Real-Time Object Detection in Occluded Environment with Background Cluttering Effects Using Deep Learning

Syed Muhammad Aamir*[1], Hongbin Ma [*2], Malak Abid Ali Khan*[3] and Muhammad Aaqib[*4]

[*1,2,3] School of Automation, Beijing Institute of Technology, Beijing 100081, China
E-mail: 3820181051@bit.edu.cn
E-mail: mathmhb@bit.edu.cn
E-mail: 3820202079@bit.edu.cn
[*4]Department of Mechatronics Engineering, University of Engineering and Technology Peshawar, Pakistan
E-mail: enggraaqib@gmail.com

Abstract. Detection of small, undetermined moving objects or objects in an occluded environment with a cluttered background is the main problem of computer vision. This greatly affects the detection accuracy of deep learning models. To overcome these problems, we concentrate on deep learning models for real-time detection of cars and tanks in an occluded environment with a cluttered background employing SSD and YOLO algorithms and improved precision of detection and reduce problems facing by these models. The developed method makes the custom dataset and employs a preprocessing technique to clean the noisy dataset. For training the developed model we apply the data augmentation technique to balance and diversify the data. We fine-tuned, trained, and evaluated these models on the established dataset by applying these techniques and highlight the results we get more accurately than without applying these techniques. The accuracy and frame per second of the SSD-Mobilenet v2 model are higher than YOLO V3 and YOLO V4. Furthermore, to employ various techniques like data enhancement, noise reduction, parameter optimization, and model fusion we improve the effectiveness of detection and recognition, we also make a graphical user interface system for the developed model with features of object counting, alerts, status, resolution, and frame per second. Subsequently, to justify the importance of the developed method analysis of YOLO V3, V4 and SSD were incorporated. Which resulted in the overall completion of the proposed method.

Keywords: Object Detection and Recognition; Data Augmentation; YOLO (You only look once); Mobilenet-SSD V2; Graphical User Interface.

1. INTRODUCTION

The techniques employed for identification and locate the object in an image are known as the object detection tasks and image understanding is a fundamental area of computer vision therefore Image classification is not enough to understand an image. Object detection and object recognition are two main terms that are used interchangeably in the area of computer vision. Object detection is a key research area of computer vision, that attracted many industries such as augmented reality [1] self-driving cars and UAVs, remote sensing [2, 3], ecology investigation [4], and medical science [5]. The key focus of the developed study is deep learning-based object detection and after a detailed literature review we could broadly divide deep learning-based object detection frameworks into three types which are region proposals (based on region selection) includes R-CNN, Fast R-CNN, and Faster R-CNN, YOLO (You Only Look Once), and SSD (Single Shot MutiBox Detector). R-CNN, Fast R-CNN, and Faster R-CNN work in two steps. In the first step they extract the features based on region selection and in the next step, it computes the convolutional neural network and gives the final detection results. Whereas YOLO (You Only Look Once) and SSD (Single Shot Multibox Detector) [6] are single-step models, they take the input image and compute the convolutional neural network directly in a single step, and gives the final detection results. The single-step models like YOLO and SSD are very fast but their accuracy lags when compared to the two-step models like R-CNN, Fast R-CNN, and Faster R-CNN. These models give very good accuracy for detection but with less frame per second. Hence, we could not use these models for real-time detection separately. YOLO and SSD work better in real-time detection. The main findings of this manuscript are to categorize the different frameworks in various ways for determining the developed improvements, we immense study various hurdles while facing the exact detection of moving objects in an occluded environment and lastly, we built a graphical user interface to show the live detection of cars and add different features to the system.

This manuscript is organized as follows; section 2 shows the related work followed by the proposed methodology in section 3. Experimentation and results are presented in section 4 and 5, and finally, section 6 conclude this manuscript.

