Available online at www.jrors.ir

# Journal Of Radar and Optical Remote Sensing

JRORS 4 (2019) 49–57

# Detection, Recognition and Tracking Cars from UAV Based implementation of MobileNet- Single Shot Detection deep neural network on the embedded system By Using Remote Sensing Techniques

Mohammad Javad Nateghi*[a]

[a] *MEng of Electronic, Imam Hossein Comprehensive University, Tehran, Iran*

Received 11 December 2019; revised 14 February 2020; accepted 10 March 2020

**Abstract**

Tracking targets from the ground is difficult due to natural and artificial barriers, and in some cases, such as car detection, is dangerous, therefore, identifying targets using remote sensing is obvious. To achieve the purpose, the desired camera is installed on the unmanned aerial vehicle (UAV). with images processing on captured images from the camera, the system has used can identify the vehicle using aerial images and follow it if it is necessary. An important issue to this matter is the accuracy of the target detection. Therefore, efficient algorithms should be used in this field, and efforts have been made to use a deep neural network in this regard because it has the best performance rather than other methods. But using this network itself will cause other problems that are especially noticeable in real-time applications of the identification system. Because this type of neural network needs a lot of time to process information. Solving this problem will using strong hardware as much as possible, but these systems cannot be installed on the UAV due to their high weight and large power consumption. For this reason, in this paper, have tried to use pre-processing methods to identify possible moving targets and illuminate other parts of images to reduce the volume of data to make processing easier, and then the system can identify and track the car with the Light MobileNet-SSD network. This method is 25 times faster than other fast methods such as yolov3, and its loss rate is 0.02.

***Keywords**:* tracking targets, car detection, UAV, deep neural network, remote sensing.

* Corresponding author
*Email address:* kpnateghi@ihu.ac.ir

## 1. Introduction

Most of the time for having a real-time surveillance system puts cameras in the place of human eyes but using artificial intelligence, the entire surveillance task as automatically as possible. The tracking of moving objects from frame to frame in real-time video sequences captured by the moving camera is a difficult task. The challenges come from complex object motion, non-rigid object, dim target, the proximity of colors and intensities, partial occlusion, change the target angle, illumination change, and real-time processing requirement. The traditional automatic system detects and tracks targets by extracting features such as points, lines, blobs, shape, etc. and adapting them to other sequential frames to track the target (Alkanat et al., 2015). In the field of real-time moving target detection and tracking has been investigated in large by the computer vision community (Iqbal et al., 2013), (Shin et al., 2020). For example, in Viola and Jones al. (Viola et al., 2005), the authors extract the simple Haar features and apply cascading supervised classifiers to detect and track the target (the target on that paper was the face of a human) in a video real-time. Notice that many car detection algorithms are developed for the surveillance systems and sometimes used in a commercial application (Sedky et al., 2005). Installing the camera on the drone and moving it will cause several problems the same as vibration on the camera that caused the received video will be exposed to noise and image stretching. The other problem is the collision between UAV with mountains, houses, trees, etc., it should flight far from the target. so, targets appear very small and it's hard to identify them (Yu et al., 2008). With Knowing these problems, there are different ways to detect and track moving objects using camera-based systems in UAVs some of these approaches extract motion pixels by Motion-based methods can then be divided into two main categories: (1) Background Subtraction and (2) Optical Flow.

Background subtraction methods find static pixels over time and then subtract these pixels to detect moving objects (Lee and Park, 2012), (Setitra and Larabi, 2014). These background subtraction methods work best when the speed of moving camera is not too fast and the size of the is target not very small. Optical flow methods find the corresponding image regions between frames and depend on the local motion vectors to detect moving objects (Waagmeester et al., 2005). However, it is computationally expensive to extract local motion vectors of all pixels in the video frame for real-time operation. This method is not suitable for small target tracking because the target will not be tracked in the image after several frames. In car detection methods using extract car features, such as the Scale Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF), and Binary Robust Independent Elementary Features (BRIEF) methods (Sakai et al., 2015). The difficulty of using traditional methods such as SIFT, SURF, BRIEF, etc. for car detection is that it is necessary to choose which features are important in each given image. This situation causes the error in these methods to be very high and Deciding Whether it is a target or not makes it difficult (O'Mahony et al., 2020), Therefore, common image processing methods cannot accurately identify targets, which is why the tendency towards the neural network is increasing. Deep neural networks have made more progress among all neural networks, and the reason for this is the huge growth in the computing power of deep learning chips and the creation of a set of tagged data such as ImageNet, PASCAL VOC, MS-COCO collections, which has led to extensive Convolutional neural network (CNN) Be studied (Jiao et al., 2019).

