. Pham, Louahdi Khoudour, Alain Crouzil, Pablo Zegers, and Sergio A. Velastin. Video-based human action recognition using deep learning: A review. *CoRR*, abs/2208.03775, 2022.

[5] Pawan Kumar Singh, Soumalya Kundu, Titir Adhikary, Ram Sarkar, and Debotosh Bhattacharjee. Progress of human action recognition research in the last ten years: A comprehensive survey. *Archives of Computational Methods in Engineering*, 29:2309 – 2349, 2021.

[6] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *CoRR*, abs/1511.08458, 2015.

[7] Basic cnn architecture. `https://www.researchgate.net/profile/Sayed-Hussain-20/publication/369477576/figure/fig1/AS:11431281129749081@1679628365418/Basic-CNN-architecture-and-kernel-A-typical-CNN.ppm`.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114, 2012.

[9] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1800–1807. IEEE Computer Society, 2017.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

[11] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1–9. IEEE Computer Society, 2015.

[12] Yuxiu Hua, Zhifeng Zhao, Rongpeng Li, Xianfu Chen, Zhiming Liu, and Honggang Zhang. Deep learning with long short-term memory for time series prediction. *IEEE Commun. Mag.*, 57(6):114–119, 2019.

[13] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1110–1118. IEEE Computer Society, 2015.

[14] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 779–788. IEEE Computer Society, 2016.

[15] Debapriya Maji, Soyeb Nagori, Manu Mathew, and Deepak Poddar. Yolopose: Enhancing YOLO for multi person pose estimation using object keypoint similarity loss. In *IEEE/CVF Conference on Computer Vision and Pattern*

Recognition Workshops, CVPR Workshops 2022, New Orleans, LA, USA, June 19-20, 2022, pages 2636–2645. IEEE, 2022.

[16] Jiaben Chen and Huaizu Jiang. Sportsslomo: A new benchmark and baselines for human-centric video frame interpolation. *CoRR*, abs/2308.16876, 2023.

[17] Jinglin Xu, Guohao Zhao, Sibo Yin, Wenhao Zhou, and Yuxin Peng. Finesports: A multi-person hierarchical sports video dataset for fine-grained action understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 21773–21782. IEEE, 2024.

[18] Seonok Kim. 3dyoga90: A hierarchical video dataset for yoga pose understanding. *CoRR*, abs/2310.10131, 2023.

[19] Tao Wu, Runyu He, Gangshan Wu, and Limin Wang. Sportshhi: A dataset for human-human interaction detection in sports videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 18537–18546. IEEE, 2024.

[20] Yixuan Li, Lei Chen, Runyu He, Zhenzhi Wang, Gangshan Wu, and Limin Wang. Multisports: A multi-person video dataset of spatio-temporally localized sports actions. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 13516–13525. IEEE, 2021.

[21] Pavlin G. Policar and Blaz Zupan. Visualizing high-dimensional temporal data using direction-aware t-sne. *CoRR*, abs/2403.19040, 2024.

[22] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, volume 8689 of *Lecture Notes in Computer Science*, pages 818–833. Springer, 2014.

[23] Karate points. `https://artpictures.club/autumn-2023.html`.

[24] Amil Khan. 3d xception model. `https://github.com/amilworks/3D-Xception`.

Erklärung

gemäss Art. 30 RSL Phil.-nat.18

Name/Vorname: Petrone Alessio

Matrikelnummer: 20-127-395

Studiengang: Bsc in Computer Science

Bachelor ☑ Master ☐ Dissertation ☐

Titel der Arbeit: Classifiying Karate Kumite Actions

LeiterIn der Arbeit: PD Dr. Kaspar Riesen

Ich erkläre hiermit, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäss aus Quellen entnommen wurden, habe ich als solche gekennzeichnet. Mir ist bekannt, dass andernfalls der Senat gemäss Artikel 36 Absatz 1 Buchstabe r des Gesetzes vom 5. September 1996 über die Universität zum Entzug des auf Grund dieser Arbeit verliehenen Titels berechtigt ist.
Für die Zwecke der Begutachtung und der Überprüfung der Einhaltung der Selbständigkeitserklärung bzw. der Reglemente betreffend Plagiate erteile ich der Universität Bern das Recht, die dazu erforderlichen Personendaten zu bearbeiten und Nutzungshandlungen vorzunehmen, insbesondere die schriftliche Arbeit zu vervielfältigen und dauerhaft in einer Datenbank zu speichern sowie diese zur Überprüfung von Arbeiten Dritter zu verwenden oder hierzu zur Verfügung zu stellen.

Bern, 16.12.2024

Ort/Datum

Unterschrift