Real-time multi-sport action tracking with convolutional neural networks

Axel Baldanza, Jean François Aujol, Yann Traonmilin, François Alary

▶ **To cite this version:**

Axel Baldanza, Jean François Aujol, Yann Traonmilin, François Alary. Real-time multi-sport action tracking with convolutional neural networks. 2022. hal-03876332v1

HAL Id: hal-03876332

https://hal.science/hal-03876332v1

Preprint submitted on 28 Nov 2022 (v1), last revised 23 Feb 2023 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

REAL-TIME MULTI-SPORT ACTION TRACKING WITH CONVOLUTIONAL NEURAL NETWORKS

Axel Baldanza[1,2], *Jean-François Aujol*[1], *Yann Traonmilin*[1] *and François Alary*[2]
[1]Univ. Bordeaux, Bordeaux INP, CNRS, IMB, UMR 5251, F-33400 Talence, France.
[2]Rematch

ABSTRACT

State-of-the-art localization and action tracking methods have shown bad performances on amateur sports videos due to the variability of acquisition conditions and occlusion problems. Moreover these methods need to be modified in order to be applied to different sports. In this paper, we present a real-time computable method that allows video action tracking in amateur sports. This method uses a convolutional neural network to analyze the players' movements instead of basing the tracking on object detection. This feature allows it to be transposed out-of-the-box to different sports.

Index Terms— Action, tracking, sport, real-time, CNN

1. INTRODUCTION

Computer vision algorithms in sports is a subject that has been growing in importance in recent years. These works aim to improve the performance of athletes or enhance the experience of spectators. In this paper, we focus on tracking methods in sports, which is a very diverse subject. It can be about multi-object tracking to compute detailed statistics of each player, complex analysis of the match, or global tracking of the region of interest to design more effective cameras. In this paper, we aim at designing a method for tracking the region of interest. We focus here on the context of amateur sports. This context has the particularity to defeat many state-of-the-art methods because of its variability. The variability of point of view and camera lenses makes object detection difficult.

One way to track the action is to follow the ball. Methods in the literature that aim to detect and track the ball such as [1, 2, 3] are efficient for the professional context. The images studied in this type of method make the detection simpler. The camera is located much higher with respect to the field and avoids ball occlusions. Many of these methods are developed for soccer only, where the contrast between the color of the ball and the grass is sharp and the ball is often dissociated from the players, which makes it more easily detectable. These algorithms show difficulties on sports like basketball and handball where the ball is mainly held by the players. [4] looks at the performance of state-of-the-art object detection methods to detect the ball on amateur videos and highlights the difficulties of this algorithm to localize the ball in this context. As explained in [5] some of these methods show unusable results on our test database and tracking the action by detecting the ball proved to be impossible. State-of-the-art real-time object detection methods such as YOLO (You Only Look Once) architecture [6, 7] and Faster R-CNN (Region-Based Convolutional Neural Network) [8, 9] pretrained on COCO dataset [10] failed to detect balls in more than 95% of the frames. A paper such as [11] presents a multi-sport ball tracking method, including sports where the ball is hand-held. However, the pipeline of this type of method is too computationally expensive to be used in real-time on embedded systems, particularly because of the precise and complete tracking of all players it requires. In the same way, some papers on tracking in sports as [12] present methods that are based on the combination of object tracking and remapping in a field plane. Remapping on a field plane is an efficient method to allow complex tracking in sports. However, these methods are also too computationally expensive in our conditions. Another way to locate the action is to localize and track the region of interest in the videos. The state-of-the-art methods for spatio-temporal action localization focus on images containing only a small number of salient objects. The videos considered in [13, 14, 15, 16] show that only a few objects are present in the images. This feature makes the localization of the action in the image easier. Our images have a large number of people and this type of algorithm is hardly efficient in our conditions. In [5], it is shown that action tracking in basketball can be done by predicting the action global motion instead of classic spatial localization. Assuming that player displacements could induce camera motion, computing optical flow in sports videos can give robust information on the position of the region of interest. This way of tracking the action allows us to avoid non-robust detection in a degraded recording context. This method is also designed to reduce the computational complexity compared to the methods mentioned above. This feature enables it to be used in embedded solutions in real-time. The method presented in [5] has shown very encouraging results. However, to be efficient, this method requires a separation of the model into different situations. Its success is thus very dependent on the classification of the action into different situations. This model has shown limitations in its ability to track long sequences and to fit some sports other than basketball where camera motions are linear and the classification of situations was complex.

