Fine grained sport action recognition with Twin spatio-temporal convolutional neural networks

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

myjournal manuscript No.
(will be inserted by the editor)

Fine grained sport action recognition with siamese spatio-temporal convolutional neural networks

Application to table tennis

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

Received: date / Accepted: date

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 (SSTCNN) 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, so called `TTStroke-21`, recorded in natural conditions at the Faculty of Sports of the University of Bordeaux. Our model takes as inputs a RGB image sequence and its computed residual Optical Flow. The proposed siamese network architecture comprises 3 spatio-temporal convolutional layers, followed by a fully connected layer where data are fused. Our method reaches an accuracy of 91.4% against 43.1% for our baseline.

Keywords Action recognition · Spatio-temporal convolutions · Siamese neural network · Sport video analysis

This work was supported by the CRISP project of the Nouvelle-Aquitaine Region and Bordeaux Idex Initiative

P.-E. Martin
LaBRI, University of Bordeaux, Talence, France
Tel.: +33-540-003-880
E-mail: pierre-etienne.martin@u-bordeaux.fr

J. Benois-Pineau
LaBRI, University of Bordeaux, Talence, France
Tel.: +33-540-008-424
E-mail: jenny.benois-pineau@u-bordeaux.fr

R. Péteri
MIA, University of La Rochelle, La Rochelle, France
E-mail: renaud.peteri@univ-lr.fr

J. Morlier
IMS, University of Bordeaux, Talence, France
E-mail: julien.morlier@u-bordeaux.fr

1 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. 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 recognizing actions contained in more general datasets, such as UCF-101 [23], DeepMind Kinetics [12] or AVA [8], which are widely used in literature for action recognition. Twenty stroke classes and an additional rejection class are considered according to the table tennis rules. We are working on videos recorded at the Faculty of Sports at the University of Bordeaux. Students of the faculty are filmed and the teachers are supervising exercises conducted during the recording sessions. 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.

There exists nowadays quite a few video datasets for action recognition, some of which contain sport actions. We can mention the UCF-101 dataset [23] with scenes shot for different sports or the Olympic Sports dataset [18] with 16 classes and 50 sequences per class, both downloaded from YouTube. For UCF-101 the source of their annotation is unknown, sometimes it is semi - automatic as stated by the authors of [8]. In our case, the coesidernd video dataset is complex for classification task as some stoke classes have only weak differences in their visual appearance leading to a low inter-class variability. Moreover, annotations are fulfilled by professional athletes, who 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 thus video indexing through the classification of strokes performed by an athlete. Our second contribution is the introduction of 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 performances with models using only RGB images or Optical Flow data and also with early and late fusion approaches. Additionally, we compare performances using our dataset with the Two-Stream I3D method recently proposed in [3] as a baseline.

The remainder of the paper is organized as follows: in section 2, related works using deep learning approaches are presented. Section 3, introduces our dataset and the way it has been recorded and annotated. Section 4 exposes the proposed classification method and results are presented in section 5. Conclusion and perspectives are drawn in section 6.

2 Related Works

The first deep learning breakthrough in image classification with AlexNet [13] has led to many improvements such as GoogLeNet [25], VGG-Net [22] and ResNet [9]. 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. However, a direct extension of these methods to 2D+T presents some difficulties. The required space for training these models is indeed far 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 [27] 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 [7]. Inspired by [22], a so-called *Tube Convnet* (T-CNN) [15] 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 [10] 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 [2] uses dynamic images as input for a CNN. Fused with the two stream networks [21], their results are promising, reaching 96% of accuracy on the UCF-101 dataset using pre-training on the ImageNet ILSVRC 2012 dataset [20]. Similary, PoTion [4] uses the movement of the human joints as features to improve the classification score of our baseline I3D [3] on UCF-101, HMDB [14] and JHMDB [11]. In [6], the temporal dimension is taken into account through a channel of 3D tensor.

The state of the art method in action recognition from videos is the Two-Stream I3D method [3], which reaches 98% and 93.5% of accuracy on UCF-101 dataset, respectively with and without pre-training on the miniKinetics dataset [12]. They follow the architecture of the two stream networks [21] but modify some of the convolutional layers with inception modules along with transfer learning. They proceed by classifying temporal sliding windows, which is a common approach for action classification [24]. In their work, the temporal window size is 64 frames which may not be long enough to classify long-term actions. To overcome this limitation, [26] 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 actions.

As pointed out in their article, this might be due to the repetition of the last frame to fill the required time window length. Our proposed model was inspired by their method, as we also use a temporal window of $T = 100$ frames, but with a frame rate of 120 per second (against 25 fps in UCF-101 dataset [23]). The choice of this window length is suitable, because actions in table tennis are fast (see statistics in section 3) and temporal aliasing should be avoided.

Note that video-based monitoring of athletes' performance is quite different from measuring fine movement. In [1] and [19], body worn inertial sensors are used. However, the use of invasive tools for monitoring might influence the performances of athletes. We recall that our goal is to develop a monitoring system based on vision only.