The image input to the convolutional network must be fixed-size, and in complex scenarios, there is a need to use a set of different images with multiple dimensions drawing, so, to solve this problem spatial pyramid matching, SPP NET is used (He et al., 2014), that allows the network to input images of various sizes and to have fixed outputs. R-FCN (Yi Li et al., 2016), FPN(Zhao et al., 2019), and Mask RCNN (Doll et al.) have improved the feature extraction methods, feature selection, and classification capabilities of convolutional networks in different ways. Some methods of identifying objects based on area proposal networks, such as R-CNN, Fast R-CNN (WALLACH, 2017), Faster R-CNN, and Mask R-CNN, performed high-precision object identification. However, these methods depended on the high computing power of the Graphics processing unit (GPU) to achieve high accuracy. But improving accuracy with heavy computing will not help real-time applications. This is because, in many applications, the ability of deep memory models to implement in real-time is an essential factor. Recently object identifiers have been so fast, such as RetinaNet (Hoang et al., 2019), SSD (Redmon, 2016), and various versions of YOLO (Redmon and Farhadi, 2018),. In the (Alganci et al., 2020), a comparison was made between the three fast architectures of deep networks, Faster RCNN, YOLOV3, and SSD to detect airplane in high-resolution satellite images, and the results showed that accuracy FasterRCNN was higher than the rest of other methods, and YOLOV3 had acceptable accuracy (of course less than FasterRCNN) and more speed than other methods. SSD does not a have good situation rather than YOLOV3 and FasterRCNN. YOLOV3 is still heavy for embedded systems and increases power consumption.

The only way to detect an object on an embedded device is to reduce the size of the deep learning model. In different versions of YOLO, SSD, and other networks, there are lighter versions of CNN called YOLO-tiny (Huang et al., 2019), YOLO-Lite, and tiny SSD (Womg et al., 2018),. However, the accuracy of these networks is greatly reduced. Therefore, how to reduce the size of the model and the distinctive floating operation without reducing the amount of accuracy is a bottleneck on embedded devices. The most popular way is to reduce components and network size by designing a more efficient network. For example, the networks SqueezeNet (Iandola et al., 2017), MobileNet (Howard et al., 2017), ShffuleNet (Mao et al., 2019), have achieved good accuracy by reducing the size of the model and Floating-point operations.

To summarize methods used for car tracking can be divided into two general categories. One category is traditional (non-deep) methods and the other is deep network-based methods. As mentioned, deep network-based methods are more accurate than traditional methods. Deep neural networks are divided into networks with low depth and high processing speed and networks with high depth and low processing speed, which can be used as needed.

Non-intelligent surveillance systems based on the visual abilities of the human user have always created unwanted errors. Also, in some places it is not possible to use the human user, so to compensate for this shortcoming, we try to use machine vision along with human supervision to make up for each other's gaps. Another point is the problem of ground imaging, which in cases where there are visual obstacles, many targets are not seen, and also with this type of imaging, it is not possible to have extensive environmental coverage. To compensate for this shortcoming, the remote sensing method will be used as a UAV.

In this paper, assumptions are considered, including the use of embedded systems without the Graphics processing unit (GPU) and only support the Central Processing Unit (CPU). Also, the distance for imaging is considered to be about 40 meters, so with this default, the targets will not be small and the car will be well visible. Also, the weather conditions are assumed to be normal and the image prepared for training and testing is selected on the same default.

The innovations used include the use of low weight deep network architecture to increase processing speed, and pre-processing methods have been used to increase the accuracy of it. The preprocessing makes the candidate areas and separates them from all of the images and this candidate area is the input of low weight deep learning. The network will identify the car from the candidate area, which will increase the speed and accuracy together.

