Action recognition in handball scenes

Kristina Host, Marina Ivasic-Kos, and Miran Pobar

Department of Informatics University of Rijeka, Center for Artificial Intelligence and Cybersecurity, University of Rijeka, Rijeka 51000, Croatia
{kristina.host, marinai, mpobar}@inf.uniri.hr

Abstract. Action recognition in sports, especially in handball, is a challenging task due to a lot of players being on the sports field performing different actions simultaneously. Training or match recordings and analysis can help an athlete, or his coach gain a better overview of statistics related to player activity, but more importantly, action recognition and analysis of action performance can indicate key elements of technique that need to be improved. In this paper the focus is on recognition of 11 actions that might occur during a handball match or practice. We compare the performance of a baseline CNN-model that classifies each frame into an action class with LSTM and MLP based models built on top of the baseline model, that additionally use the temporal information in the input video. The models were trained and tested with different lengths of input sequences ranging from 20 to 80, since the action duration varies roughly in the same range. Also, different strategies for reduction of the number of frames were tested. We found that increasing the number of frames in the input sequence improved the results for the MLP based model, while it didn't affect the performance of the LSTM model in the same way.

Keywords: Human Action Recognition, Action Recognition in Sport, Handball, Inception v3, CNN, MLP, LSTM.

1 Introduction

In the modern way of life, more and more attention is paid to healthy lifestyle, recreation and physical activity. The development of technology, especially mobile phones with various sensors and sophisticated cameras that can record fast movement, have made it possible to record the activities and performance of athletes during sports schools and recreation, and not just top athletes.

The collected data and their analysis can help the athlete or his coach to get a better insight into statistics related to the athlete's activity, but more importantly, to analyze the performance of an action, finding key elements of technique that should be adopted to improve player performance. This is especially important in team sports where a large number of players are present in the field and perform different actions at the same time, so it is difficult for the coach to follow everyone. Also, the need and opportunity for efficient ways of indexing and using the huge amount of available sports visual data has led to a growing interest in automatic analysis of visual data and sports scenes, such as automatic player detection, tracking and recognition of actions the players perform.

There are different sports on which researchers are focused today, such as basketball [1] [2], soccer [3] [4] [5], baseball [6], hockey [7] [8], volleyball [9] [10] [11], etc.

In general, to recognize human actions, one can use visual data, data obtained with sensors, or a combination of the two. Many researchers collected visual data from different online sources or recorded it themselves, defined some human actions in the domain on which they focused, and tried or proposed different approaches for action recognition. For example, in [12], the focus was on two basic player actions in broadcast tennis video, left and right swing, that were recognized using motion analysis and Support Vector Machine (SVM). Their main challenge were the far-view frames when a player figure might be only 30 pixels tall. In [7], the researchers focused on recognizing four hockey activities (penalty corner, goal, long corner, and free hit) based on video samples which they collected from International Hockey Federation and YouTube. They proposed a model that utilizes the pre-trained VGG-16 Convolutional Neural Network (CNN) which was fine-tuned for classification of hockey activities. The authors report good results, with model sometimes confusing between free hit and long corner, as both activities share mostly similar visual pattern in terms of player's position and appearance. In [2], a subset of the NCAA games available on YouTube was used to detect eleven events, and key actors in basketball games using only static information and basing their work on the Inception-7 [13] network. Authors of [14] extracted 5 broadcast videos from BadmintonWorld.tv channel on YouTube to compare the performance of different feature extraction methods for the task of recognizing 5 actions (clear, drop, lift, net shot, and smash) with a modification of Alexnet CNN and SVM. In [15], the Siamese spatio-temporal convolutional neural network that takes as an input RBG data images and Optical Flow was suggested to detect and classify actions in table tennis on a video dataset taken by GoPro cameras.

Except in sports, human action recognition is used in various domains such as video surveillance, abnormal activity identification, healthcare monitoring, and education, as detailed in a survey [16].

