Classification of Skateboard Tricks based on Deep Learning and Video Recordings

Patrick Juriga

***Abstract*—Skateboarding, an exhilarating and dynamic sport, has gained immense popularity worldwide. However, despite its growing popularity, the sport faces several challenges, including the subjective nature of judging and the lack of objective performance measures. Nevertheless, with advancements in computer vision and deep learning, there is an opportunity to automate and enhance this process. This study explores the application of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) for skateboard trick classification based on video recordings. The aim is to develop a system that is feasible of classifying skateboard tricks, overcoming challenges such as trick diversity and variations in execution styles. An experiment was conducted to gather video recordings of attempts at skateboard tricks, aiming to extract valuable information about their outcomes, in which a dataset was created comprising 439 recorded tricks. To extract robust features from the video recordings, the YOLOv8 model is employed. The dataset included labelled videos of skateboard tricks, enabling supervised training. CNNs were used for feature extraction, while RNNs were employed to capture temporal dependencies for trick identification. With the chosen neural network architecture, an accuracy of 36,36 % was achieved in classifying skateboard tricks. Based on the results obtained, the feasibility of classifying skateboard tricks based on video recordings and deep learning has not been confirmed. However, it is important to note that this outcome may be attributed to the limited training data used in the study. With a larger and more diverse training dataset, there is potential for further advancements in data-driven analysis using CNN and RNN.**

***Index Terms*—Skateboarding, Trick Classification, Deep learning, Convolutional Neural Network, Recurrent Neural Network, Computer Vision**

I. Introduction

SKATEBOARDING has become an increasingly popular sport in recent years, with millions of enthusiasts worldwide. Moreover, with the inclusion of skateboarding in the Olympic Games with its premiere at the Tokyo Summer Games in 2021 the sport has gained even more recognition and attention. Alongside this surge in popularity there has been a growing trend of integrating technology into sports, resulting in a more data-driven approach to understanding athletic performance. However, the sport faces several challenges including the subjective nature of judging and the lack of objective performance measures.

In many freestyle sports as skateboarding is one of it, performances and judgement tend to focus on subjective parameters, which can be as basic as the overall 'style factor' of an execution. This fact has so far made it difficult to gain directly obtained, objective statements about the athletes' performances (such as a goal or no goal in football). For instance a particular trick's execution can vary widely depending on factors such as the skater's individual style and the difficulty of the manoeuver. While judges can assess a skater's performance subjectively, it's challenging to quantify it objectively. Digital motion analysis can provide a solution to this challenge by complementing subjective measures with objective performance measures, offering unique feedback to competitors, judges and spectators.

In the field of action recognition, computer-aided evaluation holds significant promise as a valuable tool. However, to effectively analyze the data and recognize actions, it is crucial to first acquire the relevant data. This necessitates the use of some form of monitoring device. Common approaches involve the utilization of wearable sensors in combination with machine learning techniques. Hollaus et al. [1] developed a catch detection system for American football using a wearable sensor and machine learning. Groh et al.. [2] employed a wearable sensor to classify tricks in freestyle snowboarding. The team led by Brock [3] utilized wearable sensors to assess motion style errors in ski jumping based on machine learning methods. In skateboarding several systems have been developed that employ wearable sensors, often incorporating inertial measurement units (IMUs) attached to the skateboard, along with machine learning techniques [4], [5]. Another application of IMUs, with a focus on exergames where physical effort from the user is required, has also been explored [6]. Anlauff et al. [7] not only focused on the classification aspect but also provided visualizations of the performed tricks. Abdulah et al. [8] took a different approach by using transfer learning to classify tricks by using pre-trained models.

The major drawback from all these systems is that they need a sensor which is mounted onto the skateboard. This can be resolved by using instead of a sensor mere video recordings for the classification part. Thereby the athlete is not restricted or hindered in his activity by the recording system. Hollaus et al. [9] developed an automatic catch recognition in American football based on video and audio recordings. In this way they show the possibility in the classification in the sport action recognition. Because of this the study aims to explore the possibility of classifying skateboard tricks using video recordings. Unlike systems that require cameras to be attached to the athletes, the approach used in this research does not hinder their performance. Nevertheless, some drawbacks in contrast to the wearable sensor approach also exist for this video-system. One of them would be the fact that it requires continuous camera coverage of the entire skate park or street skateboarding area, which is not always feasible.

In the field of sport action recognition the analysis relies

P. Juriga is with the Department of Medical and Health Technologies, MCI, Innsbruck, Austria, e-mail: p.juriga@mci4me.at.

Manuscript received June 5, 2023; revised June 30, 2023.

on machine learning algorithms [10]. However, a significant challenge arises from the large size of these datasets, making traditional techniques inefficient for processing. Consequently, there is a need for time series classification (TSC) algorithms that can effectively process video files containing athletes' trick attempts. These algorithms can efficiently handle the data and detect temporal patterns. Recent studies by Karim et al. [11], [12] have demonstrated the effectiveness of multivariate long short-term memory fully convolutional networks (MLSTM-FCNs) for classifying sequences of images. Gated recurrent units (GRUs) have shown comparable performance to long short-term memory (LSTM) networks, as demonstrated by Chung et al . [13]. Elaysed et al. [14] built upon this work and proposed a GRU-FCN approach that yielded promising results. Furthermore, recent research in multivariate and univariate TSC has shown great potential using various approaches [15]–[19].

