IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 40, No. 2, April, 2023, pp. 457-468
Journal homepage: http://iieta.org/journals/ts

A New Automatic Vehicle Tracking and Detection Algorithm for Multi-Traffic Video Cameras

Sevinç Ay[1*], Murat Karabatak[2]

[1] Distance Education Center, Fırat University, Elazig 23119, Turkey
[2] Department of Software Engineering, Firat University, Elazig 23119, Turkey

Corresponding Author Email: say@firat.edu.tr

https://doi.org/10.18280/ts.400205

Received: 2 December 2022
Accepted: 11 March 2023

Keywords:
automatic vehicle tracking, automatic vehicle detection, deep learning, object recognition, faster RCNN, YOLO

ABSTRACT

Vehicle tracking systems are a vital tool in modern-day law enforcement and security operations. With the increasing threats of terrorism, organized crime, and illegal trafficking, monitoring and tracking suspicious vehicles has become a top priority for security agencies around the world. In this study, a target vehicle, which was described as suspicious, was tracked using the proposed vehicle tracking method that contains Gaussian Mixture Model (GMM) and Blob analysis. The same target vehicle was then detected using the Regions with Convolutional Neural Networks (RCNN), Faster RCNN, and You Only Look Once (YOLO) deep learning object recognition algorithms. In these applications, public traffic surveillance system images from the internet are used. Tracking is performed on images taken from more than one traffic surveillance system on the same road or route. The results from these methods were compared with each other, and the highest mean Average Precision (mAP) value was observed as 89.20% for the Faster RCNN algorithm using the Resnet101 deep learning architecture.

1. INTRODUCTION

Visual surveillance systems are frequently used in many areas for observing the movements of objects or people. Systems such as those for monitoring the flow of traffic, face recognition, detection of human behaviour, and detection of abnormal events, for example, have become an integral part of visual surveillance systems. Using these surveillance systems, which are widely deployed, especially in shopping malls and city centers, it is possible to accurately detect abnormal behaviour or the movements of suspicious vehicles or individuals among crowds. This makes it possible to detect and prevent many dangers before they arise [1, 2]. The demand for visual tracking systems in many areas of daily life has also drawn the attention of researchers toward object tracking in recent years, and recent studies have emphasized the importance of object tracking for the interpretation of surveillance systems [2, 3].

As the number of vehicles and traffic loads are increasing every day, the use of visual surveillance systems to monitor traffic has become an area of great interest in order to ensure safe cities for modern life [4]. The International Organization of Motor Vehicle Manufacturers (OICA) has reported that more than 70 million vehicles have been produced in recent years [5]. The management of such high numbers of vehicles is obviously one of the most significant problems faced by countries around the world. As traffic surveillance systems become more common, the issues of vehicle detection and the interpretation capabilities of these systems have also proved crucial. Conventional vehicle tracking techniques are not able to cope with such large amounts of data, and the need for new methods to tackle these challenging tasks has become apparent [6].

The process of examination of the records obtained by visual surveillance systems can be divided into three types, based on whether they use passive, semi-automatic, or fully automatic control. In passive control, the records captured by video recording systems are controlled by a human. In semi-automatic control, when there is a significant movement in the images, the recording camera captures this moment and the recordings are then examined in detail by a human. In fully automatic control, the examination, analysis, and reporting processes are carried out independently of human tracking [2, 3].

The use of video cameras for traffic surveillance is limited to passive control tasks or very basic semi-automatic examination. In urban traffic management, records captured throughout the day are typically examined when there are errors caused by drivers [7, 8]. Detection of vehicles is provided only in the frames in which they are detected. Vehicle detection and tracking using more than one camera are done manually, and the scope of operation of an automatic inspection system is limited [2, 3]. Today, automatic traffic management is required to perform high-level tasks such as automatic incident detection and law enforcement [6-9].

Vehicle detection and tracking is very difficult, due to the increase in the numbers of road vehicles and the need to detect errors and other situations by passive tracking methods. To carry out this process more effectively, many methods have been designed for the context of vehicle detection and tracking. Of these, deep learning approaches are particularly prominent, and vehicle detection technology has greatly improved, with deep convolutional neural networks (CNNs) with their powerful learning capabilities now widely used in the field of computer vision [10].

The CNN was designed to artificially reproduce the functional capabilities of the human cognitive system, with its powerful feature extraction capability, and has outperformed conventional methods on various computer vision tasks. In recent years, the development of graphics processing units (GPUs) in hardware technology, especially for parallel computing, has led to the development of Regions with Convolutional Neural Networks (RCNNs). Fast RCNN was developed first, and then Faster RCNN, Single Shot Detector (SSD) and You Only Look Once (YOLO), among others. Many deep-learning methods have been used for vehicle detection [4, 5, 10].

The Faster RCNN algorithm has high sensitivity and strong detection rates, and can achieve high levels of accuracy and speed when used for multi-object detection in complex traffic environments. In addition, due to its success in recognizing small moving objects, it has been used as a basis for other algorithms by many researchers. In this study, we therefore mainly focus on vehicle detection methods based on deep neural networks such as Faster RCNN and YOLO [10-12].

When studies in the field of vehicle detection and tracking are examined, it can be seen that the target vehicle is searched for from a single camera recording, or a single problem is considered, such as the obstacles and image distortions encountered when detecting and tracking an object from multiple camera recordings. In this study, we aim to provide an automatic system for the examination of records taken from video surveillance systems used for security purposes, and to detect the target vehicle from these records within a shorter time. Hence, in the current study, a new algorithm is developed for the process of tracking a target vehicle that is particularly resistant to obstacles.

The contributions of this article can be summarized as follows:

- ➢ In the literature on vehicle tracking and detection, operations are usually carried out on a single video recording taken from fixed or moving cameras. In addition, the focus is on tracking all objects in the scene. In the proposed method, when the target has been detected by one camera, the system ensures that the target is followed by other cameras in the region. In addition, target detection is performed on other cameras.
- ➢ In this study, a labeling process without human control is proposed to create a ground truth object. In the data collection phase, the first video image containing the selected target is used. From this video, the target is detected in each frame, and a ground truth object is created by recording it with bounding box information. This method can be also implemented as a new vehicle detection algorithm.
- ➢ Many factors have been found in the literature that make target tracking difficult. In these studies, the tracking process could not be carried out properly due to these difficulties. It is shown that the vehicle tracking algorithm proposed in the first stage of the study is resistant to background changes, and can continue to follow the target despite the presence of elements that block the image.
- ➢ RCNN, Faster RCNN, and YOLO, which are CNN-based modern object detection algorithms, are used and performance evaluations are compared.

The remainder of the article is organized as follows. In Section 2, a literature review of existing methods is presented. A description of the different methods used in this research is given in Section 3. Detailed information on the proposed method, the acquisition process used for the datasets, and the experiments is presented in Section 4. Finally, a conclusion is given in Section 5.

2. LITERATURE REVIEW

2.1 Object tracking

Object tracking, which forms the focus of the first stage of this study, is one of the most popular fields of research, with a wide scope and many areas of application. From the studies in the literature, we see that although many methods have been applied to object tracking, a range of difficulties are still faced in this field. Numerous natural factors such as changes in lighting, shadows, and objects blocking the view are among the difficulties that need to be considered [12].

Recent research has focused on two main areas of object-tracking applications for video analysis systems. The first of these is the improvement of optical flow estimation methods, which can achieve successful results in terms of estimation accuracy and computational quantity. The second involves improvements to background and foreground modeling methods [13].

Fei et al. [14] carried out a study involving visual tracking based on improved foreground detection. A three-frame differencing algorithm and a foreground detection method were combined with background subtraction. The authors tried to resolve problems such as shadow and foreground aperture by calculating the differences between the existing image frames. Their experimental results revealed that the proposed method yielded fast and accurate detection.

Kiratiratanapruk et al. [15] presented a gradient-based foreground detection method for a traffic monitoring system. An improved edge detection-based background modeling approach that was sensitive to light changes was proposed. This method was compared with popular background modeling techniques, and it was reported that the proposed method was resistant to lighting changes in outdoor environments and the characteristics of video camera sensors. It was also found to give high detection rates for foreground objects.

Liu et al. [16] presented an optical flow-based vehicle tracking method. In this study, the optical flow method was initially used to determine the direction of the vehicle in the first few frames of the image. Following this, the distance factor was taken into account to eliminate the problems that may occur when a similar new vehicle enters the scene. A feature template was presented for vehicle tracking in lower-resolution videos. As a result, a tracking algorithm with better performance than traditional algorithms were developed.

