Object Detection Using Synthesized Data

Matija Burić[1[0000-0003-3528-7550]], Goran Paulin[2[0000-0002-6885-5393]],

and Marina Ivašić-Kos [3[0000-0002-1940-5089]]

[1] Hrvatska elektroprivreda d.d., SIT Rijeka, Kumičićeva 13, Rijeka, Croatia
[2] Kreativni odjel d.o.o. Rijeka, Croatia
[3] Department of Informatics, University of Rijeka, Rijeka, Croatia
matija.buric@hep.hr, gp@kreativni.hr, marinai@uniri.hr

Abstract. Successful object detection, using CNN, requires lots of well-annotated training data which is currently not available for action recognition in handball domain.

Augmenting real-world image dataset with synthesized images is not a novel approach, but the effectiveness of the creation of such a dataset and the quantities of generated images required to improve the detection can be.

Starting with relatively small training dataset, by combining traditional 3D modeling with proceduralism and optimizing generator-annotator pipeline to keep rendering and annotating time under 3 FPS, we achieved 3x better detection results, using YOLO, while only tripling the training dataset.

Keywords: Object Detection, Convolutional Neural Network, YOLO, Synthesized Data, Sports, Handball.

1 Introduction

The conventional approach of the majority of published methods for object detection using neural networks requires prepared training dataset to provide successful results. These data are usually acquired from sources similar to those on which the model will finally be applied. In case of supervised learning (Fig. 1), authors are usually limited to the publicly available labeled datasets, or are forced to obtain data on their own.

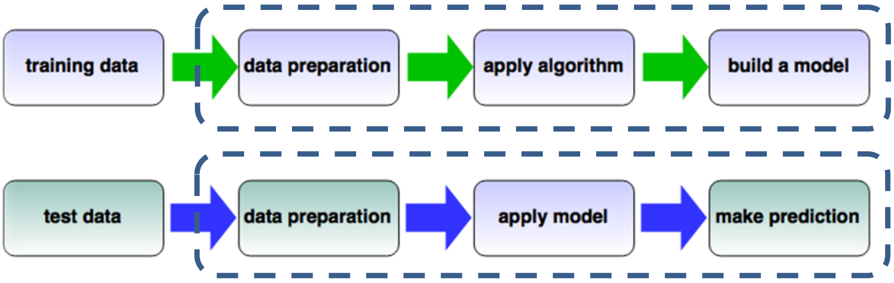

Fig. 1. The traditional supervised learning model

Publicly available datasets like MS COCO [1], PASCAL VOC [2], ImageNet [3], etc. provide an extensive set of images along with the way for extracting labels in the form suitable for a researcher. However, if someone needs a specific set of data, like in the case of handball sport, the possibilities are limited to a small number of images that are not labeled at all or are not labeled in the desired way – they can miss class or classes could be mixed, which can result in adjusting annotations or discarding them completely. An additional challenge when preparing annotations is a possible need for expert knowledge from the researched domain, which can significantly affect the cost of the whole project. Labeling of still images in object detection is usually done by drawing a bounding box around the desired object or by marking every pixel this object occupies in the image as shown in Fig 2. Information about every marked object in the image is saved in the individual file which minimally consists of object position in the image, object class, and an image filename.

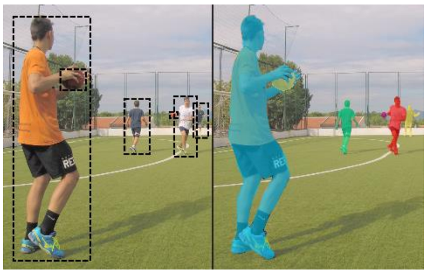

Fig. 2. Bounding box vs. instance segmentation on players and ball objects [4]

Since providing mentioned information is rather hard for a human without graphical reference, researchers have been releasing many GUI tools which makes labeling more intuitive and speed up the process of manual annotating. Tools used in this research are LabelImg [5] which provides a bounding box for YOLO format and VGG Image Annotator (VIA) [6] which provides polygon around an object for use with instance segmentation, Fig. 3. Annotations can be saved in different formats but mostly accepted are XML and JSON.

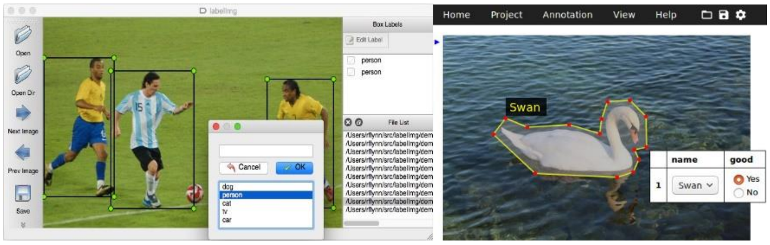

Fig. 3. LabelImg [5] and VGG Image Annotator (VIA) [6] labeling tools

Due to limitations that public datasets offer, many researchers acquire datasets themselves [7] or by a 3rd party which consumes many resources but will provide custom dataset specially designed for the project. Since the data preparation can easily exceed the time needed for building a successful model, the question arises: could this be improved somehow? One should also take into account that the bigger dataset will result in a more successful model [8].

