Sport Action Recognition with Siamese Spatio-Temporal CNNs: Application to Table Tennis

Pierre-Etienne Martin, Jenny Benois-Pineau, Renaud Péteri, Julien Morlier

Sport Action Recognition with Siamese Spatio-Temporal CNNs: Application to Table Tennis

Pierre-Etienne Martin
Univ. Bordeaux, LaBRI
Talence, France
pierre-etienne.martin@u-bordeaux.fr

Jenny Benois-Pineau
Univ. Bordeaux, LaBRI
Talence, France
jenny.benois-pineau@u-bordeaux.fr

Renaud Péteri
Univ. La Rochelle, MIA
La Rochelle, France
renaud.peteri@univ-lr.fr

Julien Morlier
Univ. Bordeaux, Bordeaux IMS
Talence, France
julien.morlier@u-bordeaux.fr

***Abstract*—Human action recognition in video is one of the key problems in visual data interpretation. Despite intensive research, the recognition of actions with low inter-class variability remains a challenge. This paper presents a new Siamese Spatio-Temporal Convolutional neural network (SSTC) for this purpose. When applied to table tennis, it is possible to detect and recognize 20 table tennis strokes. The model has been trained on a specific dataset, `TTStroke-21`, recorded in natural condition (markerless) at the Faculty of Sports of the University of Bordeaux. Our model takes as inputs a RGB image sequence and its computed Optical Flow. After 3 spatio-temporal convolutions, data are fused in a fully connected layer of a proposed siamese network architecture. Our method reaches an accuracy of 91.4% against 43.1% for our baseline.**

***Index Terms*—Action recognition, spatio-temporal convolutions, Siamese neural network, sport video analysis**

I. Introduction

Action recognition in video is one of the key problems in visual data interpretation. Despite intensive research, the recognition and differentiation of similar actions remains a challenge [1]. The target application of our research is fine grained action recognition in sports with the aim of improving athletes' performances. Without loss of generality, we are interested in recognition of strokes in table tennis. The low inter-class variability makes the task more difficult than as with usual general datasets, like UCF-101 [2] and DeepMind Kinetics [3], which are widely used in literature for action recognition. Twenty stroke classes and an additional rejection class are considered according to the rules of table tennis. This taxonomy was designed with professional table tennis teachers. We are working on videos recorded at the Faculty of Sports of the University of Bordeaux. Students are the sportsmen filmed and the teachers are supervising exercises conducted during the recording sessions. The recordings are markerless and allow the players to perform in natural conditions. The objective of this classification method is to help the teachers to focus on particular strokes performed by students. In the near future, we plan to build an automatic quality metric, measuring the similarity between an individual stroke compared to a reference one. The teacher could use this metric to efficiently correct strokes performed by students, and to help them improving their performances.

Nowadays, there exists quite a few video datasets for action recognition, some of which contains sport actions. We can mention the UCF-101 dataset [2] with sport actions shot at different scenes for different sports. They were downloaded from YouTube and the source of their annotation is unknown, sometimes it is semi - automatic as stated by the authors of [1]. In our case, the video dataset is complex for classification in the sense that the setting is almost always the same, the strokes are repetitive and annotation is fulfilled by professional athletes. The latter use quite a rich terminology. The linguistic analysis of annotations shows that for the same video and the same stroke, professionals do not employ the same degree of details in their annotations. This cannot be considered as a noise, but shows ambiguity and complexity of real-life data. This dataset is the first contribution of this paper.

The goal of our research is video indexing through the classification of strokes performed by an athlete. Our second contribution is to propose a new siamese 3D CNN architecture for this purpose. Our siamese architecture similarly processes RGB images and Optical Flow through a succession of spatio-temporal convolutions. A middle fusion is done before the calculation of the class scores. We use data augmentation in a spatial and temporal way during the training phase and compare the performances with models using only RBG images or Optical Flow data and also with early and late fusion approaches. Additionally, we compare the performances using our dataset with the baseline Two-Stream I3D method recently proposed in [4]. .

The remainder of the paper is organized as follows: in section II, related works using deep learning approaches are presented. In section III, we introduce our dataset and how it has been recorded and annotated. Section IV exposes the proposed classification method. Results are presented in section V. Conclusion and perspectives are drawn in section VI.

