Deep Learning for Domain-Specific Action Recognition in Tennis

Silvia Vinyes Mora
Department of Computing
Imperial College London
London, SW7 2AZ, UK
sv212@ic.ac.uk

William J. Knottenbelt
Department of Computing
Imperial College London
London, SW7 2AZ, UK
wjk@doc.ic.ac.uk

Abstract

Recent progress in sports analytics has been driven by the availability of spatio-temporal and high level data. Video-based action recognition in sports can significantly contribute to these advances. Good progress has been made in the field of action recognition but its application to sports mainly focuses in detecting which sport is being played. In order for action recognition to be useful in sports analytics a finer-grained action classification is needed. For this reason we focus on the fine-grained action recognition in tennis and explore the capabilities of deep neural networks for this task. In our model, videos are represented as sequences of features, extracted using the well-known Inception neural network, trained on an independent dataset. Then a 3-layered LSTM network is trained for the classification. Our main contribution is the proposed neural network architecture that achieves competitive results in the challenging THETIS dataset, comprising low-resolution monocular videos of tennis actions. We also show that the network is learning semantically meaningful information as most errors are interpretable and sensitive to player expertise.

1. Introduction

Sports analytics are on the rise, thanks to the volume and richness of data that is now available in this domain. For years, sports data was collected manually and consisted mainly of match results and coarse statistics (e.g. percentage of first serves in, in tennis). In recent years, spatio-temporal data such as locations of the players and high-level information has been made available, enabling the analysis to go a step further. In our work we are interested in classifying fine-grained tennis actions automatically from videos with the objective to bring an extra dimension to the analysis of the sport. We focus on tennis, but our work has the potential to be extended to other sports.

Current research in vision-based action recognition applied to sports is limited, mirroring the lack of benchmark datasets for this problem. Some of the most popular sports action datasets include UCF-Sport [19, 23] or more recently the Sports-1M dataset [13]. These datasets contain videos from many different sports and their labels describe which sport is being played. Different from these, we are interested in detecting finer-grained actions, such as specific tennis shots (serve, backhand and forehand) or even which sub-type of stroke (e.g. flat serve). This task accentuates the imbalance between a high intra-class variability and a low inter-class variability, bringing an additional challenge when compared to coarser action recognition. However, we also think that in order for action labels to be useful in more nuanced applications like professional training, these must be of fine-grained actions.

Some research has been done in the area of tennis action recognition. In [30, 31] the authors present a video descriptor based on optical flow and classify actions into 'left-swing' and 'right-swing' with a support vector machine. In [8] tennis actions are classified into 'non-hit', 'hit' and 'serve'. Unfortunately, the videos used in these experiments are not publicly available ([8] uses the ACASVA Actions Dataset [6] which provides features and labels, but not the RGB videos). Therefore, for our work we wanted to evaluate our methods using a publicly available dataset so that future research can be compared to our methods.

Amongst the many existing datasets for action recognition, we found one that conveniently suited our objective of fine-grained action recognition in tennis: THETIS [10]. Presented in 2013, THETIS is a complete dataset of fine-grained tennis actions comprising footage from 55 different subjects performing 12 distinct tennis shots multiple times. The videos are RGB, low-definition, monocular and shot in-the-wild, with dynamic background and occlusions. Our objective is to build a model able to classify the videos into the 12 fine-grained actions from raw footage, without the

need of pre-processing (e.g. silhouette detection) and with the ability to generalize to other tasks. For this reason, we are interested in exploring deep learning techniques instead of more traditional approaches based on hand-crafted features (both will be described later in more detail).

The proposed algorithm extracts features from each frame individually by using the well-known convolutional neural network (CNN) named Inception [25, 26], pre-trained on an independent image dataset and without fine-tuning. The resulting sequences of features are then fed to a deep neural network consisting of three stacked long short term memory units (LSTMs), a particular type of recurrent neural network (RNN).

The main contribution of this paper is the presentation of this neural network and its successful application to the challenging THETIS dataset. We also provide interesting insights from the results: first, we show that the algorithm is sensitive to the level of player expertise. Second, we compare the network's performance for classifying different levels of fine-grained actions (stroke type such as 'serve' vs sub-type such as 'flat-serve') and how it can be best trained for each task. Finally, we also show that our approach can be extended to action recognition in general by the application of our network to the HMDB dataset [16].

