Action Recognition in Sports Videos From the Ground Up

Matija Burić
University of Rijeka, Department of Informatics, Rijeka, Croatia
matija.buric@hep.hr

***Abstract* – Lately, there is a huge interest in the computer vision and its applications due to the discovery of algorithms capable of performing a variety of tasks closely related to the visual observation of surrounding environment, in some cases even better than humans do. The purpose of this paper is to build up such a system which could, with some fine tuning, achieve object detection and even action recognition, from the ground up. In order to succeed several problems must be addressed. Previous studies of this research filed must be examined which will help in determining the waypoints, specific task must be defined which computer must accomplish, right dataset for training and testing have to be selected or even built, computer-generated knowledge in a form of an algorithm needs to be applied and finally everything must be put together to make it work. By the end of this article, one will possess the knowledge to build such a system and use it as a foundation for further research.**

Keywords – computer vision; action recognition, custom dataset; YOLO; Mask R-CNN; Ball detection

I. INTRODUCTION

Action recognition in video material is too complex for a computer to handle without prior data preparation and therefore needs to be disassembled into smaller tasks, small enough to be able to process it using simple mathematical representation. This traditional approach can be figuratively described through layers. At the far bottom layer, information from the camera feed or similar sensors is transformed into data bits, usually, these are a numerical interpretation of pixel intensity, which are then by the use of simple but very time and power consuming mathematical calculus pushed to an upper layer with requests for fewer resources but more knowledge. This is propagated through all layers until reaching the last one holding the information about the observed activity or set of actions.

These mathematical layers joined together in different logical areas can be observed through workflow, like shown in the Fig 1., which takes training data, prepares it in a described way, applies learning algorithm to it and builds a model used to make a prediction in a similar fashion.

Since, like in any other problem-solving puzzles, here also are many ways to come to a solution. At the early stages of computer vision research simpler techniques were used, mostly based on handcrafted features which, for the time being, gave promising results. Even though these were successful they needed many hours of human labor, intense supervised knowledge and they were hard to adapt to a classification of another object.

Thanks to the results of AlexNet [2] introduced in 2012, deep convolution neural networks (CNN) become intensely researched. They overcome shortcomings previously mentioned methods have and even outperform them in a sense of recall and precision but require more computing power and are very time-consuming. With a constant development of computer hardware, this disadvantage has lower significance with each passing day.

This research follows the mainstream and uses latest state-of-the-art methods based on CNN to achieve its goal of providing an optimal solution for an object detection which by use of some semantics will through a further research result in action recognition.

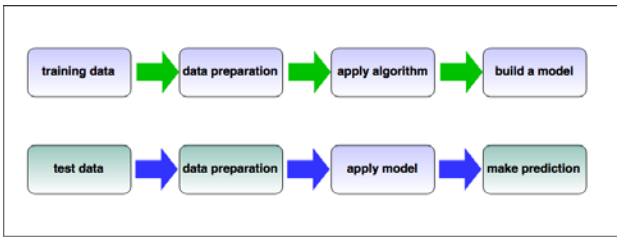

Figure 1. Traditional model of action recognition [1]

Picking the appropriate dataset for training and testing is essential for a positive outcome of the experiment and therefore part of this paper will include applicable publicly available datasets along with the dataset obtained from the real world sports footages.

II. OVERVIEW

A. *Models and Methods*

Images used for classification usually contain, due to extremely high dimensionality, too much information irrelevant for modeling. For this reason, machines perform classification based on primary elements called low-level features. The first step after dataset is acquired is features extraction. Features are, essentially, the interesting parts of the image altered in a way that machine can use. Common low-level features include edges - border between lighter and darker part of the image, corners - edges intersection, blobs – separate regions of image not sharply divided to be considered as an edge, etc.

When we talk about action recognition, features are separated into those which encode positions or trajectories of different body parts, those encoding whole body figure and into local features which are not related to a logical shape but operate on finding interest points. The whole body and body part features require additional video processing for detecting, segmenting and tracking such an object opposite to local features where no person localization is needed. For this reason, local features have been a primary choice an interest of many researchers in the field.

1) *Body based models*

Action recognition feature representations based on human body, whole or just part of it, use 2D and 3D features. Figures Fig. 2, Fig 3 and Fig. 4 shows human body features using sticks [3], silhouettes [4] and volumes [5] respectfully.

Most techniques use an explicit model of human body, such as a stick figure model, and strive to optimize the match between the model projections and an observed image frame while simultaneously keeping a correspondence of joints between frames. The resulting representation is a set of joint trajectories in 2D-time space or 3D-time space, as shown in Fig. 5.

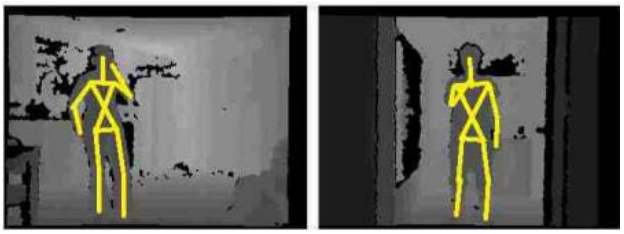

Figure 2. Stick figure from the Cornell Activity Datasets database

Figure 3. A video frame depicting a person walking (left) and the corresponding silhouette mask (right)

Figure 4. Volumes formed by stacking the silhouettes of persons while performing actions

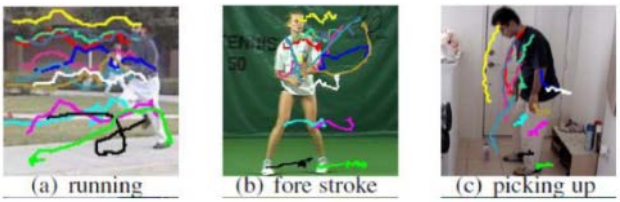

Figure 5. Tracked trajectories of joints generated by performing different actions

2D pose estimations, opposite to 3D, are less complex and therefore much easier to extract, but, since usually only one camera is used for input, occlusion makes a great difference in creating model for action recognition. Also 2D is very view-dependent meaning that the features for the same action will be very different depending on the relative orientation of the camera and the person performing the action. This can be improved by using footage from few different sources, but like mentioned before, it affects complexity and cost.

