Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 10, No. 4, 2022, 579-588.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

Vehicle Type, Color and Speed Detection Implementation by Integrating VGG Neural Network and YOLO algorithm utilizing Raspberry Pi Hardware

Mojtaba Nasehi[1], Mohsen Ashourian[2*] and Hossein Emami [2]

1. Faculty of Electrical Engineering, Islamic Azad University Majlisi Branch, Isfahan, Iran
3. Islamic Azad University Isfahan Branch(khorasgan), Isfahan, Iran

Article Info

Article History:
Received 12 May 2022
Revised 12 August 2022
Accepted 07 September 2022

DOI:10.22044/jadm.2022.11915.2338

Keywords:
Vehicle type detection, Hardware implementation, Neural network, Raspberry hardware board.

**Corresponding author: mnasehi968@gmail.com (M. Ashournia).*

Abstract

Vehicle type recognition has been widely used in practical applications such as traffic control, unmanned vehicle control, road taxation, and smuggling detection. In this work, various techniques such as data augmentation and space filtering are used to improve and enhance the data. Then a developed algorithm that integrates VGG neural network and the YOLO algorithm are used to detect and identify the vehicles. Then the implementation on the Raspberry hardware board and practically through a scenario is mentioned. The real including image datasets are analyzed. The results obtained show the good performance of the implemented algorithm is in terms of detection performance (98%), processing speed, and environmental conditions, which indicates its capability in practical applications with low cost.

1. Introduction

Vehicle type detection has gained special attention in the recent years. Different practical applications such as traffic control, smart city management, unmanned vehicles controls, road taxes, criminal vehicle detection, and smuggle detection are considered in this domain. Threshold level determination [1, 2], image edge detection [3, 4], feature integration and motion tracking [4-6], data analysis [6], image sequence analysis [6], and neural network-based methods [11, 12] are among the important algorithms and methods that have been developed and utilized in the vehicle type detection application that have some limitations in performance and implementation. In this paper, the proposed method is based on integrating the VGG neural network and YOLO algorithm by which a better theoretical performance is reached in comparison with the other methods such as the RCNN neural network. The proposed method has gained about 98% true detection performance in theory. From another viewpoint, hardware implementation of algorithms and methods are of great importance. Some algorithms are hard to be implemented since they have bottlenecks in hardware resource utilization and speed of convergence. Hardware platforms such as field programmable gate array (FPGA), digital signal processors (DSP), and single board computers (SBC) have limited internal hardware/software resources [17, 18]. In this work, we implement our proposed method on the Raspberry single board hardware, and analyze its performance in a real scenario, and compare it with another method. Practical implementation through real scenario utilizing image datasets shows a good performance of our implemented method and its capability to be utilized in single board computers. Efficient detection probability, low false alarm probability, fast running time, and being nearly an all-weather algorithm are features of the proposed method.

The rest of this paper is organized as what follows. In Section 2, the related works, and in Section 3, the proposed method is introduced and the implementation platform is described. The implementation results are provided in Section 4, and finally, the paper is concluded in Section 5.

2. Related Works

The methods that exist to identify the type and location of the vehicle are divided into three categories [10]:

2.1. Traditional method based on use of simple visual operators:

- Determining threshold value on traffic images

This method is the simplest method in detecting a moving vehicle, and is based on the fact that the moving object has a different light intensity than the background image. By setting a light intensity threshold in a small area of the image, the car can be distinguished from the background image. This method is highly dependent on the threshold value, which is determined based on the optical density of the image of a specific vehicle and its background image model. In order to optimize the threshold value, light change counting can be used but the detection and misdetection of vehicles is inevitable due to the shadows moving with the vehicle during the day that have the same light intensity as the surrounding environment [1].

- Detecting edges of traffic image.

The performance of this method is based on the features that exist in the edges of the target object in the digital image taken from the video surveillance cameras of the vehicle. Considering that the comparison method between two images is not related to two frames, this method can also detect stopped vehicles. The morphological moving object edge detection methods have wide applications. In this method, the edges of the vehicles are displayed prominently compared to the edges of the road. By using the histogram of the image, along with a series of morphological operations, the features of the edges of the image can be obtained in the spatial domain. The methods of revealing three basic types of discontinuities in image intensity, i.e. point, line, and edge, can be evaluated in this method. The most common way to search for discontinuity in this method is to apply a mask to the image.

A typical edge may be the boundary between a red section and a black section, whereas a line can be a small number of discolored pixels on a uniform background. There will be an edge on each side of the line. Edges play an important role in image processing applications [3].

2.2. Methods based on image feature extraction and an intelligent classification system:

- **Data analysis technique**

This method is adapted in the algorithm by analyzing data and using many images, having vehicles or not. Then the meaningful features are extracted from the image and the machine detects where the vehicle is located or if it is not there at all [4].

- Analysis of a sequence of images

In this method, it is assumed that the scene is dynamic. The machine has three ways to find the vehicle:

1) Remove the background with specific methods,
2) Extract the features with methods,
3) Separate different formats in the image and recognize moving objects in the image.

In this method, the images related to the camera and the three-dimensional points of the scene are simultaneously estimated in the form of a state vector in such a way that the image background is removed first, then the image features are extracted, and the three-dimensional position of the image is determined. The camera is connected to those points, and they are determined with uncertainty in the depth of 0.5 to 5 m with a uniform distribution, then in each new frame the uncertainty is reduced by repeating the features [6].