The rest of the paper is organized as follows. Section 2 revises previous work and state-of-the art techniques for action recognition. Section 3 depicts the characteristics of the two datasets used in our experiments and describes our methodology. Section 4 shows the experimental results and evaluation. Finally, section 5 concludes the paper and offers directions for future work.

2. Related work

2.1. Techniques for action recognition

Research in action recognition encompasses problems from a broad range of scenarios and their characteristics affect dramatically the choice of technique that is best suited to solve the problem. These are some of the variations that may occur:

- Action type: coarse or fine-grained (e.g. 'person playing tennis' vs 'person doing flat service in tennis').
- Scene setting: actions recorded in an experimental setting or in-the-wild. The latter may contain changes in illumination, occlusions or moving background.
- Video properties: monocular or multi-view, static or moving camera, high or low definition .

Amongst the approaches to video-based action recognition, two main categories can be be drawn: classifiers based on hand-crafted features and deep neural networks.

2.1.1 Classifiers based on hand-crafted features

Classifiers based on hand-crafted features are the most classical approach. They generally involve two main steps: feature extraction and classification. Extraction of hand-crafted features is based on domain-knowledge and some of the most popular techniques include Histogram of Oriented Gradients (HOG) [4], Harris detector [17], Motion Boundary Histograms (MBH) [5] or Cuboid detector [7]. Classifiers built on top of these features have achieved impressive results making their success undeniable but the major drawback in using them is that their selection can be problem dependent and difficult to generalize.

2.1.2 Deep architectures for action recognition

Different from hand-crafted features which are engineered and pre-defined, learned features are obtained through the performance of a machine learning task. For example, a neural network that learns to classify labeled images will contain in its hidden layers a representation of the input data that can be used as features to represent such data. These learned features have the potential of detecting structures that are more semantically meaningful and of being more generalizable. In recent years, learned features have gained popularity and they have been shown to be extremely powerful in the field of still image understanding. In particular, CNNs have exceeded any other state-of-the-art method in the domain of image classification [25, 15].

Driven by these achievements, attention has been brought to the application of deep neural networks to video classification. Unfortunately, their application to video processing has been proven to be more challenging and their success cannot yet be compared to that in still images. This can be attributed to two main limitations. First, video data is more complex than still images because of the temporal dependencies, requiring models to learn more complicated structures. Second, the availability of large datasets is reduced in comparison to still images. For instance, video classification benchmark datasets – such as KTH [20], Weizmann [3, 9], UCF Sports Action Dataset [19, 23], UCF-50 [18], HMDB-51 [16] – contain a smaller number of classes.

Progress has been made in overcoming these issues and applying deep networks to action recognition. In [2] the authors extend a traditional 2D CNN to 3D, incorporating the time domain, to learn features and then use an LSTM for classification. Their results improve upon other deep learning approaches and are competitive with hand-crafted based classifiers. Their experiments also show the benefits of using LSTMs in comparison to traditional RNNs. In [13], an end-to-end CNN video classifier is presented and evaluated in the Sports-1M dataset. They investigate different approaches of incorporating the time dimension, by fus-

ing the information across the time domain earlier or later in the network. Interestingly, their best model performs in hand with their single-frame model, opening the question of whether these features are capturing any motion information for the classification task. In [22] a two-stream CNN is presented, with a spatial stream that works on single frames and a temporal stream that utilizes optical flow. Their results outperform these presented in [13] and are competitive with state-of-the-art hand-crafted models. All of these models bring insights to the application of deep learning to videos but also highlight the difficulty of transferring their potential from still images to video sequences.

3. Methods

3.1. Experimental datasets

3.1.1 THETIS

Most of our experiments were conducted on the THETIS dataset [10]. It contains 1980 monocular RGB videos of 12 tennis actions performed three times by 55 different players (31 amateurs and 24 experienced). Actions are performed using a tennis racket but there is no tennis ball in the videos. The 12 actions are:

- backhand (with two hands)
- backhand
- backhand (slice)
- backhand (volley)
- forehand (flat)
- forehand (open stance)
- forehand (slice)
- forehand (volley)
- service (flat)
- service (kick)
- service (slice)
- smash

We used the RGB videos from the dataset but other data such as depth, skeleton 2D and 3D and silhouettes are also provided. Some challenges of this dataset are that videos contain moving background and the video sequences vary in length. Figure 1 shows a sample of frames from the dataset.