In this paper we focus on handball action recognition in video, for which different computer vision tasks should be combined. Firstly, the object or person detection can be applied to detect the players and the ball on the field [17], and secondly, the object tracking can be applied to follow the players' movements across the field [18] [19]. The active players are determined so that the key elements in a match can be followed and carriers of the game identified [20] [21]. Lastly, the action recognition can be applied on video sequences containing only the player of interest to recognize different actions across the field, in order to extract statistical data or analyze the players' performances. All experiments are performed on video sequences from the dataset of handball training sessions [22]. Since different handball actions can have different durations, here we explore how the number of frames that are input to the classifier and the way the frames are sampled from the original video affects the performance of two different classifiers for the action classification task. Furthermore, there are not many researches focusing on analysis of handball, so our idea is quite unique, but follows the main concepts from other action recognition approaches in sports mentioned above.

The paper is organized as follows: in the next section, there is an introduction on handball game, problem definition, and a description of the prepared handball dataset.

The experiment, the results of the experiment and observations are given in Section 3, followed by the conclusion and future research directions in Section 4.

2 Experiment workflow for action recognition in handball

Handball is a team sport played by 14 players divided into two teams. The point of the game is to use hands to pass the ball to each other in order to score a goal. During the game, every player is moving around the field and performing different actions.

The analysis of these actions, during or after matches and practices, would be simpler if an automated system could recognize them. For that reason, different algorithms for human action recognition are being developed [18]. Before applying them, it is necessary to collect and process visual data like video, to make same dimensionality reduction if needed, and to extract features from the data. Here, features are automatically extracted using the InceptionV3 [23] deep neural network and two approaches were used to recognize actions, one model that does not take into account the sequential steps of which an action consists but classifies each frame separately and two models which take into account the time information and the sequence of frames in action. To evaluate the model performances, we have used validation loss and accuracy metrics.

2.1 The dataset

The dataset contains a set of short high-quality video recordings of actions in handball, recorded indoors during a handball school [22]. The recordings were made using a stationary Nikon D7500 DSLR camera, with a Nikon 18-200mm VR lens, in full HD resolution (1920x1080) at 30 to 60 frames per second.

To get the subset with actions in it, it was necessary to label the recorded data and to extract parts of the video containing the chosen actions. The obtained subset consists of 2,991 short videos in .mp4 and .avi format, belonging to 11 different action classes. Considered classes are: Throw, Catch, Shot, Jump-shot, Running, Dribbling, Defense, Passing, Double-pass, Crossing and Background where action is not happening.

The subset was then split into training and testing sets in a ratio of 80:20. The distribution of videos through classes and sets is presented in Table 1.

Table 1. Distribution of videos through classes and train/test sets

Class name	No. Videos (train)	No. frames (train)	No. Videos (test)	No. frames (test)
Throw	184	4907	32	913
Catch	202	4120	50	937
Shot	83	6097	22	1655
Jump-shot	270	18018	83	5676
Running	56	6049	14	1424
Dribbling	42	3549	11	753
Defense	97	6027	30	1668

Passing	509	30618	121	7252
Double-pass	35	2122	11	654
Crossing	238	18204	59	4482
Background	684	24929	158	8342
Total	2400	124640	591	33756

The average number of frames in the short videos, depending on the action class the video belongs to, is shown in Figure 1. It can be seen that most actions contain an average number of frames of around 60 or higher, with only Throw and Catch actions that are significantly shorter, that is, the average number of frames in which those actions occur in is approximately two or more times shorter than the other ones.

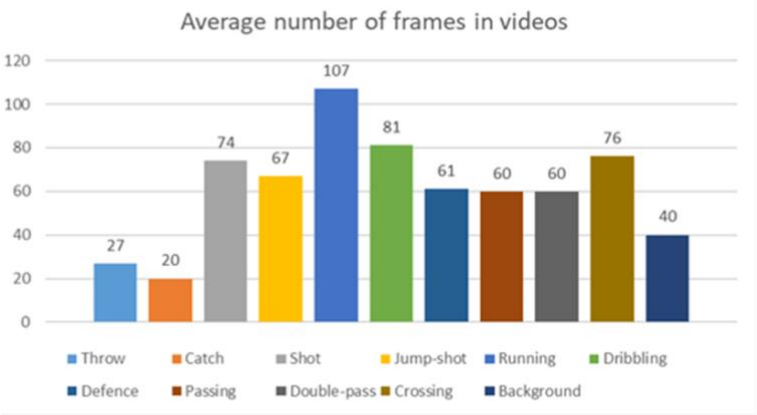


Fig. 1. Average number of frames per action from the handball dataset

An example of frame sequence for the Catch action, that is consecutive frames where a player is reaching out with his hands to catch a ball, is shown in Figure 2. Because of the different typical lengths of actions, we tested models with different input lengths ranging from 20 to 60 frames.

