Siamese Spatio-temporal convolutional neural network for stroke classification in Table Tennis games

Pierre-Etienne Martin[1], Jenny Benois-Pineau[1], Boris Mansencal[1],
Renaud Péteri[2], Julien Morlier[3]
[1]Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, F-33400, Talence, France
[2]MIA, La Rochelle University, La Rochelle, France
[3]IMS, University of Bordeaux, Talence, France
pierre-etienne.martin@u-bordeaux.fr,jenny.benois-pineau@u-bordeaux.fr,boris.mansencal@labri.fr
renaud.peteri@univ-lr.fr,julien.morlier@u-bordeaux.fr

ABSTRACT

This work presents a Table Tennis stroke classification approach through a siamese spatio-temporal convolutional neural network - SSTCNN. The videos are recorded at 120 frames per second with players performing in natural conditions. The frames are extracted, resized and processed to compute the optical flow. From the optical flow, a region of interest - ROI - is inferred. The SSTCNN is then feed by RGB and optical flow ROIs stream to give a probabilistic classification over all the table tennis strokes.

1 INTRODUCTION

In the scope of video processing, action recognition and classification is one of the main challenge. In the Sport task of MediaEval 2019 [4], this aspect is underlined by providing a dataset of Tennis table recordings, `TTStroke-21` [6], where strokes have to be extracted and classified with the aim of improving athletes performances. As a first step, videos are provided with temporal segmentation and the task is to classify those segments. However, contrary to the common datasets widely used in image and video processing such as UCF-101 [8], HMDB [3] or Kinetics [1]; this task focuses on fined grained classification with the classification of strokes highly similar. The difficulty of this task is to be able to find the characteristics of each kind of stroke using a limited dataset without over-fitting it. In this paper, we present an approach aiming at providing data with enough inter-dissimilarity and focusing on intra-similarity to feed a neural network able to classify without over-fitting on a limited dataset.

2 APPROACH

To deal with the low inter-variability of the classes in `TTStroke-21` and avoid over-fitting on this sample of the dataset, we decided to use cuboids of optical flow in addition to cuboids of RGB images with spatio-temporal convolutions processed simultaneously through a Siamese architecture as presented in [6].

2.1 Optical Flow estimator

As shown in [7], flow estimators can have a strong impact on the classification, so we tested classification using two different flow estimators: DeepFlow [9] and Dense Inversive Search - DIS [2]. Because of the strong motion artefacts observed on DIS flow, this one is smoothed with a Gaussian blur using a kernel of size 3×3 and then multiplied by the computed foreground [10] to keep only foreground motion.

2.2 Spatial segmentation

RGB and Optical Flow are spatially segmented using a region of interest - ROI - of center $\mathbf{C_{roi}} = (x_{roi}, y_{roi})$ estimated from the maximum of the optical flow norm and the center of gravity of all pixels [6] as follows:

$$\begin{aligned}
&\mathbf{C_{max}} = (x_{max}, y_{max}) = \underset{x,y}{argmax}(||\mathbf{D}||_1) \\
&\mathbf{C_g} = (x_g, y_g) = \frac{1}{\sum_{\mathbf{C}\in\Omega} \delta(\mathbf{C})} \sum_{\mathbf{C}\in\Omega} \mathbf{C}\delta(\mathbf{C}) \\
&\text{with } \delta(\mathbf{C}) = \begin{cases} 1 & \text{if } ||\mathbf{D}||_1(\mathbf{C}) \neq 0 \\ 0 & \text{otherwise} \end{cases} \\
&x_{roi} = \alpha\ f_{\omega_x}(x_{max},\ W) + (1-\alpha)\ f_{\omega_x}(x_g,\ W) \\
&y_{roi} = \alpha\ f_{\omega_y}(y_{max},\ H) + (1-\alpha)\ f_{\omega_y}(x_g,\ H)
\end{aligned} \tag{1}$$

with parameters $\alpha = 0.6$, $\Omega = (\omega_x, \omega_y) = (320 \times 180)$ the size of the resized video frames, (W, H) the size of the data inputted to our network. The function $f_\omega(u, V) = max(min(u, V - \frac{\omega}{2}),\ \frac{\omega}{2})$ allows to have input data extracted within the boundaries of our data. To avoid jittering, we apply a Gaussian blur along the time dimension to average the center position using a kernel of size 40 and scale parameter $\sigma_{blur} = 4.44$.