Although the importance of time series classification for classification in videos is evident, recent advances in the field of object detection have showcased promising results. Redmon et al. [20] introduced the YOLO model for real-time object detection, which outperforms other detection methods. Additionally, the current version, YOLOv8 [21], [22], also enables the estimation of human pose. Training neural networks has also seen advancements, with adaptive learning rate schedulers providing significant improvements in performance for various architectures. Smith's proposed work on cycling learning rate has demonstrated its effectiveness [23]. Moreover, the technique of transfer learning, which involves transferring knowledge from pre-trained models to train on custom datasets, has become a promising area in machine learning [24]. Applying transfer learning to the YOLOv8 model, Reis et al. [25] achieved real-time detection of 40 different flying objects.

The primary objective of this paper is to explore the feasibility of identifying skateboard tricks using video data and deep learning techniques. To accomplish this goal a large and diverse dataset of skate trick videos will be developed for training and testing the neural network. The results obtained from the neural network will be analyzed , and the performance will be interpreted to identify its strengths, weaknesses and potential areas for future enhancement.

II. Material and Methods

This section outlines the materials and methods used in the study. Firstly, the experimental design is presented, followed by the data acquisition phase. Next, the process of labelling and data processing is explained. The neuronal network needs some input parameters, extracted as features from the videos, in order to identify the tricks. The feature extraction process is then detailed. Finally, the model architecture, training procedures and evaluation process used in the research are shown.

A. Design of Experiment

The primary objective of the experiment was to collect a varied collection of video recordings featuring different skateboard tricks performed by numerous athletes. This data is later on processed and used to train a neural network for the trick identification. The athletes were required to perform five different tricks as follows: Ollie, Fakie Ollie, Backside (BS) Pop Shove-it, Frontside (FS) 180 and Kickflip. The selection of these tricks was due to their representation of the most fundamental movements a skater can perform. The Fakie Ollie is just slightly different from the Ollie but was included nevertheless to see if this small difference concerning the stance could be detected by the neural network. In addition, failed attempts at trick execution were classified in the sixth category No-Trick. To ensure the dataset was diverse, participants in the experiment were given minimal restrictions regarding data acquisition. The primary requirement for participants was to perform the trick in a clean manner, without excessive sketchiness. Apart from this condition, participants had the freedom to execute the trick according to their individual style and approach, allowing for a wide range of variations in the acquired data. No distinction was made between regular or goofy stance directions.

In order to achieve the design goal of the experiment a total of five participants were recruited to perform trick attempts under different conditions. All participants were male, aged between 22 and 27. Although they were skilled in performing the required range of tricks, they still are considered amateurs. Moreover, the recording sessions were conducted at two distinct locations. The study underwent a comprehensive review and received approval from the ethics board of the Management Center Innsbruck. Prior to participating, all participants were fully informed about the study's objectives, potential risks involved and how their data would be handled. Each participant signed a consent form, which is available upon request.

The experimental setup consisted of two cameras from different perspectives at a flat location, where the athletes were able to execute the tricks. The recording system was placed to record the participants from two different angles. One view was frontal to the athlete, the other view was parallel to the sagittal plane of the athlete, i.e. sideways. One condition during the recording of an attempt was that only one person and one skateboard are visible in the field of view.

B. Data Acquisition

In order to meet the requirements of the experiment, careful consideration was given to select a suitable camera for video recording. These requirements were specifically defined to record the entire body movements of the athlete and the skateboard throughout the execution of tricks. From a camera perspective, it was crucial to determine the minimum frame rate, recording time and resolution to ensure accurate data capture.

The trick attempt was defined to begin when the skateboard leaves the ground and to end with a clean landing, when the athlete is back on the skateboard and in contact with the ground. This resulted in a relevant time frame of 2 seconds for each trick attempt. Notably, the attempts for each trick category were recorded in a single continuous shot during a data recording session, allowing participants to focus solely on their respective tricks.

To ensure high temporal resolution for capturing the fast-paced movements involved, a camera capable of recording at a frame rate of 60 frames per second (FPS) was carefully selected. This approach allowed for an experimental determination of the minimum frame rate rather than relying on assumptions. The chosen camera, the GoPro Hero 9, was configured with a resolution of $1920\,\mathrm{px} \times 1080\,\mathrm{px}$.

C. Labelling and Data Processing

After the completion of the data acquisition process, the next step involved labelling the recorded data. Initially, the recorded data consisted of continuous video recordings capturing various attempts at different trick categories from the data recording sessions. Next, the recorded footage was resized using the OpenCV scaling function to a resolution $640\,\mathrm{px} \times 360\,\mathrm{px}$. This resolution was specifically chosen to meet the input size requirements of the YOLOv8 model, enhancing the efficiency and effectiveness of data processing and ensuring compatibility with the YOLOv8 model.