- **Use of Haar-like feature**

In this method, the main idea is as follows. According to the simple features that the classifier has organized in several stages, an object is defined in a cascade manner (or in a step-by-step manner). The proposed system works based on the artificial neural network. This method organizes the image in a cascade classification, and then passes through a series of filters to obtain the original image [7].

The methods based on feature extraction were used until around 2015, and had an accuracy of about 70%. The main challenges of these methods were the multiplicity of images in the input, changes in relation to light, errors in recognizing high-speed targets, and the low power in detecting image components [7].

2.3. Methods based on convolutional neural network

- **Vehicle type identification based on R-FCN network**

The idea of this method is based on the deep learning network, and it uses the R-FCN framework and the combination of vehicle database in ImageNet. In this method, online hard sampling is used to optimize the network parameters. After repeated iterations to train the network, the R-FCN model is finally obtained from vehicle target detection. According to the simulation results of this method, 87.48% is obtained [11].

- **Vehicle type classification based on convolutional neural network (CNN)**

In this method, vehicle type classification is presented based on the deep learning technique. This system consists of two stages. In the first step, they use data augmentation to increase the dataset. In the second stage, they used a CNN. CNN has different models and architectures. The network is trained using the parameters obtained from the dataset. This system is part of an integrated program that enables automatic management of traffic signals based on the type of automatic vehicle detection. In this method, according to the use of the torsion network, the accuracy is about 92% [12].

In the methods based on CNN, the accuracy of object detection reaches over 90%, and to some extent the challenges of the past methods are solved but the computational load increases, which slows down the speed of object detection [10].

3. Proposed Method

The proposed method uses a combination of VGG neural network and the YOLO algorithm, which is a suitable solution to increase the processing speed (reduce computational load) and solve challenges such as weather conditions, and diversity in a variety of vehicles. The images are very diverse from different angles of vehicles, which, in turn, increases the accuracy (object recognition). In addition, this system can speed up the process of identifying vehicles on highways and when crossing the checkpoint. This method uses different steps, which are image capture, location (area identification), vehicle type detection, classification, and evaluation. The following figure shows the training and testing process in this algorithm.

According to the Figure 1, the proposed method consists of three parts:

Images registration

One practical aspect of image processing is the processing of moving images or video. At this stage, different videos have been prepared from all over the country that are in different weather conditions. In the previous methods, the problems such as the geographical conditions of Iran were not considered, which has caused challenges such as reduced accuracy in weather, mountainous conditions, and even intolerance to high light such as desert areas. Video is a set of frames in time. The camera captures images continuously. Thus direct processing in the video is not easy, and it is a difficult task. The video is then framed and converted into images. Therefore, by framing the video, databases are provided for training but for training the database, more images from different weather conditions are required. Therefore, to increase the accuracy in different situations, the data augmentation method is needed. The data augmentation technique helps us to have new images for training by producing new images artificially from the original images. Data augmentation involves making modified versions of images that still belong to the class and statistical distribution of the original images. The goal of data augmentation is to increase the size of the collection by using new images that are plausible. Therefore, it is clear that the choice of data augmentation technique used must be done carefully and follow the content of the data and the type of problem. Modern deep-learning algorithms such as CNN can learn the features that are independent from their location in the image. However, amplifying and adding data by changing rotation, light, and color can further aid in the learning of independent features in the image. Eventually, the grid learns well to recognize the object correctly if it was turned from left to right in the image or had little light and clarity or many other things. Data augmentation works for in the following order:

- Vertical and horizontal image shifting
- Flip images from left to right and up and down
- Random image rotation
- Random image magnification
- Color space change technique
- Random wipe technique of using kernel filters
- Mixing images with noise
- Blurring images

The convolution of an image, for each pixel, processes a weighted sum of the values of that pixel and its neighbours. Depending on the choice of different weights, a wide range of image processing operations can be performed through convolution. Different canonical masks produce different results when applied to an input image. This filtering and masking operation is called a space filter. High-pass filters complement and contrast low-pass filters, meaning that they also maintain or improve high-frequency components with potential noise pixel enhancement effects. High-frequency components include fine details, dots, lines, and edges.

In other words, it highlights changes in image brightness. Using the transient filter in the proposed method, the problem of separating objects in an image is solved by using a canonization mask or in general the space filter that is applied to the image. Applying a convolution mask to the images causes the image to become smaller and the grid to look at the image as a whole. In this way, the images are prepared for pre-processing using such techniques.

