Convolutional Neural Networks based ball detection in tennis games

Vito Renò Nicola Mosca Roberto Marani Massimiliano Nitti Tiziana D'Orazio
Ettore Stella
National Research Council of Italy - Institute of Intelligent Systems for Automation
Via Amendola 122 D/O, 70126 Bari, Italy
reno@ba.issia.cnr.it

Abstract

In recent years sport video research has gained a steady interest among the scientific community. The large amount of video data available from broadcast transmissions and from dedicated camera setups, and the need of extracting meaningful information from data, pose significant research challenges. Hence, computer vision and machine learning are essential for enabling automated or semi-automated processing of big data in sports. Although sports are diverse enough to present unique challenges on their own, most of them share the need to identify active entities such as ball or players. In this paper, an innovative deep learning approach to the identification of the ball in tennis context is presented. The work exploits the potential of a convolutional neural network classifier to decide whether a ball is being observed in a single frame, overcoming the typical issues that can occur dealing with classical approaches on long video sequences (e.g. illumination changes and flickering issues). Experiments on real data confirm the validity of the proposed approach that achieves $98.77\,\%$ *accuracy and suggest its implementation and integration at a larger scale in more complex vision systems.*

1. Introduction

Sport matches have always attracted the attention of a broad audience at various levels of involvement, from players and coaches to the general public. The visibility given by broadcasted events in the last decades has further magnified this appeal. In fact, matches results and sport management decisions are usually subjects of many debates and discussions, meaning that both enthusiasts and insiders are interested in many applications like tactics analysis, highlight identification or, more generally, statistical analysis.
The advance in processing power and the growing importance of sport activities in various businesses has also attracted the attention of the sport video research community, given also the particular challenging conditions that the context provides, opening the way to new perspectives and paradigms referred to sports analysis. It is undoubted that technological progress is providing a huge amount of video data to the scientific community that needs to be processed. For this reason, computer vision plays a fundamental role for effectively exploiting big data and consequently enabling the automated or semi-automated processing of such videos. Significant information can be inferred knowing the positions of the active entities during a match or a training (i.e. balls or players), but each sport introduces different challenges due to its rules and settings.
For example, in popular team sports such as soccer or basketball, many moving players need to be identified in usually chaotic environments, while the ball is moving on the scene. Conversely, in tennis, the individual sport nature means that just a few players need to be considered during the game that evolves in an uncluttered environment in which a relatively small but fast ball is moving, providing some interesting challenges for accurately tracking the ball. Various methodologies have been applied in the last decades in game analysis, with results that continuously get better, also due to methodological or technological advancements. A pivotal research field that owes its progress to both is machine learning, with the development of deep learning and convolutional neural networks (CNNs) [5] that has been proven to be useful in many computer vision applications in the last years.
The work described in this paper is devoted to the automated identification of the ball in the tennis context with a deep learning based approach.

1.1. Related work

Ball detection and tracking are challenging problems that have attracted much interest, with a recent survey that can be found in [3]. Given their complexity, it is reasonable that some researchers proposed and built their solutions around custom-setups, seeking maximum performance or control over the data acquisition and subsequent processing. For example, by establishing the fixed location of some came-

ras, it is possible to enable ball detection techniques based on background modeling or frame differencing. This is the choice made by Pingali et al. [7] while developing a custom multi-camera installation where ball segmentation starts with frame differencing and detection relies on intensity range cues, given the monochrome nature of the high-speed cameras deployed. A custom multi-camera setup is also proposed by Conaire et al. [1] where fixed cameras enable the usage of methods relying on background modeling and blob detection. Heuristics based on visual cues and ball motion are also employed. Another multi-camera system is presented in [10] where four cameras (two for each side) are used in pair to reconstruct moving entities in the 3D space and then perform domain knowledge-based reasoning to identify balls and reconstruct their trajectories by splitting and re-merging tracklets.