2.3 Data normalization

The RGB image channels are normalized by their theoretical maximum value, 255 in our case, to map them into interval [0,1]. As done in [7] which compare different normalization methods, we decide to normalize the optical flow $\mathbf{V} = (v_x, v_y)$ using the mean μ and standard deviation σ of the maximum absolute values distribution of each optical flow components over the whole dataset. In the following equation v and v^N represent respectively one component of the OF $\mathbf{V}$ and its normalization.

$$\begin{aligned}
&v' = \frac{v}{\mu + 3\times\sigma} \\
&v^N(i,j) = \begin{cases} v'(i,j) & \text{if } |v'(i,j)| < 1 \\ SIGN(v'(i,j)) & \text{otherwise.} \end{cases}
\end{aligned} \tag{2}$$

This normalization method maps the values into interval [-1,1] and increases the magnitude of most vectors making the optical flow easier to process for classification of very similar actions such as Table Tennis strokes.

MediaEval'19, 27-29 October 2019, Sophia Antipolis, France

2.4 SSTCNN

Our Siamese Spatio-Temporal Convolutional Neural Network - SSTCNN, see Fig. 1, is constituted of 2 branches with three 3D convolutional layers with 30, 60, 80 filter response maps, followed by a fully connected layer of size 500. They take respectively cuboides of RGB values and optical flow computed from them of size ($W \times H \times T$)= ($120 \times 120 \times 100$). The 3D convolutional layers use $3 \times 3 \times 3$ space-time filters with a dense stride and padding of 1 in each direction. The two branches are fused through a final fully connected layer of size 21 followed by a Softmax function to output a probabilistic classification.

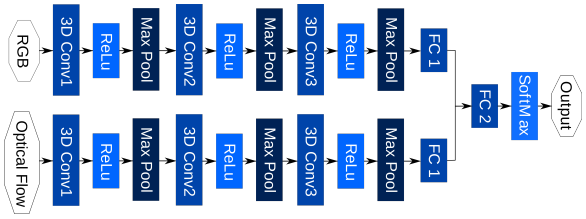

Figure 1: SSTCNN architecture

2.5 Data augmentation

Data augmentation is made online and is different for each epoch. Each stroke feed our SSTCNN once per epoch. For each stroke, we extract one video sample of size ($W \times H \times T$). The T successive frames from the RGB and Optical Flow are extracted following a normal distribution around the center of our stroke with standard deviation of $\sigma = \frac{\Delta t - T}{6}$. We also spatially augment the data by applying random rotation in the range $\pm 10°$, random translation in range ± 0.1 in x and y directions, random homothety in range 1 ± 0.1 and a 0.5 chance flip in horizontal direction and random channel swaps on the RGB data. We take extra care of applying those changing on the Optical Flow by updating its values according to the transformations. Transformations are applied and centered on the region of interest avoiding crops outside of the camera range.

2.6 Training and submitted runs

All models were trained from scratch. We used firstly 250 epochs with the data samples split randomly between all strokes and then split using only two videos for validation. However we noticed the results obtained by splitting the dataset between videos were not satisfying. After looking at the dataset in detail, this is due to the fact that most of the videos contain only one kind of stroke performed by the same player. So the model will over-fit easily to the player appearance and not the characteristics of the stroke itself. With such a limited dataset and a limited time window we preferred to focus on the random distribution of the strokes among our training and validation sets. The two first runs are the classification obtained with the model trained on the split dataset and saved on the minimum loss obtained on the validation set with two different flows presented in section 2.1. The other two runs are the same models but retrained from scratch using all data samples with the number of epochs used for obtaining best performance on the first validation set.

3 RESULTS

On the left side of the Table 1 we can see results of the first two runs from the models trained on the split database with 250 epochs; and on the right side two others runs obtained from the models trained with all the data.

Table 1: Runs results

Flow	Epochs	Train	Val	Test	Train	Test
DIS	249	70.4	52.6	19.2	61.2	17.8
DeepFlow	229	74.7	56.1	17.2	70.2	**22.9**

Compared to what has been obtained in previous work [6], the results are very low. The main differences are i) the lack of a negative class and ii) the split of the dataset in train and test sets between videos. It directly leads to an over-fitting of the dataset and makes the model much less able to do a proper classification. Best results were obtained by using DeepFlow estimator.