To our knowledge, only two publications make use of THETIS in action recognition experiments and there are no published results on the RGB videos alone. [10] presents the dataset and experiments accompanying it. They perform action recognition using state-of-the-art algorithms applied to 2D and 3D skeleton data. They achieve an average accuracy of 60.23% and 54.40% respectively, compared to a 92.99% accuracy when applied to the well-known KTH dataset [20], showing how challenging the THETIS dataset is. In [27], experiments are performed using silhouette data achieving an accuracy of 86%.

Figure 1. Samples from THETIS dataset.

Evaluation

As recommended by the authors, we performed a leave-one-out cross-validation procedure. For each experiment, all the videos from a specific subject are selected as the test set, videos from five others subjects (randomly selected) are kept as validation set and the rest are used to train the network. This procedure is repeated three times for each experiment. For the evaluation, we show a normalized confusion matrix of the results averaged between all subjects and across the three repetitions of the experiment and provide the accuracy and F1 scores, to assess the precision and recall.

3.1.2 HMDB

To show the applicability of the proposed NN to general action recognition tasks, we also show experiments performed on the HMDB dataset [16]. It contains 6849 videos from 51 actions that range from facial actions like smiling to body movements like climbing, horse riding or hand shaking. Videos are extracted mostly from movies but also from other datasets, and they can be considered in-the-wild.

Evaluation

In our experiments we use the three different splits of the data (into training and testing sets) as provided by the authors. For each split, the training set contain 3 570 examples, which we randomly divide into training and validation sets with 70% and 30% of the data respectively. Results are displayed as for THETIS with a normalized confusion matrix, accuracy and F1 scores.

3.2. Action classification

Our action classification algorithm is composed of two main steps: first feature extraction using the Inception neural network and second classification through a deep LSTM network.

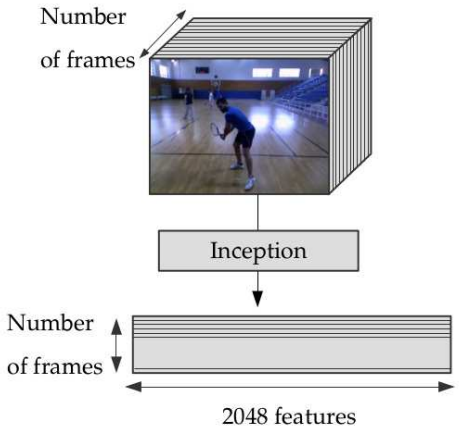

Figure 2. Feature extraction pipeline diagram.

3.2.1 Feature extraction

Inception is the name of a well-known deep CNN architecture, from [25, 26]. It was first introduced in 2015, obtaining the best results in the ImageNet Large-Scale Visual Recognition Challenge '14 (ILSVRC) – an image classification challenge of 1000 categories containing about 1.2 million images for training. Inception is a network 22 layers deep consisting of traditional convolutional layers stacked in the lower layers and 'Inception modules' stacked at higher layers. Every Inception module concatenates the output of the following operations preformed on its input (which is the result from the previous layers): 1×1 convolution, 1×1 convolution followed by 3×3 convolution, 1×1 convolution followed by 5×5 convolution and 3×3 max pooling followed by a 1×1 convolution.

Motivated by its performance in the image classification task, we chose Inception as our feature extraction algorithm. In ILSVRC '14, the network was able to attach coarse-grained labels to still images such as 'person playing tennis' but we wondered whether the features in the last layers also carried information about a person's posture or features discriminative for the action recognition in videos. Interestingly, the Inception architecture was designed to optimize computational resources so that inference could be done in settings such as mobile vision. We find this to be critically important for many action recognition applications.

In our work, Inception is used for feature extraction as shown in Figure 2. Each video clip, whose duration ranges from 90 to 150 frames, is cropped to the first 100 frames (alternate frames for HMDB, where videos are longer). At each frame, the previously mentioned network is applied to make predictions and the resulting 2048 features from the previous-to-last layer are stacked into a 2048×100 representation of the video. For videos shorter than 100 frames, we employ zero-padding. These are presented to our classification network to learn to label them with the appropriate tennis shot type.

Different from many applications in which the last layer is retrained for the specific problem, we decided not to retrain the network using THETIS. First, we wanted to see how applicable the features learned from ImageNet were in a different context. Second, given the small size of the THETIS dataset compared to the datasets used in deep architectures, we wanted to avoid as much as possible overfitting by fine-tuning the network to this particular dataset.