The purpose of using this mechanism is to immediately and accurately identify the car from a relatively long distance, which makes it possible to identify and track cars in areas where it is not possible or expensive to capture video by ground stations. The advantage of this method is the reduction or non-interference of the human user in identifying and tracking the target, which significantly reduces the probability of error. It can be used for various commercial and military scenarios.

## 2. Materials and Methods

As mentioned earlier, to implement deep network algorithms on embedded systems (especially those that use the only CPU), their size must be reduced. Decreasing the volume also causes a large error in network performance. Therefore, tried to achieve a function similar to the YOLO network by using a small deep neural network and using pre-processing methods. Compared to other existing deep networks, the YOLO network looks at the identification problem as a function of estimating. The function, meaning that like other deep networks, it does not use the sliding window technique in this technique, a window must move over the entire image. to find candidate areas for identification, this technique allows identification operations to be performed at low speeds so it's one of the best options for real-time performance. The idea in this network is taken from the way a person looks. When a person looks at a scene, he does not need to move his vision to identify different objects, and he only recognizes the surrounding objects at a glance. For this purpose, Yolo uses the function estimation to obtain the candidate areas to identify the object, i.e. all the candidate objects for the object are considered as components of one function and other areas are noise and removed. The background is considered as noise that does not affect the estimation of the desired function, this makes it possible for this identifier to be able to perform reconnaissance operations in real-time in power full GPUs.

Now, with using the idea of Yolo with pre-processing activities, the location of possible targets can be determined, and the rest of the image can be removed as noise, and the remaining image can be given as an input to a weak Deep network. But these networks have a lot of processing load, so used the pattern of these networks, and using pre-processing methods, identified the candidate areas and presented it to a weaker but faster network. So, in this paper selected light network MobileNet-SSD. The proposed algorithm is presented in the flowchart of figure1.

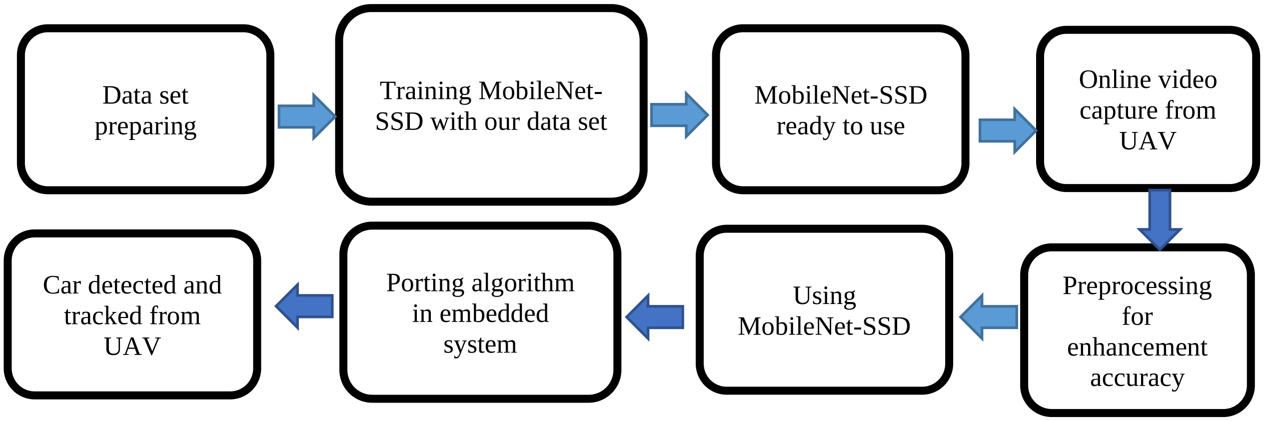
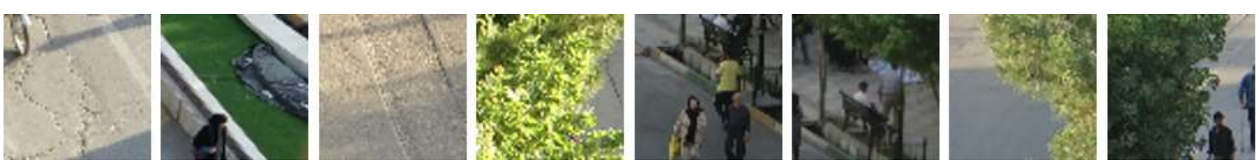



***Figure 1.*** Flowchart of research phases.

## 2.1 Data Set Preparing

The data set used to train and test our CNN includes 2350 car images and 2494 non-car images, and consists of car and non-car classes, the following of which are examples of these two classes.

a. Car classes



b. None car classes

***Figure 2.*** a. Training data set for car classes and b. for none car classes that is provided with the author.

The data production technique has also been used to increase the accuracy of the network in training data. To automatically increase the number of data in the training process. This method generates new data by rotating the data.