With a rising interest in computer vision new type of sensors are developed each day which allow capturing depth of an observing figure more easily. Some of those are RGB depth cameras which provide info about distance from camera based on infrared sensor and multiple camera feeds at fixed distances. This resembles human vising using both eyes.

2) *Bag of visual words models*

Major advantage of local features and bag of words over approaches relaying on the body model is that extraction of local features doesn't require human model or person localization. Features are extracted first by using an interest point detector which than describes local descriptor of that interest point. The descriptors are clustered into *visual words*. Swarm of visual words for each input image are called bag of visual words (BOW) and are used for learning process like shown in Fig. 6.

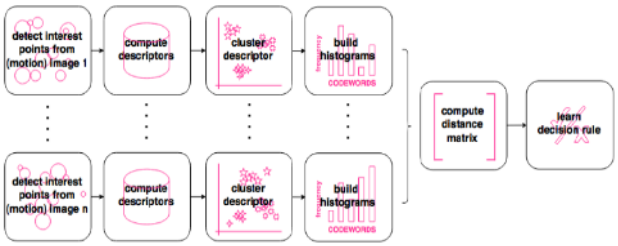

Figure 6. Bag-of-features learning diagram

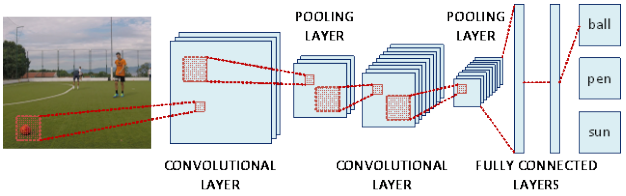

Figure 7. Typical convolutional neural networks

According to division in [6], interest point detectors are divided into contour based, intensity based and parametric based. To find edges and corners of the most interest contour based detectors are used. Intensity based detectors compute interest points solely on grey values of an image and parametric based combine previously mentioned detectors analytic approximation [7]. For example Harris 3D detector [8] computes spatio-temporal corners and determines space-time interest points by a local maximum. Cuboid detector is based on Gabor filters [9] which detects regions with spatially unique characteristics under a complex motion. Hessian [10] is used to detect space-time blobs and dense sampling [11] extracts 3D patches at fixed positions with scale variance.

When points of interest or trajectories information about form and movement are detected, interest point descriptors are used. Feature trajectories are usually extracted by matching SIFT [12] descriptors between frames or by using Kanade-Lucas-Tomasi (KTL) tracker [13].

Vast majority of descriptors are spectra descriptors based on calculated quantities, such as light and color intensity, local area gradient, statistical features and moments, surface normal and data sorted in histograms. At the time of writing this paper descriptor showing promising results, based on familiar 2D methods and capable of combining data from 3D sensors and accelerometers are 3D HOG [14], 3D SIFT [15] and HON 4D [16].

Further features processing is done with use of BOW or Fisher Vector approach [17] and then classified via common methods such as Multi-Layer Perceptron (MLP) in [18], and Support Vector Machine (SVM) [9] [8] [19] [20].

3) *Deep learning approaches*

During the last few years, the world is witnessing a steep development of neural networks, such as convolutional neural networks (CNNs) [21], used in image and video classification [22]. There are models which have a remarkable ability, compared to other previously mentioned state-of-the-art approaches, to make a prediction of the desired object in the photo and video material in non-staged, real-world conditions.

CNNs automatically extract features from the large number of images or frames inside datasets inspired by the biological neural networks that are found in human brain. These feedforward neural networks typically comprise three basic types of neural layers as shown on Fig. 7: convolutional layers, pooling layers and fully connected layers. Convolutional layer utilizes kernels (feature detectors) which when applied to the entirety of the image transform the information into a

feature maps for further processing. Due to its benefits, several studies such as [23], [24] proposed replacing fully connected layers to reduced learning time. Pooling layers takes convolution layers output and reduce its width and height before pushing it to another convolution layer. This subsampling (also called downsampling) doesn't affect depth dimension but leads to a certain information loss. Although, information is lost this behavior is favorable for its ability to decrease computational overhead and overfitting impact. The most used pooling techniques are max pooling [25] and average pooling [26]. There are also techniques like stochastic pooling [27], spatial pyramid pooling [28] and def-pooling [29]. Usually behind set of convolution and pooling layers high level reasoning is performed by fully connected layers. Opposite to convolution layers here are neurons connected to all activation in the previous layers. This

layers convert 2D feature maps into 1D feature vector which could be either forwarded to a set of categories intended for classification [30] or could be used as a feature vector in further handing [31].

There is a variety of the CNNs used for action recognition and many of them take different approach of achieving this task. For example, Authors of [32] implement the image classification by extending CNN to handle the temporal dimension of videos using several layers of 3D convolution starting with 7-frames deep cube. Others like proposed in [33] use two parallel networks capturing spatial and temporal information. First network obtains action information from a still image on individual video frame, while second one operates on the optical flow precisely describing the motion between frames and forms the temporal recognition stream. The networks outputs are merged into a final decision score using a classifier.

B. *Activity understanding*

The methods mentioned before are very accurate in object and simple action recognition still they have trouble dealing with complex and stratified actions and activities. For this reason, more descriptive model and logical operators with a help of expert knowledge should be used.

Current research in the field of activity understanding can be observed trough explanation provided in [34]. Motion image is first processed in Abstraction phase using either pixel features, objects and their properties or logical facts of knowledge. In the second phase Action modeling includes traditional classification methods via pattern recognition, for knowledge representation uses state models in a space-time domain and semantic models for understanding sequential actions. Examples of State modeling formalisms are finite state machines (FSMs), Bayesian networks (BN), HMM, etc. Semantic model requires expert knowledge of interesting subset of actions to be able to combine semantic relationships between sub-actions which is applied in cases where more

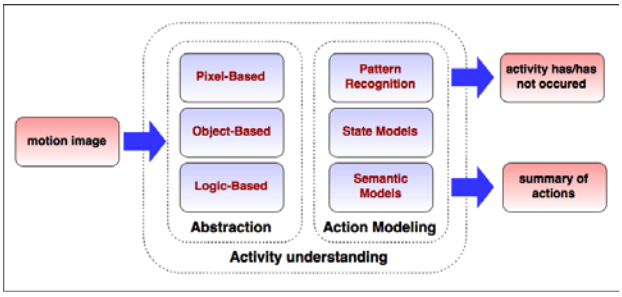

Figure 8. Activity understanding model after [34]

complex actions vary in their appearance [35] [36] [37].