3.2.2 Deep LSTM for action classification

LSTM cell architecture

RNNs are a type of neural network with the ability to learn time dependencies, making them very suitable to process sequences. At time t, the output of the RNN h_t is calculated by taking as inputs its previous output h_{t-1} and the current element of the sequence x_t as follows:

$$h_t = f(W_{xh} \times x_t + W_{hh} \times h_{t-1} + b) \quad (1)$$

where W_{xh} and W_{hh} are weight matrices, b the bias and f is the output activation function.

Although RNNs are suitable for learning time dependencies, when applied to long sequences, the gradient is likely to vanish. In 1997, a type of RNN called LSTMs were introduced to overcome this issue [12]. LSTMs are particularly suited to learn long-term dependencies in sequences, such as in video classification or speech processing. They are composed of memory cells, which contain a memory state c that is updated with the new inputs but controlled by gates determining which information to keep and what to forget.

The cell state c_t is updated by forgetting some information through the multiplication of the previous cell state c_{t-1} and the forget gate f_t and by adding new information, controlled by the input gate i_t. Finally, the output gate o_t, controls the output of the cell h_t. The LSTM implementation that we used is based on [11, 29] (Figure 3) and the calculations of the activations are as follows:

Input gate

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (2)$$

Forget gate

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (3)$$

Memory cell state

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (4)$$

Output gate

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (5)$$

Hidden state

$$h_t = o_t \tanh(c_t) \quad (6)$$

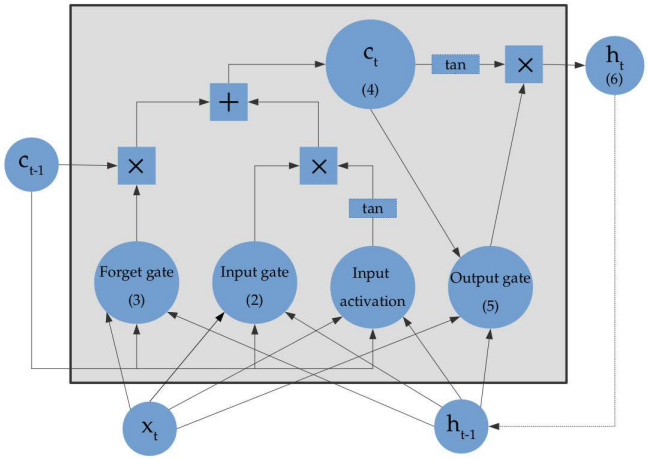

Figure 3. LSTM cell architecture. The memory of the LSTM cell is stored in c. Through time and as new inputs are fed in, the memory state is updated controlled by the forget and input gates. Both have as input: (1) the previous memory state c_{t-1}, (2) the previous output h_{t-1} and (3) the current input x_t; their activation is calculated as described in Equations (2) and (3). The cell state is modified by multiplying the old memory c_{t-1} by the forget gate, which will determine how much of the old memory to keep. Then, new memories (input activation) are added, controlled by the input gate. The output, which is then fed back to the network, is calculated as in Equation (5). Diagram inspired by [11].

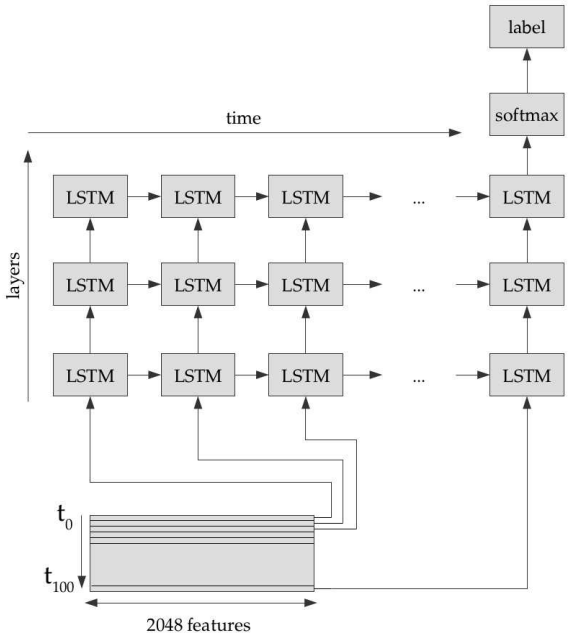

Figure 4. Architecture of our classification neural network. At the bottom is the input to the network, a sequence of 2048 features per frame for 100 frames. This is the unrolled version of the recurrent network with time going from left to right, and input processed in this direction. The input is also processed through 3 LSTMs layers, upwards. At the end, input is processed through a softmax layer to obtain the predicted label.