Since the Passing action consists of Throw and Catch actions in sequence, we also performed all experiments excluding these two classes, so we have a set of models trained on all 11 classes and a set of models trained on 9 classes.

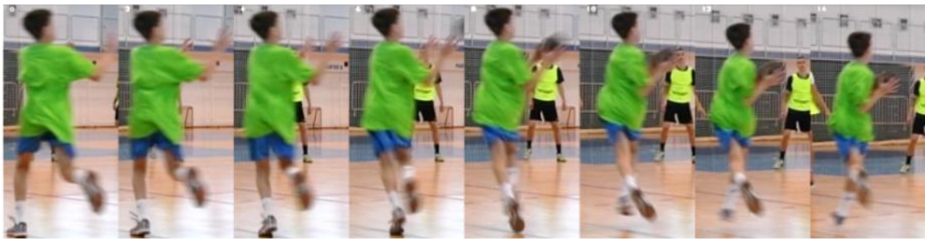


Fig. 2. An example of frame sequence for the Catch action.

2.2 The metrics

The loss function represents a measure of errors made for each example during training and validation. It is used to optimize the algorithm and to conclude how well the model is performing after each iteration of optimization. If the model's predictions are good, the loss is equal or close to zero, otherwise the loss is greater.

Accuracy is a metric for evaluating the classifier's performance, it shows how many predictions our model got right, that is, how accurate the model's prediction is compared to the true data shown in percentage or in interval [0, 1]. If the model's predictions are good the value increases towards a 100%. The point is to have low value of loss and high value in accuracy.

2.3 The action recognition models

As the baseline model, the InceptionV3 network [23] was fine-tuned on our handball dataset using the ImageNet [24] pre-trained weights as the starting point. The fine-tuning was performed so that the final output layer in the original network was replaced with a dense layer with 1,024 neurons followed by an output layer with 11 neurons with SoftMax activation function. The newly added layers were trained for 10 epochs with 12,4640 images per epoch using the RMSprop optimizer in the Keras framework with learning rate of 0.0001. Then, the top two inception layers and the newly added layers were trained for additional 25 epochs using the SGD optimizer with the learning rate of 0.0001 and momentum value of 0.9. In our experiment, this network is called CNN.

CNN voting model

The fine-tuned network was used to classify each frame in the video into one of the 11 classes, and then a majority voting scheme was used to determine the classification label for the whole sequence, so that the class label that occurs the most often among the predicted labels for each frame was selected as the label for the whole sequence. This step is presented in experiment as CNN voting.

Additionally, for comparison with other networks that only see a fixed number of frames per video, the CNN network was tested so that the same number of frames that is used as input to other networks is classified. Then the majority vote is used again to obtain the class label for the video.

LSTM and MLP models

To capture the temporal information in the video sequence, two networks with different configurations were defined. The first is a ***LSTM***-based network with one LSTM layer with 1,024 units, one fully connected layer with 512 neurons, each followed by a dropout layer with 0.5 dropout rate and a final output layer with 11 neurons. The second, hear called ***MLP***, is a neural network with two fully connected hidden layers, with 512 neurons in each hidden layer, followed by a dropout layer with dropout rate of 0.5

The input to both networks consists of a sequence of features extracted from video frames using the baseline (InceptionV3) model, tapped at the final pooling layer, yielding 2,048 features per video frame.

The networks were trained using the Adam optimizer with a learning rate of 0.00001 and decay of 10-6 for up to 100 epochs, stopping early if the validation loss does not improve for more than 20 epochs.

Frames selection strategy

Because the input to the network must be of equal length for each example, and the source videos containing actions consist of a different number of frames, ranging from 20-100, input videos have to be reduced to a fixed number of frames. Therefore, as a network input, we tested several different video sequences of 20 to 80 frames in length with different frame selection strategies.

In the videos that contained fewer frames than the network expects, copies of existing frames were inserted between frames to extend the number of frames. Conversely, to reduce the number of frames of long videos, the chosen number of frames were selected consecutively from either beginning, middle or the end of the video, or from the whole video by decimation, i.e. by skipping some frames at regular intervals. This was tested since different actions might have most distinctive characteristics in different parts of the sequence, e.g. the beginning of a jump-shot action might be similar to the running action and thus not the most informative, while for some other action like throw the beginning when the player lifts the hand with the ball might be more important.