3 The `TTStroke-21` dataset

Our dataset, the so-called `TTStroke-21`, is composed of videos of table tennis games. This dataset is continuously enriched with videos of different players at different frame rates, spatial resolutions and camera viewpoints (table 1). These sequences are recorded indoors without markers using artificial light. The player is filmed in two situations: performing repetition of the same stroke for training or in a match context. These videos have been annotated by table tennis players and experts at the Faculty of Sports, University of Bordeaux (France). The annotation process was designed as a crowdsourcing method. The sessions are supervised by professional table tennis players and teachers. A user-friendly web platform has been developed by our team for this purpose, see Fig 1, where the annotator spots and labels strokes in videos: its starting and ending frames and the stroke class. The taxonomy is built upon a shake-hand grip of the racket. In order to avoid annotation errors as much as possible, one recorded video was supposed to be annotated by at least 2 annotators. Unfortunately, this condition was hard to meet for all videos, and despite efforts for cleaning the datasets build from crowdsourced annotations like EPIC-KITCHENS[5], errors still remain.

For our first experiment,129 videos have been considered, representing 94 minutes of table tennis game at 120 fps, totaling 675 000 video frames. They represent a total of 1387 annotations. To obtain an exploitable dataset, 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

Table 1 `TTStroke-21` description

Videos	Frames	Minutes	Players	Annotators	Annotations
241	2 219 225	369	17	18	2152

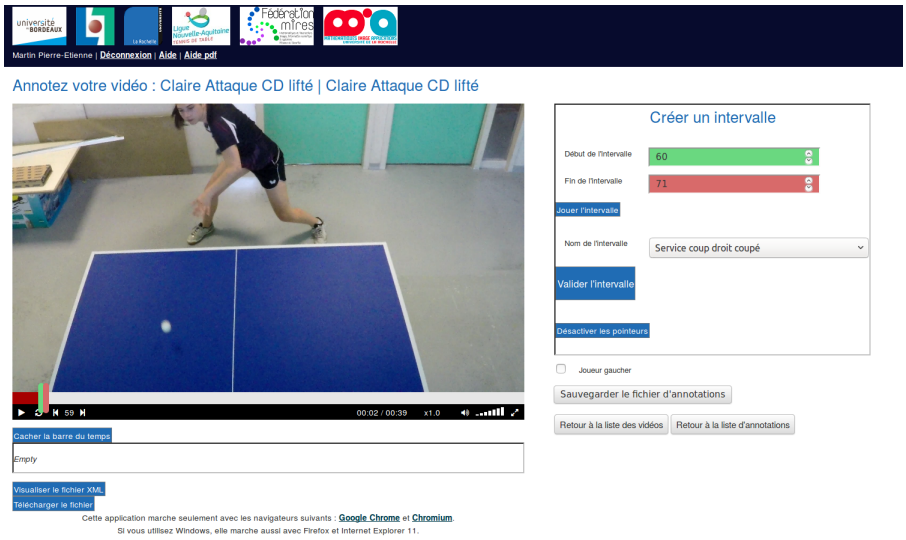


Fig. 1 Annotation platform

class, and negative examples were generated (see section 3.3). We describe here the cleansing process in details.

3.1 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 mislabels 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 accordingly 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. The peak statistics of stroke duration are $min = 0.64$s, $max = 2.27$s and the average duration is of 1.46s $\pm$ 0.36s.

3.2 Data labelling

Since a video can be annotated by several annotators, a stroke 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 annotation of 25% of stroke duration to take into account the overlap of the strokes. Above this overlap, the annotations are considered

to be part of the same stroke and are temporally fused. A last filter is applied by checking if labels of the same stroke are consistent. If not, this portion of video is not considered in our classification task. Thus, a total of 1048 strokes were conserved with a *min* duration of 0.83s,a *max* duration of 2.31s and an average duration of 1.47s $\pm$ 0.36s. This filtering, based on multiple annotations for the same recorded video, can still leave some labeling errors since multiple labeling of the same clip by different annotators was not always easy to meet in practise.

3.3 Selection of negative samples

Negative samples are created from video with more than 10 strokes detected. 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 sub-sequences between each stroke detected. We allow the overlap with the previous and the subsequent stroke of 10% of our target window length T used for classification. 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, 681 negative (non-stroke) samples with a mean duration of 2.34s $\pm$ 2.66s have been selected from the whole dataset. This high standard deviation comes from the non game activity of long period between