## 2.2 Training MobileNet-SSD with our data set

The MobileNet network (Mao et al., 2019) was constructed to increase the real-time performance of deep learning for limited CPU and GPU. This network can reduce the number of parameters without sacrificing accuracy. Previous studies have shown that MobileNet only needs 1/33 of the parameters of VGG-16 (Visual geometry group) to achieve the same classification accuracy in ImageNet-1000 classification tasks (Yiting Li et al., 2018).

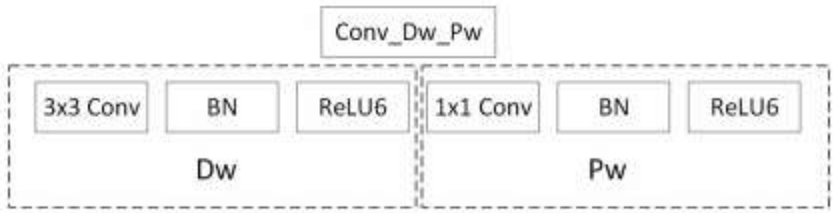



***Figure 3.*** The basic convolutional structure of MobileNet. Dw = depth-wise layers. Pw = point-wise layers. BN = batch normalization. Conv = convolution. ReLU6 = rectified liner unit 6 (Yiting Li et al., 2018).

SSD is a method for detecting objects using a single deep neural network, it has some feature same as easy to train and simple to integrate into systems that require a detection component. Moreover, experimental results on the PASCAL VOC, COCO, and ILSVRC datasets show that SSD has competitive accuracy to other slower methods. This combination of speed and accuracy make this method a very good option to use for our purpose (Science, 2018). As can be seen in the above Figure, in 12 epochs, the accuracy of the training data is about 0.99%, the accuracy of the evaluation data is close to 100, loss of the training data reached 0.01 and the loss of the evaluation data reached 0.02.

## 2.3 Preprocessing

To implement the MobileNet-SSD network training process, a desktop computer with Nvidia GTX 1080 Ti graphics card with 8 GB of RAM was used. The diagram of MobileNet network training data is shown in the figure4.

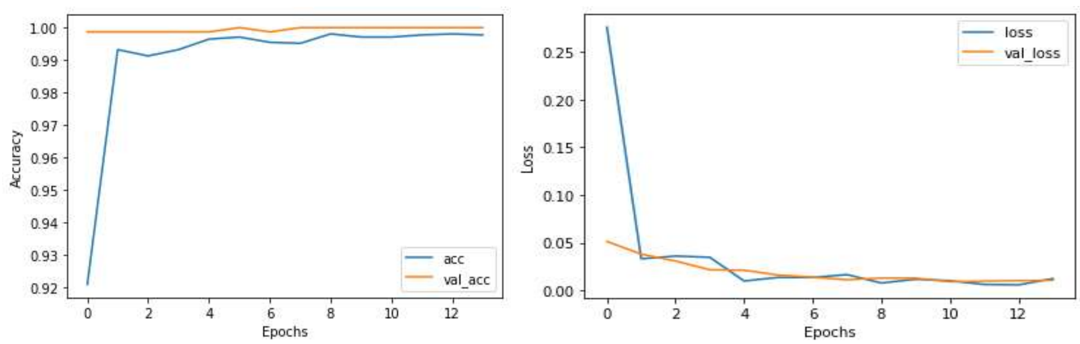



***Figure 4.*** Accuracy and loss of the training process on a desktop computer with Nvidia 1080 Ti.

To identify the moving object used the background subtraction method to separate the moving objects in the image of a static background. Then performed morphological operations to remove noise and fill cavities. also used dilation to fill more holes. To restore the contours, it has done the initial size of the erosion. Using the Chain algorithm, the background contours are found in a gray image.