Both networks were trained for each combination of string lengths and selection strategies.

3 Experimental results and observations

The results of the experiment and some observation about the performance of different models are presented below.

3.1 Comparison of the performances of action recognition models

In the first part of the experiment, we trained the CNN, LSTM and MLP models on all 11 classes and on 9 classes with different number of frames in the input sequence, obtained from the whole video by skipping frames at regular intervals (decimation of frames). The results in the terms of validation accuracy are shown in Figure 3. The results for the MLP models are marked with dots, for the LSTM models with downward pointing triangles, and with upward triangles for the CNN models. The validation accuracy values are shown on the vertical axis, and on the horizontal axis are shown the number of frames used as input to the classifier (20, 30, 35, 40, 45, 60, 80).

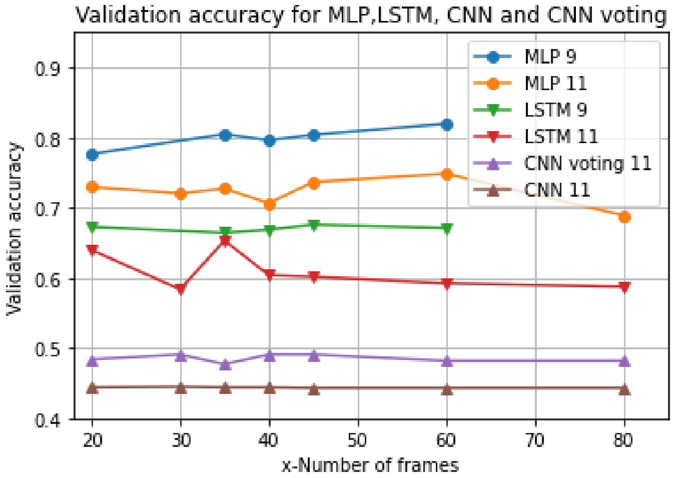


Fig. 3. Validation accuracy values obtained by skipping frames for 9 and 11 classes.

Out of all the models, the MLP model achieves the highest validation accuracy for both 9 and 11 classes. The best accuracy score of 81.95% the MLP model achieves with 60 input frames, but it achieves a similar score of 80.01% on average for all numbers of frames taken in the case of 9 classes. For 11 classes, MLP again has the highest accuracy score of 75% with the same number of input frames and the average accuracy for all tested number of input frames of approximately 72.25%. It can be seen that the validation accuracies for the model trained on 9 classes are higher than the ones trained on 11. In addition to the simpler problem with smaller number of classes, the reason for this is that Throw and Catch are fundamental parts of other actions (passing, double-pass, crossing, shot, jump-shot), which makes it more difficult to recognize the right action.

The LSTM model achieved significantly lower validation accuracy values compared to the MLP model. It achieved the best score for 9 classes of 67.58% with 45 input frames (67.05% on average), while for 11 classes the best accuracy was 65.31% for 35 input frames (60.88% on average). The worst performance is achieved by the CNN model that has no temporal dimension with the best score of 44.5% for 30 input frames, but with minimal difference between scores for different input lengths (44.37% on average). With the majority voting scheme, the score improved for about 4% to 49.1% for 45 input frames, or 48.54% on average. From this result, it can be concluded that the temporal dimension plays a major role in action recognition.

3.2 Analysis of model performance with respect to the frame selection strategy

In the second part, we analyzed the performance of the MLP and LSTM models based on the chosen number of frames sampled from either the beginning, the middle or the end of the video, or by skipping frames at regular intervals.

In the following figures (4-7), the different frame selection strategies are represented with different colors. The strategy with frames selected from the beginning is marked as *first*, the one with frames from the middle is marked as *middle*, with frames selected from the end of the video as *last*, and with frames selected by skipping frames at regular intervals as *def*.

Figure 4 and Figure 5 show the results obtained with the LSTM model trained for 11 classes, and 9 classes, respectively, taking into account different frame selection strategy.