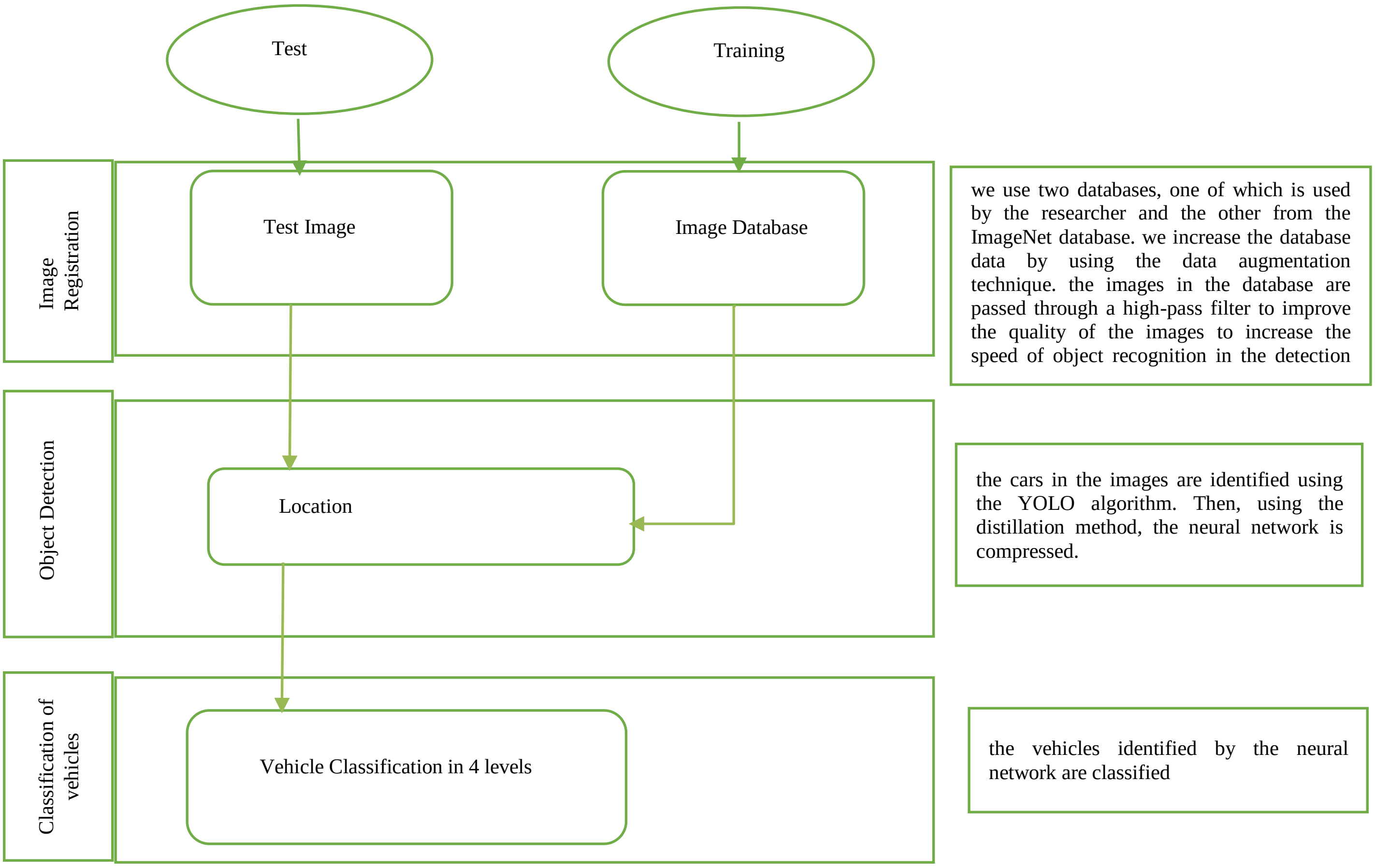

Figure 1. Steps for identifying and locating vehicles.

Object detection

All deep learning networks are object classifiers, and are not able to detect objects. Therefore, there is a need for an algorithm that can be implemented in these networks and can detect the object. The most common deep learning algorithms used to identify the type of vehicle are Yolo, SSD, Small Face, and Small Sample Segmentation Techniques, including U-Net and R-CNN. However, among the existing techniques, the Yolo algorithm has been considered due to its high speed in object recognition. This algorithm is an object recognition system. The most complete real-time system for in-depth learning and problem-solving. The algorithm first divides the image into different sections and marks each section, then runs the recognition algorithm in parallel for all of these sections to see which category each section belongs to. After fully identifying the objects, it connects them so that the two of each main object are a box. All this is done in parallel. As a result, it is real-time, and can process up to 40 images per second.

Vehicle classification

The proposed algorithm for classifying each vehicle into four levels (passenger, chassis, truck, and van) must be implemented on the neural network using the convolution neural network to perform this operation. The common convolutional neural networks that exist are Image Network, AlexNet Network, ZFNet Network, Google Network, and Resnet Network. However, among the neural networks mentioned, the VGG network is currently the most popular choice for feature extraction from images. The VGG network is specially trained in advance, which is available in the library. It is a deep learning framework designed with a focus on speed and modularity. To estimate the power, this library can process more than 60 million images per day by moving between the CPU and the graphics card. To increase the processing speed of calculations, which was one of the serious challenges in the previous methods, the transfer learning method is used. In deep learning, transitional learning is a technique in which a neural network model is trained on a problem similar to a problem that is being solved.

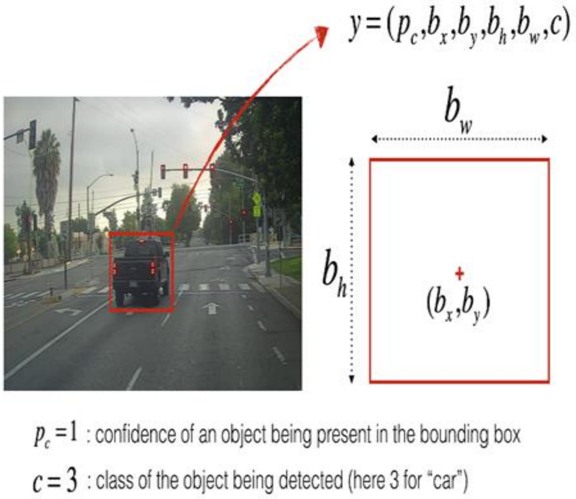

Figure 2. Applying neural network to vehicles in vehicle classification section.