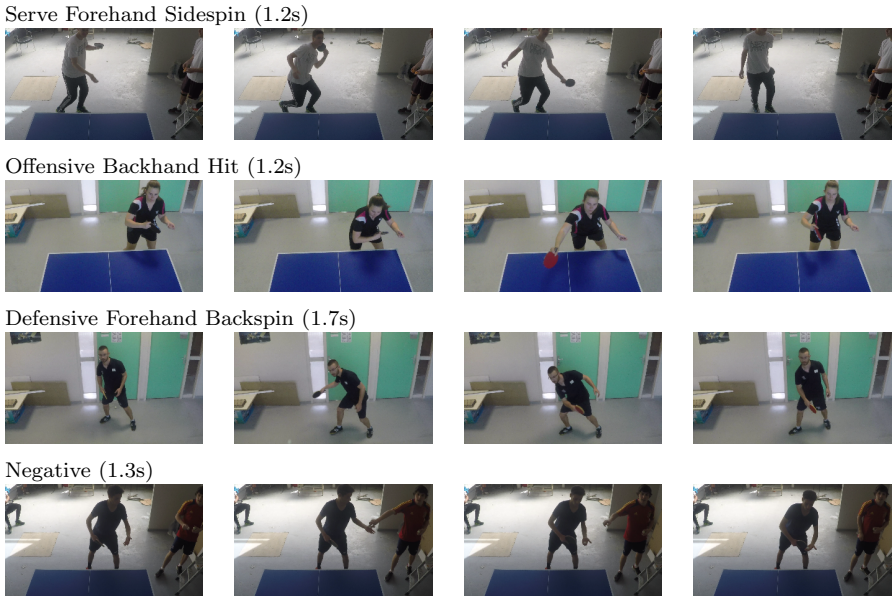


Fig. 2 Samples of `TTStroke-21` after extractions. In respective order the first frame, frames at 1/3 and 2/3 of the sample duration, and the last frame of the sample.

strokes, which can be due to a ball lost or talks between the players between games. Dataset `TTStroke-21`, with samples visible on Fig. 2, is available under request for research purposes.

4 Proposed method

To be able to classify highly similar actions, table tennis strokes in our case, a siamese 3D convolutional network model has been used to incorporate temporal features along with spatial ones. The stroke is predicted from RGB video frames and their estimated motion vectors $\mathbf{V} = (V_x, V_y)$.

We address two problems: i) classification of actions, ii) detection by classification. The *classification problem* (i) consists in assigning a label to a temporal segment corresponding to a stroke *with known temporal borders* in a given video recording. The *detection by classification problem* (ii) consists in labelling of strokes in the given video recording *without knowing their temporal borders*. In this case simultaneous partitioning of the recorded video into strokes is fulfilled. In both tasks we have to classify temporal windows of several frames. In one case the classification is done inside temporal borders, in the other case we need to slide a window with some step along the temporal axis and classify. In both cases, a deep convolution neural network classifier is proposed and we present the proposed architecture below.

4.1 Architecture of the proposed network

Our Siamese Spatio-Temporal Convolutional Neural Network (SSTCNN), Fig. 3, 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 using a bilinear interpolation of features form both branches 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.

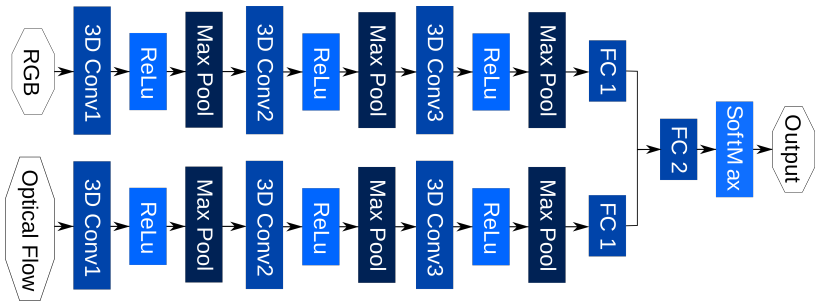


Fig. 3 Siamese Spatio-Temporal Convolutional Neural Network (SSTCNN) architecture

Strictly speaking, our network is not totally siamese since the sub-network are not entirely identical. Indeed we decided not to share the same weights because the types of the input data are heterogeneous, but the configuration in each of the branches remains the same.

4.2 Input data

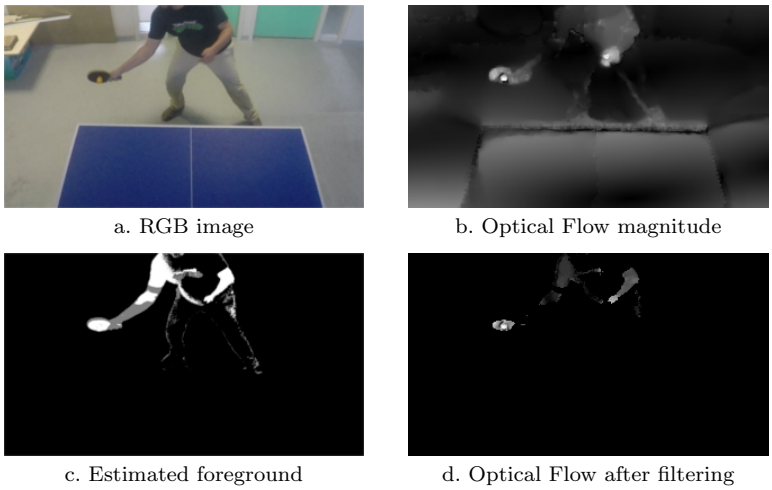
Branches of the network take RGB images and optical flow field of size ($W \times H \times T$). The optical flow is computed using method [16]. The extracted frames from the video (size 1920×1080), are resized to 320×180 for computing the optical flow field.