1.1. Our method

The method presented in this paper is based on previous work [5]. We predict the camera movements using a convolutional neural network. The use of the neural network enables us to avoid the classification of situations used in the piecewise linear model and to integrate it directly into the prediction model. Moreover, the architecture of the network presented in this paper uses a larger number of parameters, which allows a more consistent analysis of the images than with a simple matrix multiplication. The objective of this improvement is to make the model more accurate and more adapted to the problem while remaining fast to compute. This should allow the model to be extendable to long sequence videos and other sport context. Our new method is tested on basketball and handball videos. The model takes as input a sport video, computes the segmented optical flow of the foreground at each frame, and passes as input to the convolutional network 3 segmentations with k frames gap to add a temporal analysis. The algorithm returns the camera displacement necessary to track the zone of interest in the following frame. This displacement is normalized in a percentage of the field.

1.2. Contribution

The main contribution of this paper is that we have defined an accurate tracking method that is robust to most of the variations induced by the amateur conditions. The main advantage of this method is its adaptability to variations. It can be transposable out of the box to several different sports, just by re-training the network. This is one of the limitations of state-of-the-art tracking methods in sports based on object detection compared to our method. The way this model is designed and its architecture allows it to compute the tracking in real time on embedded solutions, whereas methods based on object detection and tracking are often slow and computationally expensive. The prediction of the global camera motion also allows the automatic labellization of the database while the annotation of the databases for object detection is time consuming.

2. GLOBAL MOTION PREDICTION METHOD WITH CONVOLUTIONAL NEURAL NETWORKS

In this part, we describe a method for deducing global camera motion from optical flow using convolutional neural network.

2.1. Problem formulation and previous prediction model

We recall the formulation of the problem from [5] which is the basis of this work.

The method segments foreground and background by setting to 0 all the optical flow [17] elements close enough to the 2D median (u^*, v^*) of the optical flow.

Optical flow of the foreground is defined as the matrix $F^t \in \mathbb{R}^{N \times M \times 2}$ composed by elements $f^t_{i,j}$ defined as

$$f^t_{i,j} = \begin{cases} (u_{i,j} + u^*, v_{i,j} + v^*), & \text{if } ||(u_{i,j}, v_{i,j}) - (u^*, v^*)||^2 \geq \theta. \\ 0 & \text{else .} \end{cases} \tag{1}$$

where $(u_{i,j}, v_{i,j})$ is the result of optical flow estimation at position i, j. and θ a threshold. We assume that the background displacement corresponds to camera motion and we want to differentiate it from the foreground motions that we suppose to be the player motions.

In [5], a piecewise linear supervised learning model to predict global camera motion is described. This model uses a learned weight matrix z_s for each situation occurring in the videos to predict camera global motion from the segmented optical flow of the foreground

$$d^t = \langle f^t \eta, z_s \rangle + e^t, \tag{2}$$

where $d^t \in \mathbb{R}$ is the motion predicted, $f^t \in \mathbb{R}^{NM}$ the flatten vector of segmented horizontal optical flow defined as in (1), $z_s \in \mathbb{R}^{NM}$ the learned vector for each situation, $\eta \in \mathbb{R}^{NM}$ a normalization vector, and $e^t \in \mathbb{R}$ the noise. This model needs a robust situation management algorithm to use the right matrix for what is happening in the field. This task can be challenging during long sequence videos, this is why this model has shown difficulties to track long sequence actions. Evaluation of this method shows accurate performances on short basketball videos but it starts to show bad results on long sequences and other sport videos. These errors are induced by the lack of parameters in this simple model and the complexity of situation management in other sports.

2.2. CNN architecture for action tracking

We propose the following network architecture to predict global motion: the model takes as input 3 foreground optical flows computed at times t, $t-k$, and $t-2k$ to add temporal component analysis and outputs a prediction of the background motion at time t. The gap k between the inputs can be adjusted according to the needs of each sport. This allows the model to use the evolution of players' motion across time to improve performances as it can adjust its prediction with respect to what happened previously. This also makes it more robust to the variations of amateur sport videos. The architecture of the network is summarized in Table 1. In this work, we are considering only the horizontal component of optical flow. We assume that vertical motions don't give consistent information for the evaluated sports. The model can easily be adapted to 2 dimensions optical flows if the vertical component has to be analyzed.