In this paper, the sensor is located at a distance of about 40 meters from the ground and the measurement is done by using an RGB three-band camera. According to the measurements we have made, the average size of cars from a distance of about 40 meters in pixel space is 90 x 67 pixels, so the contours whose size is about these values are taken to a MobileNet-SSD as moving object and another part is removed until prevent additional processing and speed increase.

# 3. Result and Discussion

After completing the training of MobileNet-SSD, we will port it to Oderoid XU4. To discover the speed performance of MobileNet-SSD, it compared with tiny yolov3 on an Oderoid XU4 and the result is that MobileNet-SSD is twenty-five times faster than Tiny-YOLO.

***Figure 5.*** Result of MobileNet-SSD to track moving cars and Regardless of static cars.

One of the defaults has been to use the algorithm in the CPU platform, for which purpose the performance of different networks in this field has been studied.

**Table1.** Total latency of inference in both CPU and GPU modes (Kim et al., 2019).

| # | Model | CPU Latency (S) | GPULatency (S) |
|---|---|---|---|
| 1 | Faster RCNN (ResNet-101) | 3.271 | 0.232 |
| 2 | YOLOv3-416 | 5.183 | 0.017 |
| 3 | Faster RCNN (Inception ResNet-v2) | 10.538 | 0.478 |
| 4 | YOLOv2-608 | 11.303 | 0.035 |
| 5 | Tiny YOLO-416 | 1.018 | 0.011 |
| 6 | SSD (Mobilenet v1) | **0.081** | 0.03 |
| 7 | SSD (VGG-300) | 0.361 | **0.015** |
| 8 | SSD (VGG-500) | 0.968 | **0.026** |
| 9 | R-FCN (ResNet-101) | 1.69 | 0.131 |
| 10 | Tiny YOLO-608 | 2.144 | **0.025** |
| 11 | SSD (Inception ResNet-v2) | **0.109** | 0.04 |
| 12 | SqueezeDet | **0.14** | **0.027** |
| 13 | R-FCN | 3.034 | 0.084 |

In Table1, Network MobileNet-SSD has the fastest processing compared to other networks, which makes sense for the use of this model in the embedded system. The latency of CPU is about 0.081 is suitable for real-time applications. Another thing to consider is the accuracy of this network model against networks with a volume close to them. In Table 2, a comparison is made between the networks in Table 1, this time to evaluate accuracy.

**Table 2**. Average precision (AP) at an intersection over union (IoU) 0.95 and 0.50 (Kim et al., 2019).

| # | Model | Framework | AP [IoU=0.95] | AP [IoU=0.50] |
|---|---|---|---|---|
| 1 | Faster RCNN (ResNet-101) | Tensorflow | **0.245** | **0.476** |
| 2 | YOLOv3-416 | Darknet | 0.143 | 0.367 |
| 3 | Faster RCNN (Inception ResNet-v2) | Tensorflow | **0.317** | **0.557** |
| 4 | YOLOv2-608 | Darknet | 0.198 | **0.463** |
| 5 | Tiny YOLO-416 | Darknet | 0.035 | 0.116 |
| 6 | SSD (Mobilenet v1) | Tensorflow | 0.094 | 0.233 |
| 7 | SSD (VGG-300) | Tensorflow | 0.148 | 0.307 |
| 8 | SSD (VGG-500) | Tensorflow | 0.183 | 0.403 |
| 9 | R-FCN (ResNet-101) | Tensorflow | **0.246** | **0.486** |
| 10 | Tiny YOLO-608 | Darknet | 0.06 | 0.185 |
| 11 | SSD (Inception ResNet-v2) | Tensorflow | 0.116 | 0.267 |
| 12 | SqueezeDet | Tensorflow | 0.003 | 0.012 |
| 13 | R-FCN | Tensorflow | 0.124 | 0.319 |

As can be seen, the accuracy of this network is less than that of large networks, but it is more accurate than networks that are close to it same as SqueezeDet and Tiny YOLO, in terms of volume. However, this level of accuracy has not been considered and attempts have been made to increase the accuracy to a greater extent by using pre-processing methods. Also, the pre-processing method, in addition to increasing the accuracy to about AP=0.11 in IoU=0.95, but because of image preprocessing consume times the latency increases a little and Even with these conditions, the speed of this method is still 25 times faster than Tiny YOLO. In the comparison of (Redmon et al., 2016), that results in 0.5, the present image research has better results and it showed that implemented algorithms have good capabilities in object detection.