Sushmitha et al. [17] presented a new algorithm in their study on tracking more than one vehicle in traffic. In their work, preprocessing, motion segmentation, and feature extraction methods were applied to the video frames used as input to the system. The object was determined using blob analysis and tracked with a background extraction method.

2.2 Object detection

In the field of object detection, which represents the second stage of our work, very successful results can be obtained with methods that are improving every day. Studies in the literature on this topic are reviewed below.

Maity et al. [10] worked on the detection of automatically moving vehicles in smart traffic surveillance systems. In this study, Faster RCNN and YOLO-based vehicle detection and tracking methods were compared. In addition to vehicle detection, the authors focused on the importance of tracking vehicles properly in order to avoid collisions and to provide resistance to image changes. It was reported that tracking methods should be improved, as the approaches suggested thus far are mostly manual and depend only on the camera image.

Luo et al. [11] presented a model based on Faster RCNN with NAS optimization and feature enrichment to detect vehicles from images of traffic scenes. A Retinex-based algorithm was used to reduce the changes in illumination and shadow effects in the images. Tests and experiments were performed on the UN-DETRAC dataset. At the end of the study, it was reported that the method achieved high performance in terms of accuracy, and that the detection rate should be increased.

Wang et al. [12] studied traffic signal recognition using an improved Faster RCNN with the Resnet architecture, and applied this approach to the Tsinghua-Tencent 100k dataset. It was observed that the proposed algorithm achieved higher performance than other algorithms in terms of accuracy and recall values.

Arinaldi et al. [18] presented a Faster RCNN-based traffic analysis system that included vehicle counting, vehicle type classification, and estimation of vehicle speeds from the video. In the first stage of the study, the mixture of Gaussian (MoG) and support vector machine (SVM) algorithms were used. It was shown that Faster RCNN performed better than SVM and MoG under various conditions.

Fan et al. [19] focused on Faster RCNN, in view of its high performance in object detection studies. Extensive experiments were performed on the KITII dataset. They also proposed adaptations to increase the performance of this approach.

Cao et al. [20] presented a new algorithm for this problem that capitalized on the success of Faster RCNN in detecting small objects. To solve the loss function problems, a loss function was introduced based on the intersection over union (IoU) for bounding box regression. Bilinear interpolation was used to improve the pooling of the regions of interest (RoI). The accuracy rate of the algorithm was found to be 87%. Faster RCNN was applied to a dataset consisting of images with a resolution in the range of (0.32). Zhu et al. [21] and Li et al. [22] carried out similar studies. The performance of the algorithm was better than the results of these studies.

Arcos-García et al. [23] attempted to recognize traffic signs from the COCO dataset using deep learning methods. They reported that Faster RCNN gave the best results with the Resnet V2 backbone. In their study, the Faster RCNN, SSD, and YOLO V2 object detection algorithms were compared using feature extractors such as Resnet V1 50, Resnet V1 101, Inception V2, Inception Resnet V2, Mobilenet V1, and Darknet-19.

Han et al. [24] performed real-time object detection for small vehicles using the YOLO-v2. With this study, it has been tried to find a solution to the sensitivity and performance problem. In the proposed model, convolution layers were added to different locations and the feature extraction capability of the network was strengthened. High success has been achieved in the experiments on the KITTI dataset. However, since all of the images used were under good illumination conditions, no study was carried out for scenes with insufficient lighting conditions.

Bie et al. [25] introduced a real-time vehicle detection algorithm using YOLO-v5. They developed the YOLOv5 algorithm to avoid the complex nature of existing vehicle detection algorithms. A bidirectional feature pyramid network is used to enhance the feature extraction capability. Experiments were performed on the BDD100K dataset. Compared to SSD, average average accuracy is increased by 1.7% and mAP@0.5 increases by 4%. Compared to Faster-RCNN, mAP@0.5 was reduced by 1.2%.

Li et al. [26] proposed vehicle detection for Intelligent traffic scheduling. An object detection model is proposed for identifying small sized traffic elements in UAV image sequences. Experiments on UAV image sequences have shown that the algorithm can reduce traffic congestion. The algorithm achieved better results than YOLO v3. The object detection model will be improved to deal with the dense and small objects.

Rafique et al. [27] proposed an algorithm for real-time parking management. The algorithm was developed to find empty parking spaces and generate vehicle statistics. A pretrained model of YOLO v5 was used on the MS COCO dataset. The accuracy of the work is 99.5%.

Azimjonov et al. [28] proposed a real-time vehicle detection and vehicle tracking system. YOLO and Bbox-based tracking were used. Manually, 7216 data were labeled and trained. In the study, 95.45% accuracy was obtained. Additionally, Kalman filter-based vehicle tracking was implemented and the bounding-box-based vehicle tracking algorithm was developed.

3. OVERVIEW AND METHODOLOGY

The purpose of our study was to track a target in images from more than one video camera. For this, we considered images taken from MOBESE (MOBil Electronic System Integration) cameras, which are used in Turkey for security and traffic control. Video recordings of the route along which the target vehicle passed were obtained from traffic surveillance systems.

This study is divided into two main stages. In the first stage, vehicle tracking is performed, and in the second, vehicle detection is carried out. Object tracking is made more difficult by the presence of obstacles in the image and situations where the object leaves the field of view. In our study, a background extraction method, which is one of the target tracking approaches, is developed and a new target tracking method is presented. With this method, successful tracking is achieved throughout the video sequence despite the presence of obstacles and without losing the target.

In this case, the target is a vehicle, and is selected from the first video image and framed with a bounding box. The target image obtained in this way is then followed throughout the video using the developed tracking algorithm. The tracked vehicle is limited by a bounding box. The data on the bounding box and the file path for each frame in the video are saved in a file to create a ground truth object. The ground truth data are then used as training data for the deep learning method to be used in the second video.

To detect the same target in the second video, the target vehicle and bounding box information from the first video were used. We trained RCNN, Faster RCNN, and YOLO models with different backbone networks, and the method with the best result was selected. Figure 1 shows the steps followed in our study.

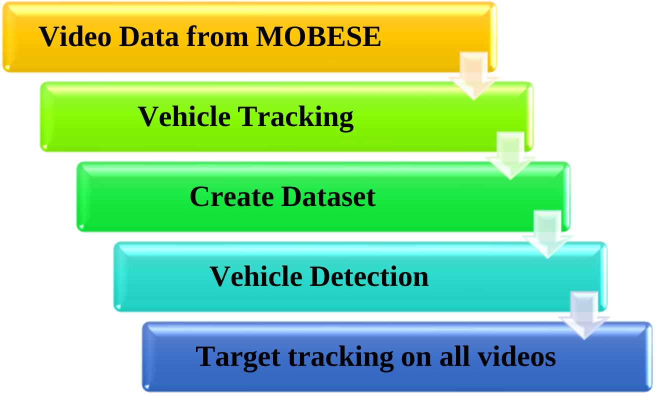


Figure 1. Vehicle detection and tracking system

3.1 Vehicle tracking

In the second part of this paper, a vehicle tracking algorithm using Gaussian mixture model (GMM)-based foreground detection is proposed. GMM is a background subtraction method that can be used to determine whether a pixel belongs to the background.

GMM was selected to model the background since it is one of the strongest models for background modeling and is resistant to changes in light and other conditions. It is a probability density function which is a weighted sum of the Gaussian component densities, as shown in Eq. (1) [24]:

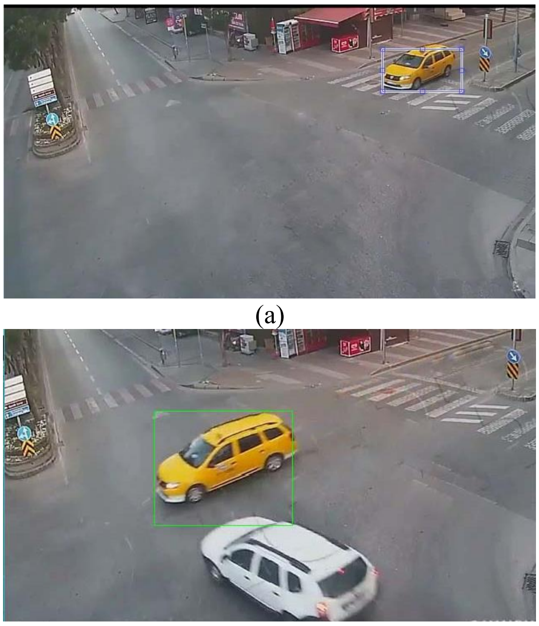
$$p(X|\lambda) = \sum_{k=1}^{M} w_k \, N\,(x|\mu_k, \Sigma_k) \qquad (1)$$

It can be seen that GMM is a weighted sum of M components.