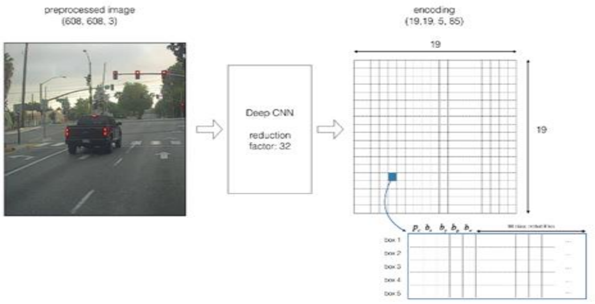

Figure 3. How to apply neural network on vehicles to classify vehicles into four levels.

Then one or more layers of the trained model are used in a new model on another issue. Transitional learning has the advantage of reducing the training time for a neural network model, and can reduce the generalization error. The pre-trained network weights are used as a starting point for the new network training process. This considers transitional learning as a kind of initial weight design. It is useful when the pre-trained network has much more labeled data than the current problem, and also the problem structure is similar in both areas. By freezing the middle edges of the neural network, this method speeds up the processing of calculations and reduces its load.

Implementation platform

To implement, due to the hardware available in the market, the Raspberry Pi board has been used. Raspberry Pi is a small computer that has been in development since 2006 and its parts are mounted on a motherboard the size of a bank card. Designing and building a low-cost, highly flexible computer was the brainchild of a group of computer programmers such as Eben Upton and David Braben, both members of the Raspberry Pi Foundation. The Raspberry Pi model used to implement the proposed method is Pi3B + Raspberry. This model is a great version of the modern Raspberry Pi models. This board is suitable for most projects because it has a 1.4 GHz processor, Bluetooth, Wi-Fi, and 4 USB ports. The technical characteristics of the board used are as follow:

Table 1. Specifications of Raspberry Pi board

Weight	Dimensions	Power	Chip	GPU
45 Gram	85.6 – 56.5- 17 m.m	2.295 - 5.661Watt	Broadcom BCM2837B0	Video Core IV

The operating system used is the Razbin operating system, which is in both Lite and Desktop versions. The Lite version has only a command line and has a smaller volume than the desktop version with a graphical interface. However, the desktop version has been used for implementation.

Python programming has been used to implement the proposed method (combination of YOLO algorithm and VGG neural network). This language is ideally designed for fast sampling of complex applications. The Keras library has been used for implementation. The Keras library is an open-source neural network library written in Python. This library runs on TensorFlow Microsoft Cognitive Toolkit, R Theano, or PlaidML libraries. In the implementation, this library allows us to define and teach neural network models in just a few lines of code. Then we install the VGG neural network and the YOLO algorithm using the existing libraries. A large number of parameters leads to better performance, and at the same time, higher computational costs. High costs will affect the training time and server costs. However, training is only the first part of the neural network life cycle. To optimize these costs, neural network compression is used .

- •Weight pruning
- •Quantify
- •Distillation

In neural network compression, there are three methods of pruning, quantization, and distillation. The distillation method has been used in our implementation. Although quantization and pruning can be effective, they do some damage. The basic idea of this approach is simple: train a large model (teacher) to achieve good performance, and then use its predictions to teach a smaller network (student). This smaller grid estimates the main function well learned by the deep grid.

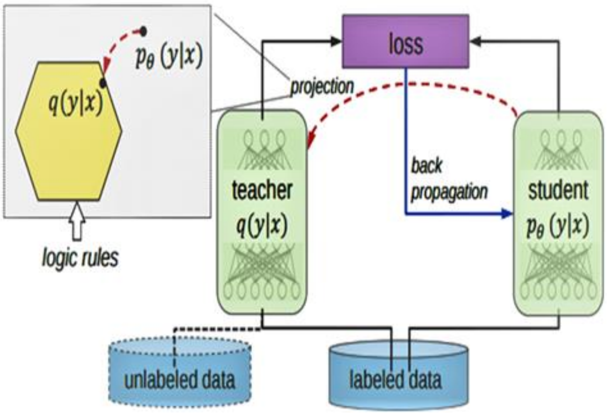

Figure 4. Graph of distillation method calculations.

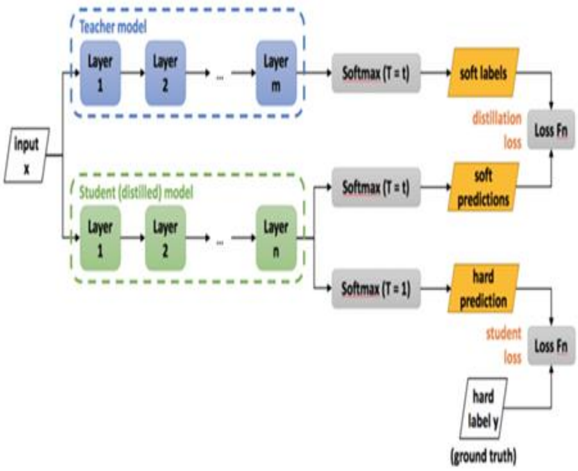

Figure 5. Distillation method implementation diagram.