Other researchers have instead focused their work on data coming from a single camera, or even more specifically from a broadcast video, that enables a broader application of the techniques, but can not be directly controlled. This kind of approach limits the amount and accuracy of the information that can be extracted: spatial and temporal resolutions can be low, cameras can move and zoom frequently, images can be over exposed or too dark, compression artifacts can be present, etc. . .. To overcome these issues, some works rely on user feedback, such as in [6] and therefore are likely best suited for manual annotation of previously acquired actions. Without user assistance, usually algorithms are tuned for a particular sport and a particular setting, due to the different challenges that most contexts provide, trying to exploit every kind of information available a priori. In tennis context, Yu et al. [14] propose a system for applications where it is needed to insert 3D virtual content for supplementing the video feed. In the same work, they devised a way to perform ball detection and tracking that is improved by exploiting the same camera auto-calibration methods used for virtual content insertion. However, as they noted, only partial 3D ball position data can be extracted from the broadcast video. In snooker games, Rea et al. [9] exploited appearance features for detection, like ball most dominant color, while deploying a particle filter for tracking tasks. In broadcast soccer videos, Yu et al. [15] use anti-models based on high-level semantic representation and domain knowledge for filtering out moving objects that are likely non-balls, so that several ball candidates can be identified on a frame by frame basis and then validated through trajectory-based reasoning. Other approaches require the knowledge of whole video sequences a priori and therefore cannot be applied in real-time contexts. For example, Yan et al. [13] perform ball detection and tracking constructing a weighted graph and optimizing an all-pairs shortest path (APSP) problem.

In this paper the authors investigate the usage of a convolutional neural network for ball detection that can work on a frame by frame basis, without requiring any image preprocessing step, like background subtraction or frame differencing.

Approaches based on neural networks have already been used for detection tasks, as in [2], where a method to detect balls in soccer videos based on Hough transform and neural networks is presented. However, one of the advantages provided by deep learning is related to its ability to automatically extract relevant features from data. In this context, the flexibility of convolutional neural networks on image recognition tasks and the different datasets [11, 4] created by the research community might suggest the availability of a dataset or CNN architecture already experimented in the tennis context. This does not look to be the case. Although the available datasets on which popular classifiers have been pre-trained often contain, among the classes, sport game equipments such as tennis balls, those datasets are general purpose by nature, since they have been employed for discriminating among a high number of objects. Up to the authors knowledge, there have not been studies focused on the use of deep learning algorithms in a real world tennis context, in which acquired images can suffer from problems that usually are not present, nor considered, in sample images obtained by digital cameras used to take static pictures, such as illumination changes and flickering issues.

This work exploits the potentiality of a convolutional neural network classifier trained on a dataset made of Ball and No Ball examples, in order to classify image portions that are likely to be a ball or not. This approach enables to perform a single frame analysis to extract ball positions without the application of specific background models, thus making the methodology also robust to illumination changes and flickering issues.

The rest of the paper is organized as follows. Section 2 presents the adopted methodology with a focus on the architecture of the deep learning classifier used in this work. In section 3 the experiments are described and the obtained results are discussed. Section 4 concludes the paper with a summary of the work and a perspective for the future.

2. Methodology

2.1. CNN Architecture

The methodology proposed in this paper makes use of a deep learning classifier to decide whether an image patch can be labeled as Ball or No Ball, namely a convolutional neural network. Figure 1 represents the CNN diagram in terms of subsequent layers, starting from the Input image layer and ending with the Class output layer [5]. The network has been designed to work with $r \times s$ RGB image patches as image inputs. CNNs have the capability of preserving the spatial relationships on the processed images by

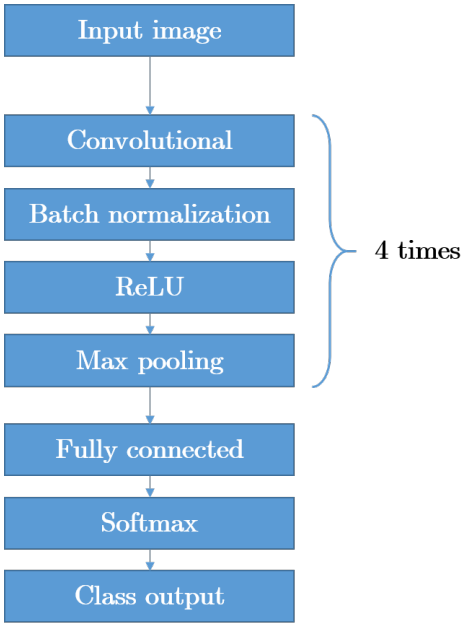

Figure 1. Diagram of the proposed deep learning network in which each box represents a different layer. The input image is a RGB $r \times s$ patch. Visual data is then processed by four deep learning blocks of layers. Finally, a classical neural network followed by a softmax and a class output gives the classification result.