4.2.1 Optical flow filtering

Due to flickering caused by artificial light during recording sessions, some artifacts appear. To keep Regions-of-Interest (ROIs) only, we filter the Optical Flow using the Hadamard product between the foreground extracted with the method of Zivkovic and Van der Heijden [28] and the optical flow previously computed (Fig 4).

4.2.2 Spatial segmentation

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:



a. RGB image b. Optical Flow magnitude

c. Estimated foreground d. Optical Flow after filtering

Fig. 4 Optical Flow filtering

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

with parameters $\alpha = 0.6$, $\mathbf{\Omega} = (\omega_x, \omega_y) = (320 \times 180)$ the size of video frames. Function $f_\omega(u, V) = max(min(u, V - \frac{\omega}{2}),\ \frac{\omega}{2})$ allows to have data inputted to our network within the boundaries of our data. To avoid jittering within our cuboids of size $(W \times H \times T)$, we apply a Gaussian filter using a kernel of size k_{size} with scale parameter $\sigma_{blur} = 0.3 * ((k_{size} - 1) * 0.5 - 1) + 0.8$ along the temporal dimension to average the center position.

4.3 Data Augmentation

For each stroke, we extract one video sample of size $(W \times H \times T)$. Without data augmentation, the T frames from the RGB and Optical Flow are centrally extracted in the temporal and spatial dimension according respectively to the duration of the stroke Δt and our spatial segmentation.

For spatial augmentation we apply random rotation in the range $\pm 10°$, a random translation in x and y direction respectively in range $\pm 0.1 * W$ and $\pm 0.1 * H$, and a random homothety in the range 1 ± 0.1. Transformations are applied and centered on the ROI computed with our spatial segmentation.

To perform temporal augmentation we extract T successive frames following a normal distribution around the center of our stroke with standard deviation of $\sigma = \frac{\Delta t - T}{6}$, which represents more than 99% of chance to be in the temporal boundaries of the stroke (Fig. 5). However, if the frames are not in the temporal boundaries, another random draw is done until the condition is satisfied.

4.4 Training step

Estimation of network parameters is fulfilled using Stochastic Gradient Descent with Nesterov Momentum. 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 strokes per class. The dataset is

split into training, validation and testing sets with the respective proportions: 70%, 20% and 10% and is describe in table 2.

We use different architectures: i)the Siamese architecture introduced in section 4.1 to train our "Siamese model", and ii)a convolution architecture using only one branch of the previous architecture form each input: RGB or Optical Flow. In the latter case, the last fully connected layer takes, as an input, only the output of the branch used. Three other models have been trained using the latter architecture to compare performances. The first model which uses RGB images only will be denoted "RGB model". The second model which is built upon Optical Flow only, will be called "Optical Flow model" and the last one using RGB images and Optical Flow concatenated together (5 channels) in the input layer of the network will be referred as "Early Fusion model". Finally, we also apply a late fusion operator such as sum of the scores, on the one-branch models "RGB" and "OptFlow". 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.

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.

4.5 Evaluation methods

Classification task To compare the performances of our models on the classification task, we use the Two-Stream I3D model introduced by Carreira and Zisserman in [3] as our baseline and apply it to our dataset following their instructions for training. 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 models of I3D are trained separately. Also, a late fusion by summing up the class scores is applied to classify the stroke. These models are referenced as "I3D (RGB)", "I3D (OptFlow)" and "I3D (RGB+OptFlow)" respectively.