Subsequently, the recorded footage was manually divided into clips of two seconds each, with each clip containing a single attempted trick. This segmentation process was carried out in Python using a combination of `PyGame` and `OpenCV` libraries. Each file was assigned a name that incorporated the location name, the trick name, a counter to ensure uniqueness and the camera perspective from which they were recorded.

After the segmentation process, a crucial step was carried out to clean the dataset and to ensure its quality and suitability for subsequent analysis. This involved removing video clips that did not meet specific criteria, ensuring that only relevant and reliable data remained. Clips that feature multiple persons or skateboards in view, tricks performed out of sight or instances where a clean landing was not achieved were identified and eliminated from the dataset.

This resulted in a recorded dataset, in a volume of 439 clips, which formed the basis for training the neural networks. An important metric for training is the ratio of attempts per class. In table I this ratio between the six classes can be seen. The BS Pop Shove-it class has the largest percentage ratio of 22,87 %, while the Kickflip class has the lowest percentage ratio of 13,21 %. This imbalance could potentially impact the performance of a deep learning model trained on this dataset, as the model may become biased toward predicting the majority classes.

Table I: Trick Attempts per Class

Class	Amount	Ratio in %
Ollie	72	16,40
Fakie Ollie	75	17,08
BS Pop Shove-it	84	22,87
FS 180	70	15,95
Kickflip	58	13,21
No-Trick	80	18,22

To increase the dataset's size and diversity augmentation techniques to the video data were applied. The different techniques were implemented by using the `OpenCV` and `imgaug` libraries in Python. Following techniques were applied: a blur filter with a kernel size of $[5, 5]$, gaussian noise with a mean of 0 and a standard deviation of 0.1, histogram equalization on the Y channel at YUV color space, horizontal flipping, random rotation by an angle within the range of $-3°$ to $3°$, random translations in both the x and y directions within the range of $-0{,}05\,\%$ to $0{,}05\,\%$. By augmenting video data, overfitting of the network can be avoided and enhance the network's ability to generalize well to unseen video recordings.

Furthermore, all network performances were judged by using separate test dataset, which was based on the raw dataset. Moreover, to study the impact of the camera perspective and if even only one perspective is necessary, different combinations of the camera views were implemented and acquired for training. The raw dataset was splitted into subset for training (67,8 %), validation (17,2 %) and testing (15,0 %). To ensure a balanced distribution of classes in the training, validation and testing subsets, the dataset was split using the `StratifiedShuffleSplit` function from the `scikit-learn` library. This function provides the advantage of maintaining a proportional representation of each class across all splits. All other generated dataset were splitted into subset for training (80 %) and validation (20 %).

D. Feature Extraction

In order to identify skateboard tricks from a video recording, additional information had to be provided as input for a neural network. This information was extracted in the form of various features. Following features were extracted to serve as input parameters: bounding box of the detected skateboard, specific keypoints of the human pose and specific keypoints from the skateboard. To obtain these features a pre-trained network called YOLOv8 [21] was used. In order to standardize the distribution of the features, a normalization technique was applied using the `StandardScaler` class from the `scikit-learn` library.

YOLOv8 includes a pre-trained object detection model capable of detecting 80 different classes [21]. In the context of skateboard detection, the relevant class ID for a detected skateboard was 36. The information about the bounding box was stored in the format $xywh$, where x and y represented the center of the box, whereas w and h denoted the width and height of the box. These informations were stored and used input parameters for the network.

As for the detection of the keypoints from the human pose, the YOLOv8 provides a model which identified 17 keypoints distributed across various body parts. Keypoints 0 to 4 correspond to facial features, while keypoints 5 to 10 represent shoulders, elbows and wrists. The set of keypoints 11 to 16 relates to the hips, knees and ankles. Since the movements of arms and face were not considered as crucial and tend to change significantly during trick execution, these features were excluded from further analysis. Instead, the keypoints 11 to 16 associated with hips, knees and ankles were utilized for additional examination.

In the specific context of studying the movement of a skateboard, a model that can detect and track relevant keypoints on the skateboard was developed. By identifying these keypoints, such as trucks and wheels on the skateboard, it bacame possible to gather more precise information about its rotation and movement. To achieve skateboard keypoint detection, the approach of transfer learning with YOLOv8 was employed. First, a dataset of single frames from the video recordings where a trick execution was done generated. This dataset sets the base for manaully annotating the keypoints of the skateboard. To achieve this the online annotation platform CVAT was used. In this case, seven keypoints were annotated and placed consistently in the same order on each skateboard image. The order was truck at the nose side, truck at the tail side, midpoint of the deck, left wheel of the truck at the nose side, right wheel of the truck at the nose side, left wheel of the truck at the tail side and right wheel of the truck at the tail side. Also the bounding box enclosing the skateboard had to be annotated. Furthermore, the dataset was augmented by using rotation and translation techniques. Subsequently, this augmented dataset was divided into subsets, with 70 % allocated for training and 30 % for validation purposes. The YOLOv8 pose model was then trained using this newly generated dataset, employing the built-in trainer function [21]. As for the hyperparameters, the default values of the YOLOv8 train function were utilized.