The maximum validation accuracy is obtained with the model trained on 11 classes with 45 frames taken from the middle of video, with the value of 70.94%, followed by 70.55% obtained with the model trained on 9 classes with 20 frames in the middle, and 70.47% for the last 45 frames and 9 classes.

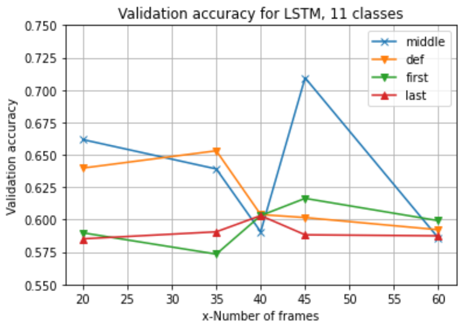


Fig. 4. Validation accuracy for the LSTM models, 11 classes

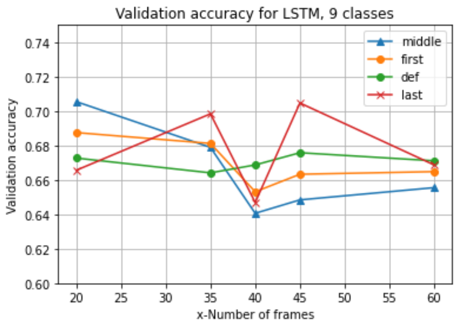
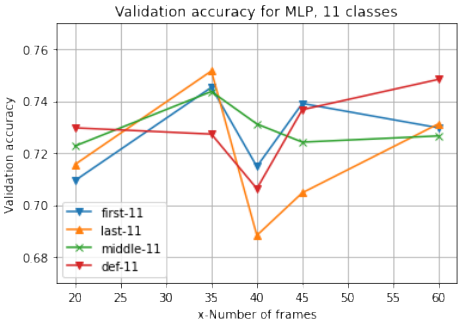


Fig. 5. Validation accuracy for the LSTM models, 9 classes

If we take into consideration the number of frames for both 9 and 11 classes, the best results are obtained with 45 frames followed by 20 frames. In most cases, (except the best result overall), for the LSTM model, additional frames in the sequence don't improve the result much over the models with 20 frames.

Looking at the way the sequence is taken and not at the number of frames, the highest average accuracy (67.69%) is achieved by the model with 9 classes taking into consideration only the last frames, followed by 67,05% by skipping frames.

Figures 6 and 7 show the results obtained with the MLP model trained for 9 and 11 classes, respectively.



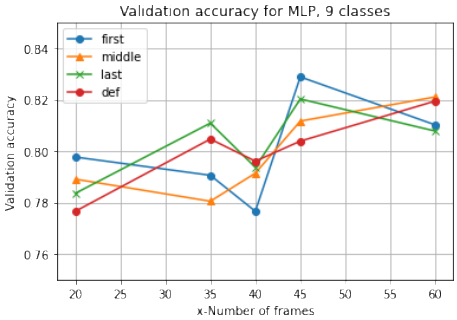


Fig. 6. Validation accuracy for the MLP models, 11 classes

Fig. 7. Validation accuracy for the MLP models, 9 classes

Model MLP achieves the maximum validation accuracy for 9 classes when is trained with 45 frames from the beginning (82.89%) followed by 82.11% obtained with 60 frames in the middle, and by 82.03% with the last 45 frames. In the case of 11 classes, the MLP model achieves the best results when it is trained with 35 frames, from the end of the video sequence (last), but comparative results achieve with the same number of frames from the middle and from the beginning of the video sequence (first). It is evident from the graph that much better validation accuracies are achieved with 9 classes than with 11 and also that the best results in case of 9 classes were achieved when 35 frames were used, and in the case of 11 classes when 45 frames were used. As opposed to the LSTM model, for the MLP model, the higher number of input frames in general increased the achieved accuracy.

Looking at the way the sequence is taken and not at the number of frames, the highest average accuracy (80.33%) is achieved by the MLP model with 9 classes taking into consideration only the last frames, followed by 80.08% looking at the first frames.

From the obtained results it is not possible to conclude exactly which frame selection strategy is the best and how many frames gives the best results because the results differ between models, and even for the same model for a different problem (9 vs. 11 classes). The number of frames and frame selection strategies appear to be highly dependent on the type of action being performed, so we will explore this in a future research.