Deep LSTM implementation details

Video-based action recognition requires the modeling of long-term time dependencies on highly complex data (images). We have seen that LSTMs are very suitable to learn these long-term dependencies. By stacking LSTMs on top of each other, it becomes possible to learn high level structures in high dimensional data such as images. Each layer uses as input the output of the previous layer, creating a hierarchical representation of the input data, where higher layers have more abstract and complex representations of the data. In [11], the authors showed that deep LSTMs greatly improved performance in speech recognition compared to one-layer LSTMs. For these reasons we decided to use a deep LSTM network.

The detailed architecture of our network is shown in Figure 4. It has 3 stacked LSTM layers, empirically found to give best results. Each of these LSTMs layers have 90 hidden units and a softmax function is applied to the last layer to obtain the predicted classification output. In learning, the cost is calculated as the cross entropy and with L2 regularization to reduce overfitting, the L2 is scaled by a λ value of 0.003. Adam optimizer is used to perform gradient descent [14] and optimize the network. We employ exponential decay of the learning rate, with a starting learning rate of 0.001 (0.005 for HMDB) and decaying with a base of 0.96 every 100 000 steps. During training, the accuracy of prediction for the validation set is calculated every 10 steps to select the best model and the parameters for the best results stored. Parameters were found to be empirically effective and we used Tensorflow for our implementation [1].

4. Results and evaluation

4.1. Action classification

The first experiment consists in classifying the videos into the correct class, amongst the 12 actions from the THETIS dataset. Figure 5 shows the confusion matrix of this experiment. The average accuracy in prediction is of 47.22%, with an F1 score of 47.05%. Figure 5 shows that for each of the 12 actions, most videos are labeled with the correct shot type and some actions, such as backhand with two hand and backhand have an accuracy of over 60%. By looking at the results in more detail, one can realize that most errors are interpretable. For instance, the network makes mistakes in discriminating between the different types of serve or smash. Videos in the THETIS dataset do not contain the tennis ball, and this could explain why smash and serve are often confused. Another source of confusion are slices and volleys, both in backhand and fore-

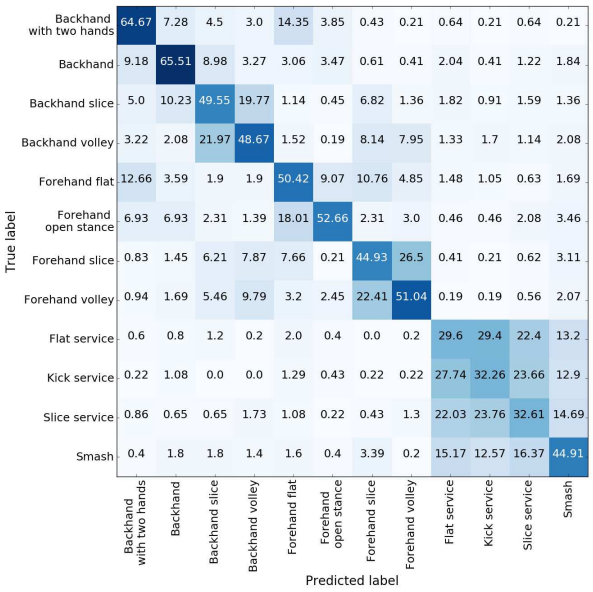

Figure 5. Confusion matrix of our model applied to the THETIS dataset.

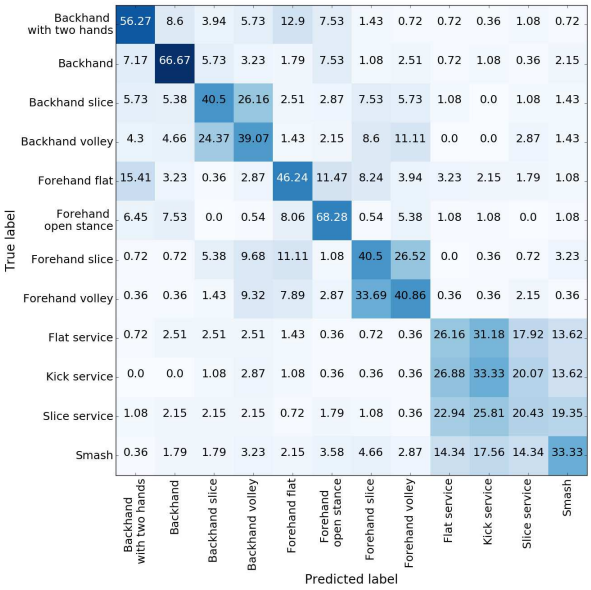

Figure 6. Confusion matrix of our model applied to the THETIS dataset, results on amateur players.

hand; these two actions are also quite similar for a human observer. Again, one main difference between volley and slice is that in the former the ball is hit before bouncing.