E. Neural Network Architecture and Implementation

In the study, the network aims to identify unique features in data sequences to classify them as the given skate tricks. Feature extraction is a common challenge in computer vision and signal processing domains, and it can be addressed by utilizing CNNs for extracting spatial features and RNNs for capturing temporal dependencies. The setup for this project involved utilizing the Python programming language in combination with the PyTorch framework on an NVIDIA GPU for accelerated computation. To handle the data effectively, the PyTorch DataLoader and Dataset classes were employed. These classes facilitated access to the samples during the training process.

To handle sequential data and perform classification tasks, the input data for the neural network consisted of a sequence of features captured over time. In this particular work, the shape of the input tensor x was determined as [sequence_length x num_features]. The sequence length corresponded to the number of frames in the video clip, which was dependent on the FPS and duration of the video clip (2 seconds in this case). As an example, if the clip has a frame rate of 30 FPS, the sequence length would be 60 frames. The number of features was determined by the bounding box coordinates, the coordinates of keypoints from the pose estimation and the skateboard pose estimation, resulting in a total of 30 features. If only one camera perspective was used, the number of features was 30. However, if both camera perspectives were utilized, the number of features was doubled. The batch size was configured as 6. Moreover, dimension shuffle was also applied to the input data. If the sequence length was smaller than the number of features, the input tensor was transposed to following shape [num_features x sequence_length].

To address this type of data, several models have been implemented, notably the multivariate time series classification approach using LSTM cells as proposed by Karim et al. [12] and the use of GRU cells by Elsayed et al. [14]. The multivariate network architecture consists of two blocks that process the input separately: the LSTM or GRU block and the FCN block. Figure 1 provides a detailed illustration of the architecture.

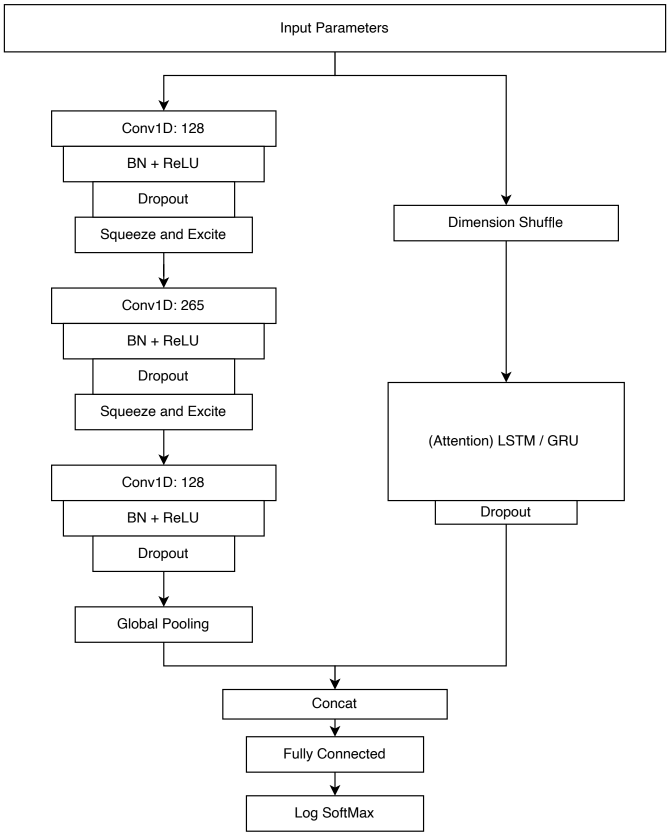

Figure 1: The structure of the developed neural network for multivariate time series classification.

The FCN block in the network architecture consists of three temporal convolutional blocks, which serve as a feature extractor. These blocks were adapted from the original FCN block proposed by Wang, Yan and Oates in 2017 [26]. Each FCN block contains a convolutional layer with specific filter numbers: 128, 256 and 128. The kernel sizes for these convolutional layers were set to 8, 5 and 3. The ReLU activation function follows the batch normalization and then a dropout with a probability of 0,1 was applied. Additionally, the first two convolutional blocks were concluded with a squeeze-and-excite block, having a reduction ratio of 16. The final temporal convolutional block was succeeded by a global average pooling layer.

Regarding the LSTM or GRU block, it undergoes a dimension shuffle layer, which transposes the temporal dimension of the input data. Afterward, the input was passed through the LSTM/GRU block. The block was set up with the following configurations: a hidden size of 512, the number of layers was fixed at 1, and the output size was defined as 128. An attention layer [12], [27] can be optionally applied. Subsequently, a dropout of 0,4 was applied. Finally, these two paths were concatenated and fed into a fully connected layer, followed by the application of the Log SoftMax function. Considering the optional attention layer, four network architectures have been implemented: MLSTM-FCN, MGRU-FCN, AMLSTM-FCN and AMGRU-FCN.

During the optimization process of the models and their hyperparameters, an empirical approach was adopted. The optimization was conducted in two stages, with the primary focus on the training dataset without the use of data duplication to reduce computation time. In the first stage, the model structure was optimized. The next stage involved fine-tuning the hyperparameters. Hyperparameters refer to the settings and configurations of the model that were not learned from the data, such as batch size, regularization parameters and optimization algorithms.