Using the probabilities generated for each class, which are produced by the heavy model as soft targets for the purpose of teaching the small model, the generalizability of the heavy model can be transferred to a smaller model. In the transfer phase, we can use the same training kit or a "transfer kit" to teach the heavy model. When the heavy model is a set of simpler models, we can use the arithmetic mean or geometric mean of the predictive distributions of each of those models as soft targets. When the entropy of soft targets is high, they provide much more information to teach each item and much less variance between them (compared to hard targets). Thus the small model can be trained on much less data than the heavy model needs, and at the same time achieve a higher learning rate. With the distillation method, the inference time of the distilled models, and not their training time, can be improved. This is the main difference between this method and the previous two methods because training can be very costly.

Recognizing color

In the proposed method, a histogram of images is used to detect the color and track the speed of the vehicle. Image histogram provides important information about the image, and has many applications in image processing. The image histogram shows the frequency of light intensity levels in the image. An image histogram is a two-dimensional graph that measures the frequency levels of the brightness of an image pixel. As shown in the figure below, the x-axis indicates the range of light intensity changes and the y-axis the frequency (number) of light intensity levels.

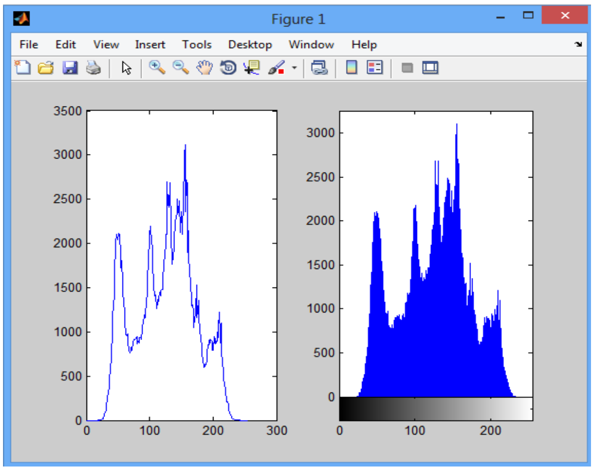

Figure 6. Histogram of vehicle images.

Due to the frequency in the histogram, the color of the vehicle is detected.

Speed tracking

However, to determine the speed of the vehicle, by calculating the displacement of the pixels in several frames, the speed of the vehicle can also be calculated.

Figure 7. Vehicle speed tracking.

4. Implementation results

To prove the performance of this device, 70,000 images of passing vehicles have been used and taught from various videos of different weather conditions, day and night in the country. In addition, the multi-threading technique has been used to reduce the program execution time and reduce the resource consumption in programming and implementation. The multi-threading technique can significantly improve the performance of any program. The multi-threading programming mechanism in Python is very simple and easy to learn.

Benefits of multi-threading programming:

• Increases code productivity and speeds up the process.

• Allows us to stay responsive in the program while performing the input and output operations. This feature is most noticeable in graphical applications (GUI).

Now to better examine the proposed method, we also implement the vehicle detection method based on the RCNN neural network and examine the performance of both. In consecutive tests with high working hours of the device, in the proposed method, the percentage of vehicle loss is less than 2% but in the method of vehicle detection based on the RCNN neural network is about 9%. The waste function actually displays the amount of error each time the neural network is executed for the training data. The loss function is calculated as follows:

$$\mathrm{Loss} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i=1}}^{\mathrm{N}}(\mathrm{y_P} - \mathrm{y_t})^2 \quad (1)$$

In addition to having a good working continuity in this device, the probability of non-detection of the vehicle has been reduced to the lowest possible value, and the efficiency of the proposed method is about 97%. In this system, because from the VGG convolution neural network with YOLO algorithm, data training with data augmentation technique and the use of spatial filter increases the accuracy, computational processing speed and thus improves accuracy performance (detection) Object) which is about 96% but in the other method implemented the efficiency is about 91%.

$$\mathrm{Accuracy} = \frac{\mathrm{T_R}}{\mathrm{T_R}} \quad (2)$$

In the proposed method, about 98 images per 100 images correctly detect that the accuracy criterion is about 98% but in the RCNN neural network method, about 90 images per 100 images correctly recognize that the accuracy criterion is about 90%.

$$\mathrm{Precision} = \frac{\mathrm{T_R}}{\mathrm{T_R}} + F_R \quad (3)$$

However, in the proposed method, due to the use of transfer learning techniques, distillation, multi-threading, and YOLO algorithm, the execution time of each algorithm is drastically reduced to about 0.1 seconds. However, in the RCNN neural network method, it is about 0.25 seconds. Execution time of each algorithm One way to estimate the performance of an algorithm is to count the number of main operations required to reach the output. In the proposed method, the learning rate (processing speed) is about 0.0010, which is divided by 10 once every 50 rounds and becomes smaller over time. The learning rate is often denoted by the symbol α and sometimes by the symbol η, and indicates the speed (step) of updating the weights, which can be a fixed or adaptable value.

Mean Squared Error (MSE)—It is a measure of the difference between the estimated value and the actual value. The mean squared error is the squared error averaged over the M*N array.

$$\mathrm{MSE} = \frac{1}{MN}\sum\sum\left(F_1(i \cdot j) - F_2(i \cdot j)\right)^2 \quad (4)$$

i = 1, j = 1

where f1 is the output image, and f2 is the input image. The smaller value of MSE, indicated the better performance of the algorithm. Another related parameter is root MSE (RMSE) that is obtained according to Equation (5):

$$\mathrm{RMSE} = \sqrt{MSE} \quad (5)$$

Peak Signal to Noise Ratio (PSNR)—It is the ratio of the maximum possible powers of signal to that of the power of noise. It can also be represented as a log function of maximum value of image and mean squared error.

$$\mathrm{PSNR} = 10\mathrm{Log}\left(\frac{255^2}{\mathrm{MSE}}\right) \quad (6)$$

The PSNR value should be high.