II. Related Works

The first deep learning breakthrough in image classification with AlexNet [5] has led to many improvements such as GoogLeNet [6], VGG-Net [7] and ResNet [8]. The next step was to extend these methods to the spatio-temporal domain for video classification. The main challenge in this task is to adapt existing works by taking into account temporal features.

978-1-5386-7021-7/18/$31.00

However, a direct extension of these methods to 2D+T presents some difficulties. The required space for training these models is indeed greater, necessitating a reduction of the batch size for training neural networks. This leads to a greater computational time, especially if models are trained from scratch. Therefore, the temporal dimension must be taken into account in a careful way.

In the work of [9] on multimodal gesture recognition, a first approach is to use 2D convolution and 3D Max Pooling on RGB-Depth images fused with Deep Belief Network using skeleton joint information. They obtain a score of 81% for the ChaLearn LAP gesture spotting challenge [10]. Inspired by [7], a so-called *Tube Convnet* (T-CNN) [11] feeds the VGGNet-16 architecture with a stack of motion-frames built with Faster R-CNN, the DBSCAN algorithm and optical flow fields. A second T-CNN introduced in [12] uses 3D convolutions and pooling. It takes as inputs 8-frame video clips performing 94.4% of accuracy on 24 classes of UCF-101. Another method developed by *Hakan Bilen et al.* [13] uses dynamic images as input for a CNN. Fused with the two stream networks [14], their results are promising, reaching 96% of accuracy on the UCF-101 dataset using pre-training on the ImageNet ILSVRC 2012 dataset [15].

The state of the art method in action recognition from videos is the Two-Stream I3D method [4], which reaches 98% and 93.5% of accuracy on UCF-101 dataset, respectively with and without pre-training on the miniKinetics dataset [3]. They follow the architecture of the two stream networks [14] but modify some of the convolutional layers with inception modules along with transfer learning. They proceed by classification of temporal sliding windows, which is a common approach for action classification [16]. In their work, the temporal window size is 64 frames which may not be enough to classify long-term actions. To overcome this limitation, [17] use Long-term Temporal Convolutions (LTC) considering as input video clips of 100 frames which improves the recognition of long-lasting actions. It uses a temporal window of 100 frames, at the expense of a less effective recognition of short term action. As pointed out in their article, this might be due to the repetition of the last frame to fill the required time window. Our proposed model was highly inspired by their method, as we also use a temporal window of $T = 100$ frames, but with a frame rate of 120 fps (against 25 fps in UCF-101 dataset [2]). The choice of this window length is suitable, because actions in table tennis are quick (see statistics in section III) and temporal aliasing should be avoided.

Note that video-based monitoring of athletes' performance is quite different from measuring fine movement. In [18], body worn inertial sensors are used. However, the use of invasive tools for monitoring might influence the performances of athletes. Their method, based on discrete wavelet transform and a random forest classifier, classifies 6 types of activities with 98% accuracy. We recall that our goal is to develop a monitoring system based on vision only.