For the multi-class classification task, the chosen loss function was cross entropy loss. As optimization algorithm Adam was employed. The number of epochs was set to 2000. To prevent overfitting and reduce training time, an early stopping configuration was implemented. In this case, the patience was set to 50 epochs and a minimum improvement of the validation loss of $1 \cdot 10^{-3}$ per epoch was required to continue training. During the training process, the learning rate was adjusted after each epoch using a cycling learning rate approach proposed by Smith in 2017 [23]. This approach involved cyclically varying the learning rate within a specific range to potentially improve the optimization process. In this study, the cycling learning rate approach was implemented with an initially higher learning rate of $1 \cdot 10^{-5}$ and a minimum learning rate of $1 \cdot 10^{-8}$.

To determine the best performing model an evaluation process was applied. The evaluation process began by assessing the performance of different models on the raw dataset, considering various combinations of perspectives such as side view and front view. This evaluation helped identify the best performing model among the options. Once the best model was determined, its performance was further analyzed by examining its ability to handle lower FPS scenarios. This analysis provided insights into how the model performs when the video footage has a reduced frame rate.

Furthermore, the evaluation included assessing the model's performance on different training datasets using video augmentation techniques. This involved applying various transformations and modifications to the training videos to augment the dataset. By evaluating the model's performance on these augmented datasets, its robustness and generalization capabilities had been assessed.

III. Results

In this section the results from the keypoint detector for the skateboard and from the trick identifications achieved through the proposed work are presented.

A. Performance of the Skateboard Keypoint Detector

The performance evaluation of skateboard keypoint detector involved analyzing visual representations of the predicted images in comparison to their corresponding manually annotated images. Several key points have been noted during this evaluation:

- Close Keypoint Matches: The detected keypoints mostly closely match the manually annotated keypoints.
- Difficulty with Wheel Locations: However, the model encounters challenges in accurately determining the location or position of the skateboard wheels, particularly when the skateboard deck is facing towards the camera or the ground.

B. Performance of the Skateboard Trick Classification

In this section the results obtained from the evaluation process of the developed neural network models for skateboard trick classification are presented.

The initial evaluation compares the performance of different neural network models, including MLSTM-FCN, AMLSTM-FCN, MGRU-FCN and AMGRU-FCN, on the raw dataset with a frame rate of 30 FPS. Table II presents the accuracy scores achieved by each model with different combinations of perspectives (side and front view). Based on the accuracy scores across all models and views, the MGRU-FCN model demonstrated the highest accuracy of 27,27 % when both side and front views were considered together. Although the AMLSTM-FCN model performed slightly better in the side view with an accuracy of 27,27 % compared to the MGRU-FCN model's accuracy of 25,76 %, the difference is relatively small. Therefore, the MGRU-FCN model was chosen for further analysis.

Table II: Comparison of trick classification accuracy with the different combinations of perspectives (side and front view) among the different models on the raw test dataset at 30 FPS.

Model	**Side view**	**Front view**	**Accuracy in %**
MLSTM-FCN	yes	-	24,24
MLSTM-FCN	-	yes	16,67
MLSTM-FCN	yes	yes	25,70
MGRU-FCN	yes	-	**25,76**
MGRU-FCN	-	yes	19,70
MGRU-FCN	yes	yes	**27,27**
AMLSTM-FCN	yes	-	**27,27**
AMLSTM-FCN	-	yes	21,21
AMLSTM-FCN	yes	yes	22,73
AMGRU-FCN	yes	-	21,21
AMGRU-FCN	-	yes	16,67
AMGRU-FCN	yes	yes	22.73

In the next step the impact of the number of frames used for training on the accuracy of the model was tested. The MGRU-FCN model was trained using different frame rates: 30 FPS, 20 FPS, 15 FPS and 10 FPS. Table III displays the accuracy scores achieved on the test dataset for each combination of frame rate and perspective. Based on these results, it can be observed that the MGRU-FCN model consistently achieved higher accuracy when considering the side view only compared to combination of side and front view. With focus on the side view alone, the model's accuracy remained relatively stable across different FPS conditions, ranging from 24,24 % to 31,82 %. Lowering the number of images improved the accuracy of the model. On the other hand, when both the side and front views were considered, the accuracy varies more significantly, ranging from

Table III: Comparison of trick classification accuracy with the MGRU-FCN model at different frame rates and perspectives.

Model	View	FPS	Accuracy in %
MGRU-FCN	Side	30	27,27
MGRU-FCN	Side & Front	30	27,27
MGRU-FCN	Side	20	24,24
MGRU-FCN	Side & Front	20	27,27
MGRU-FCN	Side	15	28,79
MGRU-FCN	Side & Front	15	21,21
MGRU-FCN	Side	10	**31,82**
MGRU-FCN	Side & Front	10	22,73

21,21 % to 27,27 %. Therefore, if the side view is the primary factor of interest and the use of the front view is not essential, selecting the MGRU-FCN model with the side view at 10 FPS would be a suitable choice, as it achieved the highest accuracy of 31,82 % among the available options.