In [10], the authors perform similar experiments but use depth videos and 3D skeletons rather than raw image and obtain 60% and 54.4% accuracy, respectively. It is reasonable to assume that classifying raw footage brings additional challenges and we consider that our results are competitive.

4.2. Expertise detection

As described in Section 3.1.1, THETIS dataset contains shots performed by 31 amateur and 24 experienced players. To assess whether our network's classification accuracy was affected by the expertise of a player, we performed two different experiments. First, we calculated the prediction accuracy for each group of players separately, when trained on the entire dataset (with the same leave-one out cross validation procedure). Interestingly, our model's accuracy is higher for professional players (54.09%) than amateurs (41.90%). A more detailed representation of the results is shown in Figures 6 and 7, for amateur and professional players respectively. One possible explanation is that professionals have a neater technique making their shots more distinct and the biggest difference, as can be seen in the figures, is within the different types of serve.

To further investigate how the network was affected by the players' expertise, we compared the network's performance when trained using only amateur players, only professionals and a mixed set of players. In order for the results to be comparable, the number of examples used for training

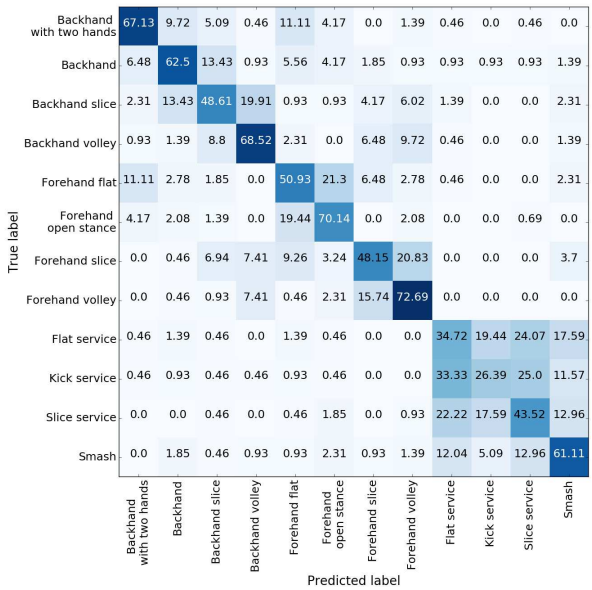

Figure 7. Confusion matrix of our model applied to the THETIS dataset, results on professional players.

should be similar. For this reason, we couldn't use previous results for the mixed set of players since they are calculated using double the amount of data compared to the data that can be used when training on only amateurs or experts. To solve the issue, we re-run the experiments selecting the test and validation sets as before but randomly selecting only 25 examples for training. Taking into account that we used videos from 5 players for validation and 1 for testing in all experiments, the final training sets contain videos from 25

Players in training set	Players in test set	Accuracy
mixed	mixed	39.65% (47.22%)
mixed	amateur	37.02% (41.90%)
mixed	professional	43.06% (54.09%)
amateur	amateur	37.70%
professional	professional	45.00%

Table 1. Accuracy of classification when training with amateurs, professionals or a mixed population.

players for the amateurs and mixed groups and 18 for professionals.

Results are summarized in Table 1. As expected, the first observation is that when training with a mixed group of players but using only 25 players for training, the classification accuracy for all groups of players is lower than when using the entire dataset (training with 49 players). When training with a reduced number of examples the accuracy is 39.65%, 37.02% and 43.06% for mixed players, amateurs and professionals respectively versus 47.22% , 41.90% and 54.09%. This results are consistent with previous experiments since classification of videos of professional players achieves the highest accuracy. Interestingly, this is further increased when only professionals are used for training – accuracy increases from 43.06% to 45.00%. Classification of videos from amateur players also benefits from training with only amateur players, increasing the classification accuracy from 37.02% to 37.70%. These results suggest that the network could be learning different features depending on the level of expertise of the players. It may be that features that help to best discriminate between different actions in amateur players are different than those to identify different shots played by professionals. Another interesting observation from these experiments is that the quality of the training data is important, but in our particular example, size of the training set is even more important as best results are achieved when using the entire dataset.