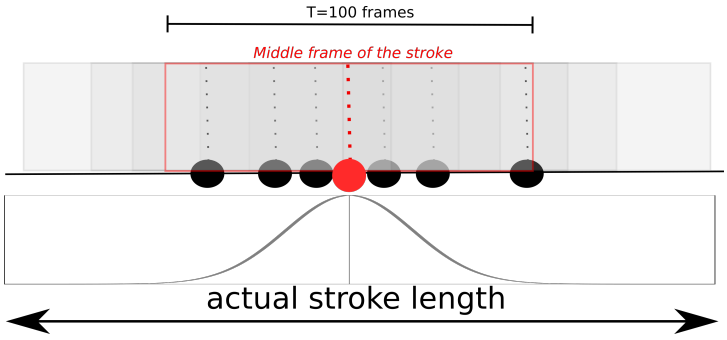


Fig. 5 Representation of 7 draw of the same stroke using temporal augmentation

Table 2 Datasets Taxonomy

	# Samples				# Frames		
Table tennis strokes	**Train**	**Val**	**Test**	**Sum**	**Min**	**Max**	**Mean**
Def. Backhand Backspin	22	6	3	31	121	233	189 ± 25
Def. Backhand Block	19	5	3	27	100	261	131 ± 37
Def. Backhand Push	6	2	1	9	121	229	155 ± 31
Def. Forehand Backspin	29	8	4	41	129	229	177 ± 25
Def. Forehand Block	8	2	2	12	100	137	115 ± 14
Def. Forehand Push	23	7	3	33	105	177	143 ± 19
Off. Backhand Flip	25	7	3	35	100	265	195 ± 49
Off. Backhand Hit	28	8	4	40	100	173	134 ± 21
Off. Backhand Loop	21	6	3	30	100	229	155 ± 32
Off. Forehand Flip	31	9	5	45	113	269	186 ± 44
Off. Forehand Hit	45	13	6	64	100	233	158 ± 34
Off. Forehand Loop	23	7	3	33	101	277	177 ± 43
Serve Backhand Backspin	56	16	8	80	133	261	188 ± 31
Serve Backhand Loop	43	12	6	61	100	265	186 ± 42
Serve Backhand Sidespin	60	17	9	86	129	269	193 ± 33
Serve Backhand Topspin	57	16	8	81	100	273	175 ± 48
Serve Forehand Backspin	58	17	8	83	125	269	182 ± 35
Serve Forehand Loop	56	16	8	80	100	273	171 ± 51
Serve Forehand Sidespin	57	16	9	82	101	273	192 ± 39
Serve Forehand Topspin	67	19	9	95	100	273	184 ± 52
Non strokes samples	74	21	11	106	100	1255	246 ± 154
Total	808	230	116	1154	100	1255	182 ± 65

We stress that in this evaluation, the goal is to recognize the class of already localized stroke. To evaluate our models on the test set, three methods have been used. The first one, used also for the validation evaluation, consists in classifying the strokes only by considering *the T frames temporally centered in each stroke*. This method does not take into account the whole stroke duration and is based on the hypothesis that the main features are centered in time. Two further methods consider *all the frames of a stroke*. For both of these methods, we perform a sliding window classification along the time dimension of the stroke with a step of $\delta t = 0.1T$ frames. We then obtain class scores for each window in the stroke. 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" and the performances are shown in table 3.

Detection by classification To evaluate the performances of our method for detection and classifications in videos, we compare our predictions with the ground truth which is built from the crowdsourced annotations of `TTStroke-21` dataset. Since the videos are limited in the diversity of strokes, experiments for this task have been conducted with the whole dataset which incorporates strokes and negative samples that were in the training, validation and test datasets.

The joint detection and classification is done through the classification of segments of video using a sliding window of size T with step 1. Different decision has been experimented to integrate classification results along the time and thus predict a label for the given time interval. The majority vote and max average decision - which average the probabilities over the classes - use a window decision of size $1.5T = 150$ and are denoted as "Vote" and "Average". Another decision is experimented which filter the predictions using a Gaussian kernel of size $2T + 1$ with scale parameter $\sigma = 0.5T$. Because the detection may not be exact in time according to the crowdsourcing annotations, the prediction is considered correct at the boundaries of strokes if it is classified as negative stroke or as one of the stroke that are overlapping. This overlap of label is set to 20% of the stroke duration. The performances are shown in table 4.

5 Experiments and Results

Our deep learning models have been trained using PyTorch framework on GPU NVIDIA Tesla P100. The size of the input data have been set to $(W \times H \times T) = (120 \times 120 \times 100)$. As explained in section 1, T has been chosen with respect to the rapidity of strokes and represents the minimum stroke duration: 0.83s as described in section 3.2.
W and H have been set according to the distance of the players to the camera, and thus to their visual appearance size in frames. The kernel size of the temporal Gaussian filter on the regions of interest is set to $\frac{1}{3}$s which represents, at 120 fps, a size of $k_{size} = 41$.

5.1 Comparison of performances

As it can be seen in Fig. 3, 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

Table 3 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	84.4	73.3	74.1	75
Late Fusion	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

with central window only is obtained. One can conclude that strokes need to be entirely considered to be better classified since the main stroke features might not always be temporally centered.

Furthermore, our models have outperformed the recent baseline model [3] 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 [3]. 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 they report their results on). Our second hypothesis is that the parameters suggested by the authors may not fit to our problem. Finally, our dataset is more challenging than the UCF-101 dataset used in their experiments. Indeed, the low inter-class variability makes the task more difficult than usual. Yet, Fig 6 shows an overfitting since the beginning of the training, that supports the first hypothesis. Moreover, a hundred frames are used as input of our model against 64 for Two-Stream I3D [3]. This has already been proven to obtain better performances for classification of long and similar actions [26], which is our case for table tennis strokes using a frame rate of 120 per second.