To increase the real data set and consequently improve the classification results, researchers have already tried to use synthesized images. In [9], using coarse 3D models of drones and aircrafts, with optimized rendering pipeline, authors have showed that using only 12 real images combined with 1500 or more synthetic images can significantly improve the performances of an image classifier.

However, in sports scenes, we are dealing with non-rigid human bodies, their posing and actions, which requires a great diversity of synthetic images to produce a usable synthetic learning data set. Pose space coverage and texture diversity proved to be the key ingredients for the effectiveness of synthetic data training in [10]. The authors generated about 5M synthetic images to train human 3D pose estimation model.

For the same purpose, more than 6M synthetic images were generated in [10] for human depth and part segmentation in real images. The authors have shown that the best results are achieved when synthetic network is fine-tuned with real images.

In [11] a parametric generative model of human action videos was introduced, relying on physics, scene composition rules, and procedural animation techniques. Approximately 6M images at 3.6 FPS was generated and combined with small datasets of real images. It was showed that combining a large set of synthetic images with small real datasets can boost recognition performance.

In the scope of this research, we will address the problem of detecting an athlete in handball scenes for which large datasets are not available. We will start with a relatively small dataset, and enlarge it with generated synthesized images to increase the training set needed to train the model for object detection and to evaluate its impact or contribution to model performance. YOLO is selected as the dedicated object detector mainly for its speed and the fact that previous researches [4, 8, 12], which are base for comparison, were made using the same YOLO methods.

The rest of the paper is organized as follows: in Section II. a synthesized data generation is presented. The experiment, comparison of the performance of YOLO object detector on custom and synthesized dataset and discussion are given in Section III. The paper ends with a conclusion and the proposal for future research.

2 Synthesized data

Our task was to train the YOLO object detector on handball scenes where players and balls are objects of interest. The recordings were captured during handball exercise in the gym with artificial illumination from the top of the field, and the sunlight from the side windows. The camera position was fixed to a few different spots covering from ¼ to a ¾ of a field. The persons in scenes are mainly young players performing some handball techniques and actions such as jump-shot, dribble and defense.

To generate the synthesized data, it was necessary to analyze the contents of the reference video scenes and to select the elements to be reproduced on the virtual scene.

The content is split into three groups: a) the environment (the sports hall), b) the player(s) and c) accessories (the ball).

Since the experiment required a single 3D environment, it was created with the traditional box modeling techniques, using Autodesk 3ds Max [13], and then textured using both Adobe Photoshop [14], for manual texture preparation, and Allegorithmic Substance Designer [15], for the procedural ones.

To generate the player a procedural modeling/texturing route was chosen to enable the generation of many virtual player variations by varying body shape, sex, hair, clothing, shoes, and accessories. Another challenge was preparing a virtual character for animation. Both problems were solved by using Adobe Fuse [16] and Mixamo [17], producing auto-rigged, textured 3D models, ready for animation.

Not being able to acquire a library of handball player's motion capture data, the virtual character was manually animated in 3ds Max (Fig. 4), adding inverse kinematics on top of the Mixamo's animation rig, resulting in 123 frames of animation. Frames from the video were used as animation reference.

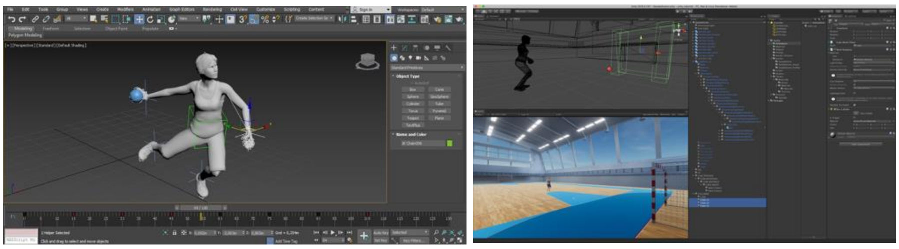

Fig. 4. Handmade animation in 3ds Max and virtual scene assembling in Unity

For the ball animation, a combination of static parenting of the ball in the player's hand and the physical simulation of shooting it, using Nvidia PhysX in Unity [18], were used.

The virtual scene was assembled in Unity, using aforementioned 3D objects, and lit with a single area light, simulating interior lighting, and the virtual Sun, Fig. 4.