The next method used for background subtraction in object tracking is blob analysis. This approach recognizes a moving object within a blob area and marks it with a bounding box. In this work, vehicle detection is the first step before more complex tasks such as tracking and classification are carried out. This approach estimates the foreground pixels in a video sequence captured from a fixed camera and creates a mask that highlights the foreground objects with the background subtracted. A foreground mask is used to compare a color or grayscale video frame with a background model. In this way, it can be determined whether each pixel in the moving image belongs to the background or the foreground.

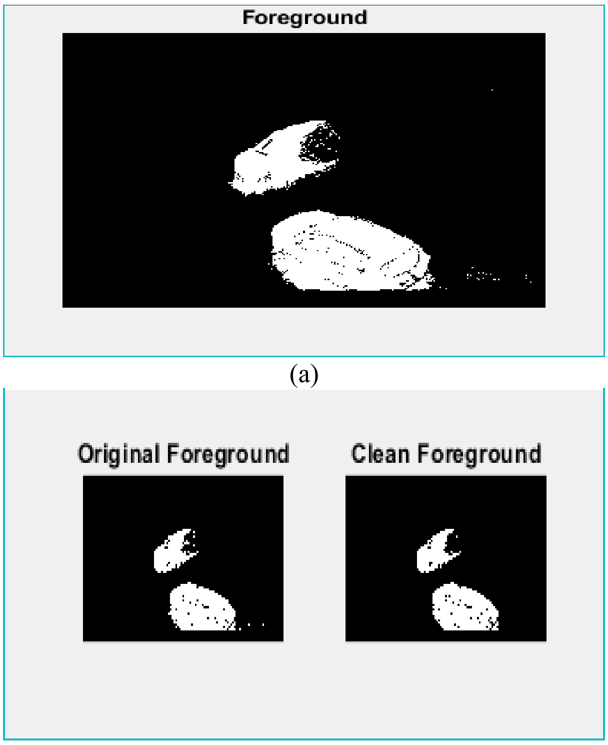
The basic idea of background subtraction is to detect the foreground object. The foreground mask uses blob analysis to create bounding boxes around vehicles. All moving vehicles are determined with bounding boxes around them and are tracked until they disappear from view in the video [29]. Figure 2(a) shows the selection of the target in the first frame, and Figure 2(b) shows the tracking of the target vehicle.

In the proposed method, rather than taking and processing the entire video, an initial frame is obtained that contains moving foreground objects that are separated from the background objects. The foreground detector considers a certain number of frames, according to the length of the video, to initiate the GMM. In this way, the background frame is learned. Figure 3(a) shows the foreground image detected by the foreground detector. Foreground detection often involves noise, and filtering and morphological opening are therefore applied to the image in order to remove noise and fill in the gaps. Figure 3(b) shows the noise-free image and its original state.


(a)

(b)

Figure 2. (a) Determining the target using the mouse in the first frame; (b) Tracking the target with blob analysis



(a)

(b)

Figure 3. Morphology process for foreground detection: (a) Foreground image; (b) Original foreground with noise, and clean foreground

Using foreground detection, all moving vehicles can be detected in the same frame of the video. However, the tracking process may fail due to the obstacles present in the scene and the proximity of the vehicles. For this reason, an improved foreground detection method based on GMM is presented. In this method, a single target is selected with the mouse in the first frame and is then tracked. It was observed that the vehicle was tracked correctly even when it was behind an obstacle or another vehicle during target tracking. Figure 4 shows a flowchart for the proposed algorithm.

Pseudocode for the vehicle tracking process based on GMM is shown in Algorithm 1 below.

Algorithm 1. The proposed vehicle tracking algorithm

```
Procedure: Vehicle Tracking
Input: Video frames n= {1,….,N}, target vehicle
Output: Target and detect vehicles, Boxes: the set of bounding boxes of vehicles in frames

for  n in amount of frame in the video
    İf n==1
        targetselection= object region in frame (n)
    else
        foreground = foregroundDetector(n)
            filteredForeground = (foreground, morphological filter for noise removal);
        boxes = each of frames (blobAnalysis, filteredForeground)
        for i= 1 to The number of bounding boxes of all vehicles in the video frame
            D(i)  = DistanceAlgorithm (targetselection , bbox at each moving car in video frames(i))
            smallindex= find min (D)
            targetselection =bbox(smallindex);
            Draw a bounding box around the target
            save every video frame with target vehicle as an image
            save bounding boxes data
        end
    end if
end
```

3.2 Vehicle detection

In this stage of the process, the vehicle, which was identified and followed in the first stage, is detected in other video images. We attempted to identify the target vehicle by training several deep learning models. When tracking the vehicle from the first video in which it is detected, the image in each video frame and bounding box data are recorded. In the second traffic video of the target vehicle, its location is determined by training different deep learning algorithms on these data. We first trained an RCNN deep learning model and then Faster RCNN and YOLO models, using the AlexNet, VGG 16, VGG 19, ResNet50, ResNet101, GoogleNet, and MobileNet architectures, and their results were compared.

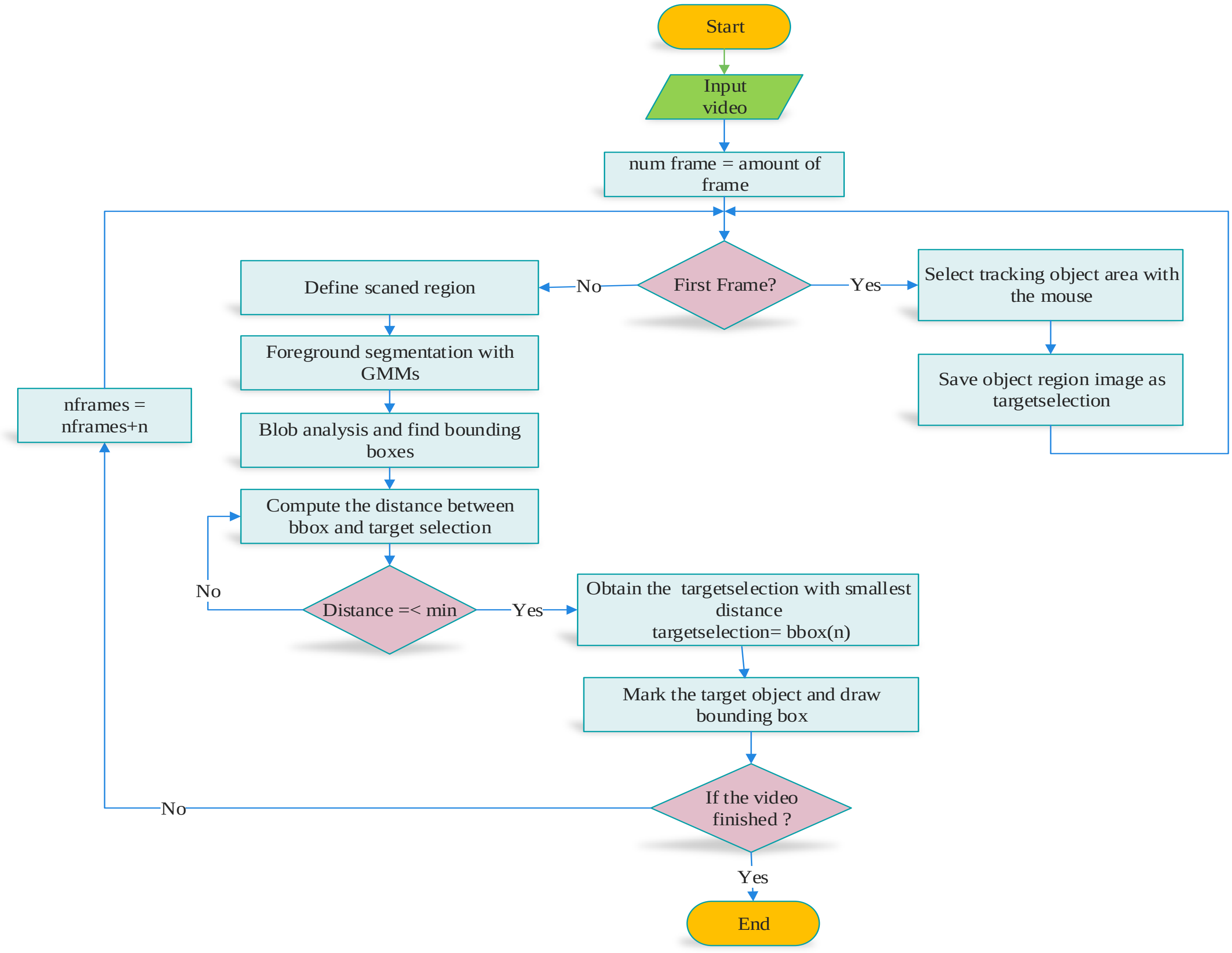

Start
Input video
num frame = amount of frame
First Frame?
No
Yes
Define scaned region
Select tracking object area with the mouse
Foreground segmentation with GMMs
Save object region image as targetselection
nframes = nframes+n
Blob analysis and find bounding boxes
Compute the distance between bbox and target selection
No
Distance =< min
Yes
Obtain the targetselection with smallest distance targetselection= bbox(n)
Mark the target object and draw bounding box
No
If the video finished ?
Yes
End

Figure 4. Flowchart for the object tracking algorithm

3.2.1 Faster RCNN

The Faster RCNN model, which has become an excellent object detection algorithm due to its low false and incomplete recognition rates, stands out in the field of vehicle tracking. This method was created from the development of RCNN [30, 31].