Semantic models include grammars, Petri nets, constraint satisfaction, etc. Inaccuracies in lower-level recognition makes semantic models more unreliable due to it deterministic nature that's why the mechanism of fuzzy reasoning is desired to deal with doubt in observation and interpretation [38]. Research in [39] [40] takes a fuzzy knowledge representation scheme which enables uncertain knowledge modeling about associations between objects that could be used for indistinct interpretation of borders between actions in video sequence.

III. EXPERIMENT

A. *Description*

In order to address a problem of action recognition it was decided to solve a problem of object detection on still images first then to deal with tracking objects in motion to make a prediction of an action.

Object detection combines both classification and localization of desired object. It strives to present this object with some sort of marking, usually it is a bounding box around it that is labeled with its corresponding class label.

Authors of Viola-Jones algorithm [41] presented one of the first effective object detectors specialized in face detection. At the time of it release it was the most precise and very fast, capable of performing detection in real-time on webcam feed based on hand-coded Haar features and a cascade of classifiers. Since then there have been a few notable methods, one of them is Histograms of Oriented Gradients (HOG) [42] with remarkable capability of detecting human figure, but still requiring a had-coded features. In 2012 with a release of [2] deep learning came into focus providing some decent results in classification [24] [43]. Later on CNNs became capable of effective object detection with a release of methods like Region with CNN features (R-CNN) [44] and its related cousins Fast R-CNN [45] and Faster R-CNN [46], Spatial Pyramid Pooling (SPP-net) [28], Single Shot Detector (SSD) [47], etc.

Because, at present time, deep learning approach is showing the most promising results CNNs with emphasis on speed and accuracy were taken into consideration. Among diversity of CNN methods YOLO [48] [49] was picked as faster representative and Mask R-CNN [50] as more precise one. There was another method used for comparison also called Mixture of Gaussians (MOG) [51] to CNN methods but since it holds only an information about "objectness" score it couldn't be used on its own to achieve desired purpose.

Experiment was performed on a custom dataset which consists of indoor and outdoor handball sports footage during practice and competition. It contains 751 videos with 1920x1080 resolution at 30 frames per second, and the total duration of the recorded material is 1990 s. The scenes were captured using stationary GoPro cameras from different angles and in different lighting conditions. The cameras in indoor scenes were mounted at a height of around 3.5 m to the left or right side of the playground. Outdoor scenes have the camera at a height of 1.5 m. Depending on the players average height, location and the camera viewpoint the size of the player in the image ranges from 40 to 240 pixels.

Both YOLO and Mask R-CNN were applied using only the CPU on the same hardware inside separate virtual machines for most reliable comparison. Publicly available pre-trained models were used with their corresponding weights build on COCO dataset, with no additional training with our own dataset.

To perform tests a high-level neural networks API Keras was applied on top of an open-source machine learning framework Tensorflow with a use of Python programing language in Ubuntu Linux environment.

According to [49] YOLO, performs real-time object detection at 45 frames per second on a Titan X GPU and a fast version runs at more than 150 fps. Mask R-CNN in addition to bounding box provides also a segmentation mask on every pixel which desired object holds. This adds a slight computational overhead [50] but offers much more information about the body posture.

Speed test in [52] using only the CPU, took on average 18.47 seconds for Mask-RCNN to process a 1920x1080 RGB color video frame, while YOLO performed much faster, with 0.94 seconds per frame.

Detectors performance were compared with the ground truth and evaluated in terms of recall, precision and F1 score [53]. For MOG all detections were considered and for YOLO and Mask R-CNN only those whose confidence is greater than 85% to avoid a large number of false positives.

Condition for a detection to be considered as true positive more than half of the area object belongs to must be inside bounding box. The factors which have a great impact on the detector efficiency are size of an object and the percentage of occlusion.

B. *Detection of players and balls using Mask R-CNN, YOLO and MOG*

Fig. 9 shows results of detection in case of simple and complex scenario. A simple scenario includes fewer objects, up to 8, close to the camera. A complex scenario is when the number of objects on the scene is equal and greater than 9, away from the camera and with the occlusions.

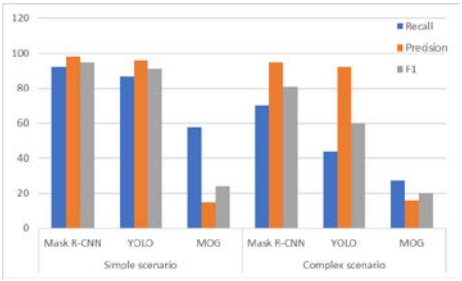

Figure 9. Evaluation results in simple and complex scenarios for Mask R-CNN, YOLO and MOG

Example indoor images of detection results, after they have been fine tuned for the best possible outcome, are presented in the Fig. 10. Bottom 3-image row shows results of detection using Mask R-CNN, middle row using YOLO and top one using MOG.

Analysis of first column shows that YOLO has difficulty detecting objects smaller than 50 pixels in height compared to Mask R-CNN but is more precise than MOG. An anomaly here is that YOLO managed to make a TP of a sitting person far back but fails to detect closer objects. MOG is more successful in detecting moving objects but since it lacks knowledge about object it is more influenced by noise due to often highly reflective playing field, light changing and shadows that players cast under artificial illumination. Since body parts of certain players move in different speed MOG detects them separately opposite to Mask R-CNN and YOLO. Also objects in a distance, small and not moving sufficiently are not detected by MOG.

YOLO detection in second row successfully detects less than half individuals count including coach dresses in blue. This is interesting since players further away are detected. It seems that YOLO have problems distinguishing floor color from coach's wear. MOG was more successful in this case but far best results were achieved using Mask R-CNN.

In the last row YOLO outperformed other methods. It even detected sports ball through the net. Mask R-CNN had a problem from the occlusion caused by the net and made a FP based on the ground reflection. The most unproductive method was MOG which detected only net segments moving and no objects of interest at all.