2. RELATED WORKS

Before Deep Learning, the task of object detection would be done through various steps like edge detection and extraction of features from images. Histogram of oriented gradients (HOG) [7] is one of the widely used feature extractor techniques in computer vision and image processing applications, the primary concept behind working of the (HOG) descriptor is that the target object's appearance and shape inside an image could be represented by the distribution of intensity gradients and the direction of edges. One of the preliminary neural networks-based object detection algorithms is the sliding window algorithm [8]. In the object detection algorithm, not only want to know whether the object is in the image but also wants to know where the object is in the image. The sliding window is one of the earliest object detection algorithms which could detect and locate an object in the image with better accuracy. Many applications need real-time object detection [9], for which the sliding window algorithm is not appropriate. In 2016, an algorithm [10] YOLO (You Only Look Once) is much faster than a sliding window algorithm because YOLO allows the image through the convolutional neural network only once as its name suggests, and similarly, YOLO V2 [11] is an upgraded version of YOLO V1 because YOLO V1 had some limitations, which YOLO V2 tried to improve. The convolutional neural architecture used for feature extraction in YOLO V2 is darknet-19, which is a custom architecture and uses an additional 11 layers for object detection. First, YOLO V1 could only predict two bounding boxes for each grid cell however, if there are more than two objects whose center lies in the same grid cell, then YOLO will only be able to predict bounding boxes for two objects. YOLO also struggles to detect small bounding boxes in the image as it's a very deep convolutional neural network and can't keep information of small objects because of the max-pooling layers in the network which reduces the receptive field of the network. Some of the object detectors [12] which take less memory of processing power can't be used in UAVs because their detection accuracy is low and they also fail to detect smaller objects. For this purpose, a new network called Slim-YOLO V3 [13] was proposed and it performs two steps over YOLO V3 to make it lighter and faster. First, it does L1 regularization on the layers to decrease sparsity in the channel, secondly, it does prunes of less informative feature channels which makes the network lighter and reduces the number of parameters, and also decreases the number of trainable parameters and floating-point operations in the network as compared to the YOLO V3 network. This results in a much lighter network and two times faster than the YOLO V3 network, with a slight decrease in inaccuracy. Ren et al [14], developed a further region proposal network because RPN is a key computation for enhancing the efficiency of the model. The already established method acts in a close way of free of cost to share full-image convolutional features with network detection. They achieved with FCN and have the ability to identify the bound of objects and rate each state respectively. A single-shot multi-box object detector (SSD) [15] is one of the earliest single shot detectors. Single-shot detection means that the objects were detected and localized on one forward pass. SSD works in two steps, first, it uses a VGG neural network for feature extraction to creates feature maps for the objects. After creating feature maps, it does convolutional filtering on the feature maps for object detection. A Multi-box detector is a regression technique for creating bounding boxes, SSD is much similar to YOLO algorithms the only difference comes when both predict the bounding box information. Wang et al [16] execute to perform an approach based on adversarial network and spatial dropout network for the accomplishment of deformation and occlusion characteristics and conclude a small change among some of the classes which would be needed for better object detection. In the method presented in [17], to enhanced some of the regional proposal networks of Faster R CNN and set a proper channel to identify the smaller region employing a single feature map of high level, in addition, they also improve the accomplishment of small object detection. According to [18], an offline deep neural network was employed for the detection of different tasks and an action-driven technique for tracking. However, Bae et al [19], suggested a technique of learning-based detection and tracking to discriminate model of appearance while employing deep appearance for learning of huge training datasets for a solid and reliable connection among the detection and tracking.

3. PROPOSED METHOD

A method is presented for real-time detection of cars and tanks in an occluded environment with a cluttered background employing SSD and YOLO techniques. The proposed method is fully automatic, real-time, and makes incorporates various techniques to achieve the objective of this research. The method performs data construction, the input is then pre-processed and augment data to increase the samples of training. Lastly, we trained YOLO V3 and YOLO V4 models for real-time object detection. The following figure 1 shows the overall process of the proposed system.