Block	Layers	output size
Conv1	Conv : 32 3x3 filter Maxpool : 2x2 filter	(32, 90, 160)
Conv2	Conv : 64 3x3 filter Maxpool : 2x2 filter	(64, 45, 80)
Conv3	Conv : 128 3x3 filter Maxpool : 2x2 filter	(128, 22, 40)
Conv4	Conv : 256 3x3 filter Maxpool : 2x2 filter	(256, 11, 20)
FC1	FC : 128x11x20	(1,128)
FC2	FC : 128x64	(1,64)
FC3	FC : 64x32	(1,32)
FC4	FC : 32x1	(1,1)

Table 1: Network architecture: This table shows the components of our motion prediction network and the different sizes of output for horizontal flows of size 90x160. The input is processed by 4 convolutional layers Conv1, Conv2, Conv3, and Conv4 producing convolutional feature maps. Then the result is flattened to pass through 4 fully connected layers (FC1, FC2, FC3, and FC4).

The neural network takes as input 3 foreground segmented optical flows of size (160,90). Thus, it can be applied at each time onto videos of any length that is longer than $2k$ frames. The output is the predicted camera motion normalized in a percentage of the field. The advantage of this architecture is that it can be computed in real-time by embedded solutions. The architecture of the network has been designed to limit the complexity of the calculations while maximizing the accuracy of the model. This network architecture has achieved very fast processing times. On common laptop CPUs, it is calculated at 70,5 fps. It allows being calculated in real-time for classic video qualities but also for very high-quality solutions. Figure 1 shows an illustration of our proposed pipeline to summarize and understand how the method works.

2.3. Training and dataset

The evaluation is performed on videos from two sports: basketball and handball. As the two sports require different behaviors, we separated the training database into two bases (60494 optical flow matrices for basketball and 42654 for handball). Videos from this database match the constraint that the camera is placed close to the middle of the field to make our model consistent. The networks are learned using ADAM [18] optimizer.

As explained in [5], the main advantage of this work is that the database can be annotated automatically. The camera displacement is calculated as the 2D median of the horizontal optical flow assumed as the background global motion calculated at time t. The motion to be predicted is then annotated with this value, normalized as a percentage of the total field.

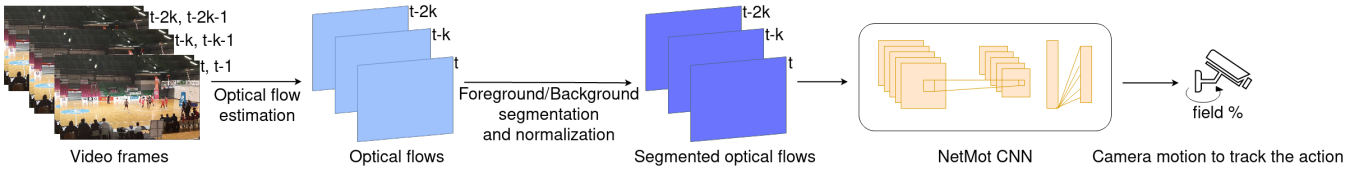

Fig. 1: Detailed pipeline of the proposed algorithm.

3. EXPERIMENTAL RESULTS

In this section, we study the effectiveness of our optical-flow-based method for tracking interesting content in sport amateur games. We will compare the method presented in this paper (NetMot) with a version of the CNN that takes as input a single segmented optical flow computed at time t (NetMotV0) to show the benefits of the addition of temporal analysis, and the method presented in [5] (PEN). As there is no other out-of-the-box method in the literature that allows this type of prediction, it is difficult to numerically compare our method with others. Performances are evaluated on 3 different test datasets. First, to compare our model to the PEN model, on short basketball videos, we tested it on the 15 basketball videos presented in the paper [5]. To evaluate the ability of the model to track long sequences, we tested it on 7 basketball videos ranging from 24 seconds to more than two minutes. Finally, to evaluate the adaptation to another sport, we also evaluated the model on 14 handball videos.