finding a huge number of small filters, mimicking the human vision system, through linear and non linear operations. Linearity is represented by a Convolutional layer devoted to the identification of a bank of filters, i.e. the feature maps, defined as 64 kernels of dimension 5×5 followed by a Batch normalization. Non linearity is then introduced with a Rectified Linear Unit function that supports the classifier to work with non linearly separable classes. A 2×2 non overlapping Max pooling layer, also called downsampling layer, progressively reduces the dimensionality of the data.

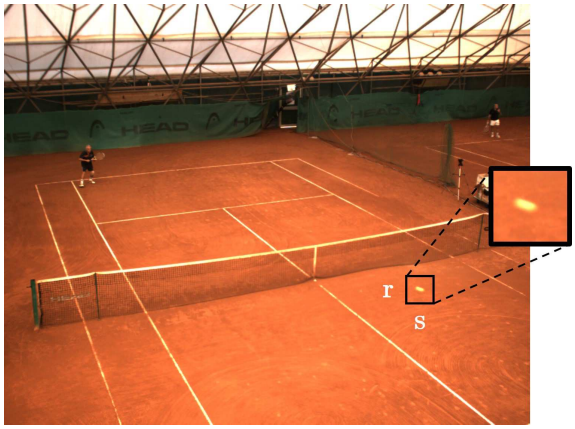

Figure 2. Example of frame with a zoomed detail of a $r \times s$ patch that contains a moving ball.

This means that the deep learning pipeline can be repeated only a limited number of times (in our implementation 4, as shown in figure 1). The parameters of the filters as well as the total number of layers have been empirically determined to find a trade-off between computational complexity and accuracy. The CNN architecture is completed by a 2 outputs Fully connected layer, followed by Softmax and finally the Classification output layer. A final remark should be made about the rectangular patch size to be fed to the CNN, that should be properly tuned in order to take into account all the possible ways the ball appears on the scene: big or small due to the perspective, sharp or blurred due to its speed. An example is given in figure 2 where a moving ball is represented.

2.2. CNN Training and Outputs

The training of the CNN has been performed using the Stochastic Gradient Descent with Momentum algorithm [8], with an initial learning rate of 0.05 and a progressive reduction of this value given by a drop factor of 0.5. The dataset is first randomly split in Training, Validation and Test set. Then, the learning algorithm starts updating the CNN parameters relying on all the Training images during an epoch. The accuracy of the CNN is computed at the end of each epoch on the Validation set, never used during the learning phase, in order to avoid training bias. The drop factor progressively reduces the learning rate at each epoch to lower convergence time and prevent data overfitting.

It is worth pointing out the different kind of information that can be extracted from the CNN in the final layers. First, the classification of the whole patch as Ball or No Ball after the Softmax evaluation is the classical CNN output. However, this feature always labels a patch with the most probable class label without taking into account the fact that the same pixel belongs to many different patches. For this reason, the pixel-wise average probability value for the class Ball has been computed exploiting the output of the Fully Connected Layer. In more details, let $I \in \mathbb{N}^{h \times w \times 3}$ be a RGB image to be analyzed by the CNN, P a generic $r \times s \times 3$ patch, κ the number of patches belonging to each pixel of coordinates (u, v) and $B : \mathbb{N}^{r \times s \times 3} \rightarrow [0, \ldots, 1]$ the function that returns the Ball probability value for each RGB patch. The Probability Image can then be defined as follows:

$$PI(u,v) = \frac{\sum_{i=1}^{\kappa} B(P_i)}{\kappa} \qquad u = 1, \ldots, w \; v = 1, \ldots, h \tag{1}$$

giving a pixel-based evaluation of Ball probability.