Correlation Coefficient (COC)—It is the measure between the predicted values and the actual values ca

lculated earlier. The value of correlation coefficient lies between 0 and 1. The value of the correlation coefficient is 0 when there is no match between the values. It increases as the relationship strength between the predicted values and actual values increases. A perfect match of the values will have coefficient of 1.0. The maximum value of correlation is preferred here. COC is calculated using Equation (7):

$$\mathrm{Correlation(r)} = \frac{\mathrm{N}\sum\mathrm{XY}-(\sum\mathrm{X})(\sum\mathrm{Y})}{\mathrm{sqrt([N}\sum\mathrm{X2}-(\sum\mathrm{X})2][\mathrm{N}\sum\mathrm{Y2}-(\sum\mathrm{Y})2])} \quad (7)$$

where,

N = number of pixel of the image

X = input image, Y = output

In Table 2 and Figure 8, the calculated values of PSNR, RMSE and COC for the proposed method and RCNN method are depicted.

Table 2. Values calculated in two implemented methods.

Method	PSNR	RMSE	COC
Proposed method	38.0135	10.3545	0.0456
RCNN method	34.4664	23.4339	-0.0521

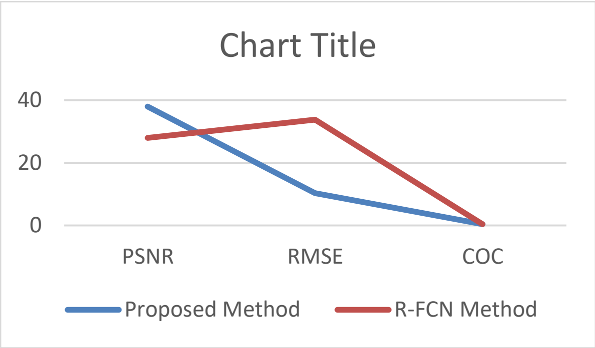

Figure 8. Trends of PSNR, RMSE, and COC for two implemented methods.

Table 3. Results of proposed method.

Vehicle type	Loss	Accuracy	Rate learning
Riding	0.3749	%97.97	0.0010
Truck	0.3641	%98.22	0.0010
Long undercarriage	0.3689	%98.55	0.0010
Bus	0.3687	%97.63	0.0010

Table 4. Results of R-CNN method.

Vehicle type	Loss	Accuracy	Rate learning
Riding	0.7736	%87.47	0.02
Truck	0.7841	%86.79	0.02
Long undercarriage	0.7753	%88.13	0.02
Bus	0.7743	%87.48	0.02

Table 5. Results of calculations of proposed method.

Epoch	Iteration	Time elapsed	RMSE
1	1	00:00:04	
1	200	00:01:30	
2	400	00:02:56	0.05
3	600	00:04:20	
3	800	00:05:43	
4	1000	00:07:07	0.15
5	1200	00:08:30	0.11
5	1400	00:09:50	
6	1600	00:11:10	0.14
7	1800	00:12:30	
7	2000	00:13:50	0.14
7	2065	00:14:15	

Table 6. Results of calculations of R-CNN method.

Epoch	Iteration	Time elapsed	RMSE
1	1	00:00:03	0.088
1	200	00:01:32	
2	400	00:02:48	
3	600	00:04:27	
3	800	00:05:47	
4	1000	00:07:33	0.25
5	1200	00:08:23	0.22
5	1400	00:09:53	
6	1600	00:11:16	0.27
7	1800	00:12:37	0.26
7	2000	00:13:51	
7	2065	00:14:17	

This system can be used operationally on mountainous and desert routes. This system has properly shown its efficiency during the period of operation in various atmospheric and light conditions so that this system can perform its tasks even in snowy weather and blizzard.

To prove the performance of this device, 74040 images of passing vehicles have been tested in different weather conditions, day and night. At the same time, multi-threading technique has been used in order to reduce the program execution time and reduce the resources consumed in programming and implementation.

In the consecutive tests with high working hours, the percentage of vehicle loss in the proposed method is less than 3% but in the vehicle detection method based on R-CNN, it is about 9%.

At the same time, the presence of proper working continuity in this device has reduced the probability of not recognizing the vehicle to the lowest possible value, and the efficiency of the proposed method is about 97%. In this system, since the VGG convolution neural network is used together with the Yolo algorithm, the speed of calculation processing is increased, and as a result, it improves the performance of accuracy (object recognition) to about 97%. In other methods made, the efficiency is about 91%.

In the presented method, about 97 images out of every 100 images are recognized correctly; in fact, the accuracy standard is around 97% but in the R-CNN method, about 91 images are correctly recognized out of every 100 images; the accuracy standard is around 91%.

In the proposed method, due to the use of the distillation technique, multi-threading and YOLO algorithm, the execution time of each algorithm has been greatly reduced to about 0.1 seconds. However, in the method, it is about 0.25 seconds. The execution time of each algorithm is a method of estimating the efficiency of the algorithm, counting the number of main operations required to reach the output.