3.1. Evaluation metrics

To evaluate the numerical results of the model, we use the mean absolute error (MAE) on predictions and time integrated predictions (i.e. position with respect to the start of the video). We are analyzing time integration of prediction results because it highlights the shift between the position of the predicted camera tracking at time t and the ground truth. We will name the MAE on integrated predictions as the Tracking Error and we consider that when it is over 15% of the field, the algorithm has lost the location of the action. On long videos, we also need to analyze the MAE on sub-sequences of 10 seconds to make sure that the camera never loses track of the match before recovering it. The section MAX TE (Max Tracking Error) corresponds to the maximum MAE computed on 10-second subsequences. Analyzing MAE on standard prediction allows us to see if predictions are locally close to ground truth. However, a model can have predictions closer to the ground truth but a less accurate tracking, because of drift and compensation phenomena.

3.2. Evaluation results

Performances on short basketball videos are summarized in Table 2. The evaluation results on this test database show that NetMot is more accurate than the PEN model. Focusing on the standard predictions, we see that the MAE is significantly lower for NetMot. 13 of the 15 videos have lower MAEs with NetMot than with PEN. This result means that the model predicts displacements that are closer to the ground truth. It means that the model uses less compensation to keep track of the game. This makes the global motion of the camera smoother for the viewer. Looking at the Tracking Error, we see that NetMot is more accurate than PEN across time too. The average MAE is significantly lower for the new model. This means that the camera position is on average closer to the ground truth. The action is better centered and the loss of the game thread is less likely. 13 of the 15 videos have lower MAEs on integrated predictions with NetMot than with PEN. These results show that using a convolutional neural network, adding temporal dimension and the increase in the number of parameters have enhanced the precision and robustness of the model. These characteristics allow the model to analyze and understand the players' motions in a much more complex way. They also allow being more robust to variations in the acquisition mode. The tracking error of the PEN model on this test database is higher than that presented in [5]. This result is explained by the fact that in this work, this model was trained on a larger and more varied database. These results highlight a lack of flexibility of the previous model [5] which had difficulties in adapting to more varied data. Comparing NetMot with the NetMotV0 network, we can see that the addition of the temporal dimension in the inputs increases the accuracy of the network across time. The analysis of the evolution of the movements between frames t, $t-10$, and $t-20$ allows the network to perform a more advanced analysis of the players' displacements. This analysis allows it to better contextualize the movements and it makes them less dependent on the acquisition parameters. This result is highlighted on the Tracking Error.

	MAE			Tracking Error		
	PEN	NetMotV0	**NetMot**	PEN	NetMotV0	**NetMot**
MEAN	0.18	0.11	**0.07**	12.04	7.15	**4.06**
BEST	2	**7**	**7**	2	6	**7**
NUMBER OF MAE >15				3	4	**1**

Table 2: Summary of the performance of the different algorithms on the short basketball videos database. Mean absolute error on motion prediction (MAE) and Tracking Error with the two versions of the model (NetMot and NetMotV0) and the model presented in [5] (PEN). NetMot provides the best results.

The results on the handball video database presented in Table 3 again show that the NetMot model is the most accurate method across time. It was able to track the actions of all the videos in the test database without losing the location. The NetMotV0 method obtained more accurate local prediction but it was much more prone to the drift phenomenon and it obtained much worse across-time results. These results highlight the interest in using the temporal component for prediction. The PEN piecewise linear method showed accurate results for tracking handball videos. However, the results are on average significantly less accurate than those obtained by NetMot.

The performances of the method on long sequences are presented in Table 4. These results and the curves presented in Figure 2 show that NetMot can follow long sequences of matches without losing the thread. To be consistent with the evaluation metric, and to know if the algorithm has lost track of the game during the sequence, we need to study the MAE on 10-second sub-sequences. This value must be less than 15% of the field for the result to be considered accurate. In this database, the MAX MAE on the subsequences section of Table 4 shows that no 10-second subsequence has been tracked with an MAE higher than 15% of the field. This table also highlights the improvement that the use of convolutional neural