The camera rig was built to allow changing the camera position and orientation from the code, reframing the scene after a single animation sequence is played. Camera rig's rotational pivot was placed in the center between the player and the goal. The camera is positioned 8 meters away from the rotational pivot, at the height of 1.5 meters, to match camera placement from the reference video.

To produce 984 synthesized images, the camera rig is rotated by 45 degrees every 123 frames, rendering 8 different views.

Along with beauty pass rendering (photorealistic PNG image), pixel-perfect masks, representing the player and the ball positions, were generated and stored in the red (the ball) and the green (the player) channels of a PNG image, Fig. 5.

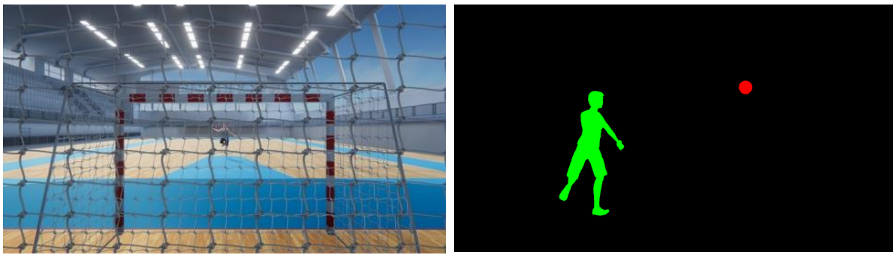

Fig. 5. Rendered beauty pass (photorealistic image) and the mask (from a different frame)

Average times for rendering both photorealistic image and the accompanying mask, using SSD for storage, are 178 ms and 84 ms, respectively.

After rendering, the rendered mask sequence is processed using Python, calculating bounding boxes for the green and the red channels, and writing them in XML files, using PASCAL VOC format. Average time for creating a single annotation is 40 ms.

3 Experiment

3.1 Datasets

For training models, two types of datasets were used. Images input size in both cases is 1920x1080 px (full HD) resolution.

The first one (real dataset) was obtained via GoPro cameras at heights of 1.5 m and 3.5 m at the border of the field during one week in handball school without adjusting the scene to preserve real-world conditions. The videos were captured mostly in the indoor hall with artificial lighting, but part of the footage was collected outdoors during clear weather. The ball and players' clothing are of a different color, not following two team rule. The persons in scenes are mainly young players with some spectators, and there are many ball occurrences in the single shot. Dataset has 418 images which are divided roughly at 70:30 training: test ratio.

The second synthesized dataset is computer generated to reassemble real-world conditions as close as possible. Point of view is from different positions, but 1.5 m camera height is preserved. It consists of images of one-player-one-ball at different stages of handball jump-shot action. The player is described as a young female player, dressed in team colors, with the ball. Dataset has 984 images divided in the same ratio between train and test part as a real dataset.

3.2 Models

YOLO (You Only Look Once) [19] object detector is used for training and testing the performance of synthesized and real data. It is a specially designed neural network that first divides the input image into a grid of smaller segments, and at the same time produces the probability distribution of the object classes and several candidate boundary boxes with the associated reliability for each segment. If there is no object inside the bounding box, the value of confidence will result in zero opposite to a case where

YOLO will take into account intersection-over-union to determine confidence level. In case when object spans over two or more segments, YOLO will elect the center cell to be the prediction candidate for an object. YOLO achieves satisfactory accuracy, although not at the level of methods such as Mask R-CNN [20, 21] – the advantage is that it is significantly faster [12].

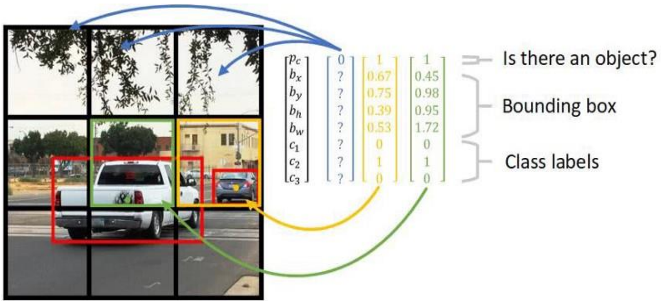

Fig. 6. The way YOLO presents object detection data [22]

The problem that arises according to research results of [4, 12] is that pre-trained YOLO shows insufficient results of ball detection in handball sport which can be improved by training model on more handball sports-related data. Also, YOLO, along with other tested methods in [4] have difficulty detecting all smaller objects or objects far away from the camera. For this reason, based on research in [8] configuration for training a new model was adopted to train only two classes – player and ball object, with an input images size of 1024x1024 px. These settings lower detection speed but partly handles mentioned problem. Since computer hardware is generally improving with every passing day, it is reasonable to believe this will have little impact in the long run.