Instead of the selective search used by Fast RCNN, Faster RCNN also uses a region proposal network (RPN). In this way, the time required for object detection with Faster RCNN can be reduced to about 0.3 s [30, 31].

Faster RCNN consists of two modules: the first is the RPN, a deep convolutional neural network that enables region propositions, and the second is the Fast RCNN detector, which uses the regional proposals from RPN. Faster RCNN is an end-to-end training model [31, 32].

Region Proposal Network

A Faster RCNN network can take an input image of any size. After preprocessing, the input image is passed to the backbone to extract the features. The feature map is then given to the RPN to carry out region proposal. After the network regions have been resized, they are passed to the fully connected layer and classified. RPN offers region recommendations to separate foreground data from background data. In Faster RCNN, anchor boxes are used; the RPN can predict the possibility of an anchor being background or foreground, and can refine the anchor [33].

Anchor Boxes

Anchor boxes are important parameters of deep learning object detectors such as Faster RCNN. They consist of bounding boxes of a certain height and width, and play an important role. These boxes are selected based on the sizes of the objects in the training data, and are defined to determine the aspect ratio of the object we want to detect. Through the use of anchor boxes, prediction and detection information can be obtained for multiple objects of different sizes.

3.2.2 YoLo

YOLO, which performs detection and classification in a single step, has become an important tool for object detection [34]. Bounding box and class predictions are made after the input image is evaluated; this approach differs from traditional methods, since the bounding box and class predictions are made simultaneously. YOLO is particularly successful for the real-time detection of targets, although its detection success for small objects is not as high as other methods. It is also sensitive to poor lighting [35, 36]. In this study, a YOLO-v2 network using the MobileNet architecture was considered.

3.2.3 Backbone networks

AlexNet

In the ImageNet competition held in 2012, AlexNet surpassed all its competitors. It was able to yield significantly increased classification accuracy, and the error rate was reduced to 15.3%. This architecture consists of five convolutional layers and three fully connected layers [37].

VGGNet

This is one of two architectures that were shown to be successful against other approaches in the ImageNet competition of 2014. Developed by Simonyan and Zisserman, VGGnet is offered by the VGG Group (Oxford). There are two architectures, which are known as VGG 16 and VGG 19; the former has 16 convolutional layers while the latter has 19, and they are named for the numbers of layers. VGGNet fixed the high kernel sizes encountered in AlexNet by reducing [30].

GoogLeNet

GoogLeNet was proposed by Google, and was launched after winning the competition in 2014. It was inspired by LeNet, and uses the inception module differently from previous architectures. It achieved an error rate of 6.67%. GoogleNet consists of 22 layers, more than all its predecessors [38].

ResNet

This architecture, which won the ImageNet competition in 2015, takes its name from a residual network. ResNet with deeper features has 152 layers. In this model, residual blocks are used to reduce the training error, and in order to reduce the depth, random dropping is applied in the training layers. The error threshold can be reduced below the human error threshold, and has reached 3.57%. There are several varieties, such as Resnet50, Resnet101, and Resnet152, which are named based on the numbers of layers [38, 39].

MobileNet

This is designed to be used in mobile applications, and relies on deeply separable convolutions, meaning that memory can be used more efficiently. It is relatively fast compared to similar architectures with the same level of complexity [40].

4. EXPERIMENTS AND ANALYSIS OF RESULTS

4.1 Training platform and parameter settings

Matlab 2021a version was used as our experimental platform. All of the experiments were performed on a computer equipped with an Intel(R) Xeon(R) W-2245 CPU @ 3.90 GHz processor, 32 GB RAM, and an NVIDIA Quadro RTX 40000 graphics card. The computer used the Windows 10 operating system. The average training time for each of the experiments was 10 h. Table 1 shows the parameter settings used for Faster RCNN and YOLO.

Table 1. Parameter settings for deep learning models

Options	Value
Epoch	50
Iteration	8000
Initial Learning Rate	0.001
Mini Batch Size	1
Momentum	0.9

4.2 Data collection

Video observation was carried out under different weather conditions and at different times of the day. In the data collection step, MOBESE images captured in the province of Elazig in Turkey were used. Video recordings with views of different (or the same) parts of the road were selected from the cameras, which were placed sequentially. The resulting video recordings were numbered. The target vehicle was selected from the first video using the mouse, and vehicle tracking was performed on this video sequence.

Using the proposed tracking algorithm, the target vehicle could be observed throughout the video, and each frame was recorded along with the bounding box coordinates of the vehicle. The video was split into images of size 74 kb with resolution 1280 x 720. Video frames were saved in a dedicated folder with names based on consecutive numbers. Rotation was applied to the images to replicate the training data. The data obtained in this way were used for training several deep learning models. We used 80% of the data for training and 20% for testing. Using the detector created as a result of this training process, the target was detected in the video from the second camera.

Ground Truth

Ground truth is a term that describes the data used to train and test artificial intelligence (AI) models. Ground truth labeling is required in order to generate ground truth data. Labeling is the process of assigning data so that it can be perceived by a deep learning model. In this study, as target tracking was being carried out in the first video, the ground truth data were also created. With the developed algorithm, a labeling process is carried out quickly and without human control. Each frame in the video where target tracking took place was recorded in a folder, and a sequential number and name combination was assigned to it. There was only one tracked vehicle in each video frame, and its bounding box coordinates were recorded. In each line of the files, the file path of the video frame and the bounding box data of the target vehicle in that frame were recorded.

There are several applications that can be used for labeling in Matlab, such as Image Labeler or Video Labeler. In future studies, since the proposed tracking algorithm was found to successfully perform target tracking, it can also be used in labeling applications in cases where considerable time is required to create ground truth data manually. In these applications, the target determined in the first video frame is automatically labeled with the bounding box throughout the video sequence using the selected tracking algorithm. The proposed tracking algorithm, which will be added to the application with the Select Algorithm option, was shown to be very successful in terms of automatically labeling the target in all frames.

4.3 Evaluation metrics

For target detection, the most commonly used evaluation metrics are the precision, recall, mean average precision (mAP), and frame rate per second (FPS). For object detection tasks, we calculate the precision and recall using the intersection over union (IoU), which is determined based on the true positive (TP), false positive (FP), false negative (FN), and true negative (TN) rates. TP means correctly classified, FP means incorrectly classified, and FN represents the missed examples in this category.

4.3.1 Recall

This is the true positive (TP) rate for all predictions, and is defined as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{2}$$

4.3.2 Precision

This is the true positive (TP) rate for all positive predictions, and is defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{3}$$

4.3.3 Mean Average Precision (mAP)

Many object detection algorithms, such as Faster RCNN and YOLO, use mAP to evaluate their models. Before calculating the mAP value, it is necessary to calculate the precision and recall. The mAP is the area under the precision-recall curve, and is defined as:

$$\text{mAP} = \frac{1}{\text{n}}\sum_{\text{k=1}}^{\text{k=n}} \text{AP}_\text{k} \tag{4}$$

4.3.4 Intersection over Union

IoU is defined as the area of intersection between the predicted bounding box coordinates and the ground truth boxes. It is calculated by dividing the overlap of the boxes by the total area (Figure 6).

A TP is determined by the IoU threshold. This means that the bounding box is considered to be a TP if the IoU is greater than a specified threshold. Hence, the value of the average precision (AP) depends on the IoU threshold setting. In this study, we set a value of 0.5 as the IoU threshold to evaluate the detection results; if the IoU was greater than or equal to 0.5, the detected box was considered to be a TP. IoU is defined as:

$$\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}} \tag{5}$$

4.3.5 Frame rate per-second

The fps refers to the number of frames detected per second; it is affected not only by the structure of the algorithm but also by the hardware configuration of the experimental equipment.