According to table 3, 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

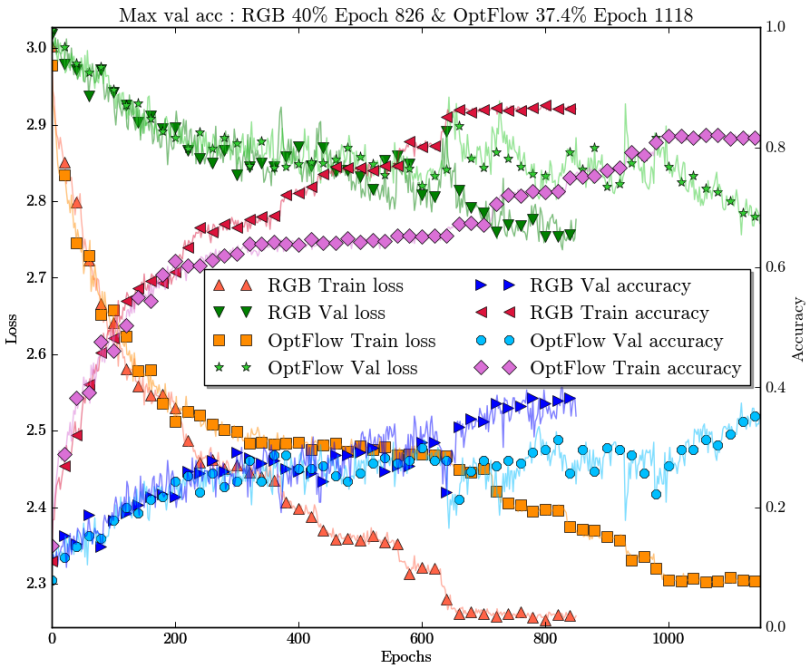


Fig. 6 Training process of the Two-Stream I3D models

model could still be improved when combined with RGB model, as it has been done in [26].

5.2 Analysis of the classification results

As it can be seen from the confusion matrix (Fig. 7), some classes are entirely wrongly predicted. This must be due to the lack of training data in those classes. As shown in table 2, the presence of the "`Defensive Forehand Block`" class is poor within the dataset. Moreover, since the annotations are crowdsourced, some wrongly labeled strokes 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 being annotated as "`Defensive Forehand Push`".

However, according to Fig. 8, it can be noticed that our model does not overfit the training dataset in contrast to the I3D models (see Fig. 6). Data augmentation did not improve our scores. This is certainly due to the length

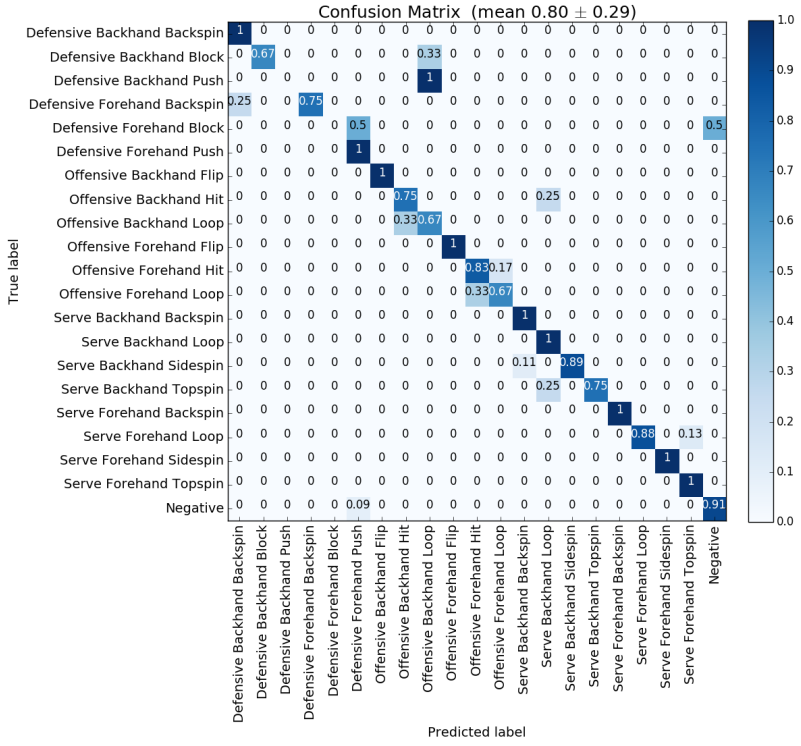


Fig. 7 Confusion Matrix on the test dataset using our Siamese model

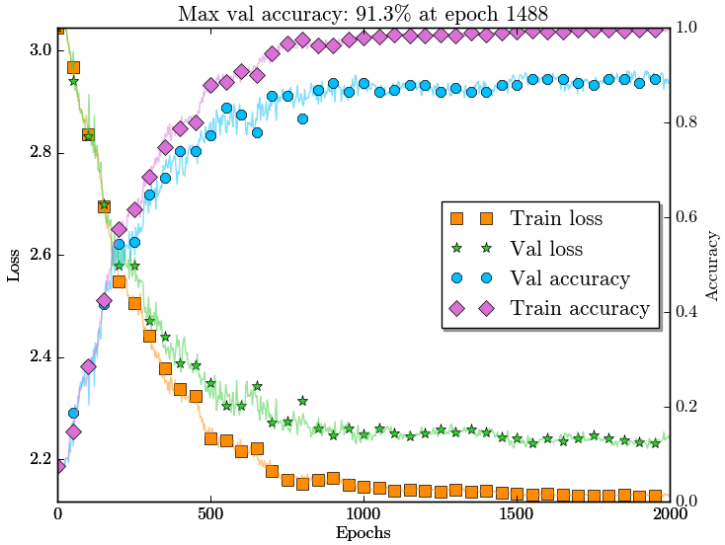


Fig. 8 Training process of our Siamese model

of the strokes (maximum 2.3s i.e. 276 frames) compared to our time window ($T = 100$) which leads our model to learn non representative features.