For the sake of the comparison of the results of training with synthesized and real data, three models were trained. In order to optimize the training process, transfer learning was performed. This way, training basic features is avoided [23].

To optimize the training process variable learning rate was applied. In the first epoch learning rate (LR) was set to a step of 0.001 and later was reduced to a step of 0.00001. If LR is too small converging to a minimum would require significant computation and, consequently, a lot of time, opposite to a too big LR where convergence is faster, but it is much harder to reach a minimum. Too high LR is likely to overshoot and too low LR is likely to get stuck at a local minimum, unable to converge to a global minimum. Optimized LR combines the speed of higher LR at the beginning and precision of a smaller LR at the later time during a process (Fig 7).

Further training is improved by variable batch size where multiple samples are passed through the network at once. Since input image size is increased to 1024x1024 px, higher batch size requires a lot of memory and therefore maximum batch size applied in the experiment was set to the value of 8.

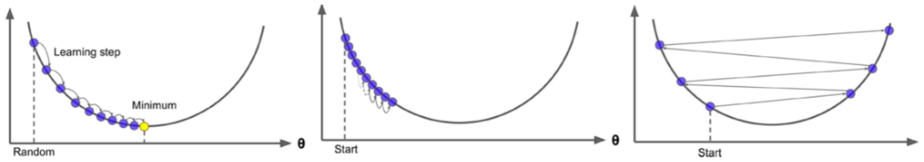

Fig. 7. From left to right: optimized, too small and too big LR. Too small LR is slow and takes a lot of resources. Too big LR, in this case, moves further away from the minimum.

The first reference model (Y-R) was trained using real images for approximately 20 epochs and is used as the base point for comparison.

The second model (Y-S) was trained using only synthesized data for 20 epochs.

The third model (Y-RS) was trained with both real and synthesized data, altogether 987 images, slightly less than 20 epochs.

Y-R and Y-RS models were unstable during training. This is most likely because input images contain objects that are small in size, occupying only a fraction of an image. Since the described method rescales those images during preprocessing, annotated bounding boxes become too small, which sometimes result in an error. Such behavior was manifested in [7].

3.3 Environment and metric

The model was trained using 12 core E5-2680v3 CPU server with one GeForce GTX TITAN X 12GB memory GPU. YOLOv2 was implemented on Debian Linux operating system with some additional programming using Python language.

Testing was performed on a different server using only 12 core E5-2680v3 CPU.

Training and testing speed were not significantly different from the expected train/test per image ratio.

Performance is measured using Average Precision (AP) [24], and the detection for each model is compared to the ground truth considering that the Intersection Over Union (IoU) should be equal to or greater than 50%. For reference, observe Fig 8.

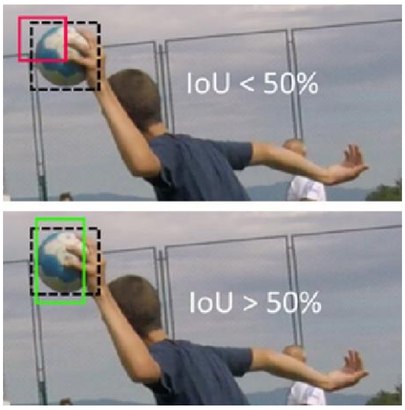

Fig. 8. Visual description of Intersection Over Union (IoU) [8]

To avoid multiple detections of the same object, True Positive (TP) detection is considered only if an object was not detected yet. The precision-recall curve is calculated for the person and ball individually. AP is finally computed as an occupying area underneath the curve. Example in Fig. 9. shows AP of a person detection using Y-RS trained on both synthesized and real datasets and tested on a real dataset.