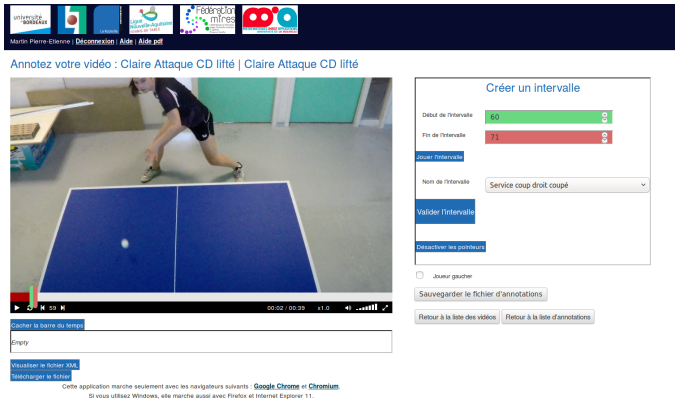


Fig. 1. Annotation platform

III. The TTStroke-21 dataset

Our dataset, TTStroke-21, is composed of 129 videos representing 94 hours of table tennis game at 120 fps, totaling 675 000 video frames. Sequences are recorded in a so-called *ecological* situation (no markers or sensors on the player). A player is filmed in two situations: performing repetition of the same stroke for training or in a match context. Sequences have been recorded indoors using artificial light. These videos have been annotated by table tennis players at the Faculty of Sports, University of Bordeaux (France). They represent a total of 1387 annotations. In order to avoid annotation errors as much as possible, one video recording was supposed to be annotated by at least 2 annotators. Unfortunately, this condition was hard to meet for all videos and despite the effort that went into cleaning the datasets build from crowdsourced annotations like EPIC-KITCHENS [19], errors still remain. The annotation process was designed as a crowdsourcing method where annotations where done during simultaneous sessions. The sessions were supervised by professional table tennis players and teachers. A user friendly web platform was developed by our team for this purpose (see Fig. 1). To obtain an exploitable dataset, the annotations had to be processed by different filters to remove annotation errors such as i) too long or too short duration, ii) mislabeling, iii) lack of labels. After that, each annotated stroke was considered as a positive example of its class, and negative examples were generated. We describe here the cleansing process in details.

A. Crowdsourcing filtering

In all crowdsourced applications, possible errors of the annotators should be taken into account. As the annotators were not familiar with the annotation platform at the beginning of the annotation sessions, there were some mislabelled portions of the videos. These mislabellings have been filtered out automatically by considering only annotations not starting at first frame (default parameter), annotations ending after the end of the video and annotations out of the time range (set between 0.6 and 2.3 seconds). The length of the time range was set up according to the domain knowledge of professional table tennis players of the Faculty of Sports. This allowed the

isolation of strokes ranging from a fast hit to a long serve. After filtering, 1074 annotations were retained.

B. Data organization

Since a video can be annotated several times by annotators, an action detection over all the annotations has been done. Our dataset is player-centered, with only one player in each video. In the case of two players in one video, we allow an overlap between each action of 0.25 to take into account the overlap of strokes. These actions are used in the classification problem. A last filter is applied by checking if all the annotations in one action are the same. If not, this action is not considered in our classification. Thus, a total of 1048 actions were conserved. The peak statistics of duration are $min = 0.63$s, $max = 2.27$s and the average duration is of 1.45s $\pm$ 0.36s. This filtering, supposing multiple annotations of the same video recording, still left some labeling errors as multiple labeling of the same clip by different annotators was not possible.

C. Selection of negative samples

Negative samples are created from videos with at least 11 detected actions. The other videos are not fully annotated most of the time and would lead to incorporation of strokes in the negative samples. The negative samples are video subsequences between each action. We allow the overlap with the previous and the subsequent action of 10 frames. It represents 10% of our target window length in the classification framework. However, this approach was still selecting wrong negative samples because of videos that were only partially annotated. This has been manually cleared to avoid the incorporation of strokes in negative samples. After these steps, 272 negative (non-stroke) samples have been selected from the whole dataset. Dataset `TTStroke-21` is available under request for research purposes.

IV. Proposed method

To be able to classify highly similar actions, a siamese 3D convolutional network model has been used to incorporate temporal features along with spatial ones (the temporal windows size has been set to $T = 100$). The action class is predicted from RGB video frames and their estimated motion vectors $\mathbf{D} = (V_x, V_y)$.

A. Architecture of the proposed network

Our Siamese Spatio-Temporal Convolution network (SSTC), Fig. 2, 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. All 3D convolutional layers use $3 \times 3 \times 3$ space-time filters with stride and padding of 1 in all directions. The two branches are combined in a second fully connected layer of size 21 (corresponding to the number of considered classes). A Softmax layer is finally applied at the end of our network to obtain a classification score.

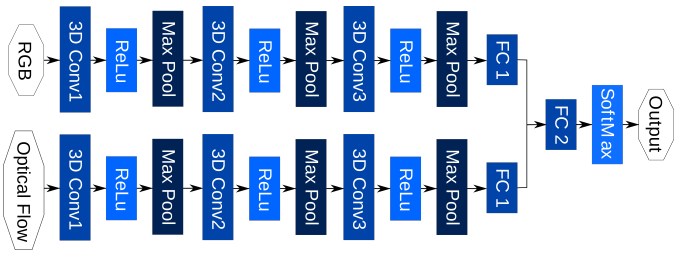


Fig. 2. Our Siamese (SSTC) architecture.

B. Input data

Branches of the network take RGB images and optical flow field as input. The optical flow (size 120×120) is computed using method [20]. The extracted frames from the video (size 1920×1080), are resized to 320×180 for computing the optical flow field.

1) Optical flow denoising: Due to flickering caused by artificial light during recording, some artifacts appear. To keep areas of interest only, we filter the Optical Flow using the Hadamard product between the foreground calculated using the method of Zivkovic and Van der Heijden [21] and the optical flow previously computed (Fig 3).