4.3. From fine-grained actions to stroke types

Observing the results from Figure 5, we noticed that our learning algorithm was not only able to identify the 12 fine-grained actions from the THETIS dataset but, when doing mistakes, these were generally between actions that can be grouped into the same type of tennis stroke. For instance, even though a slice service is confused with a flat or kick service, the network is still recognizing that it is a serve. For this reason, we looked at the accuracy of the prediction when grouping classes into more general stroke types, as shown in Table 2. We still consider these actions to be fine-grained as they are within the domain of tennis actions and fine-grained when compared to general action recognition (e.g. detecting which sport is being played).

For this, we used the results from the first experiments of action classification and grouped the labels into the 4 main actions in Table 2: backhand, forehand, service and smash. The result from this is shown in Figure 8. In this setting the action detection accuracy reaches 76.92% with and F1 score of 76.90%. As can be seen from the Figure 5, the mean accuracy is brought down by the smash detection. In fact, most errors come from a confusion between smash and serve. These are quite similar in terms of body movement and the main differences are the state of the game (serve is played at the beginning of a point), player position in the court and ball trajectory before hitting the ball. THETIS videos do not contain the tennis ball, which could help in discriminating between the two actions. Also, in real-world applications we might expect to know the players position or state of the game, helping to further discriminate between the two actions.

Having obtained these results, we wondered how predictions would compare if we trained on the main strokes directly. For this, we grouped smash and serves into the same category as they are very similar in terms of body movement and it helps balancing the classes. Table 3 shows the results by category, and we can see that training for the specific task, which is detecting one of the three main strokes in this case, produces better detection results than training for finer-grained actions and then regrouping the actions into their more general categories. The classification accuracy improves from 84.10% to 88.16% for players of mixed abilities, from 81.23% to 84.33% for amateurs and from 87.82% to 89.42% for professionals, when trained using the entire dataset.

Two main observations can be derived from that. First, the network performs best when trained for the specific task in which it is evaluated and second the features relevant to discriminate between stroke type and between finer-grained actions might be different.

4.4. Applicability to general action recognition tasks

To investigate the ability of generalization of our network, we evaluated it on the HMDB dataset. We achieved an accuracy of 43.19% and F1 score of 42.48%. In Table 4, our performance in HMDB is compared to existing models that, as ourselves, use exclusively RGB data. For instance, we do not include: the two-stream ConvNet of [22] (59.4%) which uses optical flow information, models in [28] that use Fisher Vectors (53.3%) and a combination of HOG, HOF and MBH (60.1%). The results presented here show that our network has the potential to be applied to other tasks, further supporting that it could be applicable to other sports.

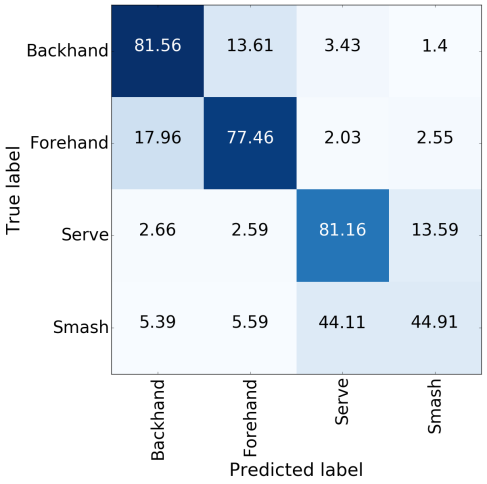

Figure 8. Confusion matrix of our model on the THETIS dataset, grouped classes.

Action group	Actions
Backhand	backhand (with two hands) backhand backhand (slice) backhand (volley)
Forehand	forehand (flat) forehand (open stance) forehand (slice) forehand (volley)
Service	service (flat) service (kick) service (slice)
Smash	smash

Table 2. Fine-grained actions grouped into stroke types.

5. Conclusions and future work

In this work we have presented a 3-layered LSTM network able to classify fine-grained tennis actions and which uncovered a number of interesting points.

First, our network achieved good results by using features extracted through the application of the Inception neural network, trained on an independent dataset and without the need of fine-tuning. This suggests that it is a robust data representation with the potential to be transferable to multiple tasks and domains. Second, the networks' classification errors were interpretable, suggesting it was learning semantically meaningful information. Endorsing this idea, the network performed better when trained with only amateur or only professional players rather than a mixed population. It is possible that it learned different features when looking at amateurs and professional players and it would be interesting to investigate this further. Third, the network performed better for professional players than amateurs, when trained on a mixed population. A possible cause is that professionals have a better techniques that makes their strokes more distinct. In the future, we would like to consider whether this can be exploited to assess a player's expertise. Fourth, the same network architecture was able to detect the three main strokes with an 88.16% accuracy, and it performed better than when an indirect inference was made from finer-grained actions. This further supports the robustness and transferability of the Inception features. Finally, we also showed how the proposed approach can be applied to general action recognition tasks, by evaluating it with the HMDB dataset.