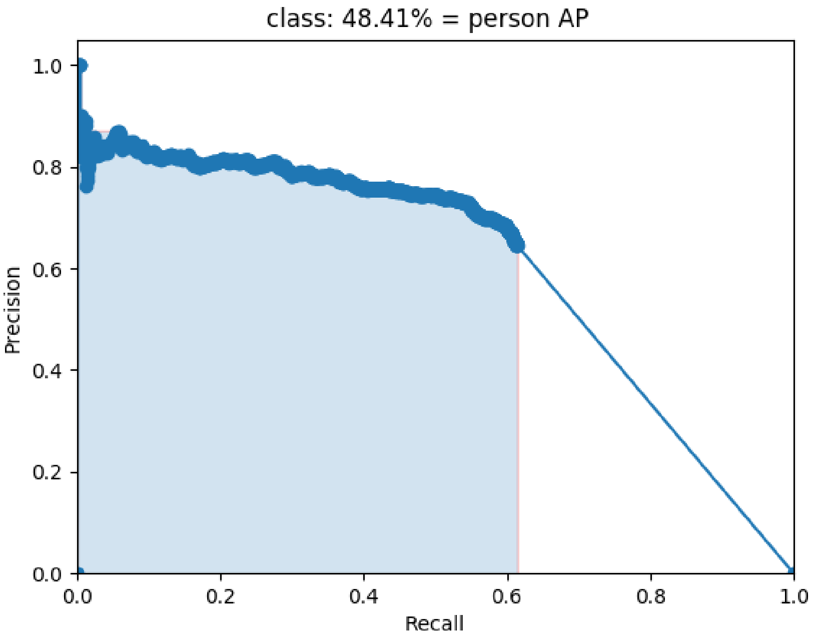

Fig. 9. The precision-recall curve for person detection with Y-RS model where AP is occupying area underneath the curve

3.4 Testing

Table 1. shows the results of tested models using mean average precision (mAP) and AP for individual class on real and synthesized test sets. The higher percentage means better results.

The object of our interest was to get the best possible results of the object detector for a player and a ball in a set of real scenes. Since players resemble greatly to individuals in the audience, the authors decided to treat all person object detections as positives whether player or spectator is detected. Due to a variety of players clothing at this point of research, it is hard even for humans to distinguish spectator from player based solely on the appearance so it was decided to disregard FP made by spectators pretending to be players and treat all player-like objects as players.

The experiment has shown that the model Y-R, trained only on the real dataset, has the worst detection results (8.49% mAP) compared to other models trained in the experiment. Real ball objects were poorly detected compared to the average human observation and there were many false positives of player objects.

Training the model on synthesized data (Y-S) achieved the 41.16% mAP on a synthesized test set. This was expected since training samples resemble test set more closely then real datasets. Results on the synthesized test set were mainly used to validate efficiency and ability to perform detection in the artificial dataset. The results on a real set, which was far more interesting for our task, were significantly worse (9.76% mAP). It is important to note that the synthesized data set contained only one player handling the ball. Also, the sports jersey of the player and the color of the ball were the same in all scenes, which leads to overfitting the model and problems with real-world detection. With small variations, such as color change of the athlete's clothing, the model would probably improve.

Combining the synthetic data with real data proved to be a good idea, and model Y-RS trained on both synthesized and real images achieved the 25.75% mAP which is the best detection results on a real data set. The precision of ball object more than doubled and the precision of player object roughly tripled. Considering that authors had only a limited number of real samples at disposal, mAP was significantly improved, in fact, tripled, when roughly additional 1000 synthesized images were included in the model.

Table 1. Class AP and mAP of tested models on the real and synthesized test set

Model	Test set	Ball AP	Player AP	mAP
Y-R	real	1.39 %	15.59 %	8.49 %
Y-S	synthesized	6.36 %	75.95 %	41.16 %
Y-S	real	1.14 %	18.37 %	9.76 %
Y-RS	real	**3.05 %**	**48.41 %**	**25.73 %**

Below are some typical examples of handball scenes and the results of the player and the ball detection for trained models. Detection results of trained models are analyzed on the same image examples to enable an accurate comparison of the performance of the model. Fig. 10 shows detection results in the form of green squares around the object which describe true positive (TP), red squares describe false positive (FP) and blue describe Ground Truth (GT).

Y-R handles detection of players which are not affected by occlusion with more ease than Y-S. Still, it has problems detecting small objects along with person sitting far left. When combined in Y-RS, results of detection improve slightly like in the case where a coach is detected even though it is behind one of the players, but it generates more FP like a detected ball in the right part of the image.