2) Spatial segmentation: Considering the obtained frame resolution and player position in our setting, the size of ROI inputted to the network was set to 120×120. The ROI center $\mathbf{X_{roi}} = (x_{roi}, y_{roi})$ is estimated from the maximum of the optical flow norm and the center of gravity of all pixels with non-null optical flow norm as follows:

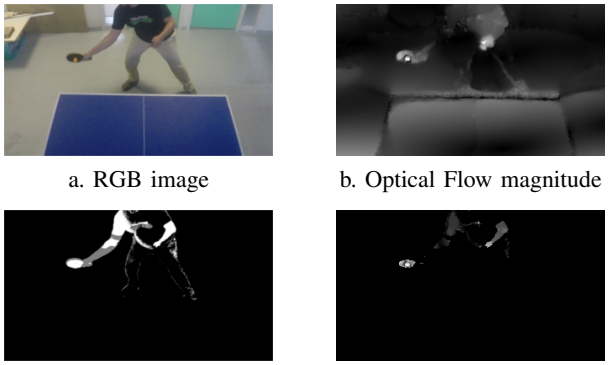
$$
\begin{aligned}
&\mathbf{X_{max}} = (x_{max}, y_{max}) = \underset{x,y}{argmax}(||\mathbf{D}||_1) \\
&\mathbf{X_g} = (x_g, y_g) = \frac{1}{\sum_{\mathbf{X}\in\Omega}\delta(\mathbf{X})}\sum_{\mathbf{X}\in\Omega}\mathbf{X}\delta(\mathbf{X}) \\
&\text{with} \quad \delta(\mathbf{X}) = \begin{cases} 1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

with parameters $\alpha = 0.6$, $\mathbf{\Omega} = (\omega_x, \omega_y) = (320 \times 180)$ the size of video frames, $(W,\ H)$ the size of the data inputted to our network (120×120). 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 (kernel of size 40 and scale parameter $\sigma_{blur} = 4.44$).

C. Data Augmentation

For each action, we extract one video sample of size ($W \times H \times T$)= ($120 \times 120 \times 100$). We extract the 100 frames from the RGB and Optical Flow temporally centered according to



a. RGB image b. Optical Flow magnitude

c. Estimated foreground d. Optical Flow after filtering

Fig. 3. Optical Flow filtering

the length of the action and spatially centered according to our spatial segmentation.

For spatial augmentation we apply random rotation in the range $\pm 10°$, a random translation in range ± 0.1 in x and y directions, and a random homothety in range 1 ± 0.1. Transformations are applied and centered on the region of interest.

To perform temporal augmentation we extract 100 successive frames following a normal distribution around the center of our action with standard deviation of $\sigma = 0.3 * ((w_t - 1) * 0.5 - 1) + 0.8$ with $w_t = \frac{2 * fps + 1}{W_s}$ (with $W_s = 6$ being the observation window size around our center). If the frames are not in the temporal boundaries of our actions, another random draw is done until the condition is satisfied.

D. Training step

Estimation of network parameters is fulfilled with Stochastic Gradient descent with Nesterov Momentum as in [5].We use a momentum of 0.5 and decrease it to 0.1 and 0.05 at epoch 1000 and 1500 respectively, as the momentum methods are known to oscillate at the beginning of the iterative process. We use a weight decay of 0.005. The maximum number of epochs is set to 2000. Cross-entropy loss is used as objective function. The batch size is relatively low for memory matter and is set to 10. The number of negative samples is chosen twice bigger than the mean of the number of actions per class. The dataset is split into training, validation and testing sets with the respective proportions: 70%, 20% and 10% (table I). We use different architectures: the Siamese architecture introduced in section IV-A to train our "Siamese model", and a convolution architecture using only one branch of the previous architecture. The last fully connected layer takes as an input only the output of the branch used. Three other models have been trained using this last architecture to compare performances. One using RGB images only will be denoted "RGB model", another one using only Optical Flow will be called "Optical Flow model" and the last one using RGB images and Optical Flow concatenated together (5 channels) will be referred as "Early Fusion model". For the Siamese model we use a learning rate of 0.001 and for the other models the learning rate is set to 0.01.