In order to improve the model's robustness and generalization, augmented training datasets were utilized. The performance of the MGRU-FCN model at 10 FPS was then evaluated using these augmented datasets to assess the impact of specific augmentations on classification accuracy. The achieved accuracies by the model on different training datasets are presented in Table IV. From the results, it was observed that the highest accuracy of 36,36 % was obtained when using the combination of raw data, horizontal flip, rotation and translation, as well as when all augmentations were combined. On the other hand, the training dataset with histogram equalization and horizontal flip yielded the lowest accuracy. The second-best accuracy of 34,85 % was achieved when combining the raw data with rotation and translation. The dataset including blur, noise and histogram equalization obtained an accuracy of 33,33 %.

Table IV: Achieved accuracies of trick classification on the test dataset, using different combinations of the training dataset.

Raw Data	Blur	Noise	Histogram	Horizontal Flip	Rotation	Translation	Accuracy in %
Yes	-	-	-	-	-	-	31,82
Yes	-	-	-	Yes	-	-	28,79
Yes	Yes	Yes	Yes	-	-	-	33,33
Yes	-	-	Yes	Yes	-	-	27,27
Yes	-	-	-	-	Yes	Yes	34,85
Yes	-	-	-	Yes	Yes	Yes	**36,36**
Yes	Yes	Yes	Yes	Yes	Yes	Yes	**36,36**

Based on these results, it is recommended to select the MGRU-FCN model trained with the augmented dataset consisting of raw data, horizontal flip, rotation and translation for further analysis, as it achieved the highest accuracy of 36,36 %. Adding the additional augmentations of blur, noise and histogram equalization only increased computational cost without improving accuracy.

The initial accuracy of the MGRU-FCN model on the raw dataset at 30 FPS with a side view perspective was 27,27 %. Through the subsequent analysis and improvements, the accuracy of the model was increased to 36,36 %.

Table V provides an evaluation of the MGRU-FCN model with the side view perspective at 10 FPS, trained using the augmented dataset consisting of raw data, horizontal flip, rotation and translation. The evaluation was based on precision, recall and F1 measures for each class in the trick identification task. The performance of the model varied across different classes.

Table V: MGRU-FCN model evaluation based on the test dataset at 10 FPS with side view perspective and the augmented training dataset.

Class	Precision in %	Recall in %	F1 Measure in %	Number of Data
Ollie	27	27	27	11
Fakie Ollie	18	18	18	11
BS Pop Shove-it	57	92	71	13
FS 180	0	0	0	10
Kickflip	55	67	60	9
No-Trick	12	8	10	12

For the Ollie class, the precision, recall and F1 measure were all 27 %, indicating poor model prediction. The model correctly predicted only 27 % of the instances in this class. Similarly, for the Fakie Ollie class, the precision, recall and F1 measure were all 18 %, indicating poor model prediction. The model correctly predicted only 18 % of the instances in this class. For the BS Pop Shove-it class, the precision was 57 %, indicating moderate model predictions. However, the recall was 92 %, suggesting that the model was able to correctly identify a high proportion of instances belonging to this class. The precision, recall and F1 measure for the FS 180 class were all 0 %, indicating that the model did not predicted any instances correctly for this class. The Kickflip class had a precision of 55 %, indicating somewhat accurate model predictions. The recall was 67 %, suggesting that the model could identify a relatively high proportion of instances belonging to this class. For the No-Trick class, the precision was 12 %, indicating that the model's predictions for this class were not very accurate. The recall was 8 %, suggesting that the model struggled to identify instances belonging to this class.

IV. Discussion

The goal of this work is to develop and analyze a neural network for the classification of skateboard tricks based on video recordings. The objectives of the study are to develop a large and diverse dataset of skate trick videos for training and testing the neural network and to analyze the results to interpret the performance of the model. The aim is to identify the factors that influence the performance of the model and provide insights for future improvements.

To achieve the first objective a dataset consisting of skateboard trick videos was created. The dataset comprises 439 clips belonging to the following classes: Ollie, Fakie Ollie, Kickflip, Backside Pop Shove-it, Frontside 180 and No-Trick. The collection process involved gathering videos from two distinct locations. There were no specific restrictions placed on the participants while executing the tricks, except for requiring a clean landing. This approach aimed to introduce diversity into the dataset. However, it is important to note that the limited number of samples in the dataset might not encompass the full range of variation in skater's style and filming conditions. This limitation could potentially impact the model's ability to generalize effectively to unseen data and may introduce biases or constraints on its performance. To address these limitations, it would be beneficial to gather a larger and more diverse dataset.

The neural networks are using video recording features as inputs. The YOLOv8 model was used to extract skateboard bounding box and human pose keypoints. For complex maneuvers like a Kickflip, a custom keypoint detector based on transfer learning from YOLOv8 pose model was developed. This detector showed promise in identifying skateboard keypoints accurately. However, challenges remained when the skateboard faced the camera or the ground. To improve performance, a larger and more diverse dataset can be used, including different angles and locations. This will enhance the detector's robustness and accuracy. Additionally, exploring the option of setting visibility flags for keypoints in the annotation platform could offer valuable insights and improvements to the process.