In the proposed method, the speed of the learning rate (processing speed) is about 0.0010, which is divided by 10 once every 50 rounds and becomes smaller over time. The learning rate is often represented by the symbol α and sometimes by the symbol η, and represents the speed (step) of updating the weights, which can be a fixed value or change to an adaptive one.

This proposed method can be used operationally in the mountainous and desert route. During the period of operation, this system has shown its efficiency in various atmospheric and light

conditions in such a way that this system is able to perform its tasks even in a snowy and blizzard weather.
In the figure below, image (a) is the image obtained from the camera, and image (b) are the reconstructed images in different conditions that are recognized by the device.

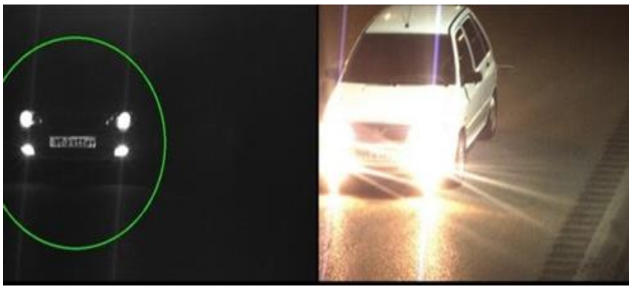

Figure 9. Detection of a vehicle in poor lighting conditions.

Figure 10. Detection of vehicle in snow conditions.

Figure 11. Detection of vehicle in fog and snow conditions.

In this method, about 7737 images at night and 3614 images in noise mode are used from videos that are framed; these videos are from different atmospheric and environmental conditions of our country. According to the results of the above figure, it can be seen that the presented approach to detect vehicles at night or images that have noise has a high accuracy of 89%. However, it also has a loss, which is caused by the similarity of the images of the vehicles, which causes the neural network to make a mistake but in the R-CNN method, the efficiency in such conditions is about 81%.
One of the problems of vehicle recognition is the imaging of a part of the vehicle, which will cause the type of vehicle not to be recognized. The proposed method can use wide imaging to identify and separate vehicles. In the proposed method, the sensitivity to the overlap between several vehicles has been reduced to the lowest possible state. They can be identified and separated to the extent that they can be seen.

Figure 12. Detection of a heavy vehicle by an implementation device.

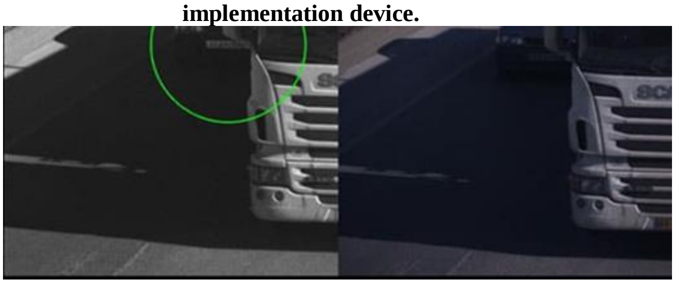

Figure 13. Separation of types of vehicles implemented by the system.

5. Conclusion

A new method for vehicle type detection was designed and implemented based on the integration of the VGG artificial neural network and the YOLO algorithm on the Raspberry Pi hardware platform. Implementation is based on the Python programming language and the Rasbin operating system. The proposed method uses a low hardware and a high speed. The implementation results show that the good performance of the proposed method is the possibility of misdiagnosis and warning. The proposed method has innovations compared to other domestic and foreign samples including increasing the detection accuracy by up to 98%, increasing the processing speed (reducing the computational load), separating the overlapping components, increasing the accuracy in different weather conditions, and reduced cost compared to other examples noted.

Acknowledgment

Thanks to Behyar Sanat Danesh Bonyan Company for helping us preparing the vehicle detection system project.

References

[1] B. Stewart, I. Reading, M. Thomson, T. Binnie, K. Dickinson, and C. Wan, "*Adaptive lane finding in road traffic image analysis*," in Seventh International Conference on Road Traffic Monitoring and Control, 1994., 1994, pp. 133-136: IET.

[2] Enkelmann, W. (1991). "*Obstacle detection by evaluation of optical flow fields from image sequences.*" Image and Vision Computing,1991, 9(3), 160-168.

[3] Won Y, Nam J, Lee BH. "*Image pattern recognition in natural environment using*

morphological feature extraction." InJoint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR) 2000 Aug 30 (pp. 806-815). Springer, Berlin, Heidelberg.

[4] Li J, Liu Y, Tageldin A, Zaki MH, Mori G, Sayed T. *"Automated region-based vehicle conflict detection using computer vision techniques."* Transportation Research Record. 2015 Jan;2528(1):49-59.

[5] Kim JB, Park HS, Park MH, Kim HJ. *"A real-time region-based motion segmentation using adaptive thresholding and K-means clustering."* InAustralian joint conference on artificial intelligence 2001 Dec 10 (pp. 213-224). Springer, Berlin, Heidelberg.

[6] Hadi RA, Sulong G, George LE. *"Vehicle detection and tracking techniques:"* a concise review. arXiv preprint arXiv:1410.5894. 2014 Oct 22.

[7] Oliveira M, Santos V. *"Automatic detection of cars in real roads using haar-like features."* Department of Mechanical Engineering, University of Aveiro. 2008 Jul;3810.