4.4 Training process

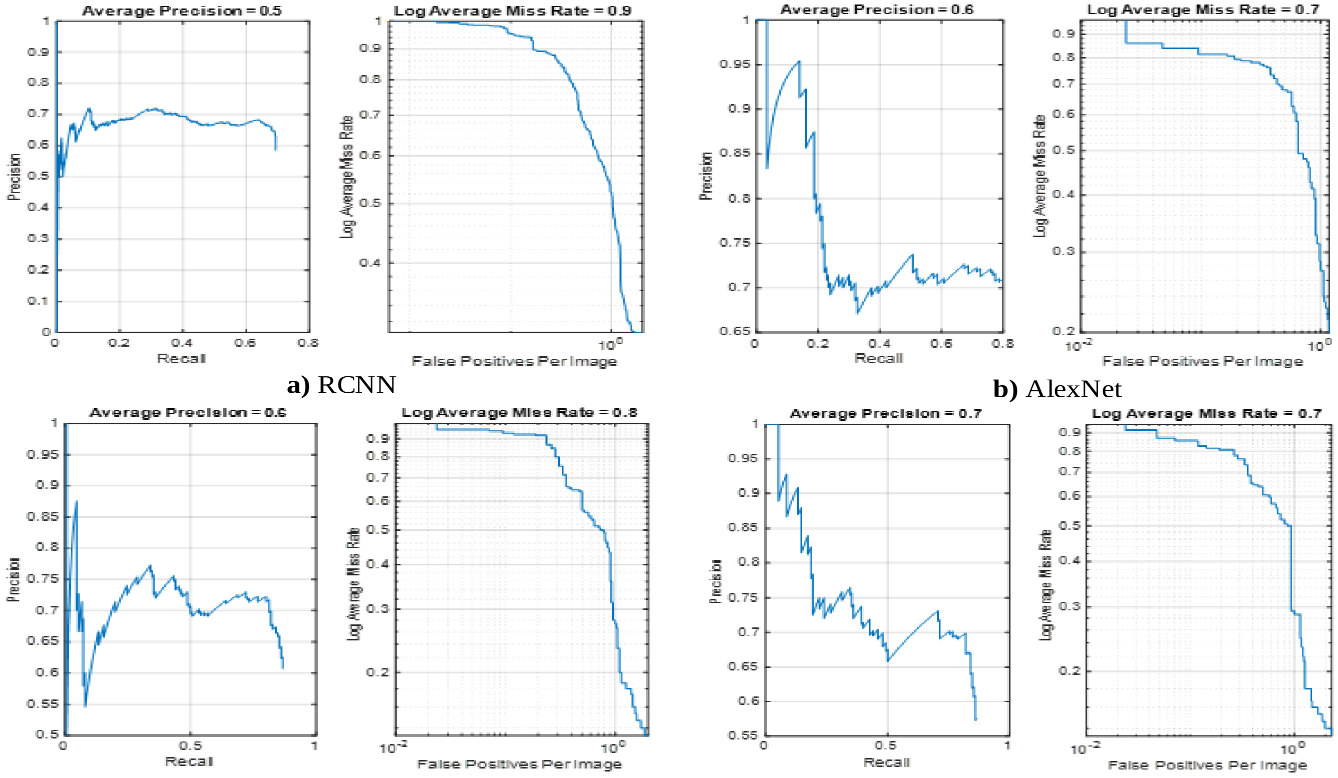
At the object detection stage, the target vehicle was detected from other video images. We also compared the detection results at this stage from different backbone architectures, which were used with various deep learning models, in order to evaluate the different approaches to object detection. We analyzed the effect of this situation on the detection results. Table 2 presents a comparison of these models.

Table 2. Detection results from different models and backbone networks

Deep Learning	Backbone	Fps	Mean Average Precision (mAP) (%)	Average Miss Rate (mMR) (%)
RCNN	AlexNet	2.4	42.90	89.44
Faster RCNN	VGG 16	2.26	62.04	76.47
Faster RCNN	VGG 19	3.08	64.47	74.70
Faster RCNN	AlexNet	2.3	61.22	73.56
Faster RCNN	ResNet 50	8.10	85.16	32.80
Faster RCNN	ResNet 101	9.24	89.20	31.10
Faster RCNN	GoogleNet	3.05	59.02	77.87
YOLO	MobileNet	10.04	74.20	73.82

The mAP and miss rate ratios are used to determine the performance at the training stage. Curves of the log average miss rate and precision vs. recall for the models are shown in Figure 5 and Figure 6.

The results of the study showed that small vehicles could be detected, including those far from the cameras. Our system was also able to identify vehicles going in different directions, in blurry images, and under rainy conditions, at night, etc. However, the detection results were sometimes insufficient in crowded scenes or in situations where vehicles were side by side. In general, the proposed method performed very well for vehicles of various sizes, and could be extended in future studies.



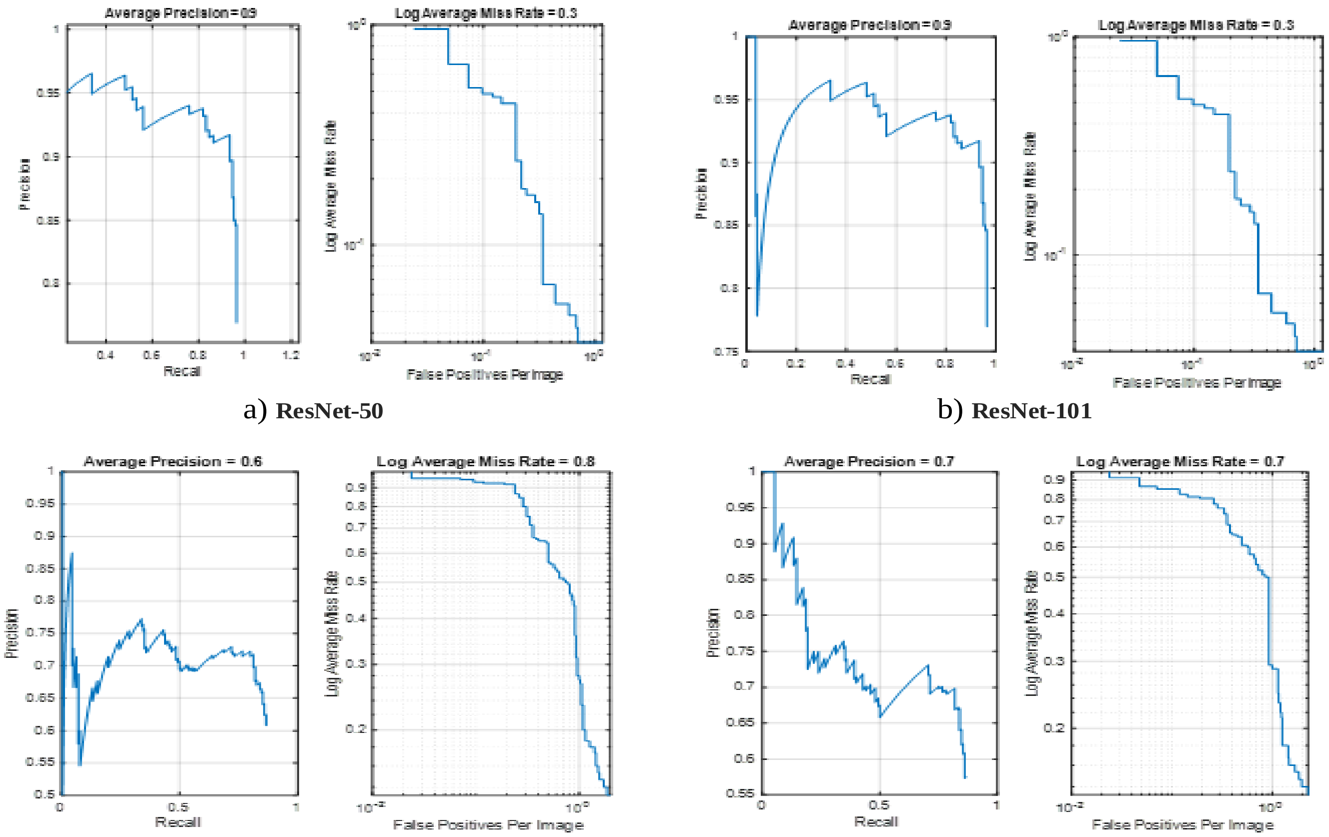
a) RCNN

b) AlexNet

c) VGG16

d) VGG19

Figure 5. Precision vs. recall curves for various deep learning models: (a) RCNN; (b) AlexNet; (c) VGG-16; (d) VGG-19



a) **ResNet-50**

b) **ResNet-101**

c) **GoogleNet**

d) **YOLO**

Figure 6. Precision vs. recall curves for various deep learning models: (a) ResNet-50; (b) Resnet-101; (c) GoogleNet; (d) YOLO