Fig. 11. shows results in a simple outdoor scene. The figure contains three players, with no occlusion, a ball and car partly visible but non important for the sport of interest. Object detectors have performed well, but background extractor resulted with few FPs and missed detections (FN).

All methods struggle with sports balls detection. Fig 12. Describes this behavior where Mask R-CNN was unable to detect the ball, while YOLO and MOG detected one out of two balls. Mask R-CNN however detected one more person at a distance than YOLO. It is important to notice that shadows which can be misdetected as real objects have not confused YOLO and Mask R-CNN.

Obtained results shows that Mask R-CNN is more appropriate in the footages of team sports due to its ability to successfully detect individual players even when they are inside a group and further away from the camera. An additional benefit Mask R-CNN provides is a mask around the detected object, which can be obtained with slightly more computation power. The advantage of YOLO method lays in speed performance allowing more time for testing and tuning in final model. Also it has proven to be sufficient and in a case of occlusion even better than Mask R-CNN. MOG was the fastest method but has proved to have too many FP in comparison with other two methods.

This is expected due to fact that MOG is a binary background foreground distinguisher working only on motion data.

The object all three methods have problem detecting is sports ball. Since ball detection is a vital aspect of the further research it was decided to improve performance of tested methods by adopting models to this unique object. MOG was taken out of the further experiment as the worst of the three and because it can't be used on its own to achieve additional improvements.

C. *Ball detection using custom trained Mask R-CNN and YOLO models*

To stay consistent with a previous experiment both YOLO and MASK R-CNN were trained and tested using CPU only on the same hardware inside same virtual environment (VMware) but on separate virtual machines with installed software as before for most reliable comparison.

Figure 10. Indoor sport detection results of MOG (upper row), YOLO (middle row) and Mask R-CNN (lower row)

Figure 12. Outdoor sport detection results of MOG (up), YOLO (middle) and Mask R-CNN (down)

In order to improve ball detection, models were trained using dataset specifically annotated for sports ball classification. Dataset consists of approximately 800 images divided in even ration on custom and public dataset.

Custom dataset was acquired out of the frames randomly selected from the video footages previously mentioned with resolution of 1920x1080 pixels and publicly available images sizing from 174x174 up to 5184x3456 pixels from various sources. Publicly available dataset is used to avoid overfitting.

Annotations were generated manually with both masks and square bounding boxes. Due to fact that ball objects vary in size, to achieve optimal results, rescaling of input images is needed. For this reason and for sake of equality CNN architecture was altered in a way to have comparable input size. YOLO has input image 1088x1088 and Mask R-CNN 1024x1024 RGB input size. Transfer learning [54] is applied to both methods which reduce training requirements. Weights trained on COCO dataset [55] are used to avoid training a model from scratch. COCO dataset consists of over 123 000 images including sports ball class, therefore the features usually found in images are already fused into trained weights. This way information learned in the experiment from the custom dataset is just add up on the top of the publicly available COCO weights.

In case of YOLO to avoid excessive speed variance tiny-yolo is used for training.

Training was performed through series of 5000 steps where weights at each hundredth step is saved to test its performance. One with the smallest loss was picked for final model. Batch size i.e. the number of samples that are passed through the network at one time was dynamically changed for better efficiency. First 2000 steps were trained with a lower value (YOLO: 2 and Mask R-CNN:1) and the second part with a higher value (up to 32). As with batch size, learning rates were also made variable by using higher learning rate at the beginning of training to more quickly descend to a local minimum and lower learning rate at the end to avoid

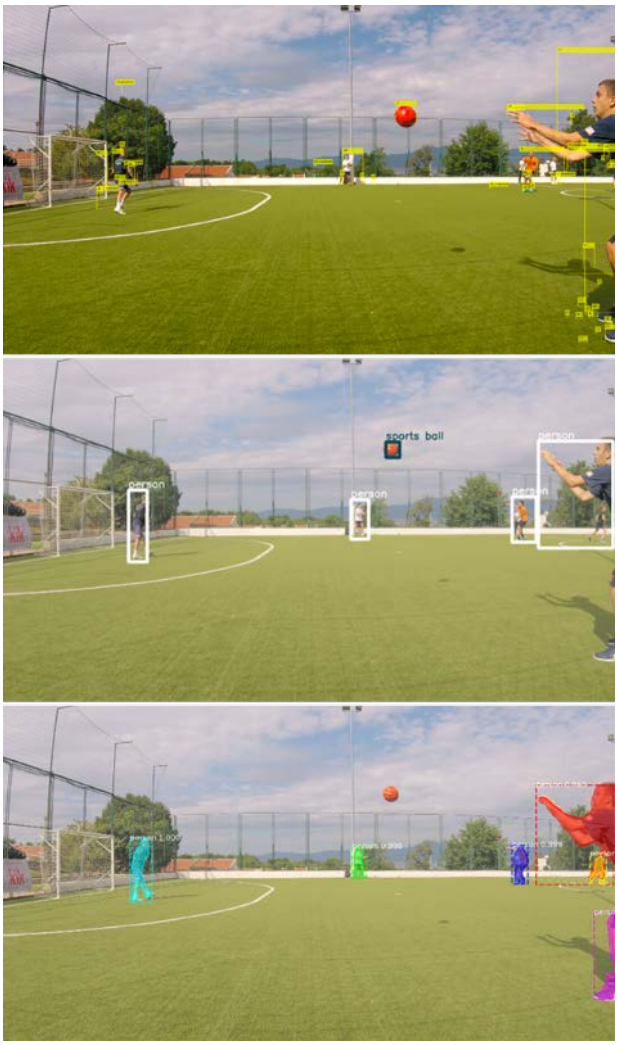

Figure 11.Sports ball detection results of MOG (up), YOLO (middle) and Mask R-CNN (down)

overshooting minimum loss. YOLO took approximately 97 hours and Mask R-CNN around 25 hours for a complete cycle of training. Performance of sports ball detectors is measured as before with a same threshold values of 85% for both methods.

Results are divided in 2 groups describing F1 score values for custom and publicly available model for each method as described in a Fig. 13. Custom trained models have better overall results thanks to improvement of detection on custom dataset. Recall values are higher and precision values are lower when models trained on a custom dataset are used as seen on Fig. 14.