[8] Dubuisson MP, Jain AK. *"Contour extraction of moving objects in complex outdoor scenes."* International Journal of Computer Vision. 1995 Jan;14(1):83-105.

[9] Hicham B, Ahmed A, Mohammed M. *"Vehicle type classification using convolutional neural network."* In2018 IEEE 5th International Congress on Information Science and Technology (CiSt) 2018 Oct 21 (pp. 313-316). IEEE.

[10] Nasehi M, Ashourian M, Moalem P. *"An Overview of the Type of Vehicle Detection Techniques."* Majlesi Journal of Telecommunication Devices. 2020 Sep 1;9(3).

[11] Zhigang Z, Huan L, Pengcheng D, Guangbing Z, Nan W, Wei-Kun Z. *"Vehicle target detection based on R-FCN."* In2018 Chinese Control And Decision Conference (CCDC) 2018 Jun 9 (pp. 5739-5743). IEEE.

[12] Li Y, Song B, Kang X, Du X, Guizani M. *"Vehicle-type detection based on compressed sensing and deep learning in vehicular networks."* Sensors. 2018 Dec 19;18(12):4500.

[13] Wang X, Zhang W, Wu X, Xiao L, Qian Y, Fang Z. *"Real-time vehicle type classification with deep convolutional neural networks."* Journal of Real-Time Image Processing. 2019 Feb;16(1):5-14.

[14] Sheng M, Liu C, Zhang Q, Lou L, Zheng Y. *"Vehicle detection and classification using convolutional neural networks."* In2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS) 2018 May 25 (pp. 581-587). IEEE.

[15] Vedaldi A, Zisserman A. *"Vgg convolutional neural networks practical."* Department of Engineering Science, University of Oxford. 2016 Oct;66.

[16] Huang R, Pedoeem J, Chen C. *"YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers."* In2018 IEEE International Conference on Big Data (Big Data) 2018 Dec 10 (pp. 2503-2510). IEEE.

[17] Sharifi-Tehrani O. *"Hardware Design of Image Channel Denoiser for FPGA Embedded Systems."* Przegląd Elektrotechniczny. 2012 Jan 1;88(3b):165-7.

[18] Sharifi-Tehrani O. *"Novel hardware-efficient design of LMS-based adaptive FIR filter utilizing Finite State Machine and Block-RAM."* Przeglad Elektrotechniczny. 2011 Jul;87(7):240-4.

[19] Tang T, Zhou S, Deng Z, Zou H, Lei L. *"Vehicle detection in aerial images based on region convolutional neural networks and hard negative example mining."* Sensors. 2017 Feb 10;17(2):336.

پیاده سازی تشخیص نوع، رنگ و سرعت وسایل‌نقلیه با ادغام شبکه عصبی وی‌جی‌جی و الگوریتم یولو با استفاده از سخت افزار رزبری‌پای

مجتبی ناصحی[1]، محسن عشوریان[2,*] و حسین امامی[2]

[1] دانشکده مهندسی برق، دانشگاه آزاد اسلامی، واحد مجلسی، اصفهان، ایران.

[2] دانشگاه آزاد اسلامی واحد اصفهان (خوراسگان)، اصفهان، ایران.

ارسال ۲۰۲۲/۰۵/۱۲؛ بازنگری ۲۰۲۲/۰۸/۱۲؛ پذیرش ۲۰۲۲/۰۹/۰۷

چکیده:

تشخیص نوع وسیله نقلیه به طور گسترده در کاربردهای عملی مانند کنترل ترافیک، کنترل وسایل نقلیه بدون سرنشین، عوارضی‌ها و کشف قاچاق استفاده می‌شود. در سال‌های اخیر استفاده از شبکه‌های عصبی عمیق به‌عنوان ابزاری کارآمد در شناسایی باوجود تنوع شرایط محیطی و اجسام مطرح‌شده‌اند. اما همچنان بعضی از چالش‌ها مانند تعدد تصویر در یک صحنه، به‌هم‌پیوستگی تصویر وسیله نقلیه و زمینه تصویر، وجود نویز در تصاویر، تلرانس نسبت به تغییرات نور و افزایش بار محاسبات وجود دارد. در روش پیشنهادی از تکنیک‌های مختلفی مانند افزایش داده و فیلتر فضایی برای بهبود و ارتقای پایگاه داده تصاویر استفاده شده است، سپس یک الگوریتم توسعه یافته که شبکه عصبی وی‌جی‌جی و الگوریتم یولو را ادغام می‌کند برای تشخیص و طبقه‌بندی وسایل نقلیه استفاده می‌شود و در نهایت جهت پیاده سازی از برد سخت افزاری رزبری و عملاً از طریق یک سناریو ذکر شده استفاده می‌گردد. مجموعه داده‌های واقعی از جمله تصویر تجزیه و تحلیل می شوند. نتایج به‌دست‌آمده نشان می‌دهد تشخیص وسایل نقلیه افزایش یافته و به حدود ۹۸ درصد رسیده است، سرعت پردازش(کاهش بار محاسبات) کاهش یافته، قابلیت تعمیم با شرایط محیطی دارد که نشان‌دهنده قابلیت آن در کاربردهای عملی است و هزینه ساخت کمتر نسبت به دیگر نمونه‌های داخلی و خارجی دارد.

کلمات کلیدی: تشخیص نوع وسایل‌نقلیه، پیاده سازی سخت افزار، شبکه عصبی، برد سخت افزار رزبری‌پای.