The execution of a skate trick is time-dependent, requiring time series classification to identify the trick by processing the data from the trick attempt. For this task a neural network architecture combining RNN and FCN has been chosen. Several models were developed, including MLSTM-FCN, AMLSTM-FCN, MGRU-FCN and AMGRU-FCN. An evaluation process was applied to identify factors influencing the performance of these models, comparing them for further selection. The models were evaluated based on their performance using the raw dataset, considering different views (side and front) and analyzing their accuracy scores. Based on the evaluation results, the MGRU-FCN model consistently achieved higher accuracy scores compared to the other models, especially when the side view was considered. Consequently, the MGRU-FCN model with the side view was selected for further investigations. To understand how different frames per second affect the performance of the model, the MGRU-FCN model was evaluated under varying FPS settings. The results indicated that the model's accuracy remained relatively stable across different FPS conditions. However, it was observed that the reduction of the FPS slightly improved the accuracy of the model. To enhance the model's robustness and generalization capabilities, augmented training datasets were utilized. The evaluation of these augmented datasets revealed the importance of specific augmentations in improving accuracy. Notably, the combination of raw data with horizontal flip, rotation and translation resulted in the highest accuracy of 36,36 %.

The analysis of the neural network's performance has highlighted both strengths and weaknesses. It suggests that using only the side view rather than a combination of side and front view can simplify the data acquisition process since only one camera is needed. The placement of the front view camera presented certain difficulties, as it was challenging for a rider to perform tricks while riding towards this camera, resulting in a higher variability in the rider's path in the video recording. On the other hand, the side view camera captured the rider passing by, leading to a more consistent drive-through direction in the video recording. Additionally, reducing the frames per second not only decreased computational costs but also improved the model's performance. However, despite these adjustments, the developed system's classification accuracy remained low and unreliable, both with and without augmentation techniques. The results suggest that there is potential for deep learning to classify skateboard tricks, as some tricks have a higher recall value, indicating the model's ability to correctly identify a significant proportion of samples for those classes.

However, further investigations are required to address the current limitations and improve the model's performance. The existing dataset and system might not be sufficient to achieve reliable classification accuracy and additional data collection and model refinement are needed to enhance the classification performance for this set of skate tricks.

To enhance the data collection process, it is recommended to improve the participant recruitment process. The subjective perception of the author of a certain resistance to technology within the skateboard scene may have made it difficult to find willing participants for this study. Exploring alternative methods of participant recruitment beyond relying solely on the social media profile of the local skate club, may help generate a more diverse dataset.

Regarding the neural network architecture, alternative models could be considered for the classification task. In this study only the combination of CNN and RNN models was tested. However, for the next evaluation, exploring other architectures such as transformer-based models, ensemble models (combining transformer-based with CNN and RNN) or 3D CNNs might be beneficial. These architectures have shown promise in various applications and could potentially improve the accuracy and performance of the classification task.

Another aspect to explore is the relevance of the features used in the classification task and the potential for improvement by incorporating more features. The current set of features might contain misleading information, which could affect the classification accuracy. Conversely, it is also possible that the selected features do not provide enough insight to effectively differentiate between various skate tricks. Additionally, if the dataset is sufficiently large, an alternative approach worth investigating is training a network directly on the raw videos without extracting features. This could potentially provide a more comprehensive understanding of the skateboard tricks and lead to improved classification results.

Automation of data processing and labelling could be a valuable improvement. For example, in the side view perspective, identifying the maximum height of a trick could be used to automatically cut the video into clips centered around the peak moment of the trick. Such automation could lead to the development of a program that reads the video, extracts features from each clip and saves them into a single file for further analysis. This automation process could streamline data processing and facilitate the creation of a larger dataset for training and evaluation.

V. Conclusion

In conclusion, this research aimed to classify skateboard tricks using deep learning algorithms, specifically CNNs and RNNs. With an accuracy of 36,36 % the feasibility of the classification task based on video data is not given, which could be solved through a larger and more diverse dataset.

The study introduced a custom keypoint detector for accurate skateboard movement analysis during tricks. While effective, challenges remained, especially when the skateboard deck was facing towards the camera. Overcoming these challenges may involve using a more diverse dataset and exploring visibility flags in the annotation process.

The deep learning architecture chosen, MGRU-FCN, demonstrated promise in time series classification, capturing temporal dependencies and the sequential nature of trick execution. This model outperformed alternative architectures, particularly when considering the side view camera. Classifying side view data may be beneficial in future data acquisition as it requires fewer cameras and provides a more consistent and standardized view of the rider's path.

Although some skate tricks achieved higher recall values, indicating the model's ability to correctly identify significant samples, the overall classification accuracy remained a significant challenge. To address this, further investigation into alternative feature sets, transformer-based models, ensemble models or 3D CNNs is recommended. Additionally, investigating the relevance of features, direct training on raw videos and automating data preprocessing could contribute to a more comprehensive understanding of skateboard tricks and further enhance the classification results.

Acknowledgment

The author would like to thank everyone who helped to create this paper. Special acknowledgements go to Bernhard Hollaus for supervising this work.

References

[1] B. Hollaus, S. Stabinger, A. Mehrle, and C. Raschner, "Using Wearable Sensors and a Convolutional Neural Network for Catch Detection in American Football," *Sensors*, vol. 20, no. 23, p. 6722, nov 2020.