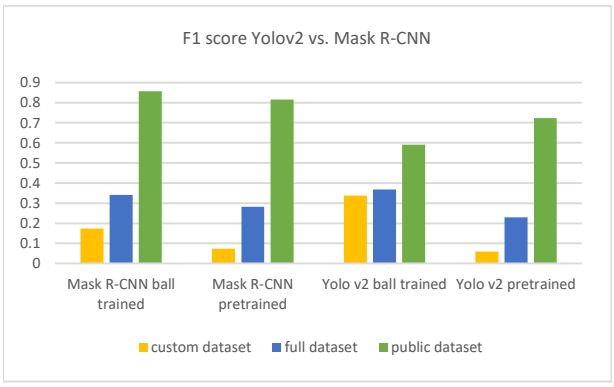

Figure 13. Score on custom and publicly available dataset for both methods on ball objects

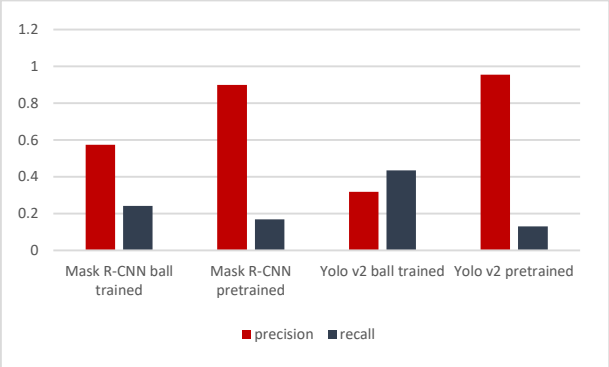

Figure 14. Precision and recall of trained and pretrained models on public and custom dataset combined

On this example image in Fig. 15 all models failed to detect ball close to the camera easily distinguished to a human observer. Pretrained models failed to detect all ball objects while custom trained models showed significant improvement by correctly detecting one of two closest balls. Compared to Mask R-CNN, YOLO has much higher number of FP but can handle further object with better effectiveness than before as described on Fig 16.

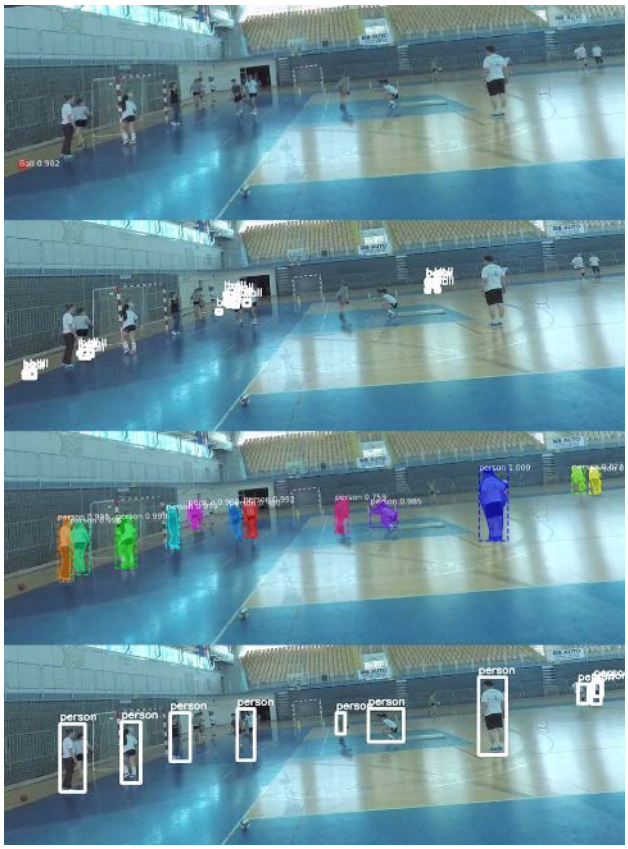

Figure 15. Indoor detection using (starting from the top) custom Mask R-CNN and YOLO following pretrained ones

Figure 16. Custom Mask R-CNN model unable to detect ball objects compared to YOLO

Some degradation in precision are noticed with models trained with only ball objects. Fig 17 is one of the examples. Object correctly detected as person using pretrained model is partly and falsely detected as ball with custom trained model.

Figure 17. Pretrained and custom trained Mask R-CNN model differences

In a case of a public dataset Mask R-CNN performs better and handles closer objects more successfully even in case they are heavily occluded.

Figure 18. Custom Mask R-CNN and YOLO on publicly available image

Detection speed decreased for YOLO model by 43% but still performs quicker than Mask R-CNN. Speed difference between Mask R-CNN models is in favor to custom trained model by 37%.

IV. DATASETS

Besides custom dataset obtained personally by authors of the [1] [52], public datasets are available for scientific research of action recognition. They include constantly growing datasets with additional information acquired using RGB-D sensors, accelerometers and position markers placed directly on an observed model, multiple sources, etc. Following datasets are good starting ground when no custom dataset is available.

Princeton Tracking Benchmark [56] datasets introduced in 2013 include real world footage of variety of actions performed by humans, pets and object presentations in form of RGB images with its accompanying depth. Along with 100 RGB-D tracking datasets comes tracking software. Annotations are for every frame in a form of bounding box around target object. Example is shown on Fig. 19.

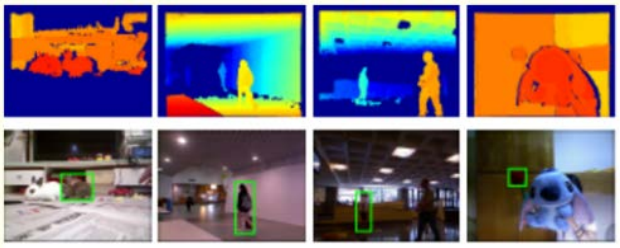

Figure 19. Princeton Tracking Benchmark

Cornell Activity Datasets: CAD-60 & CAD-120 [57] [3]. They come with 60 RGB-D and 120 RGB-D videos receptively. CAD-60 includes 2 male and 2 female persons in usual domestic environment (kitchen, bedroom, bathroom, and living room) performing 12 activities: rinsing mouth, brushing teeth, wearing contact lens, talking on the phone, drinking water, opening pill container, cooking (chopping), cooking (stirring), talking on couch, relaxing on couch, writing on whiteboard, working on computer, like in Fig. 20. CAD-120 consists of videos with same number of people in similar environment. Activities are divided into 10 high-level activities (making cereal, taking medicine, stacking objects, unstacking objects, microwaving food, picking objects, cleaning objects, taking food, arranging objects, having a meal) and 10 sub-activity labels (reaching, moving, pouring, eating, drinking, opening, placing, closing, scrubbing, null) with 12 object affordance labels (reachable, movable, pourable, pourto, containable, drinkable, openable, placeable, closable, scrubbable, scrubber, stationary). Skeleton joint position and orientation is labelled on each frame. RGBD data has resolution of 240 by 320. RGB is saved as three-channel 8-bit PNG file and depth is saved as single-channel 16-bit PNG file.