In Figure 8, validation accuracies for all combinations of MLP and LSTM models, frame selection strategies and number of frames for 9 and 11 classes, are presented. The validation accuracy values for the MLP models trained on 9 classes are marked with red dots, and for the models trained with 11 classes are marked with blue cross. The results for LSTM models are marked with upward pointing triangles for the model trained on 9 classes, and downward pointing triangles for the model with 11 classes. The number of frames used as input to the classifier are shown on the horizontal axis.

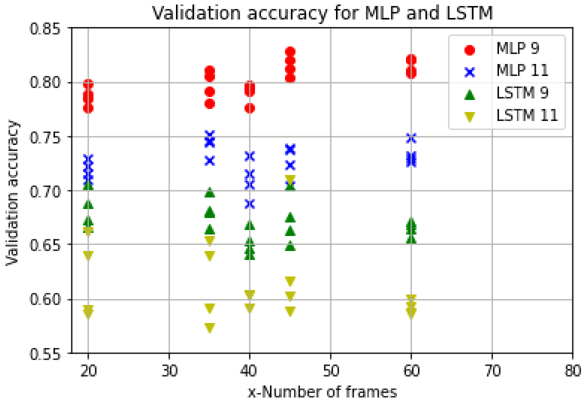


Fig. 8. Validation accuracy for MLP and LSTM

We can see that the MLP algorithm trained on 9 classes obtains best results regardless of the number of frames, or the way they are taken. The validation accuracies for the model trained on 9 classes are generally higher than the ones trained on 11 classes but MLP model with 11 classes performs better than LSTM with both 9 and 11 classes. The best overall result of 82.89% of accuracy is the one obtained with MLP model for 9 classes with 45 input frames.

If we observe the average accuracy per number of frames for all observed strategies of frame selection, the highest average accuracy (81.62%) is achieved by the MLP model with 9 classes and 45 frames, followed by 60 frames (81.47%). By the same principle looking at the number of frames rather than the way the frame sequence is taken, the LSTM model achieves the highest average accuracy (68.28%) with 9 classes and with 20 frames, followed by 45 frames (68.07%).

4 Conclusion

The goal of this paper was to test the performance of different algorithms for action recognition on our handball dataset containing short videos of 11 action classes.

We compared the validation accuracies of a CNN-based classification model used as baseline that doesn't use the temporal dimension of input videos but classifies the frames individually, with two models that do (LSTM and MLP). Both LSTM and MLP models are built on top of the CNN model, so that the input features to the LSTM and MLP are calculated using the CNN model. For LSTM and MLP models we tried different numbers of classes (9 and 11), different numbers of frames (20, 35, 40, 45, and 60) taken in different ways (from the beginning, the middle, the end, and by frame decimation).

Overall, we obtained the best results with the MLP model with validation accuracy 82.89% for 9 classes trained on the first 45 frames of the video, significantly higher than the best result obtained with the baseline classifier (49.1%). We can conclude that the MLP and LSTM models successfully exploit the information in the temporal dimension to for recognizing handball actions.

Our data set contains handball actions that differ significantly in the duration of the performance, so to train the model they had to be reduced to the same length. Experiments have shown that the number of frames and frame selection strategies can significantly affect the accuracy of the action recognition, however, an exact conclusion about the best strategy for frame selection from the obtained results is difficult because the strategy that gives the best results differs between models, and even for the same model for a different problem (9 vs. 11 classes). Nevertheless, for the LSTM model it can be concluded that increasing the number of input frames does not contribute to a better result, regardless of the frame selection strategy, while for MLP it could be concluded that a larger number of frames can positively affects the results of action recognition. The number of frames and frame selection strategies appear to be highly dependent on the type of action being performed, so the question of the ideal number of frames, and the question of best selection strategies remains opened. Therefore, it will be more deeply analyzed in further work, where each action will also be considered separately. Likewise, we plan to enlarge our dataset, especially for those actions for which we have fewer videos, to achieve a more balanced distribution of videos for different action classes.

Acknowledgment

This research was fully supported by the Croatian Science Foundation under the project IP-2016-06-8345 “Automatic recognition of actions and activities in multimedia content from the sports domain” (RAASS) and by the University of Rijeka under the project number uniri-drustv-18-222.