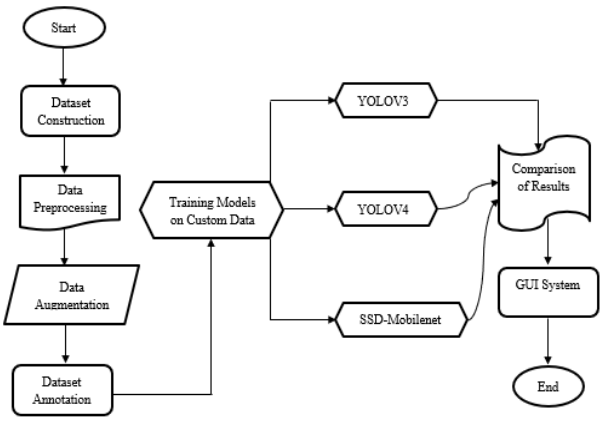

Fig. 1 Block Diagram of Proposed System

3.1. Data Preprocessing and Augmentation

Data preprocessing has a significant impact on the performance of a supervised learning model due to which unpredictable samples are most probably lead to erroneous output. The data may be noisy and influenced by several unpredictable external factors. Therefore, the preprocessing technique is employed to avoid such types of issues, i.e., we preprocess the data and delete the uncleaned and wrongly annotated data after a deep analysis of the dataset. By cleaning the unwanted noisy data from our dataset and resizing our images in ordered form and enhancing the contrast we get proper data for training the model. While the data augmentation technique is used for enlarging the existing dataset, the data are crucial and it's very difficult to get enough training data to achieve close to 100% accuracy on the prediction. In our dataset, there was not enough diversity due to which it gives unsatisfactory results when the model was trained on this dataset. By producing diversity in the dataset, we tried some data augmentation techniques like cropping, resizing, padding etc.

3.2. Dataset Construction

We had two types of objects in our image data-set, one type of data set had cars and the other had tanks in the images, these frames are extracted from the video sequences. The car's video is taken from the side angle of a traffic signal; therefore, the videos are small as they were recorded from some distance and all the cars shown in the images are from the side angle. The tank video is taken in an open field and only one tank in the video is roaming around. As the images are taken from the video and mostly, they have the same type of cars and tanks. Therefore, in the case of the tank dataset, we have low diversity and had the bounding box information for the objects saved in text file format. For each image, its bounding box information is saved in a separate text file with the same name as the image, the bounding box information includes the center point of the height and width of the bounding box then the information is normalized by the height and width of the image. Bounding box normalization is done so that there is not a very big difference in the loss of small and large bounding boxes. Dataset construction is the most challenging part of computer vision. Therefore, to construct a formal dataset, we set a camera far away from the road at an angle that the objects look small, occluded, and with a cluttering effect of the background and take frames of the video and collect around 60000 images of cars and tanks from those frames.

4. EXPERIMENTATION

The experimentation phase for object detection in an occluded environment was trained on deep learning models due to high accuracy, the SSD-Mobile Net V2 model was chosen for experimentation purposes. In this research work, the model has first fine-tuned three versions of object detection algorithm YOLO V2, YOLO V3, and Slim-YOLO V3 without data augmentation and data preprocessing. During training, the proposed model achieved unsatisfactory results however, the model properly detects, tracks, and recognize the objects in images and videos. The accuracy of YOLO V2, YOLO V3, and Slim-YOLO V3 was 25%, 60.4%, and 9.1% respectively. These three object detection frameworks are one of the best object detection algorithms however, it gives unsatisfactory results on the custom dataset due to without employing of data augmentation and preprocessing techniques. While employed the data preprocessing method the proposed model achieved pretty results for the algorithm of YOLO V3, YOLO V4, and SSD-MobileNetv2.