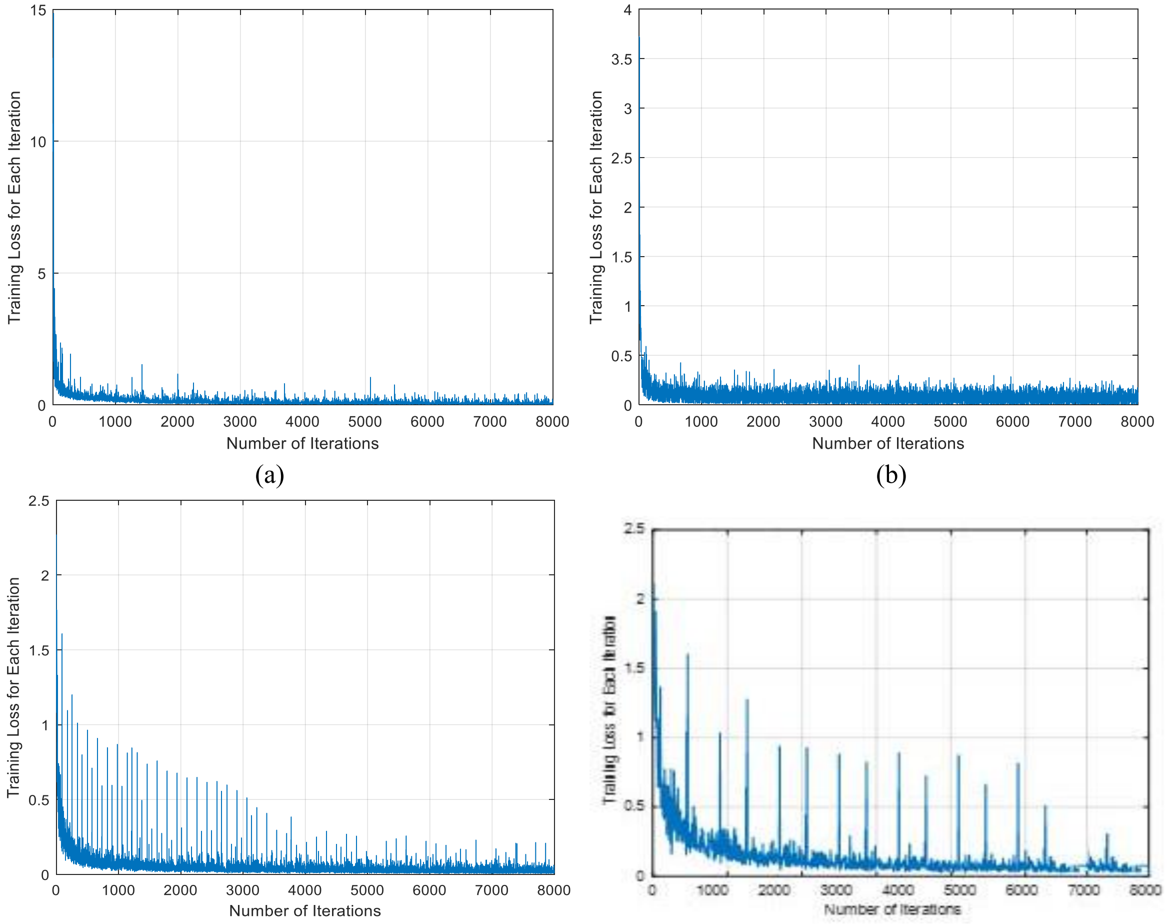
4.5 Loss function for the training process

Multitasking loss in deep learning is expressed using a loss function. The loss function for an image is defined as:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N}\sum i\, L_{cls}(p_i, p_i{}^*) + \alpha \frac{1}{N_{reg}}\sum i p_i{}^* L_{reg}\,(t_i, t_i{}^*) \tag{6}$$

Here, i is the index of an anchor in a mini-batch and p_i is the positive probability value for the example. L_{cls} represents the classification loss for all samples.

The loss function is made up of a classification loss and a regression loss, and represents the difference between the predicted value for the model and the training sample. The smaller the value, the closer the predicted sample is to the real sample, and the better the robustness of the model [41].



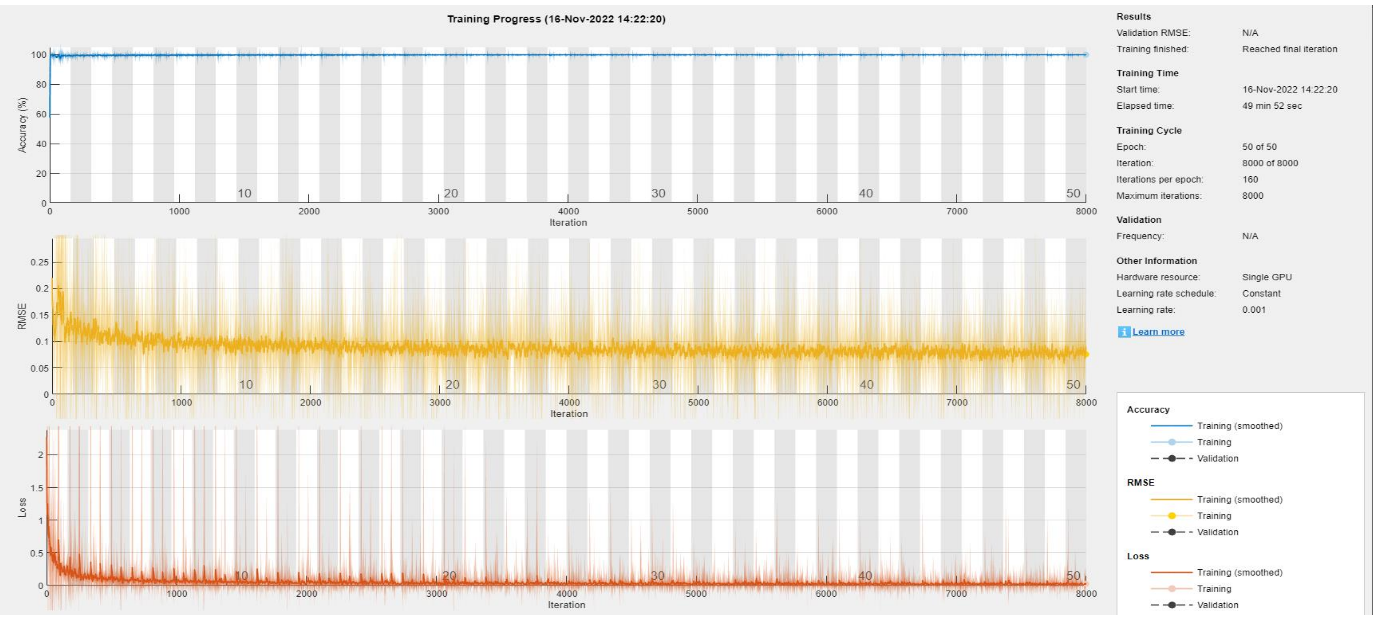
(a)

(b)

(c)

(d)

Figure 7. Total loss while training the network: (a) AlexNet; (b) Resnet 101; (c) Resnet-50; (d) YOLO