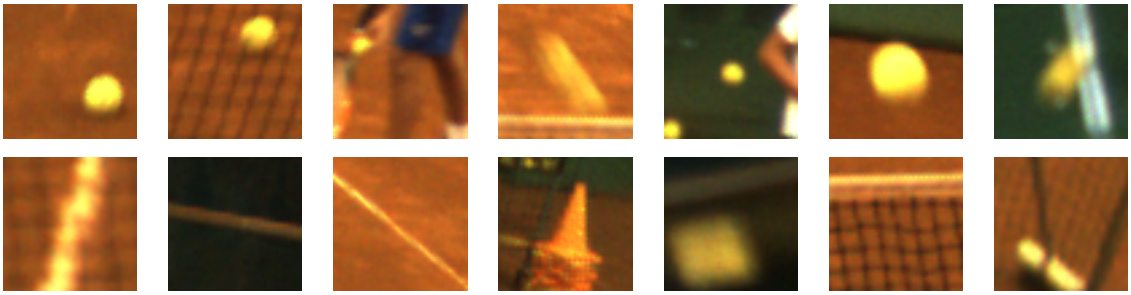

Figure 3. Examples of Ball images (first row) and No Ball images (second row) extracted from the dataset considered in this paper. An image patch has been labeled as Ball without imposing constraints about the position and the size of the ball and including blurred moving balls. The other class is made of examples that are likely to be confused with a ball, such as the court lines (that in image patches can be easily misclassified as a fast moving ball), as well as other sport equipments usually present during training sessions.

3. Experiments and results

3.1. Dataset description

A total number $N = 116385$ of $r = 50 \times s = 50$ RGB images have been collected starting from RAW videos acquired on a real setup by AVT Prosilica GT 1920C cameras. The images have been acquired both during a tennis training session and a friendly match performed on a clay court, and then manually labeled as Ball or No Ball. Among the N images, 27600 patches have been labeled as Ball and 88785 are considered as No Ball. Examples of images belonging to the two classes are shown in figure 3, where the first row reports Ball samples and the second row shows No Ball patches. A first observation should be made looking to some of the Ball sample images selected and shown on the first row: the ball is visible under different conditions, such as alone, behind the net, in the hand of a player or blurred as a consequence of a fast shot. Moreover, there is no restriction about the ball position relative to the image patch, meaning that its presence can be observed on the corners of the patch as well as on its center. This is an essential feature for the classifier to be more robust to real scenarios, especially when it needs to be used on unknown images to predict the position of a ball. In fact, in such situations, also due to performance considerations, the whole image is generally divided in partially overlapping patches to be fed to the CNN, thus reducing the probability of capturing a ball in the middle of the patch. Hence, no *a priori* knowledge about the ball position and or size can be used. Looking at the No Ball samples row it is immediate to observe various situations, such as an over-exposed court line that might be misclassified as a Ball, blurred backgrounds of the court, cones and other equipments often present during a tennis training session. It is worth noting that Ball and No Ball samples are unbalanced in terms of cardinality, due to the fact that, for each image, there are naturally less examples of Ball with respect to the other class.

3.2. Quantitative and qualitative analysis

All the experiments have been performed in Matlab r2017 using the Neural Network toolbox to design, build, train, validate and test the CNN described in the previous section. As stated in the previous section, the whole dataset has been randomly partitioned according to the following criteria: 65% for the Training set, 15% for the Validation set and finally 20% for the Test set.

Table 1 shows the quantitative results of the CNN performance computed on the Test set that has never been fed to the classifier during the training phase. The metrics used to evaluate the results are computed in terms of True Positives, False Positives, True Negatives and False Negatives, namely Precision (P), Recall (R), True Negative Rate (TNR), Accuracy (A) and Balanced Accuracy (BA) [12]. The first result to analyze is that, for both classes, the CNN achieves a score greater than 96% on test images, in particular the overall accuracy is 98.77%. Looking at the results for Ball classification, particularly useful on a realistic use case of a vision system aimed to locate a ball on a scene, it is interesting to discuss the percentages of Precision (98.77%) and Recall (96.01%) that are strictly related to the performance of the CNN in terms of false positives and false negatives. These two metrics are useful to quantify how many selected

Table 1. Quantitative results of the CNN computed on the test set in terms of Precision, Recall, True Negative Rate, Accuracy and Balanced Accuracy computed for both Ball and No Ball classes.