5. RESULTS

A series of experimentation were employed in an occluded environment of object detection with background cluttering effects. In order to exhibit the generalization, this study was implemented employing python with anaconda-based TensorFlow. The custom dataset was used which was achieved from a video by taking its frames and provided labels to objects is the requirement of the model and then fine-tuned three object detection frameworks and trained them on the custom dataset for two classes i.e., tanks and cars. As the dataset has not had enough diversity and the objects were occluded with background cluttering effects in an uncontrolled outdoor environment therefore, some pre-processing and data augmentation steps were done for better results. We used the transfer learning method for SSD-Mobilenetv2 and fine-tuning method for training the different versions of YOLO. These versions of YOLO and SSD are among the best object detection algorithms. Then trained YOLO V2, YOLO V3, and Slim-YOLO V3 models on the custom dataset without preprocessing and data augmentation but without applying these techniques we achieved unsatisfactory results. In this work, we evaluated and compared the results of these two versions of YOLO and SSD-Mobilenet-v2 on the dataset and detect cars and tanks in real-time in an occluded environment with background cluttering effects.

5.1. Implementing YOLO V3

We utilized the darknet implementation for fine-tuning YOLO V3 on the custom dataset. The algorithm can detect objects in images at three stages to detect small, medium, and large objects. Then modified the configuration file of YOLO V3 to detect two classes, set the learning rate to 0.001 for the stability of losses, and keep the batch size of 8 so that batch will have enough examples to get different features of the images. Similarly, tested the model for detections after every 100 epochs and saved the best model. As presented in Fig.2 the error training loss and evaluation graph of YOLO V3 and the number of training epochs was kept at 20k with an initial learning rate (LR) of 0.001. The LR has been

approaching 0.1 times of initial LR at 16k and 18k epochs respectively.

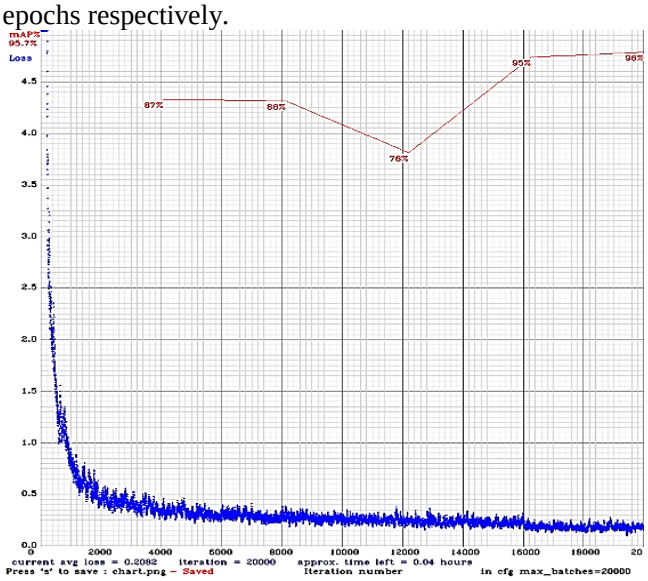

Fig. 2 Training Graph of YOLO V3

The following Fig.3 are the real-time detection results of the model YOLO V3 in an occluded environment with a background cluttering effect which is trained on the custom dataset. For training the model we employ preprocessing to clean the dataset and then data augmentation to produce diversity in the dataset and after that fine-tune the model according to respective classes and get pretty results while testing the data. To test the capabilities of the proposed model the mean average precision (mAP) of 82.6% was achieved.

Fig. 3 Real-Time Detection in Occluded Scenario and Cluttered Background

To justify the effectiveness of the proposed experimentation a comparative analysis was taken into account without employing preprocessing and data augmentation and it represents the mAp of YOLO V2, YOLO V3, and Slim-YOLO V3 results in fig.4. Similarly, the experimentation was done with the help of preprocessing to remove the unwanted data and data augmentation for generating diversity in the dataset. Subsequently, the proposed model was again trained on YOLO and SSD-Mobilenet-V2 and achieved more accurate and better results than the models applied without the steps of preprocessing and data augmentation techniques.