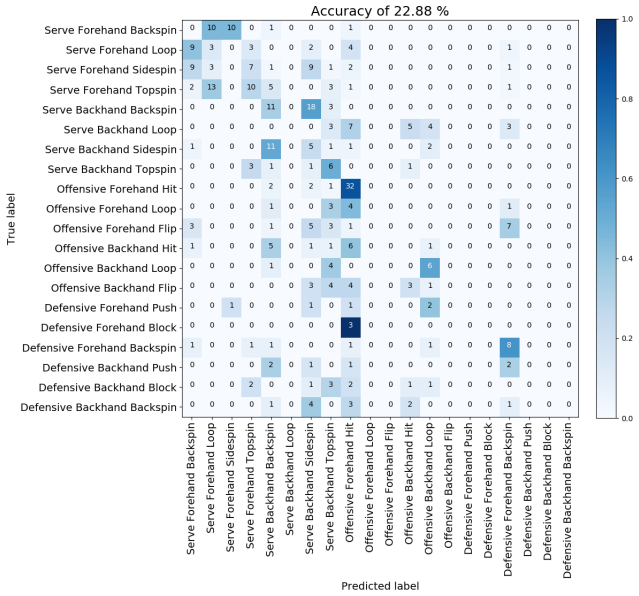

Figure 2: Confusion Matrix of our best run

Furthermore, if we consider the confusion matrix of our best run, Fig. 2, and group strokes in larger classes as: 'Forehand', 'Backhand' or 'Service', 'Offensive', 'Defensive' or their intersection (6 classes), we respectively get accuracies of 76.8%, 65.8% and 54.8%.

4 CONCLUSION

Despite a strong over-fitting, by grouping strokes together in larger classes, we can notice that some characteristics to recognize strokes are still learned. Furthermore, the work on `TTStroke-21` [5] is still in progress and the enrichment of the dataset will be a big contribution in the domain of action detection and classification especially for very similar actions.

ACKNOWLEDGMENTS

This work was supported by Region of Nouvelle Aquitaine grant CRISP and Bordeaux Idex Initiative.

REFERENCES

[1] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. 2017. The Kinetics Human Action Video Dataset. *CoRR* abs/1705.06950 (2017).

[2] Till Kroeger, Radu Timofte, Dengxin Dai, and Luc Van Gool. 2016. Fast Optical Flow Using Dense Inverse Search. In *ECCV (LNCS)*, Vol. 9908. Springer, 471–488.

[3] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso A. Poggio, and Thomas Serre. 2011. HMDB: A large video database for human motion recognition. In *ICCV*. IEEE Computer Society, 2556–2563.

[4] Pierre-Etienne Martin, Jenny Benois-Pineau, Boris Mansencal, Renaud Péteri, Laurent Mascarilla, Jordan Calandre, and Julien Morlier. 2019. Sports Video Annotation: Detection of Strokes in Table Tennis task for MediaEval 2019. In *Proc. of the MediaEval 2019 Workshop, Sophia Antipolis, France, 27-29 October 2019.*

[5] Pierre-Etienne Martin, Jenny Benois-Pineau, and Renaud Péteri. 2019. Fine-Grained Action Detection and Classification in Table Tennis with Siamese Spatio-Temporal Convolutional Neural Network. In *ICIP 2019*. IEEE, 3027–3028.

[6] Pierre-Etienne Martin, Jenny Benois-Pineau, Renaud Péteri, and Julien Morlier. 2018. Sport Action Recognition with Siamese Spatio-Temporal CNNs: Application to Table Tennis. In *CBMI 2018*. IEEE, 1–6.

[7] Pierre-Etienne Martin, Jenny Benois-Pineau, Renaud Péteri, and Julien Morlier. 2019. Optimal choice of motion estimation methods for fine-grained action classification with 3D convolutional networks. In *ICIP 2019*. IEEE, 554–558.

[8] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *CoRR* 1212.0402 (2012). arXiv:1212.0402

[9] Philippe Weinzaepfel, Jérôme Revaud, Zaïd Harchaoui, and Cordelia Schmid. 2013. DeepFlow: Large Displacement Optical Flow with Deep Matching. In *IEEE ICCV*. 1385–1392.

[10] Zoran Zivkovic and Ferdinand van der Heijden. 2006. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters* 27, 7 (2006), 773–780.