Figure 20. Cornell Activity Datasets: CAD-60 & CAD-120

Northwestern-UCLA Multiview Action 3D Dataset [58] contains RGB, depth and human skeleton data captured simultaneously by three Kinect cameras. This dataset includes 10 action categories: pick up with one hand, pick up with two hands, drop trash, walk around, sit down, stand up, donning, doffing, throw, carry (Fig. 21). Each action is performed by 10 actors in a library from a variety of viewpoints.

Figure 21. Northwestern-UCLA Multiview Action 3D Dataset

RGB-D People Dataset [59] was gathered by a three vertically mounted Kinect sensors on a tower at approximately 1.50 m height. It contains of 3000+ RGB-D frames acquired in a university hall and contains mostly upright walking and standing persons seen from different orientations and with different levels of occlusions, Fig. 22. Annotations are made in a form of a square box. Depth images are saved as 16 bits, 1 channel PGM images - 640 by 480. They contain the raw data content from the Kinect sensor. Namely, each pixel has value between [0, 1084]. RGB images are saved as 8 bits, 3 channels PPM images - 640 by 480. Dataset doesn't provide activity annotations but offers material for an art gallery research [60].

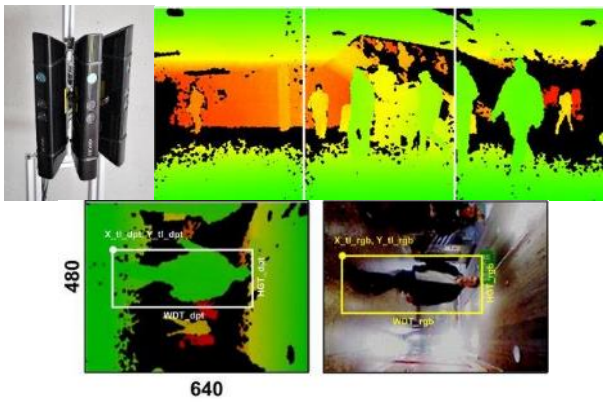

Figure 22. RGB-D People Dataset

UTD Multimodal Human Action Dataset (UTD-MHAD) [61] is a collection of videos using a Kinect sensor and a wearable inertial sensor in an indoor environment. The dataset contains of 27 actions performed by 4 females and 4 males with 4 times action repetition. The dataset includes 861 data sequences. Four data modalities of RGB videos, depth videos, skeleton joint positions, and the inertial sensor signals were recorded in three channels or threads (Fig. 23). One channel was used for simultaneous capture of depth videos and skeleton positions, one channel for RGB videos, and one channel for the inertial sensor signals (3-axis acceleration and 3-axis rotation signals). For data synchronization, a time stamp for each sample was recorded. The inertial sensor was worn on the subject's right wrist or the right thigh (see the figure below) depending on whether the action was mostly an arm or a leg type of action.

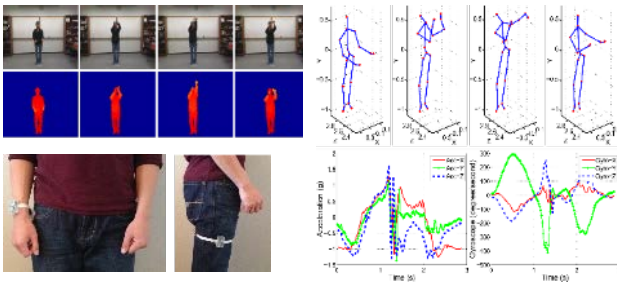

Figure 23. UTD Multimodal Human Action Dataset (UTD-MHAD)

Berkeley Multimodal Human Action Database (MHAD) [62] contains 11 actions performed by 12 subjects (7 male and 5 female) in the range 23-30 years of age with an exception of one elderly subject. All the subjects performed 5 repetitions of each action, coming to an about 660 action sequences (around 82 minutes of video). In addition, a T-pose for each subject was recorded which can be used for the skeleton extraction along with the background data (with and without the chair used in some of the activities). The specified set of actions comprises of the actions with movement in both upper and lower extremities, actions with high dynamics in upper extremities and actions with high dynamics in lower extremities. Each action was simultaneously captured by five different systems: optical motion capture system, four multi-view stereo vision camera arrays, two Microsoft Kinect cameras, six wireless accelerometers and four microphones (Fig. 24).

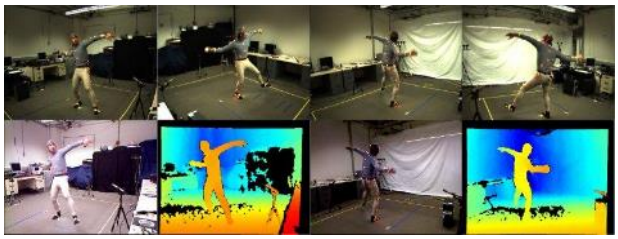

Figure 24. Berkeley Multimodal Human Action Database (MHAD)

Dataset of a human performing daily life activities in a scene with occlusions [63] consists of 12 RGB-D video sequences of a person moving in front of a Kinect in a scene with obstacles, Fig.25. In addition to the depth and RGB image, each sequence contains the synchronized ground truth data obtained from a Qualisys motion capture system with 8 infrared cameras. 3D representation of a human model is achieved by using 15 position markers: one for a head, neck and torso and 2 for shoulders, elbows, wrists, hips and knees.

Figure 25. Dataset of a human performing daily life activities in a scene with occlusions

Building custom dataset by recording video is very time consuming and it requires significant resources to collect desired footages however, using private dataset allows customization to adopt to a desired method.