Results

Validation RMSE: N/A

Training finished: Reached final iteration

Training Time

Start time: 16-Nov-2022 14:22:20

Elapsed time: 49 min 52 sec

Training Cycle

Epoch: 50 of 50

Iteration: 8000 of 8000

Iterations per epoch: 160

Maximum iterations: 8000

Validation

Frequency: N/A

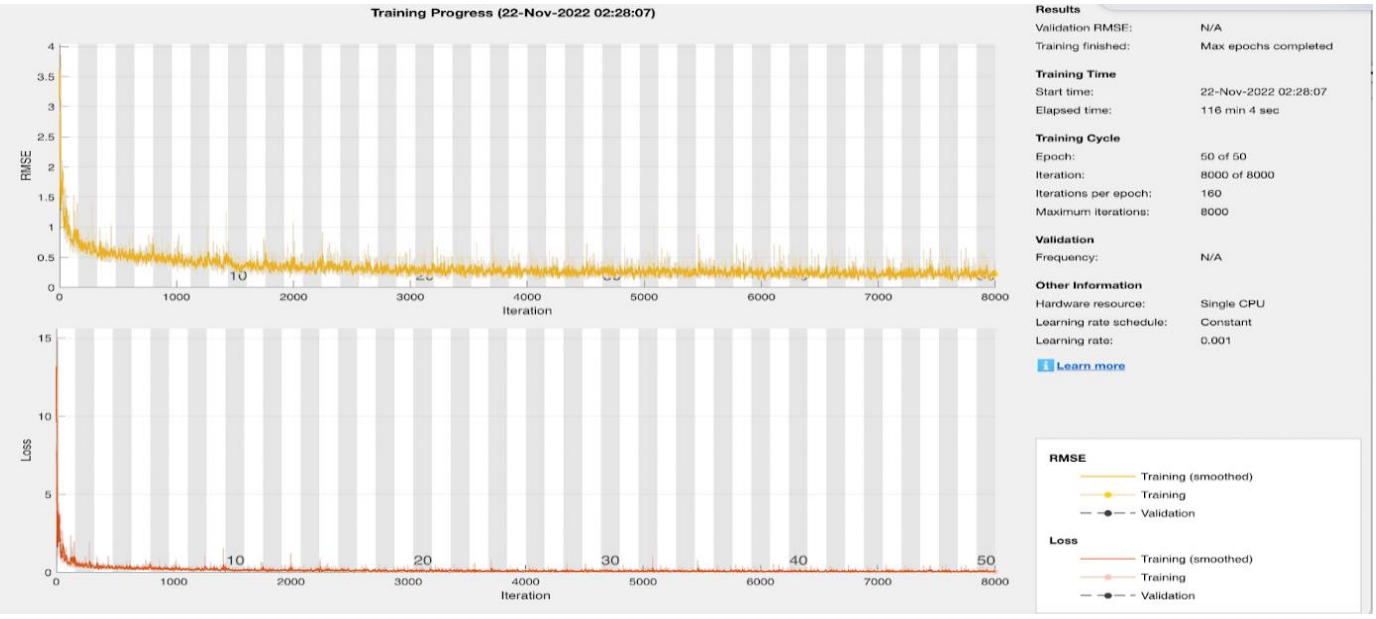
Other Information

Hardware resource: Single GPU

Learning rate schedule: Constant

Learning rate: 0.001

(a)



Results

Validation RMSE:	N/A
Training finished:	Max epochs completed

Training Time

Start time:	22-Nov-2022 02:28:07
Elapsed time:	116 min 4 sec

Training Cycle

Epoch:	50 of 50
Iteration:	8000 of 8000
Iterations per epoch:	160
Maximum iterations:	8000

Validation

Frequency:	N/A

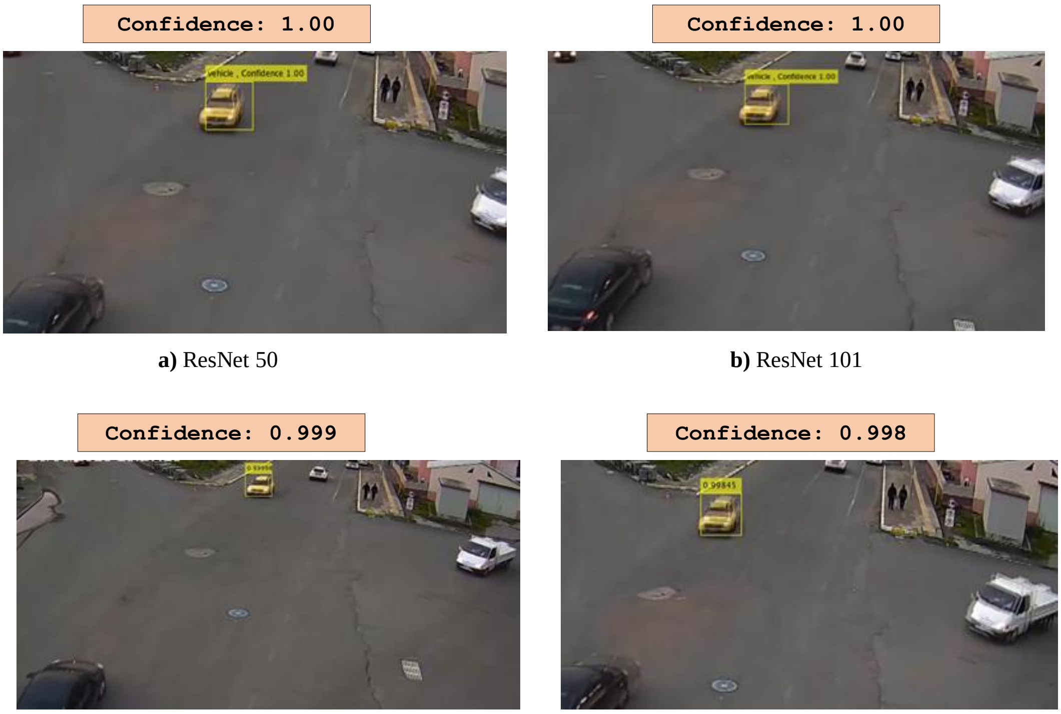
Other Information

Hardware resource:	Single CPU
Learning rate schedule:	Constant
Learning rate:	0.001

Learn more

(b)

Figure 8. Training progress for two algorithms: (a) MobileNet network trained with YOLO; (b) Resnet 101 network trained with Faster RCNN



a) ResNet 50

b) ResNet 101

c)VGG 19

d) YOLO

Figure 9. Vehicle images detected with different models: (a) ResNet 50; (b) ResNet 101; (c) VGG 19; (d) YOLO

Figure 7 shows the variation in the loss function over the last 8,000 iterations of the system, which allows us to observe the details of the oscillation and convergence of the curve. The loss value for the overall training process is constantly decreasing, and there is no negative trend of up and down vibration over the whole process; it therefore seems that the network parameters for model training are optimally selected.

When deep learning networks are trained, information can be obtained about the training progress of the network by monitoring. This information includes the progress of the network in terms of accuracy, how fast it is developing, and the loss rate for the training data of the network. Figure 8 shows the training progress over 8,000 iterations for the Faster RCNN and YOLO networks, as these were found to give the best results after comparing the different deep learning models.

Target tracking and then target detection were performed on the video recordings obtained from the video surveillance systems for a given route. The target vehicle selected in the first video was detected in the second video, where the vehicle was determined to pass along the same road route. The different deep learning networks were trained on the task of vehicle detection and their results were compared, and it was found that the vehicles were detected correctly with an accuracy of 99% and above. Images of the detected vehicles are shown in Figure 9.

5. CONCLUSION

In this study, we have presented a new approach to automatic target recognition for video surveillance systems. The selected target vehicle was detected from multiple video images taken from different cameras along the same route, from traffic surveillance systems called MOBESE. An improved target tracking method was developed to obtain ground truth data from the video in which the target was first detected. With this approach, ground truth data can be created automatically, and this method can also be used as a new tracking algorithm for image labeling applications. The target vehicle was detected with a high level of accuracy by training with deep learning methods in the video images obtained from other cameras along the routes followed by the target vehicle.

Based on the mAP results obtained in the study, the order of effectiveness of the models was as follows: Faster RCNN ResNet 101 > Faster RCNN ResNet 50 > YOLO > Faster RCNN VGG19 > Faster RCNN VGG16 > Faster RCNN AlexNet > Faster RCNN GoogleNet > RCNN.

The most successful results were obtained with the Faster RCNN and YOLO models. It has been reported in previous studies that YOLO is sensitive to changes in illumination, and is incapable of recognizing small objects [35, 36]. For this reason, it was insufficient when used for images of receding vehicles and MOBESE images with strong variation in lighting. The detection results obtained with Faster RCNN were found to be more successful.

REFERENCES

[1] Akbulut, O. (2014). Video için bölgesel ortak değişim betimleyici tabanlı iyileştirilmiş hedef takibi. Doctoral Thesis, Kocaeli Ünivercity.

[2] Talu, F. (2010). İnsan hareketlerinin takibinde karşılaşılan problemlerin çözümüne yeni yaklaşımlar. Doctoral Thesis, Fırat Ünivercity.

[3] Yılmaz, M. (2008). Birden fazla kamera ile çoklu hedef takibi. Master's Thesis, Middle East Technical University.

[4] Jeon, H.J., Pham, C.C., Nguyen, V.D., Jeon, J.W. (2018). High-speed car detection using resnet-based recurrent rolling convolution. In 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 286-291. https://doi.org/10.1109/SMC.2018.00059

[5] Organization of Motor Vehicle Manufacturers. (2022). 2022 production statistics. http://www.oica.net/category/production-statistics/2017-statistics/.