TABLE I
DATASETS TAXONOMY

Table tennis strokes	**Train**	**Val**	**Test**	**Total samples**
Def. Backhand Backspin	22	6	3	31
Def. Backhand Block	19	5	3	27
Def. Backhand Push	6	2	1	9
Def. Forehand Backspin	29	8	4	41
Def. Forehand Block	8	2	2	12
Def. Forehand Push	23	7	3	33
Off. Backhand Flip	25	7	3	35
Off. Backhand Hit	28	8	4	40
Off. Backhand Loop	21	6	3	30
Off. Forehand Flip	31	9	5	45
Off. Forehand Hit	45	13	6	64
Off. Forehand Loop	23	7	3	33
Serve Backhand Backspin	56	16	8	80
Serve Backhand Loop	43	12	6	61
Serve Backhand Sidespin	60	17	9	86
Serve Backhand Topspin	57	16	8	81
Serve Forehand Backspin	58	17	8	83
Serve Forehand Loop	56	16	8	80
Serve Forehand Sidespin	57	16	9	82
Serve Forehand Topspin	67	19	9	95
Non strokes samples	74	21	11	106
Total length	808	230	116	1154

TABLE II
PERFORMANCE COMPARISON OF THE DIFFERENT MODELS

	Accuracies			
Models	**Val**	**Test**	**TestVote**	**TestAvg**
I3D (RGB)	40	40.5		
I3D (OptFlow)	37.4	30.2		
I3D (RGB + OptFlow)	41.7	43.1		
RGB	88.7	78.5	78.5	81.9
Optical Flow	47.8	44	44	44.8
Early Fusion (RGB + OptFlow)	84.4	73.3	74.1	75
Late Fusion (RGB + OptFlow)	62.2	57.7	59.5	70.7
Siamese (without data aug)	90.43	87.9	88.8	**91.4**
Siamese	91.3	87.9	88.8	89.7

We use data augmentation on our training set for all the models and evaluate them at each epoch with the accuracy on the validation dataset without augmentation. Models with the best accuracy are saved for the next evaluations on the test set.

V. EXPERIMENTS AND RESULTS

Our deep learning models have been trained using Pytorch framework on GPU NVIDIA Tesla P100. To compare the performances of our models, we use the Two-Stream I3D model introduced by Carreira and Zisserman in [4] as our baseline and apply it to our dataset following their instructions for training (table II). The first max polling layer has been discarded because of the size of our input data which are twice smaller than theirs. The RGB images and Optical Flow streams are trained separately and a late fusion by addition of the class scores is performed to classify the action.

A. Evaluation methods

We stress that in our experiments the goal was to recognize the class of already localized stroke. We do not perform action detection but action classification. To evaluate our models on the test set, several methods have been used. The the first one, used also for the validation evaluation, consists in classifying the actions only by considering the T frames centered in each action. This method does not take into account the whole action and is based on the hypothesis that the main features are temporally centered. Two further methods consider all the frames of an action. For both of these methods, we perform a sliding window classification along the time dimension of the action with a step of 10 frames. We then obtain class scores for each window in the action. Our second method uses majority vote whereas our third method uses the average score of the obtained class scores. The three methods are respectively referred as "Test", "TestVote", and "TestAvg" in table II. As it can seen on Fig. II, the average score method performs the best. A gain of 12.9 % on the late fusion method and of 3.5 % on the siamese model with central window only is obtained. It stresses the fact that actions need to be entirely considered to be better classified since the main stroke features might not always be temporally centered.

B. Comparison with the baseline

Our models have outperformed the recent baseline model [4] which we have trained from scratch on our dataset, exactly as we did it with all our models. The maximum accuracy obtained on our dataset with our method is 91.4% against 43.1% with the I3D from [4]. One hypothesis to explain this behavior is that the Two-Stream I3D model is deeper than ours, and may overfit our dataset (which is more limited than UCF-101 and HMDB-51 datasets). Our second hypothesis is that the parameters advised by the authors may not fit to our problem. And even if it has been proven to be efficient on UCF-101, the low inter-class variability makes the task more difficult than usual. Yet, Fig 4 shows an overfitting since the beginning of the training, that supports the first hypothesis. Also the use of a 100 frames as input of our model in our method against 64 for Two-Stream I3D [4] has already proven to obtain better performances for classification of long and similar actions [17].

C. Architectures comparison

According to table II, our Siamese model outperforms all the other models, even though our RGB model performs quite similarly. The RGB model also outperformed the late fusion method, meaning the training of our Optical Flow model could still be improved when combined with RGB model, as it has been done in [17].