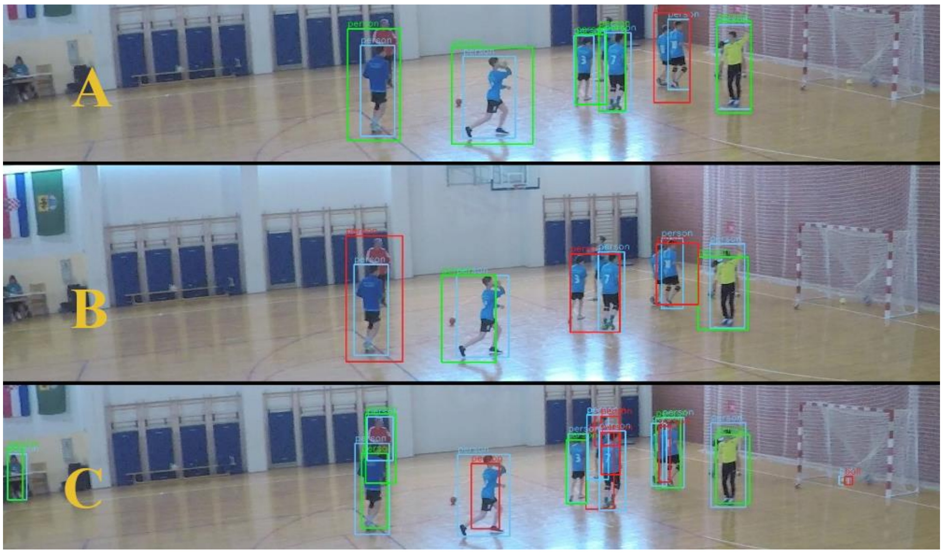

Fig. 10. Performance of Y-R (A), Y-S (B) and Y-RS (C) model top to bottom

When model Y-S is tested on synthesized dataset similar to the one it was trained on, overall results are much better as seen in Fig. 11, however, one should take into account that our goal is to train a model that will be able to detect players and balls in real handball scenes successfully.

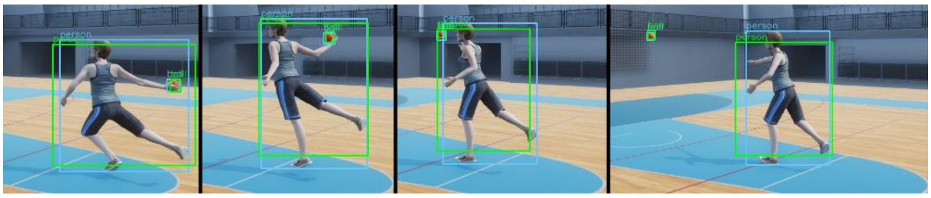

Fig. 11. Time lapse of the Y-S detection of synthesized data

The example in Fig. 12 emphasizes the benefit of training on both training sets since the top and middle parts of the image have similar results where Y-R manages to find one player behind the glass partly obstacle and Y-S manages to find another close by. When combined, in Y-RS, both players are detected successfully plus one further back behind them. Also notable are partial detections of players behind the net. Compared to Fig. 10, in Fig. 12 action ball and static ball on the right are partly detected even though humans would have more problem distinguish action ball from the background noise in this case.

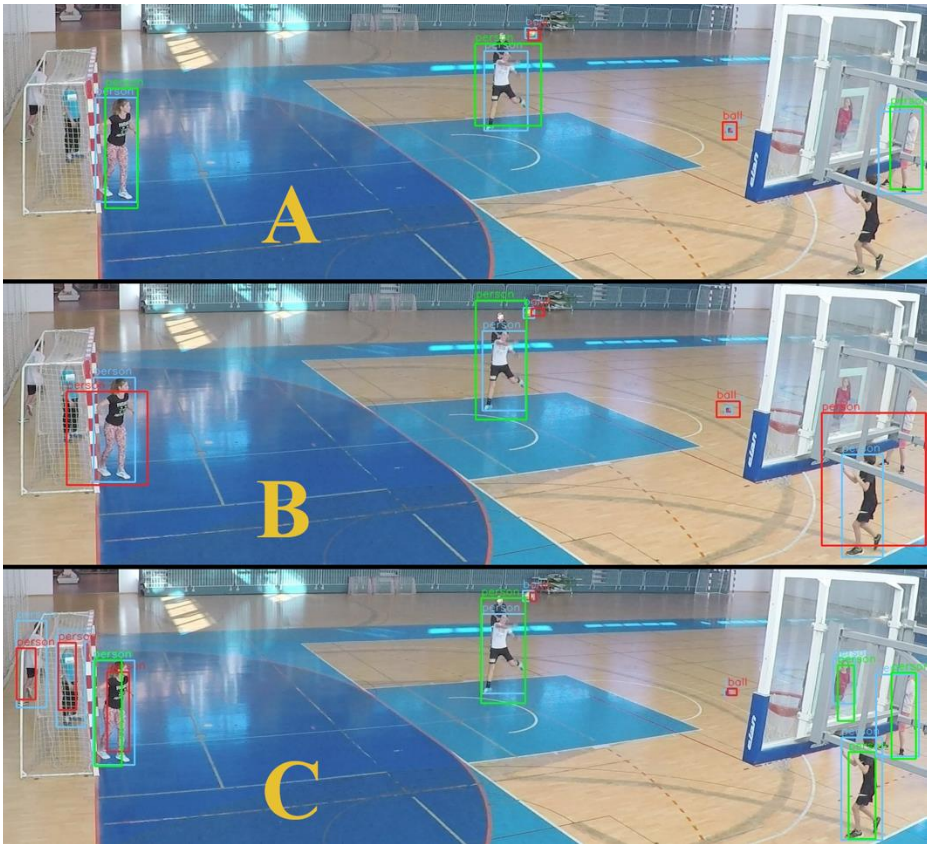

Fig. 12. Similar results of Y-R (top - A) and Y-S (middle - B) and improvement of detection when Y-RS (bottom – C) is used

The last example in Fig. 13 shows the improvement of both Y-S and Y-RS over Y-R method where players nearby were not detected at all or have been partly detected. Another observation is that the ball object detected using Y-S method is very similar by color and texture to the ball used in training the specific model. It can be compared to the ball object of resulting detection in Fig 11.