Class	Ball	No Ball
P	0.9877	0.98772
R	0.96014	0.99628
TNR	0.99628	0.96014
A	0.98771	0.98771
BA	0.97821	0.97821

elements are relevant (P) and how many relevant elements have been selected (R), namely the No Ball patches labeled as Ball and vice versa.

Figure 4 shows some examples of this phenomenon. On the left side Ball false positives are shown, i.e. those images labeled as No Ball but classified as Ball during the test, mainly due to the fact that the lines of the court can be confused with the motion blur of a fast ball, or because a stopped ball left outside the court has been labeled as No Ball in the dataset, as shown on the last row of figure 4. The remaining images represent Ball false negatives, i.e. patches manually labeled as Ball but classified as No Ball by the CNN. The misclassification causes are largely ascribable to the presence of a player or a racquet on the patch, since several instances of racquet appearance were used as negative examples during the training phase. In other cases, a blurred image of a moving ball showing a striking similarity with edge lines on the clay court can result in false negative classification.

Anyway, the overall score of the CNN suggests that the amount of false positives and negatives can be reasonably neglected in a real setup, where additional information, deliberately not used in these experiments, are available. For example, the domain knowledge about the scene can be considered to filter the output of the CNN and infer robust information. To further investigate the effectiveness of the methodology, also qualitative results of the proposed approach can be considered, as shown in figure 5 where the CNN is tested with some never before seen images that were not used to create the dataset. In this experiment, the input image is split in 50×50 overlapping patches that are individually processed by the CNN. The cyan rectangles in the example frames (figure 5 (a) and (d)) highlight the patches labeled as Ball and show a certain number of false positives in correspondence of the court lines. The first thing to observe is that a higher number of patches is labeled as Ball in the neighborhood of the true positive patches if compared to the number of Ball patches in false positive zones. This behavior suggests a deeper investigation on the probability values produced by the fully connected layer of the CNN that have been exploited to visualize the Probability Image defined in equation 1, as shown in figure 5 (b) and (e), where the $[0, \ldots, 1]$ range is mapped on a blue to yellow colormap. The evidence is that both the court lines on (b) and the cluttered portion on the upper right part of (e) are not depicted with high probability values. For this reason, the two images have been filtered discarding the pixels whose values are under a threshold τ defined as the 99^{th} percentile of all the Ball probability values of the image. Finally, the filtered probability image is shown in (c) and (f), proving that a certain number of false positives can be effectively filtered out with negligible impact on previously legitimate patches labeled as balls.

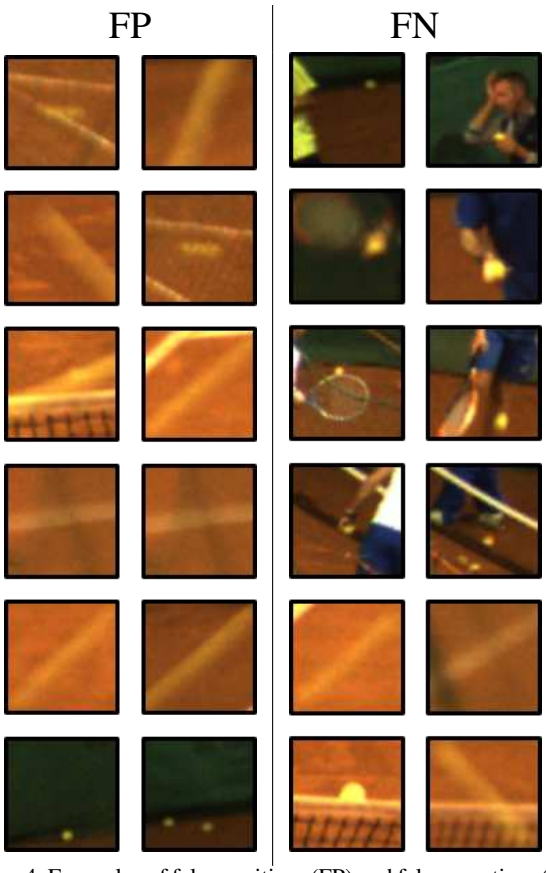

Figure 4. Examples of false positives (FP) and false negatives (FN) samples, with respect to the Ball class, extracted from the randomly defined test set.