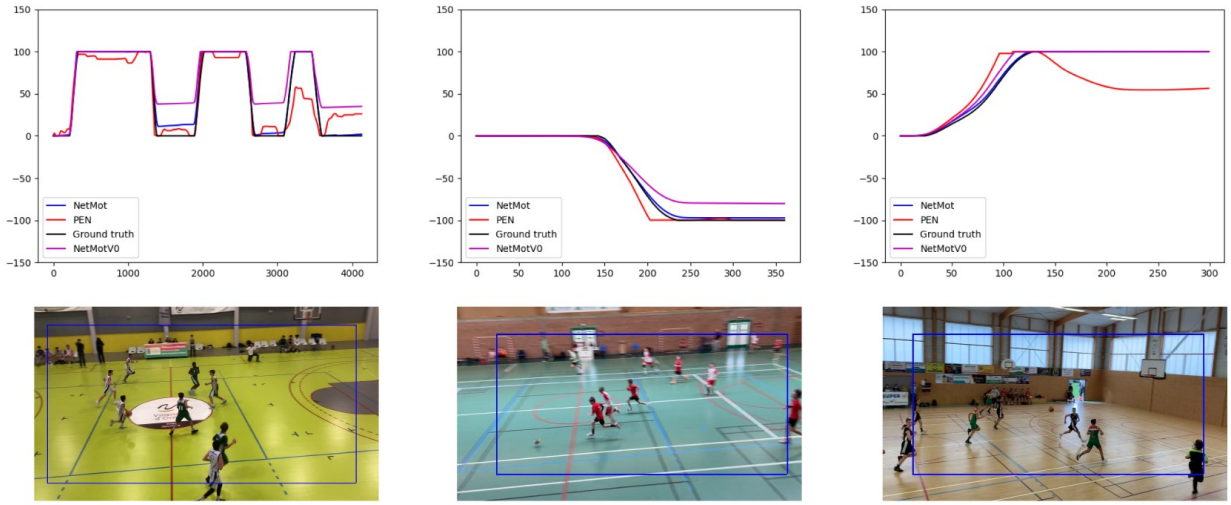

Fig. 2: Examples of well-tracked videos from long basketball sequence (left), handball sequence (middle) and short basketball sequence (right). The blue box shows the position of the (virtual) camera using our tracking method.

	MAE			Tracking Error		
Video	PEN	NetMotV0	**NetMot**	PEN	NetMotV0	**NetMot**
1	0.21	0.11	**0.08**	**4.92**	10.03	5.68
2	0.31	0.13	**0.11**	**10.68**	17.86	10.79
3	0.10	**0.06**	0.11	14.04	**3.14**	5.31
4	0.1	**0.07**	0.16	4.02	3.68	**2.99**
5	0.19	**0.05**	0.29	9.27	**3.97**	9.09
6	0.15	0.13	**0.1**	**4.82**	21.53	11.6
7	0.1	**0.07**	0.08	5.14	12.66	**2.61**
8	0.26	**0.08**	0.17	9.36	7.96	**5.49**
9	0.14	**0.1**	**0.1**	**2.97**	20.45	6.27
10	0.15	**0.12**	0.13	**6.17**	22.35	11.22
11	0.24	**0.06**	0.11	9.65	13.3	**6.62**
12	0.22	**0.09**	0.1	7.02	10.43	**2.71**
13	0.14	**0.05**	0.12	4.1	8.25	**2.8**
14	0.15	**0.08**	0.07	4.80	13.13	**0.57**
MEAN	0.18	**0.08**	0.12	6.92	12.05	**5.98**
BEST	0	**11**	4	5	2	**7**
NUMBER OF MAE >15				**0**	4	**0**

Table 3: Mean absolute error on motion prediction (MAE) and Tracking Error with the model NetMot on handball videos. The table shows that our model NetMot tracks the action on short handball sequences without losing the match thread (Tracking Error ≤ 15).

networks brings over the piecewise linear method. One of the limits of the method presented in [5] is its capacity to track long sequences, in particular, because of the classification of situations. The results of PEN on the base of long videos show that this method lost the position of the action several times. These results show that the method presented in this paper is robust to long game sequences.

The results presented in the curves of the Figure 2 show how the Netmot model (blue curve) is able to predict values close to the ground truth (black curve) compared to other models. The number of frames is different between short and long sequences. This explains why the transitions look sharper on the curves associated with the long sequences. On the images, we can see that the sequences are well tracked by the algorithm because the blue rectangle is not shifted from the capture camera. This means that the predicted camera has followed the same movements as the real camera.