## 4. Conclusion

Car detection through remote sensing methods has a wide range of commercial and military applications. In this paper, the sensor installs on the UAV. UAV some Restrictions in power and weight so cannot carry heavy loads, and therefore light embedded boards should be used to image processing. This sort of board has less processing capability than large industrial computers. For this reason, in applications that require the use of real-time systems, low-volume algorithms should be used that will reduce the quality of work. Therefore, there is a trade-off between accuracy and speed. It is very important to find a method that has the appropriate response to all three of these challenges. This method is using image processing algorithms alongside the deep neural network. The image processing algorithm has a low processing volume, and with its operations, it detects moving objects and provides input tothe deep network, which is the same as the operation of the YOLO network to candidate target as moving vehicle. Deep Network MobileNet, which has previously been trained using data sets, identifies the car from outside the car. The results are appropriate using this innovative method and compared to fast networks such as TINY YOLO, has seen a speed of ۲۵ times. That is suitable for real-time applications and the loss rate has reached ۰,۰۲, which indicates the proper performance of the proposed method.

Based on the hypotheses announced in this study, the use of the GPU system improves the performance of algorithms. The assumption of a distance between the camera and the target, which was 40 meters, showed that the algorithm used had a high ability to detect the car from a distance. The assumption of system efficiency in normal weather conditions is also confirmed. Based on the results obtained, this hypothesis is proved.

## Reference

Alganci, U., Soydas, M., & Sertel, E. (2020). Comparative research on deep learning approaches for airplane detection from very high-resolution satellite images. *Remote Sensing*, *12*(3). https://doi.org/10.3390/rs12030458

Alkanat, T., Tunali, E., & Öz, S. (2015). A real-time, automatic target detection and tracking method for variable number of targets in airborne imagery. *VISAPP 2015 - 10th International Conference on Computer Vision Theory and Applications; VISIGRAPP, Proceedings*, *2*, 61–69. https://doi.org/10.5220/0005298400610069

Doll, P., Girshick, R., & Ai, F. (n.d.). *Mask R-CNN ar*.

He, K., Zhang, X., Ren, S., & Sun, J. (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8691 LNCS*(PART 3), 346–361. https://doi.org/10.1007/978-3-319-10578-9_23

Hoang, T. M., Nguyen, P. H., Truong, N. Q., Lee, Y. W., & Park, K. R. (2019). Deep RetinaNet-Based Detection and Classification of Road Markings by Visible Light Camera Sensors. *Sensors (Basel, Switzerland)*, *19*(2), 281. https://doi.org/10.3390/s19020281

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. http://arxiv.org/abs/1704.04861

Huang, R., Pedoeem, J., & Chen, C. (2019). YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers. *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*, 2503–2510. https://doi.org/10.1109/BigData.2018.8621865

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2017). 50 X Fewer Parameters and < 0 . 5Mb Model Size. *Iclr*, 1–13. https://doi.org/10.1007/978-3-319-24553-9

Iqbal, J., Pasha, M., Riaz-un-Nabi, Khan, H., & Iqbal, J. (2013). Real-time target detection and tracking: A comparative in-depth review of strategies. *Life Science Journal*, *10*(3), 804–813. https://doi.org/10.7537/marslsj100313.121

Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., & Qu, R. (2019). A survey of deep learning-based object detection. *IEEE Access*, *7*(3), 128837–128868. https://doi.org/10.1109/ACCESS.2019.2939201

Kim, C. E., Dar Oghaz, M. M., Fajtl, J., Argyriou, V., & Remagnino, P. (2019). A comparison of embedded deep learning methods for person detection. *VISIGRAPP 2019 - Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, *5*, 459–465.