During this process one of the things that needs consideration is number of capturing sources. If one camera is used depth perception and occlusion are problems which are hard to solve. Multiple cameras, on the other hand, concentrate on an object from different angles to give much more information. The position of cameras can be calculated like in [64] for an optimal solution. Multiple cameras can be set up two ways. Camera fields can overlap, which is more suitable for detail action examination, or side by side in sequence (art gallery), mostly used in surveillance mode. Art gallery, also, allows greater filed coverage with a same resource but shares the problem as one camera approach, only partial image is visible. Overlapping cameras provide info about the object from different perspectives but there is still an occlusion problem. This can be avoided by with approach described in [65] where different cameras are used sequentially which also require less samples and computational power.

With development in game industry affordable RGB-D sensors like Microsoft Kinect and Asus Xtion give depth based on two cameras and infrared spectrum sensor. Even though these sensors give more flexibility they are still inferior compared to marker based systems [64].

V. CONCLUSION

Thanks to constantly developing hardware and accompanying software solutions computer vision is becoming more and more similar to a human vision and even better in some cases. Recent approaches based on deep learning give another dimension of machines learning to distinguishing objects and actions with little or non prior knowledge which is expected to continue even faster with constantly bigger and bigger “Big Data”.

Overview presented in this paper gives a good foundation for experimenting with different approaches to extend computer vision even further. Approach used in this paper shows the possibility of implementing player object detection and even improving sports ball object detection, essential for further research of activity recognition in sport video, by taking published studies in combination with a custom dataset. It also describes difficulties and diversity of obtaining training datasets and what should be considered when one is doing so. There is still a lot of space for improvement but that is reasonable if taken in consideration how young this research field is. For more complex activity recognition and detection, it is advisable to use some sort of semantic models based on expert knowledge. This is also the next step author of this paper is intend to do.

ACKNOWLEDGMENT

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

REFERENCES

[1] M. Burić, M. Ivašić-Kos and M. Pobar, "An Overview of action recognition in vidos," *MIPRO Opatija*, pp. 1098-1103, 2017.

[2] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional," *Advances in neural information processing systems*, pp. 1097-1105, 2012.

[3] H. S. Koppula, R. Gupta and A. Saxena, "Learning human activities and object affordances from rgb-d videos," *The International Journal of Robotics Research*, pp. 951-970, 2013.

[4] I.-K. Marina, A. Iosifidis, A. Tefas and Pitas, "Person De-Identification in Activity Videos," *BiForD, Opatija, MIPRO*, pp. 75-80, 2014.

[5] L. Gorelick, M. Blank, E. Shechtman and M. Irani, "Actions as space-time shapes," *EEE transactions on pattern analysis and*, pp. 2247-2253, 2007.

[6] C. Schmid, R. Mohr and C. Bauckhage, "Evaluation of Interest Point Detectors," *International Journal of Computer Vision*, vol. 37, no. 2, p. 151, 2000.

[7] K. Rohr, "Recognizing Corners by Fitting Parametric Models," *Arbeitsbereich Kognitive Systeme, FB Informatik, Universitat Hamburg, International Journal of Computer Vision*, pp. 213-230, 1992.

[8] I. Lindeberg and T. Laptev, "Space-time interest points," *ICCV*, 2003.

[9] P. Dollar, V. Rabaud, G. Cottrell and S. Belongie, "Behavior recognition via sparse spatio-temporal features," *VS-PETS*, 2005.

[10] G. Willems, T. Tuytelaars and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," *ECCV*, 2008.

[11] H. Nakayama, T. Harada and Y. Kuniyoshi, "Dense Sampling Low-Level Statistics of Local Features," *CIVR*, 2009.

[12] D. Lowe, "Method and apparatus for identifying scale invariant features," *U.S. Patent 6,711,293*, 2004.

[13] B. Kanade and D. Lucas , "An iterative image registration technique with an application to stereo vision," *International Joint Conference on*, 1981.

[14] A. Klaser, M. Marszalek and C. Schmid, "A Spatio-temporal Descriptor Based on 3d-gradients," *British Machine Vision Conference*, 2008.

[15] P. Scovanner, S. Ali and M. Shah, "A 3-dimensional SIFT Descriptor and its Application to Action Recognition," *ACM Proceedings of the 15th International Conference on Multimedia*, p. 357–360, 2007.

[16] O. Liu and Z. Oreifej, "HON4D: Histogram of Oriented 4D Normals for Activity Recognition from Depth Sequences," *Conference on Computer Vision and Pattern Recognition*, 2013.

[17] F. Perronnin, J. S´anchez and T. Mensink, "Improving the Fisher Kernel for Large-Scale Image Classification," *Xerox Research Centre Europe (XRCE)*, 2010.

[18] A. Tefas and A. Iosifidis , "View-invariant action recognition based on Artificial Neural Networks," *Neural Networks and Learning Systems IEEE Transactions*, pp. 412-424, 2012.

[19] A. Karpathy and e. al., "Large-scale Video Classification with Convolutional Neural Networks," *Computer Vision and Pattern Recognition (CVPR)*, pp. 1725-1732, 2014.

[20] I. Laptev, M. Marszałek, C. Schmid and B. Rozenfe, "Learning realistic human actions from movies," *Computer Vision and Pattern CVPR 2008. IEEE Conference*, 2008.

[21] H. Wang, A. Klaser, C. Schmid and L. Cheng-Lin, "Action Recognition by Dense Trajectories," *CVPR 2011 - IEEE Conference on Computer Vision & Pattern Recognition*, pp. 3169-3176, 2011.

[22] Y. LeCun, B. Yoshua and H. Geoffrey, "Deep learning," *Nature 521.7553*, pp. 436-444, 2015.

[23] M. Oquab, L. Bottou, I. Laptev and J. Sivic, "Is object localization for free? - Weakly-supervised learning with convolutional neural networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 685–694, 2015.

[24] C. Szegedy, W. Liu and Y. Jia, "Going deeper with convolutions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 1–9, 2015.

[25] D. Scherer, A. Müller and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition,," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, p. 92–101, 2010.

[26] Y. L. Boureau, J. Ponce and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," *Proceedings of the ICML*, 2010.