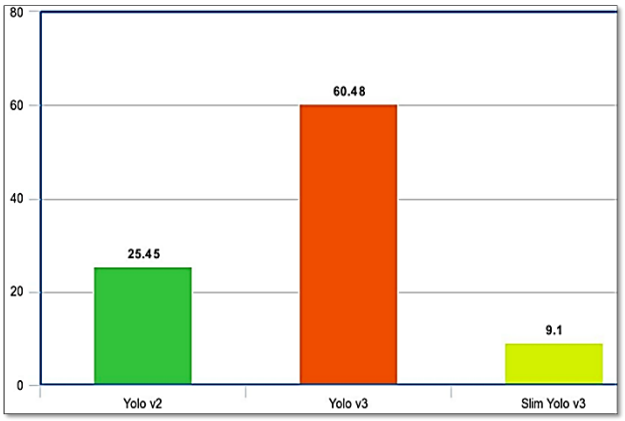

Fig. 4 Map of YOLO Without Preprocessing and Augmentation

Similarly, in Fig.5 the detection results of the YOLO V3 model in which black car is occluded and some part of the car is covered by the white car and some part is covered by the tree and wall and it's not detected successfully but the white car is detected with a low score. While in the condition of the tank, it is also detected but the detection rate is low which is not acceptable in the case of real-time detection.

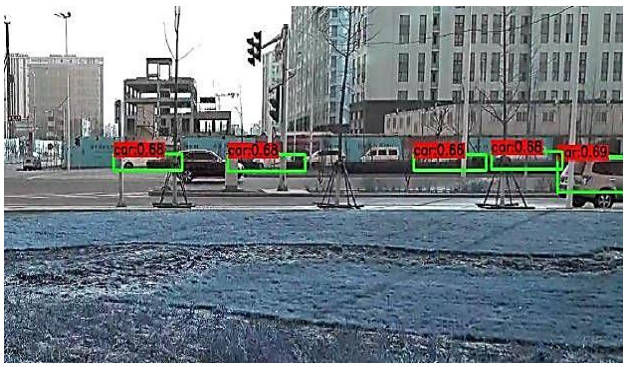

Fig. 5 YOLOV3 Detection Results Without Preprocessing and Augmentation

5.2. YOLO V4 Results

We employed the YOLO V4 model and trained with the same learning rate. However, network response was different because of its core structures are different. We used CSPDarknet53 as a backbone, SPP, and PAN models as neck and YOLO-v3 as ahead. LR has been reduced to .1 times at 16k and 18k. After 18k iterations, the model accuracy is not improving further. If two objects have the same color and are merged or close to each other then it's more difficult for a model to detect and recognized the objects in such a condition therefore, the proposed model would detect and classify the data with good scores in an occluded environment. It is depicted in fig. 6 that some cars are very closed with each other and a lot of factors have been seemed in front of the car and covered some portion of the car but still, it detected and classified the car accurately.

Fig. 6 Real-Time Detection of Cars in Occluded Environment

Further computations were performed to get the values from the confusion matrix and the threshold value of 0.25 true positive is 1734 and false negative is 201, the average intersection over union is 67.21%. Similarly, the intersection over the union threshold is 75% used under the curve for each unique recall. Mean average precision (mAp @ 0.75) is equal to 0.823662. After the evaluation of the SSD-Mobilenet v2 model, the proposed technique achieved 73.92% average precision of cars at a value of 50 for intersection over union threshold and 100% average precision for tanks at a threshold value of 50 IOU. The mean average precision we get at 0.5 IOU (intersection over union) is 0.866121 which is equal to 86.6%. After evaluation of the model, the classification loss and localization loss were 0.79 and 0.39 respectively. Here in the following Fig.7 is the detection results of our model which we trained on our custom dataset of cars and tanks after preprocessing and data augmentation for better detection results Here we could notice the cars are parked very near with each other and even on the naked eye we can't see the full image of every car, as some parts of the cars can be seen but still the model detects the cars more efficiently with better accuracy rate.