Lee, J., & Park, M. (2012). An Adaptive Background Subtraction Method Based on Kernel Density Estimation. *Sensors (Basel, Switzerland)*, *12*(9), 12279–12300. https://doi.org/10.3390/s120912279

Li, Yi, He, K., Sun, J., & others. (2016). R-FCN: Object detection via region-based fully convolutional networks. *Advances in Neural Information Processing Systems*, *Nips*, 379–387. http://papers.nips.cc/paper/6465-r-fcn-object-detection-via-region-based-fully-convolutional-networks.pdf

Li, Yiting, Huang, H., Xie, Q., Yao, L., & Chen, Q. (2018). Research on a surface defect detection algorithm based on MobileNet-SSD. *Applied Sciences (Switzerland)*, *8*(9). https://doi.org/10.3390/app8091678

Mao, Q. C., Sun, H. M., Liu, Y. B., & Jia, R. S. (2019). Fast and Efficient Non-Contact Ball Detector for Picking Robots. *IEEE Access*, *7*, 175487–175498. https://doi.org/10.1109/ACCESS.2019.2955834

Mao, Q., Sun, H., Liu, Y., & Jia, R. (2019). Mini-YOLOv3: Real-Time Object Detector for Embedded Applications. *IEEE Access*, *7*, 133529–133538. https://doi.org/10.1109/ACCESS.2019.2941547

O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G. V., Krpalkova, L., Riordan, D., & Walsh, J. (2020). Deep Learning vs. Traditional Computer Vision. *Advances in Intelligent Systems and Computing*, *943*(Cv), 128–144. https://doi.org/10.1007/978-3-030-17795-9_10

Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*.

Redmon, J. S. D. R. G. A. F. (2016). (YOLO) You Only Look Once. *Cvpr*. https://doi.org/10.1109/CVPR.2016.91

Sakai, Y., Oda, T., Ikeda, M., & Barolli, L. (2015). An Object Tracking System Based on SIFT and SURF Feature Extraction Methods. *2015 18th International Conference on Network-Based Information Systems*, 561–565. https://doi.org/10.1109/NBiS.2015.121

Science, C. (2018). *Faculty of Mathematics and Computer Science A follow-me algorithm for AR . Drone* *using MobileNet-SSD and PID control*.

Sedky, M. H., Moniri, M., & Chibelushi, C. C. (2005). Classification of smart video surveillance systems for commercial applications. *IEEE Conference on Advanced Video and Signal Based Surveillance, 2005.*, 638–643. https://doi.org/10.1109/AVSS.2005.1577343

Setitra, I., & Larabi, S. (2014). Background subtraction algorithms with post-processing: A review. *Proceedings - International Conference on Pattern Recognition*, *March 2019*, 2436–2441. https://doi.org/10.1109/ICPR.2014.421

Shin, J., Kim, H., Kim, D., & Paik, J. (2020). Fast and robust object tracking using tracking failure detection in kernelized correlation filter. *Applied Sciences (Switzerland)*, *10*(2). https://doi.org/10.3390/app10020713

Viola, P., Way, O. M., Jones, M. J., & Snow, D. (2005). *T61K38U53J53134*. *63*(2), 153–161.

Waagmeester, A., Thompson, J., & Reyrat, J. M. (2005). Identifying sigma factors in Mycobacterium smegmatis by comparative genomic analysis. *Trends in Microbiology*, *13*(11), 505–509. https://doi.org/10.1016/j.tim.2005.08.009

WALLACH, B. (2017). Developing. *A World Made for Money*, 241–294. https://doi.org/10.2307/j.ctt1d98bxx.10

Womg, A., Shafiee, M. J., Li, F., & Chwyl, B. (2018). Tiny SSD: A Tiny Single-Shot Detection Deep Convolutional Neural Network for Real-Time Embedded Object Detection. *2018 15th Conference on Computer and Robot Vision (CRV)*, 95–101. https://doi.org/10.1109/CRV.2018.00023

Yu, W., Yu, X., Zhang, P., & Zhou, J. (2008). a New Framework of Moving Target Detection and Tracking for. *Archives*, *XXXVII*(B3b, Commission 3), 606–614.

Zhao, Y., Han, R., & Rao, Y. (2019). A new feature pyramid network for object detection. *Proceedings - 2019 International Conference on Virtual Reality and Intelligent Systems, ICVRIS 2019*, 428–431. https://doi.org/10.1109/ICVRIS.2019.00110.