References

1. R. Ji: Research on Basketball Shooting Action Based on Image Feature Extraction and Machine Learning, IEEE Access, vol. 8, pp. 138743-138751 (2020).
2. V. Ramanathan, J. Huang, S. Abu-El-Haija, A. Gorban, K. Murphy i L. Fei-Fei: Detecting Events and Key Actors in Multi-person Videos, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas (2016)
3. R. Sanford, S. Gorji, L. G. Hafemann, B. Pourbabaee i M. Javan: Group Activity Detection from Trajectory and Video Data in Soccer, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle (2020).
4. B. Gerats: Individual action and group activity recognition in soccer videos, Faculty of EEMCS, University of Twente, Twente (2020).
5. K. Bonenkamp: Action Recognition in Soccer Videos, Informatics Institute, Faculty of Science, University of Amsterdam, Amsterdam (2014)
6. A. Piergiovanni i M. S. Ryoo: Fine-Grained Activity Recognition in Baseball Videos, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City (2018).
7. K. Rangasamy, M. As’ari, N. Rahmad i N. F. Ghazali: Hockey activity recognition using pre-trained deep learning model, ICT Express (2020).

8. K. Sozykin, S. Protasov, A. Khan, R. Hussain i J. Lee: Multi-label Class-imbalanced Action Recognition in Hockey Videos via 3D Convolutional Neural Networks, 2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Busan (2018).
9. F. Haider, F. Salim, D. Postma, R. Van Delden, D. Reidsma, B.-J. BEIJNUM i S. Luz: A Super-Bagging Method for Volleyball Action Recognition Using Wearable Sensors, Multimodal Technologies and Interaction, (2020)
10. M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat i G. Mori: A Hierarchical Deep Temporal Model for Group Activity Recognition,2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas (2016).
11. T. Bagautdinov, A. Alahi, F. Fleuret, P. Fua i S. Savarese: Social Scene Understanding: End-to-End Multi-person Action Localization and Collective Activity Recognition, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, (2017)
12. G. Zhu, C. Xu, Q. Huang, W. Gao i L. Xing: Player Action Recognition in Broadcast Tennis Video with Applications to Semantic Analysis of Sports Game,Association for Computing Machinery, New York (2006).
13. Ioffe, S., Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv:1502.03167 [cs]. (2015).
14. N. A. Rahmad i M. A. As'ari: The new Convolutional Neural Network (CNN) local feature extractor for automated badminton action recognition on vision based data, Journal of Physics Conference Series (2020).
15. P. Martin, J. Benois-Pineau i R. Péteri: Fine-Grained Action Detection and Classification in Table Tennis with Siamese Spatio-Temporal Convolutional Neural Network, 2019 IEEE International Conference on Image Processing (ICIP), Taipei (2019).
16. P. Pareek i A. Thakkar: A survey on video-based Human Action Recognition: recent updates, datasets, challenges, and applications, Springer Nature, Artificial Intelligence Review (2020).
17. Burić, M., Pobar, M., Ivašić-Kos, M.: Adapting YOLO network for Ball and Player Detection. In: 8th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2019). pp. 845–851 (2019).
18. K Host, M Ivasic-Kos, M Pobar: Tracking Handball Players with the DeepSORT Algorithm, ICPRAM, 593-599, 2020
19. Buric, M., Ivasic-Kos, M., Pobar, M.: Player Tracking in Sports Videos. In: 2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom). pp. 334–340 (2019).
20. M Pobar, M Ivasic-Kos: Active Player Detection in Handball Scenes Based on Activity Measures, Sensors 20 (5), 1475
21. Ivasic-Kos, M., Pobar, M., Gonzàlez, J.: Active Player Detection in Handball Videos Using Optical Flow and STIPs Based Measures. In: 2019 13th International Conference on Signal Processing and Communication Systems (ICSPCS). pp. 1–8 (2019).
22. M. Ivasic-Kos i M. Pobar: Building a labeled dataset for recognition of handball actions using mask R-CNN and STIPS, 7th IEEE European Workshop on Visual Information Processing (EUVIP), Tampere (2018).
23. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the Inception Architecture for Computer Vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2818–2826. IEEE, Las Vegas, NV, USA (2016).
24. J. Deng, W. Dong, R. Socher, K. L. L. Li i L. Fei-Fei: ImageNet: A large-scale hierarchical image database, 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami (2009).