Fig. 7 Detection Result of SSD-MOBILE NET-V2 in Occluded Environment

In the following figure some part of the white car is covered by the black car which is known as occluded but still the algorithm is efficient to detect all the vehicles successfully. We get these results after applying to preprocess and data augmentation techniques. We test our models with both custom and standard datasets. After applying data augmentation and preprocessing we fine-tune the algorithm and set the best parameters for training the model and also the addition of extra features to the algorithm like counting, class id, recording, database, and status. The cars in the following figure are mostly covered by trees, traffic light poles, walls, and some parts of cars are covered by the same color cars which are known as occluded vehicles.

Fig. 8 Results of Improved Algorithm

A summary of multiple models is shown in the following table. Different networks and frameworks were utilized to get better rates of performance. SSD, YOLO V4, and YOLO V3 have 86%, 82%, and 78% mAp respectively.

Table 1 Different models result from summary

Network	**Map**	**FPS**	**Network**
SSD–Mobile net V2	86.6%	104	SSD – Mobile net V2
YOLO V4	82.2	58	YOLO V4
YOLO V3	78	63	YOLO V3

5.3. GUI RESULTS

It is depicted in Fig. 9, that the graphical user interface shows the live detection of cars because of adding different functionalities and features to the system such as it shows the frame per second of the video as well as the resolution and give us the counting results which is one of the best parts of experimentation and have different controls like start, stop, start recording, and stop recording.

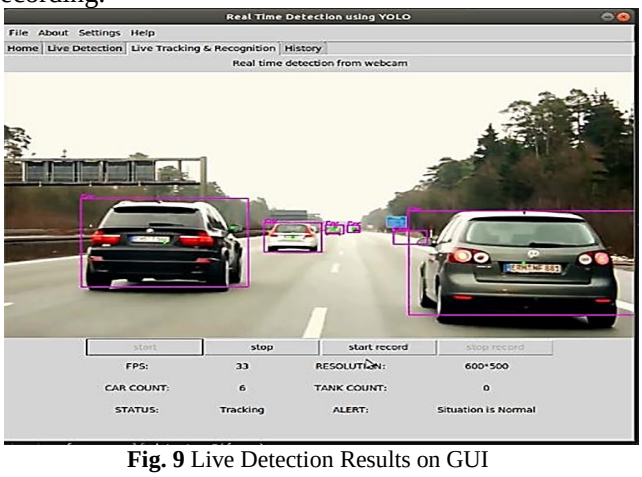

Fig. 9 Live Detection Results on GUI

6. CONCLUSION AND FUTURE WORK

This study involved multiple algorithms and hence fine-tuned YOLO V2, YOLO V3, and Slim-YOLO V3 on a custom dataset without employing the technique of data preprocessing and data augmentation and achieved unsatisfactory results. Similarly, performing data augmentation and preprocessing techniques and fine-tune YOLO V3, YOLO V4 models, additionally used the method of transfer learning for Mobilenet-SSD and trained these models on the custom dataset to achieved good accuracy with satisfactory frame per second for real-time detection. We diversify the dataset for better training to get good accuracy of detection while the objects are small, occluded, and also with background cluttering effect. After applying the proposed model, we get 78% mAp with 58 FPS on the YOLO V3 model, 82.3 % mAp with 63 FPS on the YOLO V4 model, and 86.6% mAp with 104 FPS on SSD-Mobilenet V2. Therefore, the developed SSD-Mobilenet v2 is more accurate and robust in this scenario. The training was performed on different configurations of YOLO and was tested on the test split of the data, then the results were compared with the proposed and already established method. Among these models, SSD-Mobilenet v2 gives better accuracy with more robustness, which is 86.6 mAp with 104 frames per second. For real-time detection and tracking, we build a graphical user interface system that shows different functionalities and features like real-time detection, real-time tracking, and recognition, car count, tanks count, alerts, frame per second, resolution, status, and record. In the future, we would focus on developing the developed algorithm for faster, accurate, and robust detection. We would be able to change the base of the model to develop a new model according to our need and could add more features to the graphical user interface system which would be used in more applications to solve real-world problems.

Acknowledgements

This work was partially funded by the National Key Research and Development Plan of China (No. 2018AAA0101000) and the National Natural Science Foundation of China under grant 62076028.