5.3 Joint stroke detection and classification

In the first part of table 4, we reached 79.6% of accuracy for detection and classification in videos with the siamese model. However, a video can be mostly constituted of negative samples (none stroke class), which can make our performance evaluation biased. This is why in the second part of Table 4 we show performances without counting the Negative labels or their overlaps, making the evaluation method much more discriminant since even the overlap will not be considered. Results are lower but still very satisfying since we reach 75.6% of accuracy for stroke detection and classification with the RGB model. Additionally, in the second part of the table 4, we can notice the weighted method using Gaussian kernel can be useful for decision when we are not taking into account the negative samples in videos. This is certainly due to the part of the video where an action is started but not ended because the ball do not reach the player, leading to high probability of stroke presence for short video samples. Surprisingly, we can also notice how the performances decrease for the Optical Flow and Siamese model, whereas RGB model maintain a correct score. We can explain this behavior by the composition of the dataset and the presence of negative samples in the videos. Negative samples will be much more easier to classify for our Siamese and optical Flow model since the motion is lower than in the strokes samples and may lead to slow learning of

Table 4 Performance of stroke detection and classification

	Accuracies		
Models	**Vote**	**Average**	**Gaussian**
RGB	76.6	78.4	77.9
Optical Flow	75	75.1	75
Siamese	79.1	**79.6**	79.2
without taking into account negative labels			
RGB	71.4	73.4	**75.6**
Optical Flow	19.3	20.1	22.1
Siamese	66.5	67.1	72.2

the strokes. It also underlines the influence of the optical flow field on features computed for classification in the siamese model.

6 Conclusion and Perspectives

In the challenging task of action recognition in sport video, with a weak inter-class variability, this paper presented a Siamese spatio-temporal convolutional neural network (SSTCNN) to complete this task. With an accuracy of 91.4%, our SSTCNN model has performed the best on a new dataset of table tennis strokes, `TTStroke-21`, recorded in real-world conditions and annotated with crowdsourcing. The dataset has also been used to test the performance of our models on the detection and classification of the stroke through classification of cuboids of videos. Our SSTCNN performs the best when considering all the classes but if the negative samples are not taken into account, our RGB model takes the lead. Furthermore in recent work [17], we show the effects of the Optical Flow normalization on the performance and recent results proves Siamese model to be more effective on the classification and detection task. Experimentation are still being conducted to understand and improve our results. The dataset is continuously enriched with new acquisitions, different players and camera viewpoints, in order robustify the model with respect to the acquisition conditions. We plan to develop pedagogical tools using our model to help students and teachers in the training sessions.

Acknowledgements We would like to thank Alain Coupet from sport faculty, expert and teacher in table tennis, for the proposed table tennis strokes taxonomy and all the players and annotators for their involvement in the acquisition and annotation processes leading to `TTStroke-21`.

References

1. Ahmadi, A., Mitchell, E., Richter, C., Destelle, F., Gowing, M., O'Connor, N.E., Moran, K.: Toward automatic activity classification and movement assessment during a sports training session. IEEE Internet of Things Journal **2**(1), 23–32 (2015)

2. Bilen, H., Fernando, B., Gavves, E., Vedaldi, A.: Action recognition with dynamic image networks. CoRR **abs/1612.00738** (2016)
3. Carreira, J., Zisserman, A.: Quo vadis, action recognition? A new model and the kinetics dataset. CoRR **abs/1705.07750** (2017)
4. Choutas, V., Weinzaepfel, P., Revaud, J., Schmid, C.: Potion: Pose motion representation for action recognition. In: CVPR 2018, 2018, pp. 7024–7033. IEEE Computer Society (2018)
5. Damen, D., Doughty, H., Farinella, G.M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., Wray, M.: Scaling egocentric vision: The EPIC-KITCHENS dataset. CoRR **abs/1804.02748** (2018)
6. Debard, Q., Wolf, C., Canu, S., Arné, J.: Learning to recognize touch gestures: Recurrent vs. convolutional features and dynamic sampling. In: 13th IEEE International Conference on Automatic Face & Gesture Recognition, 2018, pp. 114–121 (2018)
7. Escalera, S., Baró, X., Gonzàlez, J., Bautista, M.Á., Madadi, M., Reyes, M., Ponce-López, V., Escalante, H.J., Shotton, J., Guyon, I.: 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, pp. 459–473 (2014)
8. Gu, C., Sun, C., Vijayanarasimhan, S., Pantofaru, C., Ross, D.A., Toderici, G., Li, Y., Ricco, S., Sukthankar, R., Schmid, C., Malik, J.: AVA: A video dataset of spatio-temporally localized atomic visual actions. CoRR **abs/1705.08421** (2017)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR **abs/1512.03385** (2015)
10. Hou, R., Chen, C., Shah, M.: 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, pp. 5823–5832 (2017)
11. Jhuang, H., Gall, J., Zuffi, S., Schmid, C., Black, M.J.: Towards understanding action recognition. In: International Conf. on Computer Vision (ICCV), pp. 3192–3199 (2013)
12. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., Zisserman, A.: The kinetics human action video dataset. CoRR **abs/1705.06950** (2017)
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS, Lake Tahoe, Nevada, United States., pp. 1106–1114 (2012)
14. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T.A., Serre, T.: HMDB: A large video database for human motion recognition. In: ICCV, pp. 2556–2563. IEEE Computer Society (2011)
15. Li, Z., Wang, W., Li, N., Wang, J.: 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, pp. 3056–3060 (2016)
16. Liu, C.: Beyond pixels: Exploring new representations and applications for motion analysis. Ph.D. thesis, Massachusetts Institute of Technology (2009)
17. Martin, P., Benois-Pineau, J., Péteri, R., Morlier, J.: Optimal choice of motion estimation methods for fine-grained action classification with 3d convolutional networks. In: Submitted to ICIP 2019. IEEE (2019)
18. Niebles, J.C., Chen, C., Li, F.: Modeling temporal structure of decomposable motion segments for activity classification. In: ECCV 2010, pp. 392–405 (2010)
19. Noiumkar, S., Tirakoat, S.: Use of optical motion capture in sports science: A case study of golf swing. In: ICICM, pp. 310–313 (2013)
20. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Li, F.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115**(3), 211–252 (2015)
21. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NIPS, pp. 568–576 (2014)
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014)
23. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild. CoRR **1212.0402** (2012)

