Two Stream Network for Stroke Detection in Table Tennis

Anam Zahra, Pierre-Etienne Martin

To cite this version:

Anam Zahra, Pierre-Etienne Martin. Two Stream Network for Stroke Detection in Table Tennis. MediaEval 2021, Dec 2021, Online, Germany. hal-03482217

HAL Id: hal-03482217
https://hal.science/hal-03482217
Submitted on 15 Dec 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Two Stream Network for Stroke Detection in Table Tennis

Anam Zahra, Pierre-Etienne Martin
CCP Department, Max Planck Institute for Evolutionary Anthropology, D-04103 Leipzig, Germany
anam_zahra@eva.mpg.de, pierre_etienne_martin@eva.mpg.de

Figure 1: Pipeline method for stroke detection from videos. Cuboids of RGB and optical flow are fed to the network and classified as stroke or non-stroke. The feature dimension is described as follow: \( R \times G \times B \times \text{channels} \times \text{temporal} \times \text{height} \times \text{width} \).

ABSTRACT

This paper presents a table tennis stroke detection method from videos. The method relies on a two-stream Convolutional Neural Network processing in parallel the RGB Stream and its computed optical flow. The method has been developed as part of the MediaEval 2021 benchmark for the Sport task. Our contribution did not outperform the provided baseline on the test set but has performed the best among the other participants with regard to the mAP metric.

1 INTRODUCTION

With the advent of Convolutional Neural Networks (CNNs), especially after the success of AlexNet [9], object detection, localization, and classification from images and videos have greatly progressed [3, 5, 9, 22]. The development of computer vision methods has motivated broader applications in the academic world. Our team is currently working on egocentric recordings from children in kindergarten and at home. The analysis of these recordings shall give us an automatic overview of their interactions on a daily basis. We hope to link these interactions with their cognitive development and, thereby, better understand early child development. With our participation in the Sports Video Task [12], in the stroke detection subtask, we hope to perfect our knowledge in event detection and transpose it to our project.

The diversity of applications and visual data in sport, makes sports video analysis attractive for researchers. Automated sport event detection and action classification, especially from low-resolution videos, are helpful for monitoring and training purposes. For example in [6, 10], the authors automate the performance analysis for the training optimization of players. Similarly, Sports Video task at MediaEval 2021 benchmark aims at improving athlete performance and training experience through the first steps of stroke detection and classification from videos.

Event detection in videos is the first step to many other hot-topics such as video summarizing [7], automated semantic segmentation [1] and action recognition [4, 16]. These methods may be used to build summary, selecting highlights, and assisting players in training sessions. One way to approach the problem of event detection in sports with balls, can be through ball detection and tracking. Several researchers have tried to get the 2D, and 3D ball trajectories in order to achieve so [17, 18, 20].

Inspired from [8, 14, 15, 21], this method combines the optical flow and features learned from the RGB stream in order to detect a stroke in table tennis and assess its duration. This implementation is an extension of the baseline code provided by the Sport Task organizers [11].

2 APPROACH

Initially, we sought to use ball detection and tracking to perform stroke detection. The first implementation used the pretrained model TTNet [21]. However, the model failed to adapt to the acquisition conditions from TTStroke-21 [13], on which the task is built upon, and no fine-tuning was possible since no ball coordinates are available in the provided annotations. Therefore we decided to train a model from scratch.
In this section, we first present the preparation of the videos and then the model presenting the processed data. Both processes are depicted in Fig. 1. Post processing is performed to form a final decision.

2.1 Data Preparation
In video content analysis, the motion of objects of interest between frames can be of significant interest in order to understand their evolution in space. As such, we decided to use optical flow as a modality to perform stroke detection. Inspired by [14], we decided to use DeepFlow method [23] to compute the optical flow from consecutive frames. The optical flow is computed from frames resized to $320 \times 128$. This size was initially chosen to keep the ball at least two pixels big, as it has previously been done in [21]. Both the RGB and optical flow frames are consecutively stacked in a tensor of length 75. As in [11], stroke detection is tackled as a classification problem with two classes: "Stroke" and "Non-stroke".

2.2 Model
As shown in figure 1, our Two-Stream model is composed of two branches of the same length. Each branch is a succession of four blocks and each block is composed of a convolutional layer with $3 \times 3 \times 3$ filters, followed by a ReLU activation function, and a $2 \times 2 \times 2$ pooling layer. The output of each branch is then flattened and fed to a fully connected layer that outputs a feature vector of length 500. Both feature vectors are then concatenated and fed into a final fully connected layer of length two to predict the "Stroke" and "Non-stroke" classes. One branch takes RGB frames of the video and the other computed optical flow. The model is trained using a stochastic gradient descent method over 250 epochs with a learning rate of 0.001, a batch size of 10, a weight decay of 0.005, and a Nesterov momentum [19] of 0.5. The negative samples creation and input processing is the same as the baseline [11].

2.3 Post Processing
Our model classifies 75 consecutive frames. In order to create stroke segments over the whole video, we classify every 75 frames of the videos, which leads to applying a sliding window without overlap. If two consecutive segments are classified as stroke, the segments are fused to create only one stroke.

3 RESULTS AND ANALYSIS
The metrics for evaluating the detection performance are described in [12]. Our approach reached a mean Average Precision (mAP) of 0.00124 and a Global Intersection over Union (G-IoU) of 0.0700. It falls behind the baseline which reaches respectively 0.0173 and 0.144. Our other attempts using early concatenation of the RGB and Optical Flow modalities - meaning an input of size $5 \times 320 \times 128$ in one branch model - or training method without shuffling of the data, reached even lesser performance.

Table 1: Stroke concentration and duration in frame per set.

<table>
<thead>
<tr>
<th>Set</th>
<th># Strokes/1K frames</th>
<th>Mean</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>1.85</td>
<td>143.2</td>
<td>52</td>
<td>296</td>
</tr>
<tr>
<td>Valid</td>
<td>2.28</td>
<td>134.3</td>
<td>72</td>
<td>292</td>
</tr>
<tr>
<td>Test</td>
<td>0.57</td>
<td>361.0</td>
<td>75</td>
<td>4500</td>
</tr>
</tbody>
</table>

Indeed, by looking at the stroke distribution across the different sets, see table 1, we may notice how little the inferred stroke ratio is on the test set: 0.57 strokes for 1000 frames, whereas the stroke rate is 1.85 and 2.28 for 1000 frames in the training and validation sets. Furthermore, our post processing was not limited in term of stroke duration, leading to everlasting strokes: 4500 frames - meaning the fusions of 60 consecutive video segments. These points indicate that our post-processing method can be improved.

A better separation of the stroke may be reached by defining the event using ball tracking and the ball motion [2]. This was our initial attempt, inspired by [21], but the available pretrained model considers a different point of view and was unable to adapt to the TTStroke-21 videos point of view.

4 CONCLUSION
The Sports Video Task, and more specifically the stroke detection subtask, has proven to be challenging. Even if our implementation has learned to classify strokes, we were not able to outperform the baseline performance. We have underlined the importance of the post processing step through a stroke concentration and duration analysis. Furthermore, our failure to adapt a pretrained model on similar dataset, but with a different acquisition point of view, stresses the difficulty of the deep trained models to adapt to a change of scene, which is inherent to the fine-grained aspect of the classification subtask. As first time participants, we thought to tackle only one task to ease our submission. However, we now believe that a method tackling both the detection and classification may be the best for solving the Sport Video subtasks.
REFERENCES