With this work we wish to motivate the exploration of deep neural networks in the sports domain and the use and production of benchmark datasets in sports action recognition. In the future, it would be interesting to investigate how to incorporate spatio-temporal data to our network to improve action detection and how to combine action recognition with statistical data in order to push forward the field of tennis analytics.

Players tested	Trained actions	Accuracy
all	all	84.10%
all	3	88.16%
amateur	all	81.23%
amateur	3	84.33%
professional	all	87.82%
professional	3	89.42%

Table 3. Accuracy of detection when training with fine-grained actions and then re-grouping vs training directly with the three main strokes classes.

Model	HMDB-51 accuracy
Spatial stream ConvNet [22]	40.5%
Soft attention model [21]	41.3%
Our model	43.2%
Composite LSTM [24]	44.1%

Table 4. HMDB-51 classification accuracy by state-of-the-art models from RGB data exclusively.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential deep learning for human action recognition. In *International Workshop on Human Behavior Understanding*, pages 29–39. Springer, 2011.

[3] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, volume 2, pages 1395–1402. IEEE, 2005.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CPVR)*, volume 1, pages 886–893. IEEE, 2005.

[5] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *European conference on computer vision*, pages 428–441. Springer, 2006.

[6] T. De Campos, M. Barnard, K. Mikolajczyk, J. Kittler, F. Yan, W. Christmas, and D. Windridge. An evaluation of bags-of-words and spatio-temporal shapes for action recognition. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 344–351. IEEE, 2011.

[7] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72. IEEE, 2005.

[8] N. FarajiDavar, T. De Campos, J. Kittler, and F. Yan. Transductive transfer learning for action recognition in tennis games. In *ICCV*, pages 1548–1553. IEEE, 2011.

[9] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE transactions on pattern analysis and machine intelligence*, 29(12):2247–2253, 2007.

[10] S. Gourgari, G. Goudelis, K. Karpouzis, and S. Kollias. THETIS: Three dimensional tennis shots a human action dataset. In *CVPR Workshops*, pages 676–681. IEEE, 2013.

[11] A. Graves, A.-R. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649. IEEE, 2013.

[12] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[13] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732. IEEE, 2014.

[14] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[16] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *ICCV*, pages 2556–2563. IEEE, 2011.

[17] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, pages 1–8. IEEE, 2008.

[18] K. K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, 24(5):971–981, 2013.

[19] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, pages 1–8. IEEE, 2008.

[20] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *ICPR*, volume 3, pages 32–36. IEEE, 2004.

[21] S. Sharma, R. Kiros, and R. Salakhutdinov. Action recognition using visual attention. *arXiv preprint arXiv:1511.04119*, 2015.

[22] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014.

[23] K. Soomro and A. R. Zamir. Action recognition in realistic sports videos. In *Computer Vision in Sports*, pages 181–208. Springer, 2014.

[24] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using LSTMs. In *ICML*, pages 843–852, 2015.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9. IEEE, 2015.

[26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826. IEEE, 2016.

[27] J. Vainstein, J. F. Manera, P. Negri, C. Delrieux, and A. Maguitman. Modeling video activity with dynamic phrases and its application to action recognition in tennis videos. In *Iberoamerican Congress on Pattern Recognition*, pages 909–916. Springer, 2014.

[28] H. Wang, D. Oneata, J. Verbeek, and C. Schmid. A robust and efficient video representation for action recognition. *IJCV*, 119(3):219–238, 2016.

[29] W. Zaremba, I. Sutskever, and O. Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

[30] G. Zhu, C. Xu, W. Gao, and Q. Huang. Action recognition in broadcast tennis video using optical flow and support vector machine. In *European Conference on Computer Vision*, pages 89–98. Springer, 2006.

[31] G. Zhu, C. Xu, Q. Huang, W. Gao, and L. Xing. Player action recognition in broadcast tennis video with applications to semantic analysis of sports game. In *Proceedings of the 14th ACM international conference on Multimedia*, pages 431–440. ACM, 2006.