[27] H. Gu and X. Wu, "Max-Pooling Dropout for Regularization of Convolutional Neural Networks," *Neural Information Processing, vol. 9489 of Lecture Notes in Computer Science*, p. 46–54, 2015.

[28] K. He, x. Zhang, S. Ren and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," *Computer Vision – ECCV*, p. 346–361, 2014.

[29] W. Ouyang, X. Wang and Z. Zeng, "DeepID-Net: Deformable deep convolutional neural networks for object detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 2403–2412, 2015.

[30] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*, p. 1097–1105, 2012.

[31] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, p. 580–587, 2014.

[32] S. Ji, W. Xu, M. Yang and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, pp. 221-231, 2013.

[33] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *Advances in neural information processing systems*, pp. 568-576, 2014.

[34] G. Lavee, E. Rivlin and M. Rudzsky, "Understanding Video Events: A Survey of Methods for Automatic Interpretation of Semantic Occurrences in Video," *Technion - Computer Science Department - Tehnical Report CIS-2009-06*, 2009.

[35] V. T. Vu, F. Bremond and M. Thonnat, "Automatic video interpretation: A recognition algorithm for temporal scenarios based on pre-compiled scenario models," *nternational Conference on Computer Vision Systems*, p. 523–533, 2003.

[36] A. Borzin, E. Rivlin and M. Rudzsky, "Surveillance interpretation using Generalized Stochastic Petri Nets," *The International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, 2007.

[37] A. Davis and J. Bobick, "The recognition of human movement using temporal templates," *IEEE Transactions on Pattern Analysis and Machine*, vol. 23, no. 3, p. 257–267, 2001.

[38] M. S. Aggarwal and J. K. Ryoo, "Semantic Representation and Recognition of Continued and Recursive Human Activities," *Springer Science+Business Media, Int J Comput Vis*, pp. 1-24, 2009.

[39] M. Ivašić-Kos, I. Ipšić and S. Ribarić, "A knowledge-based multi-layered image annotation system," *Expert systems with applications. 42*, pp. 9539-9553, 2015.

[40] M. Ivašić-Kos, M. Pobar and S. Ribarić, "Two-tier image annotation model based on a multi-label classifier and fuzzy-knowledge representation scheme," *Pattern recognition*, vol. 52, pp. 287-305, 2016.

[41] P. A. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features," *CVPR*, no. 1, pp. 511-518, 2001.

[42] D. Navneet and B. Triggs, "Histograms of Oriented Gradients for human detection," *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE,*, 2005.

[43] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks For Large-Scale Image Recognition," *arXiv:1409.1556.*, 2014.

[44] J. Girshick, T. Donahue, J. Darrell and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," *IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH*, pp. 580-587, 2014.

[45] R. Girshick, "Fast r-cnn," *Proceedings of the IEEE international conference on computer vision*, 2015.

[46] R. Shaoqing and et al, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, 2015.

[47] W. Liu and et al., "Ssd: Single shot multibox detector," *European conference on computer vision. Springer, Cham*, 2016.

[48] J. Farhadi, J and A. Redmon, "YOLO9000: Better, Faster, Stronger," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI*, pp. 6517-6525, 2017.

[49] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV*, pp. 779-788, 2016.

[50] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," *2017 IEEE International Conference on Computer Vision (ICCV), Venice*, pp. 2980-2988, 2017.

[51] C. Grimson, C. Stauffer and E. L. W, "Adaptive background mixture models for real-time tracking," *1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Fort Collins, CO*, vol. 2, pp. 246-252, 1999.

[52] M. Burić, M. Pobar and M. Ivašić-Kos, "Object Detection in Sports Videos," *41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2018.

[53] M. Ivašić-Kos and M. Pobar, "Multi-label Classification of Movie Posters into Genres with Rakel Ensemble Method," *Artificial Intelligence XXXIV. SGAI 2017. Lecture Notes in Computer Science, vol 10630; Chambridge : Springer*, pp. 370-383, 2017.

[54] D. Cook, K. D. Feuz and N. C. Krishnan, "Transfer learning for activity recognition: a survey," *Springer London*, 2013.

[55] T. Lin and et al, "Microsoft COCO: common objects in context," *European conference on computer vision, Springer, Cham*, pp. 740-755, 2014.

[56] S. Song and J. Xiao, "Tracking Revisited using RGBD Camera: Unified Benchmark and Baselines," *Princeton University Proceedings of 14th IEEE International Conference on Computer Vision (ICCV2013)* , 2013.

[57] J. Sung, C. Ponce, B. Selman and A. Saxena, "Unstructured human activity detection from rgbd images," *Robotics and Automation (ICRA), 2012 IEEE International Conference on. IEEE*, 2012.

[58] J. Wang and N. B. Xiaohan, "UCLA Datasets," Department of Electrical Engineering and Computer Science Northwestern University, [Online]. Available: http://users.eecs.northwestern.edu/~jwa368/my_data.html. [Accessed 28 09 2018].

[59] L. Spinello and K. O. Arras, "People Detection in RGB-D Data," *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[60] F. Faion, S. Friedberger, A. Zea and U. D. Hanebeck, "Intelligent Sensor-Scheduling for Multi-Kinect-Tracking," *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference*, 2012.

[61] C. Chen, R. Jafari and N. Kehtarnavaz, "UTD-MHAD: A Multimodal Dataset for Human Action Recognition Utilizing a Depth Camera and a Wearable Inertial Sensor," *Proceedings of IEEE International Conference on Image Processing, Canada,* 2015.

[62] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal and R. Bajcsy, "Berkeley MHAD: A Comprehensive Multimodal Human Action Database," *In Proceedings of the IEEE Workshop on Applications on Computer Vision (WACV),* 2013.

[63] A. Dib and F. Charpillet, "Pose Estimation For A Partially Observable Human Body From RGB-D Cameras," *Proc. of the International Conference on Intelligent Robot Systems (IROS),* 2015.

[64] E. J. Almazan and G. A. Jones, "Tracking People across Multiple Non-Overlapping RGB-D Sensors," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 831-837, 2013.

[65] S. Han, M. Achar, S. Lee and F. Peña-Mora, "Empirical assessment of a RGB-D sensor on motion capture and action recognition for construction worker monitoring," *SpringerOpen Journal, Visualization in Engineering,* 2013.