24. Stoian, A., Ferecatu, M., Benois-Pineau, J., Crucianu, M.: Fast action localization in large-scale video archives. IEEE Trans. Circuits Syst. Video Techn. **26**(10), 1917–1930 (2016)
25. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015, pp. 1–9 (2015)
26. Varol, G., Laptev, I., Schmid, C.: Long-term temporal convolutions for action recognition. IEEE Trans. Pattern Anal. Mach. Intell. **40**(6), 1510–1517 (2018)
27. Wu, D., Pigou, L., Kindermans, P., Le, N.D., Shao, L., Dambre, J., Odobez, J.: Deep dynamic neural networks for multimodal gesture segmentation and recognition. IEEE Trans. Pattern Anal. Mach. Intell. **38**(8), 1583–1597 (2016)
28. Zivkovic, Z., van der Heijden, F.: Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recognition Letters **27**(7), 773–780 (2006)

Mr. Pierre-Etienne Martin received his M.S. degree from the University of Bordeaux, Pázmány Péter Catholic University and Autonomous University of Madrid through the Master degree Image Processing and Computer vision Master promotion 2015-2017. He is pursuing his PhD thesis on "Fined grained action recognition in table tennis for athletes performance improvement" at the University of Bordeaux in Talence, France. He is supervised by Jenny Benois-Pineau and co-supervised by Renaud Péteri.

Jenny Benois-Pineau Jenny Benois-Pineau is a full professor of Computer

science at the University Bordeaux and chair of Video Analysis and Indexing research group in Image and Sound Department of LABRI UMR 58000 Université Bordeaux/CNRS/IPB-ENSEIRB. She is now a chair of international relations at College of Sciences and Technologies at University Bordeaux.
She obtained her PhD degree in Signals and Systems in Moscou and her *Habilitation Diriger la Recherche* in Computer Science and Image Processing from University of Nantes France.
Her topics of interest include visual content mining, image and video analysis and indexing, artificial intelligence for visual content analysis, healthcare, cultural applications. She is the author and co-author of more than 180 papers in international journals, conference proceedings, book chapters. She is Senior Acciotated Editor JEI SPIE and associated editor of EURASIP Signal Processing: Image Communication, Elsevier, Multimedia Tools and applications, Springer, journals. She has served in numerous program committees in international conferences and workshops: ACM MM, CIVR, ICIP, ICMCE, CBMI, AMR, IPTA, SAMT, ECMCS. She gave invited lectures at the universities of Sussex (GB), UPC, UAM (Spain), UNAM, IPN (Mexico), University of North Carolina at Chapel Hill, Brooklynn Polytechnic, NJTI (USA), Firenze (Italy), Klagenfurt(Austria). In 2016 she was nominated Knight of the French Order of Academic Palms.

Renaud Péteri received the engineering degree in physics and image processing from Telecom Physique Strasbourg, France, the M.S. degree in photonics and image processing from the University of Strasbourg, in 2000, and the Ph.D. degree in image and signal processing from MINES ParisTech, France, in 2003. Since 2005, he has been an Associate Professor with the University of La Rochelle, La Rochelle, France, and a member of the Mathematics, Image and Applications Laboratory.
His current research interests include signal and image processing, dynamic textures, video analysis, computer vision and machine learning.
Dr. Péteri was an ERCIM Post-Doctoral fellow at the Hungarian Academy of Sciences in 2004 and at the Mathematics and Computer Science Institute, Amsterdam, The Netherlands, in 2005. He was also an invited scholar at the University of California, San Diego in the SVCL laboratory in 2013.