D. Analysis of the classifications

As it can be seen from the confusion matrix (Fig. 6), some classes are entirely incorrectly predicted. On the one hand, this is due to the lack of data in some classes. As shown in table I, the presence of the "Defensive Forehand Block" class is poor within the dataset. On the other hand, since the annotations are crowdsourced, some wrongly labeled actions are still present in the dataset leading to mislearned strokes. We noticed afterwards that this is the case with the "Defensive Backhand Push" stroke, some of which are annotated as "Defensive Forehand Push".

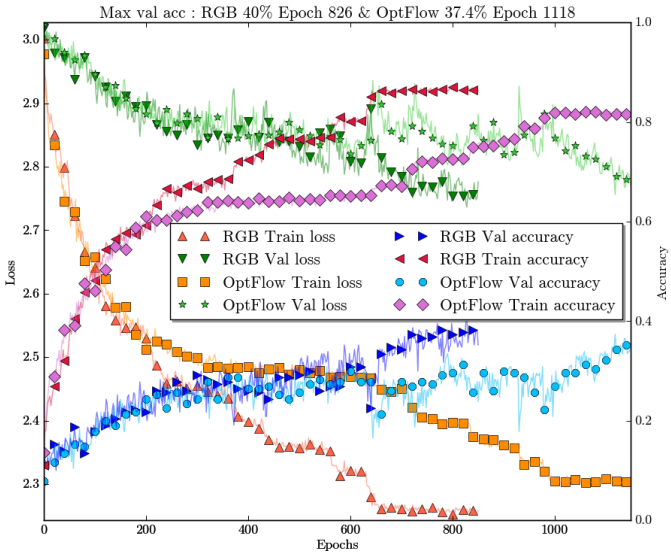


Fig. 4. Training process of the Two-Stream I3D models

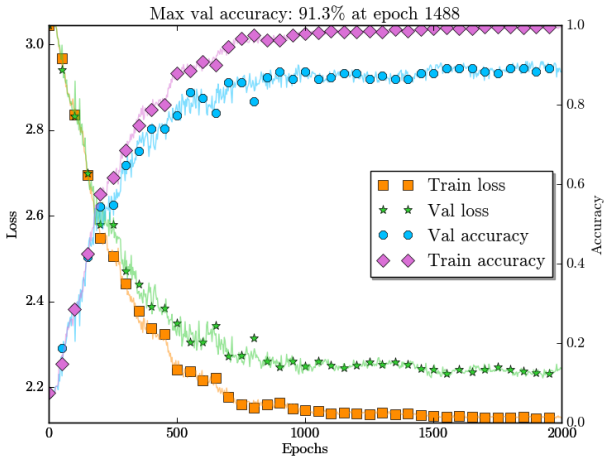


Fig. 5. Training process of our Siamese model

However, according to Fig. 5, it can be noticed that our model does not overfit the training dataset in contrast to the I3D models (see Fig. 4). Data augmentation did not improve our scores. This is certainly due to the length of the actions (maximum 2.3s i.e. 276 frames) compared to our time window ($T = 100$) which leads our model to learn non representative features.

With a final maximum score of 91.4, it is not obvious that we can obtain a significantly better score with our dataset that still has noisy crowdsourced annotations.