REFERENCES:

[1] Shahrokni, L. Vacchetti, V. Lepetit, P. Fua. Polyhedral Object Detection and Pose Estimation for Augmented Reality Applications. In Proceedings of the Computer Animation 2002, Geneva, Switzerland, 21–21 June 2002; pp. 65–69.

[2] J. Han, D. Zhang, G. Cheng, L. Guo, J. Ren. Object Detection in Optical Remote Sensing Images Based on Weakly Supervised Learning and High-Level Feature Learning. IEEE Transaction on Geoscience and Remote Sensing, vol. 53, no.6, pp. 3325-337, June 2015.

[3] Tang, S. Zhou, Z. Deng, H. Zou, L. Lei. Vehicle Detection in Aerial Images Based on Region Convolutional Neural Networks and Hard Negative Example Mining. Sensors 2017, 17, 336.

[4] A. Takeki, T. T. Trinh, R. Yoshihashi, R. Kawakami, M. Iida, T. Naemura. Combining deep features for object detection at various scales: finding small birds in landscape images. IPSJ Transaction on Computer Vision and Applications 8, 5 (2016).

[5] B. Solaiman, B. Burdsall, C. Roux. Hough Transform and Uncertainty Handling. Application to Circular Object Detection in Ultrasound Medical Images. In Proceedings of the 1998 International Conference on Image Processing, Chicago, IL, USA, 7–7 October 1998; pp. 828–831.

[6] Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, A. C. Berg. SSD: Single Shot Multi-Box Detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.

[7] N. Dalal and B. Triggs. Histogram of Oriented Gradients for Human Detection. In Computer Vision and Pattern Recognition, 2005. CVRP 2005. IEEE Computer Society Conference 2005: 886-893.

[8] Zhang, Y. Zhong, and X Li. Slim-YOLO V3: Narrower, Faster and Better for Real-Time UAV Applications. In Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW), Seoul, Korea (South), 2019, pp. 37-45.

[9] M. Li, B. Kim, A. Mourikis. Real-Time Motion Tracking on a Cellphone Using Inertial Sensing and a Rolling-Shutter Camera. Proceedings-IEEE International Conference on Robotics and Automation. 4712-4719.

[10] J. Redmon, S. Divvala, R. Girshick, A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.

[11] J. Redmon, and A. Farhadi. YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Honolulu, 2017 pp. 7263-7271.

[12] S. Belongie, J. Malik, and J. Puzicha. Shape Context: A new descriptor for shape matching and object recognition. In Advances in neural information processing systems, pp. 831-837. 2001.

[13] J. Han, D. Zhang, G. Cheng, L. Guo, J. Ren. Object Detection in Optical Remote Sensing Images Based on Weakly Supervised Learning and High-Level Feature Learning. IEEE Transaction on Geoscience and Remote Sensing, vol. 53, no.6, pp. 3325-337, June 2015.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-CNN: Towards real-time object detection with region proposal networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, 2017.

[15] Tang, S. Zhou, Z. Deng, H. Zou, L. Lei. Vehicle Detection in Aerial Images Based on Region Convolutional Neural Networks and Hard Negative Example Mining. Sensors 2017, 17, 336

[16] Wang, X.; Shrivastava, Gupta; A-fast-RCNN: Hard positive generation via adversary for object detection. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2606–2615

[17] Ren, Y.; Zhu, C.; Xiao, S. Small object detection in optical remote sensing images via modified faster R-CNN. Appl. Sci. 2018, 8, 813

[18] Yun, S.; Choi, J.; Yoo, Y.; Yun, K.; Choi, J.Y. Action-Driven Visual Object Tracking with Deep Reinforcement Learning. IEEE Trans. Neural Net. Learn. Syst. 2018, 29, 2239–2252.

[19] Bae, S.H.; Yoon, K.J. Confidence-Based Data Association Deep Appearance Learning for Robust Multi-Object Tracking. IEEE Trans. Pattern Anal. Mach. Intell. 2018, 40, 595–610.