[6] Fernández, J., Cañas, J.M., Fernández, V., Paniego, S. (2021). Robust real-time traffic surveillance with deep learning. Computational Intelligence and Neuroscience, 2021: 4632353. https://doi.org/10.1155/2021/4632353

[7] Ergezer, H. (2007). Hareketli Nesnelerin Görsel Tespiti ve İzlenmesi. Master's Thesis, Middle East Technical University.

[8] Harris, C., Stephens, M. (1988). A combined corner and edge detector. In Alvey Vision Conference, 15(50): 10-5244.

[9] Kanade, T., Collins, R., Lipton, A., Burt, P., Wixson, L. (1998). Advances in cooperative multi-sensor video surveillance. In Proceedings of DARPA Image Understanding Workshop, 1(3-24): 2.

[10] Maity, M., Banerjee, S., Chaudhuri, S.S. (2021). Faster r-cnn and yolo based vehicle detection: A survey. In 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, pp. 1442-1447. https://doi.org/10.1109/ICCMC51019.2021.9418274

[11] Luo, J., Fang, H., Shao, F., Zhong, Y., Hua, X. (2021). Multi-scale traffic vehicle detection based on faster R–CNN with NAS optimization and feature enrichment. Def. Technol., 17(4): 1542-1554. https://doi.org/10.1016/J.DT.2020.10.006

[12] Wang, F., Li, Y., Wei, Y., Dong, H. (2020). Improved faster rcnn for traffic sign detection. In 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, pp. 1-6. https://doi.org/10.1109/ITSC45102.2020.9294270

[13] Alpatov, B.A., Babayan, P.V., Ershov, M.D. (2018). Vehicle detection and counting system for real-time traffic surveillance. In 2018 7th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, pp. 1-4. https://doi.org/10.1109/MECO.2018.8406017

[14] Fei, M., Li, J., Liu, H. (2015). Visual tracking based on improved foreground detection and perceptual hashing. Neurocomputing, 152: 413-428. https://doi.org/10.1016/j.neucom.2014.09.060

[15] Kiratiratanapruk, K., Dubey, P., Siddhichai, S. (2005). A gradient-based foreground detection technique for object tracking in a traffic monitoring system. In IEEE Conference on Advanced Video and Signal Based Surveillance, Como, Italy, pp. 377-381. https://doi.org/10.1109/AVSS.2005.1577298

[16] Liu, Y., Lu, Y., Shi, Q., Ding, J. (2013). Optical flow based urban road vehicle tracking. In 2013 Ninth International Conference on Computational Intelligence and Security, Emeishan, China, pp. 391-395. https://doi.org/10.1109/CIS.2013.89

[17] Sushmitha, S., Satheesh, N., Kanchana, V. (2020). Multiple car detection, recognition and tracking in traffic. In 2020 International Conference for Emerging Technology (INCET), Belgaum, India, pp. 1-5. https://doi.org/10.1109/INCET49848.2020.9154107

[18] Arinaldi, A., Pradana, J.A., Gurusinga, A.A. (2018). Detection and classification of vehicles for traffic video analytics. Procedia Computer Science, 144: 259-268. https://doi.org/10.1016/j.procs.2018.10.527

[19] Fan, Q., Brown, L., Smith, J. (2016). A closer look at Faster R-CNN for vehicle detection. In 2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, Sweden, pp. 124-129. https://doi.org/10.1109/IVS.2016.7535375

[20] Cao, C., Wang, B., Zhang, W., Zeng, X., Yan, X., Feng, Z., Liu, Y., Wu, Z. (2019). An improved faster R-CNN for small object detection. IEEE Access, 7: 106838-106846. https://doi.org/10.1109/ACCESS.2019.2932731

[21] Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., Hu, S. (2016). Traffic-sign detection and classification in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas,

NV, USA, pp. 2110-2118. https://doi.org/10.1109/CVPR.2016.232

[22] Li, J., Liang, X., Wei, Y., Xu, T., Feng, J., Yan, S. (2017). Perceptual generative adversarial networks for small object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, Honolulu, HI, USA, pp. 1222-1230. https://doi.org/10.1109/CVPR.2017.211

[23] Arcos-García, Á., Álvarez-García, J.A., Soria-Morillo, L.M. (2018). Evaluation of deep neural networks for traffic sign detection systems. Neurocomputing, 316: 332-344. https://doi.org/10.1016/j.neucom.2018.08.009

[24] Han, X., Chang, J., Wang, K. (2021). Real-time object detection based on YOLO-v2 for tiny vehicle object. Procedia Computer Science, 183: 61-72. https://doi.org/10.1016/j.procs.2021.02.031

[25] Bie, M., Liu, Y., Li, G., Hong, J., Li, J. (2023). Real-time vehicle detection algorithm based on a lightweight You-Only-Look-Once (YOLOv5n-L) approach. Expert Systems with Applications, 213: 119108. https://doi.org/10.1016/j.eswa.2022.119108

[26] Li, Y., Chen, Y., Yuan, S., Liu, J., Zhao, X., Yang, Y., Liu, Y. (2021). Vehicle detection from road image sequences for intelligent traffic scheduling. Computers and Electrical Engineering, 95: 107406. https://doi.org/10.1016/j.compeleceng.2021.107406

[27] Rafique, S., Gul, S., Jan, K., Khan, G.M. (2023). Optimized real-time parking management framework using deep learning. Expert Systems with Applications, 220: 119686. https://doi.org/10.1016/j.eswa.2023.119686

[28] Azimjonov, J., Özmen, A. (2021). A real-time vehicle detection and a novel vehicle tracking systems for estimating and monitoring traffic flow on highways. Advanced Engineering Informatics, 50: 101393. https://doi.org/10.1016/j.aei.2021.101393

[29] Aslam, N., Sharma, V. (2017). Foreground detection of moving object using Gaussian mixture model. In 2017 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, pp. 1071-1074. https://doi.org/10.1109/ICCSP.2017.8286540

[30] Xie, T., Li, X., Zhang, X., Hu, J., Fang, Y. (2021). Detection of Atlantic salmon bone residues using machine vision technology. Food Control, 123: 107787. https://doi.org/10.1016/j.foodcont.2020.107787

[31] Girshick, R. (2015). Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, pp. 1440-1448. https://doi.org/10.1109/ICCV.2015.169

[32] Ren, S., He, K., Girshick, R., J. Sun, J. (2017). Faster RCNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6): 1137-1149. https://doi.org/10.1109/TPAMI.2016.2577031

[33] Chorowski, J.K., Bahdanau, D., Serdyuk, D., Cho, K., Bengio, Y. (2015). Attention-based models for speech recognition. NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems, pp. 577-585.

[34] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779-788. https://doi.org/10.1109/CVPR.2016.91

[35] Xue, Y., Ju, Z., Li, Y., Zhang, W. (2021). MAF-YOLO: Multi-modal attention fusion based YOLO for pedestrian detection. Infrared Physics & Technology, 118: 103906. https://doi.org/10.1016/j.infrared.2021.103906

[36] Huang, R., Pedoeem, J., Chen, C. (2018). YOLO-LITE: A real-time object detection algorithm optimized for non-GPU computers. In 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, pp. 2503-2510. https://doi.org/10.1109/BigData.2018.8621865

[37] Toğaçar, M., Ergen, B., Özyurt, F. (2020). Evrişimsel sinir aği modellerinde özellik seçim yöntemlerini kullanarak çiçek görüntülerinin siniflandirilması. Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 32(1): 47-56. https://doi.org/10.35234/fumbd.573630

[38] Gürkahraman, K., Karakiş, R. (2021). Brain tumors classification with deep learning using data augmentation. Journal of the Faculty of Engineering and Architecture of Gazi University, 36(2): 997-1011. https://doi.org/10.17341/gazimmfd.762056

[39] Kaya, E.C. (2020). Yapay zeka tabanli drone optimizasyonu. Master's Thesis, Department of Electrical and Electronics Engineering, Graduate School of Natural and Applied Sciences, Gazi University.

[40] Qin, Z., Zhang, Z., Chen, X., Wang, C., Peng, Y. (2018). Fd-mobilenet: Improved mobilenet with a fast downsampling strategy. In 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, pp. 1363-1367. https://doi.org/10.1109/ICIP.2018.8451355

[41] Jiang, D., Li, G., Tan, C., Huang, L., Sun, Y., Kong, J. (2021). Semantic segmentation for multiscale target based on object recognition using the improved Faster-RCNN model. Future Generation Computer Systems, 123: 94-104. https://doi.org/10.1016/j.future.2021.04.019