[2] B. H. Groh, M. Fleckenstein, and B. M. Eskofier, "Wearable Trick Classification in Freestyle Snowboarding," in *IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. IEEE, 2016.

[3] H. Brock and Y. Ohgi, "Assessing Motion Style Errors in Ski Jumping Using Inertial Sensor Devices," *IEEE Sensors Journal*, vol. 17, no. 12, pp. 3794 – 3804, Jun. 2017.

[4] B. H. Groh, M. Fleckenstein, T. Kautz, and B. M. Eskofier, "Classification and Visualization of Skateboard Tricks using Wearable Sensors," *Pervasive and Mobile Computing*, vol. 40, pp. 42–55, 2017.

[5] M. A. Abdullah *et al.*, "The Classification of Skateboarding Trick Manoeuvres Through the Integration of IMU and Machine Learning," in *Intelligent Manufacturing and Mechatronics*, Z. Jamaludin and M. N. Ali Mokhtar, Eds. Singapore: Springer Singapore, 2020, pp. 67–74.

[6] N. K. Corrêa, J. C. M. de Lima, T. Russomano, and M. A. dos Santos, "Development of a Skateboarding Trick Classifier using Accelerometry and Machine Learning," *Research on Biomedical Engineering*, vol. 33, no. 4, pp. 362–369, Dec. 2017.

[7] J. Anlauff *et al.*, "A Method for Outdoor Skateboarding Video Games," in *Proceedings of the 7th International Conference on Advances in Computer Entertainment Technology*. ACM, nov 2010, pp. 40–44.

[8] M. A. Abdullah *et al.*, "The Classification of Skateboarding Tricks via Transfer Learning Pipelines," *PeerJ Computer Science*, vol. 7, 2021.

[9] B. Hollaus, B. Reiter, and J. C. Volmer, "Catch Recognition in Automated American Football Training Using Machine Learning," *Sensors*, vol. 23, 2023.

[10] S. P. Sahoo, S. Ari, K. Mahapatra, and S. P. Mohanty, "HAR-Depth: A Novel Framework for Human Action Recognition using Sequential Learning and Depth Estimated History Images," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 5, pp. 813 – 825, Oct. 2020.

[11] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "LSTM Fully Convolutional Networks for Time Series Classification," *IEEE Access*, vol. 6, pp. 1662–1669, Dec. 2017.

[12] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate LSTM-FCNs for Time Series Classification," *Neural Networks*, vol. 116, pp. 237–245, 2019.

[13] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," in *NIPS 2014 Workshop on Deep Learning*, 2014.

[14] N. Elsayed, A. S. Maida, and M. Bayoumi, "Deep Gated Recurrent and Convolutional Network Hybrid Model for Univariate Time Series Classification," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 10, no. 5, 2019.

[15] X. Zou, Z. Wang, Q. Li, and W. Sheng, "Integration of Residual Network and Convolutional Neural Network Along with Various Activation Functions and Global Pooling for Time Series Classification," *Neurocomputing*, vol. 367, pp. 39–45, 2019.

[16] W. Tang *et al.*, "Rethinking 1D-CNN for Time Series Classification: A Stronger Baseline," *CoRR*, vol. abs/2002.10061, 2020. [Online]. Available: https://arxiv.org/abs/2002.10061

[17] K. Fauvel, T. Lin, V. Masson, Élisa Fromont, and A. Termier, "XCM: An Explainable Convolutional Neural Network for Multivariate Time Series Classification," *Mathematics*, vol. 9, no. 23, pp. 1–20, Dec. 2021.

[18] H. Liu, Z. Dai, D. R. So, and Q. V. Le, "Pay Attention to MLPs," *CoRR*, vol. abs/2105.08050, 2021. [Online]. Available: https://arxiv.org/abs/2105.08050

[19] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A Transformer-based Framework for Multivariate Time Series Representation Learning," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, aug 2021, pp. 2114–2124.

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[21] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics," 2023, version 8.0.0. [Online]. Available: https://github.com/ultralytics/ultralytics/

[22] J. Solawetz and Francesco, "What is YOLOv8? The Ultimate Guide." 2023, accessed on 09.07, 2023. [Online]. Available: https://blog.roboflow.com/whats-new-in-yolov8/

[23] L. N. Smith, "Cyclical Learning Rates for Training Neural Networks," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017.

[24] F. Zhuang *et al.*, "A Comprehensive Survey on Transfer Learning," in *Proceedings of the IEEE*, vol. 109, no. 1. IEEE, 2020, pp. 43–76.

[25] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-Time Flying Object Detection with YOLOv8," 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2305.09972

[26] Z. Wang, W. Yan, and T. Oates, "Time Series Classification from Scratch with Deep Neural Networks: A Strong Baseline," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, May 2017.

[27] A. Vaswani *et al.*, "Attention Is All You Need," in *Advances in Neural Information Processing Systems*, Long Beach, CA, USA, 2017, pp. 5998–6008.

Patrick Juriga is with the Department of Medical and Health Technologies, MCI, Innsbruck, Austria. Patrick possesses extensive experience in various fields of mechatronics, with a particular emphasis on hardware programming.