		MAE		Tracking Error		MAX TE	
Video	Duration	PEN	**NetMot**	PEN	**NetMot**	PEN	**NetMot**
1	0:34	0.15	**0.09**	6.67	**2.42**	10.69	**3.76**
2	0:24	**0.001**	0.003	0.05	**0.005**	0.057	**0.005**
3	2:18	0.1	**0.07**	14.14	**0.03**	45.86	**12.58**
4	0:30	**0.001**	0.004	0.11	**0.01**	0.12	**0.009**
5	1:42	0.12	**0.04**	23.08	**5.68**	39.50	**8.87**
6	0:53	0.16	**0.07**	11.21	**4.44**	17.72	**10.83**
7	0:56	0.14	**0.08**	24.15	**4.64**	53.82	**9.24**
	MEAN	2.93	**0.04**	11.34	**2.46**	-	-
	BEST	2	**5**	0	**7**	0	**7**

Table 4: Mean absolute error on motion prediction (MAE) and Tracking Error with the model NetMot on long videos. The table shows that NetMot tracks the action on long sequences without losing the match thread (MAX TE is always ≤ 15).

4. CONCLUSION AND FUTURE WORKS

In this paper, we presented a method based on convolutional neural networks to track the actions of various amateur sports in real-time. This method differs from other state-of-the-art methods by several points. First, it is reusable out of the box in several sports. The use of player movement analysis rather than precise object location makes it robust to non-professional conditions. The automatic annotation of the training databases also allows a significant time saving compared to methods where the position of objects in the images is necessary for the training. Finally, the low computational cost makes it possible to compute the tracking in real-time on low computational power embedded processors. Our method has shown very good results in its ability to track actions. It is able to track actions in several different sports as well as long sequences of matches. However, in real conditions, losing the thread of the game can lead to real difficulties to find it quickly. Designing a tracking method based on player movements in amateur videos captured with a wide-angle camera could help limit this factor.

Acknowledgements

This work was co-funded by Rematch Company, the Ministère en charge de l'Enseignement Supérieur, de la Recherche et de l'Innovation and ANRT.

5. REFERENCES

[1] Paresh R Kamble, Avinash G Keskar, and Kishor M Bhurchandi, "A deep learning ball tracking system in soccer videos," *Opto-Electronics Review*, vol. 27, no. 1, pp. 58–69, 2019.

[2] Xinguo Yu, Hon Wai Leong, Changsheng Xu, and Qi Tian, "Trajectory-based ball detection and tracking in broadcast soccer video," *IEEE Transactions on multimedia*, vol. 8, no. 6, pp. 1164–1178, 2006.

[3] Tiziana D'Orazio, Cataldo Guaragnella, Marco Leo, and Arcangelo Distante, "A new algorithm for ball recognition using circle hough transform and neural classifier," *Pattern recognition*, vol. 37, no. 3, pp. 393–408, 2004.

[4] Matija Burić, Miran Pobar, and Marina Ivašić-Kos, "Object detection in sports videos," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2018, pp. 1034–1039.

[5] Axel Baldanza, Jean-François Aujol, Yann Traonmilin, and François Alary, "Piecewise linear prediction model for action tracking in sports," *Proceedings of the 30th European Signal Processing Conference (EUSIPCO 2022)*, 2022.

[6] Glenn Jocher, K Nishimura, T Mineeva, and R Vilariño, "Yolov5," *Code repository https://github.com/ultralytics/yolov5*, 2020.

[7] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[8] Francisco Massa and Ross Girshick, "maskrcnn-benchmark: Fast, modular reference implementation of instance segmentation and object detection algorithms in pytorch," *Google Scholar*, 2018.

[9] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, pp. 91–99, 2015.

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[11] Xinchao Wang, Vitaly Ablavsky, Horesh Ben Shitrit, and Pascal Fua, "Take your eyes off the ball: Improving ball-tracking by focusing on team play," *Computer Vision and Image Understanding*, vol. 119, pp. 102–115, 2014.

[12] Tianzhu Zhang, Bernard Ghanem, and Narendra Ahuja, "Robust multi-object tracking via cross-domain contextual information for sports video analysis," in *2012 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2012, pp. 985–988.

[13] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar, "Rethinking the faster r-cnn architecture for temporal action localization," in *proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1130–1139.

[14] Khurram Soomro, Haroon Idrees, and Mubarak Shah, "Predicting the where and what of actors and actions through online action localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2648–2657.

[15] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid, "Action tubelet detector for spatio-temporal action localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4405–4413.

[16] Xiaolong Wang and Abhinav Gupta, "Videos as space-time region graphs," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 399–417.

[17] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2758–2766.

[18] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.