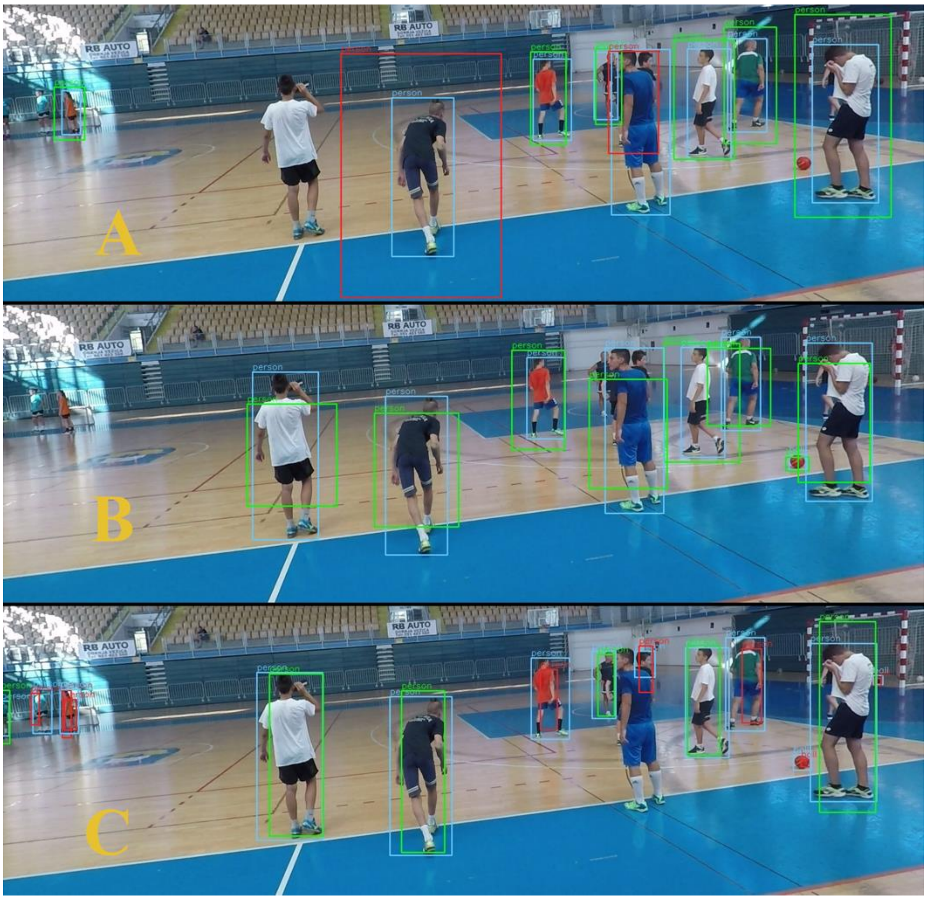

Fig. 13. Example of improved result with Y-S (A) over Y-R (B) and even more improved using Y-RS (C)

Ball far back in the right part of the image was also detected considering its size and partial visibility.

4 Conclusion

In some cases, when a set of training data is not large enough, or does not exist, the synthesized data can provide an appropriate alternative that, besides the ability to collect data and increase the set of images, also allows for different scenes variations with respect to camera position, framing, brightness, and the like.

In this paper, we used a training set for YOLO object detector that consisted of real and synthesized handball scenes. The functionality of three YOLO models trained on real (Y-R), synthesized (Y-S), and mixed training set (Y-RS) were compared on the real test set. When only one type of data was used for training the detection model, with a relatively small dataset, quite poor detection results were obtained (Y-S and Y-R model). Y-R model shows to be more successful with overlapping objects but fails to detect obvious ones, probably because its missing more training data. Y-S model manages to detect usual player postures and balls with distinguished edge but seems to be overfitted. It misses detection of some obvious objects. Significantly better results were achieved when both synthesized and real data were used for training, as in the Y-RS model. The objects are detected more precisely with greater IoU and object which are neither recognized with Y-R or Y-S, whether they are too small or occluded, gets detected. It confirms that a larger number of images in a training set will result in a more efficient object detector even when part of the set is synthesized.