4. Conclusion and future works

In this paper, an innovative approach to ball detection in tennis assisted by a convolutional neural network has been presented. This is one of the first application of deep learning techniques devoted to the sport analysis and represents a step towards the inclusion of such methodologies in more complex systems for game analysis. Experiments on real data have demonstrated that the classifier achieves very high accuracy values, suggesting the feasibility of the approach. Future directions of this research will surely regard the integration of the CNN based ball detection in complex vision systems, to be effectively used in conjunction with classical approaches where domain knowledge or cinematic considerations can be used to further enhance robustness introducing, as an example, the distinction between active and inactive balls during a game. Finally, the extension of the CNN application range will be investigated too, reinforcing the classifier with more data and enabling its use on different game settings.

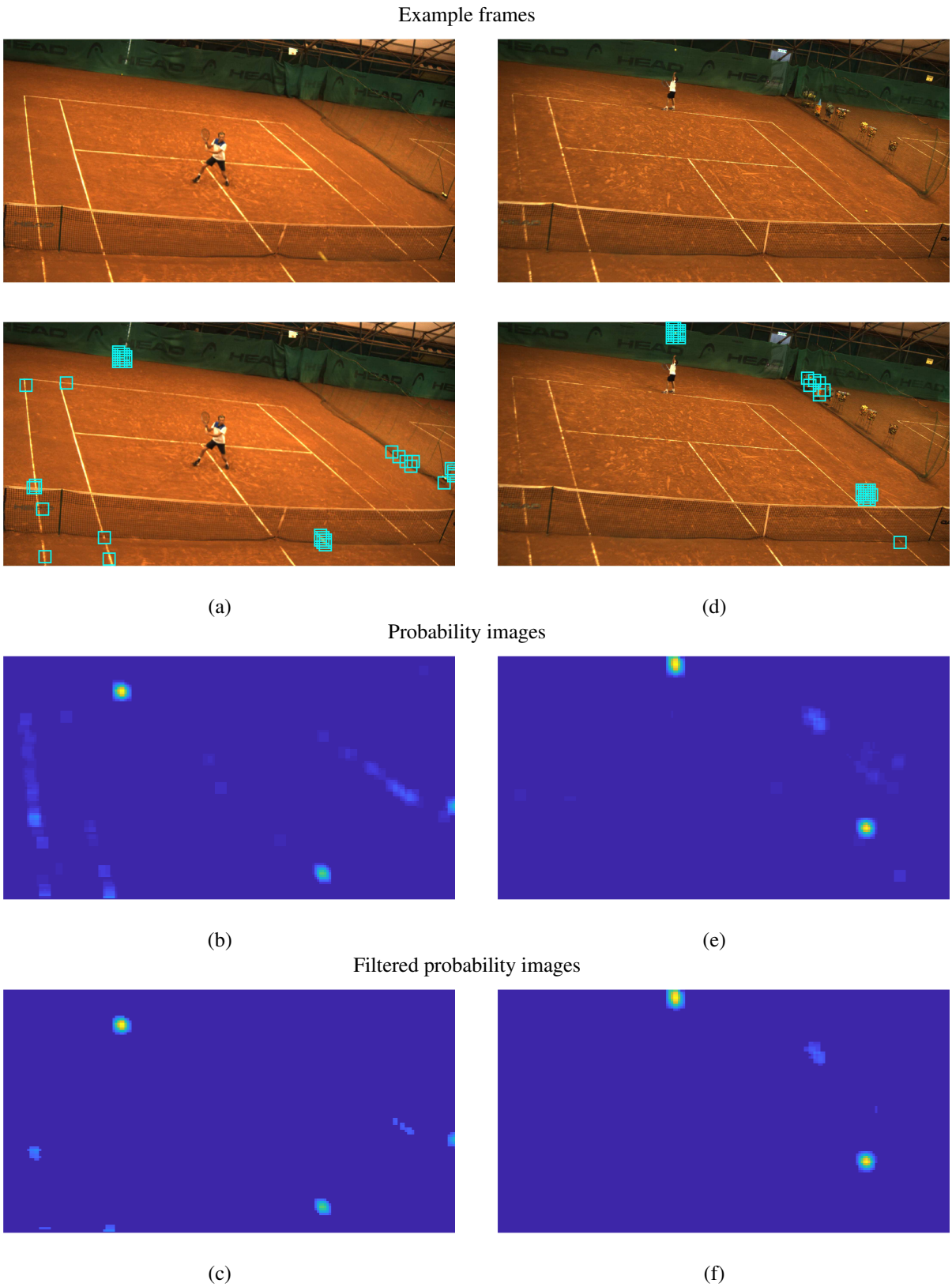

Figure 5. Qualitative results of the proposed approach for two example frames. The full frame input images are first divided in overlapping patches and then each patch is processed by the CNN. In the first column, cyan rectangles represent image patches classified as balls. The second column contains the Ball probability image as defined in equation 1, while the last one shows only the probability values greater than the 99th percentile computed on the whole probability image.

References

[1] C. O. Conaire, P. Kelly, D. Connaghan, and N. E. O'Connor. Tennissense: A platform for extracting semantic information from multi-camera tennis data. In *2009 16th International Conference on Digital Signal Processing*, pages 1–6, July 2009. 2

[2] T. D'Orazio, C. Guaragnella, M. Leo, and A. Distante. A new algorithm for ball recognition using circle hough transform and neural classifier. *Pattern Recognition*, 37(3):393 – 408, 2004. 2