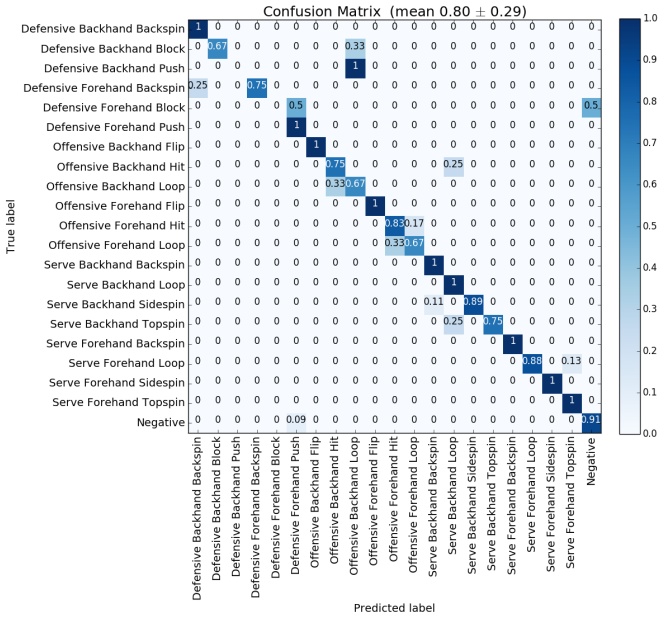


Fig. 6. Confusion Matrix on the test dataset using our Siamese model

VI. Conclusion and Perspectives

In the challenging task of table tennis strokes classification, this paper presented a Siamese spatio-temporal convolutional (SSTC) method along with other methods to complete this task. With an accuracy of 91.4%, our SSTC model has performed the best on a new dataset of table tennis strokes, `TTStroke-21`, recorded in real-world conditions and annotated with crowdsourcing. Before attempting to improve our results, this dataset must by enlarged and cleaned to obtain non noisy crowdsourced annotations since the impact of labeling error cannot be calculated without a full review of the whole dataset. Furthermore, this work is still in progress: the dataset is continuously enriched, and the next step is the joint detection and classification of strokes. In the future, we plan to obtain a qualitative measurement of the classified strokes with the aim to improve athlete performances and to develop pedagogical tools.

Acknowledgment

We would like to thank Alain Coupet, Ronan Le Merrer and Mathieu Dubos for their involvement in the acquisition and annotations of the table tennis dataset, and the reviewers for their constructive comments. This work was supported by Region of Nouvelle Aquitaine grant CRISP and Bordeaux Idex Initiative.

References

[1] C. Gu, C. Sun, S. Vijayanarasimhan, C. Pantofaru, D. A. Ross, G. Toderici, Y. Li, S. Ricco, R. Sukthankar, C. Schmid, and J. Malik, "AVA: A video dataset of spatio-temporally localized atomic visual actions," *CoRR*, vol. abs/1705.08421, 2017.

[2] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, vol. abs/1212.0402, 2012.

[3] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," *CoRR*, vol. abs/1705.06950, 2017.

[4] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," *CoRR*, vol. abs/1705.07750, 2017.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS, Lake Tahoe, Nevada, United States.*, 2012, pp. 1106–1114.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, 2015, pp. 1–9.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[9] D. Wu, L. Pigou, P. Kindermans, N. D. Le, L. Shao, J. Dambre, and J. Odobez, "Deep dynamic neural networks for multimodal gesture segmentation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1583–1597, 2016.

[10] S. Escalera, X. Baró, J. Gonzàlez, M. Á. Bautista, M. Madadi, M. Reyes, V. Ponce-López, H. J. Escalante, J. Shotton, and I. Guyon, "Chalearn looking at people challenge 2014: Dataset and results," in *Computer Vision - ECCV 2014 Workshops - Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part I*, ser. Lecture Notes in Computer Science, L. Agapito, M. M. Bronstein, and C. Rother, Eds., vol. 8925. Springer, 2014, pp. 459–473.

[11] Z. Li, W. Wang, N. Li, and J. Wang, "Tube convnets: Better exploiting motion for action recognition," in *2016 IEEE International Conference on Image Processing, ICIP 2016, Phoenix, AZ, USA, September 25-28, 2016*. IEEE, 2016, pp. 3056–3060.

[12] R. Hou, C. Chen, and M. Shah, "Tube convolutional neural network (T-CNN) for action detection in videos," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2017, pp. 5823–5832.

[13] H. Bilen, B. Fernando, E. Gavves, and A. Vedaldi, "Action recognition with dynamic image networks," *CoRR*, vol. abs/1612.00738, 2016.

[14] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., 2014, pp. 568–576.

[15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[16] A. Stoian, M. Ferecatu, J. Benois-Pineau, and M. Crucianu, "Fast action localization in large-scale video archives," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 26, no. 10, pp. 1917–1930, 2016.

[17] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1510–1517, 2018.

[18] A. Ahmadi, E. Mitchell, C. Richter, F. Destelle, M. Gowing, N. E. O'Connor, and K. Moran, "Toward automatic activity classification and movement assessment during a sports training session," *IEEE Internet of Things Journal*, vol. 2, no. 1, pp. 23–32, 2015.

[19] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, "Scaling egocentric vision: The epic-kitchens dataset," in *European Conference on Computer Vision (ECCV)*, 2018.

[20] C. Liu, "Beyond pixels: Exploring new representations and applications for motion analysis," Ph.D. dissertation, Massachusetts Institute of Technology, 5 2009.

[21] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognition Letters*, vol. 27, no. 7, pp. 773–780, 2006.