Also, the synthesized images used in this experiment were significantly simplified real-world simulation, with only one player, always in the same outfit, and a single ball, featuring only different camera views. That means that, at this point, less data is available from ten synthesized image than from one real image containing ten players. On the other hand, this could help in distinguishing between players and audience, because players during game usually are dresses to team colors, different than spectators. It should be noted that in this research all human objects were treated as players due to fact that our real dataset consists from casually dressed players from which some are active on the field and some are static as audience waiting their turn during competition and practice. Regardless of this, real and synthetized dataset contributed to improving the detection results of the player and ball in a realistic scenario.

Similar models could be applied in other sports as well with smaller adjustment. Major problems which arise in the handball sport is also present in other sports like occlusion, small/distant objects and to this point there is no universal solution. Some can be handled with even higher resolution of input images, different point of view or by use of multiple view sources but this extends out of the scope of this research.

When taken into account previous observations and the fact that with slightly more effort at the beginning of building synthesized dataset one can accumulate a much large number of images for training, results are looking promising for use in future research.

Additional action which could improve results and it is considered for future research is to apply a newer version of YOLO. The newer YOLOv3 was reported to be more accurate in detecting smaller objects which can be beneficial in handball sport detection.

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 18-222-1385.

References

1. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 740–755 (2014).
2. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. 88, 303–338 (2010).
3. Jia Deng, Wei Dong, Socher, R., Li-Jia Li, Kai Li, Li Fei-Fei: ImageNet: A large-scale hierarchical image database. In: researchgate.net. pp. 248–255 (2009).
4. Buric, M., Pobar, M., Ivasic-kos, M.: Ball detection using Yolo and Mask R-CNN. 5th Annu. Conf. Comput. Sci. Comput. Intell. (2018).
5. GitHub LabelImg, https://github.com/tzutalin/labelImg, last accessed 2019/07/29.
6. Dutta, A., Zisserman, A.: The VIA Annotation Software for Images, Audio and Video. Proceedings of the 27th ACM International Conference on Multimedia. (2019).
7. Ivasic-Kos, M., Pobar, M.: Building a labeled dataset for recognition of handball actions using mask R-CNN and STIPS. 7th European Workshop on Visual Information Processing (EUVIP). IEEE. (2018).
8. Burić, M., Pobar, M., Ivašić-Kos, M.: Adapting YOLO Network for Ball and Player Detection. ICPRAM -. 845–851 (2019).
9. Chen, W., Wang, H., Li, Y., Su, H., Z.W.: Synthesizing training images for boosting human 3d pose estimation. Fourth International Conference on 3D Vision (3DV). (2016).
10. Souza, R., Gaidon, A., Cabon, Y., López, A.: Procedural generation of videos to train deep action recognition networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017).
11. Varol, G., Romero, J.: Learning from synthetic humans. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 109-117).(2017)
12. Buric, M., Pobar, M., Ivasic-Kos, M.: Object detection in sports videos. 2018 41st Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2018 - Proc. 1034–1039 (2018).
13. 3ds Max 2019 Help, http://help.autodesk.com/view/3DSMAX/2019/ENU, last accessed 2019/07/29.
14. Photoshop User Guide, https://helpx.adobe.com/photoshop/user-guide.html, last accessed 2019/07/29.
15. Substance Designer User Guide - Substance Designer, https://docs.substance3d.com/sddoc/substance-designer-user-guide-102400008.html, last accessed 2019/07/29.
16. Adobe Fuse CC (Beta) help topics, https://helpx.adobe.com/beta/fuse/topics.html, last accessed 2019/07/29.
17. Animate 3D characters with Mixamo, https://helpx.adobe.com/creative-cloud/help/animate-characters-mixamo.html, last accessed 2019/07/29.
18. Unity - Manual: Unity User Manual (2018.3), https://docs.unity3d.com/2018.4/Documentation/Manual/, last accessed 2019/07/29.
19. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. Proceedings of the IEEE conference on computer vision and pattern recognition. (2017).

20. He, K., Gkioxari, G., Dollar, P.: Mask r-cnn. Proceedings of the IEEE international conference on computer vision. (2017).
21. Pobar, M., Ivašić-Kos, M.: Mask R-CNN and Optical Flow Based Method for Detection and Marking of Handball Actions. 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI). (2018).
22. AI zone, Understanding Object Detection Using YOLO, https://dzone.com/articles/understanding-object-detection-using-yolo, last accessed 2019/07/29.
23. Cook, D., Feuz, K.D., Krishnan, N.C.: Transfer learning for activity recognition: a survey. Knowl. Inf. Syst. 36, 537–556 (2013).
24. Ivasic-Kos, M., Ipsic, I., Ribarić, S.: A knowledge-based multi-layered image annotation system. Expert systems with applications 42.24. (2015).