[3] P. R. Kamble, A. G. Keskar, and K. M. Bhurchandi. Ball tracking in sports: a survey. *Artificial Intelligence Review*, Oct 2017. 1

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. 2

[5] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015. 1, 2

[6] Y. Liu, D. Liang, Q. Huang, and W. Gao. Extracting 3d information from broadcast soccer video. *Image and Vision Computing*, 24(10):1146 – 1162, 2006. 2

[7] G. Pingali, A. Opalach, and Y. Jean. Ball tracking and virtual replays for innovative tennis broadcasts. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 4, pages 152–156 vol.4, 2000. 2

[8] N. Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999. 3

[9] N. Rea, R. Dahyot, and A. Kokaram. Semantic event detection in sports through motion understanding. In P. Enser, Y. Kompatsiaris, N. E. O'Connor, A. F. Smeaton, and A. W. M. Smeulders, editors, *Image and Video Retrieval*, pages 88–97, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. 2

[10] V. Renò, N. Mosca, M. Nitti, T. D'Orazio, C. Guaragnella, D. Campagnoli, A. Prati, and E. Stella. A technology platform for automatic high-level tennis game analysis. *Computer Vision and Image Understanding*, 159:164 – 175, 2017. Computer Vision in Sports. 2

[11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 2

[12] M. Sokolova and G. Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009. 4

[13] F. Yan, W. Christmas, and J. Kittler. Layered data association using graph-theoretic formulation with application to tennis ball tracking in monocular sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1814–1830, Oct 2008. 2

[14] X. Yu, N. Jiang, L.-F. Cheong, H. W. Leong, and X. Yan. Automatic camera calibration of broadcast tennis video with applications to 3d virtual content insertion and ball detection and tracking. *Computer Vision and Image Understanding*, 113(5):643 – 652, 2009. Computer Vision Based Analysis in Sport Environments. 2

[15] X. Yu, H. W. Leong, C. Xu, and Q. Tian. Trajectory-based ball detection and tracking in broadcast soccer video. *IEEE Transactions on Multimedia*, 8(6):1164–1178, Dec 2006. 2