ADVANCES IN CYBER-PHYSICAL SYSTEMS

Vol. 6, Num. 2, 2021

METHODS OF VEHICLE RECOGNITION AND DETECTING TRAFFIC RULES VIOLATIONS ON MOTION PICTURE BASED ON OPENCV FRAMEWORK

Yevhen Fastiuk, Ruslan Bachynskyy, Nataliia Huzynets

Lviv Polytechnic National University, 12, Bandera Str, Lviv, 79013, Ukraine.
Authors' e-mail: *yevfast@gmail.com, bac_ruslan@ukr.net, natdemchuk@ukr.net*

Submitted on 10.10.2021

***Abstract*: In this era, people using vehicles is getting increased day by day. As pedestrians leading a dog for a walk, or hurrying to their workplace in the morning, we've all experienced unsafe, fast-moving vehicles operated by inattentive drivers that nearly mow us down. Many of us live in apartment complexes or housing neighborhoods where ignorant drivers disregard safety and zoom by, going way too fast. To plan, monitor and also control these vehicles is becoming a big challenge. In the article, we have come up with a solution to the above problem using the video surveillance considering the video data from the traffic cameras. Using computer vision and deep learning technology we will be able to recognize violations of rules. This article will describe modern CV and DL methods to recognize vehicle on the road and traffic violations of rules by them. Implementation of methods can be done using OpenCV Python as a tool. Our proposed solution can recognize vehicles, track their speed and help in counting the objects precisely.**

***Index Terms*: OpenCV, Object Detection, Object Tracking, Deep Neural Network, CV.**

I. INTRODUCTION

The increasing number of cars in cities can cause high volume of traffic, and implies that traffic violations become more critical nowadays around the world. This causes severe destruction of property and more accidents that may endanger the lives of the people. To solve the alarming problem and prevent such unfathomable consequences, traffic violation detection systems are needed. For this reason, the system enforces proper traffic regulations at all times and "catches" those who does not comply. A traffic violation detection system must be realized in real-time as the authorities track the roads all the time. Hence, traffic enforcers will not only be at ease in implementing safe roads accurately, but also efficiently; as the traffic detection system detects violations faster than humans. This system can detect traffic light violation in real-time. A user-friendly graphical interface is associated with the system to make it simple for the user to operate the system, monitor traffic and take action against the violations of traffic rules.

II. RELEVANCE OF THE RESEARCH

The criticality of the problem of traffic violations is very different depending on the country – in some countries, its monitoring process is very deeply controlled by government, in others – not so much. Unfortunately, our country – Ukraine – in the second group. Nowadays, in Ukraine, a traffic violation detection system is deployed in places with high traffic, but in other places – with less traffic – there are no such devices and there a lot of traffic rules violations occur. With these tools and methods that to be studied in the article, it may be much easier to develop and deploy traffic violation detection system due to reduced cost and easier future improvements.

III. RELATED WORKS

In the modern world, small computing devices have reached a very high level in their computational and photographic capabilities. Computer vision became one of the widespread technology used in small computing devices.

Nowadays computer vision helps in solving a lot of different problems.

Mr. Chaku A. with co-authors presented a concept of software service for garbage type recognition using mobile devices and computer vision technology. It uses smartphone hardware (camera for capturing objects and then using neural network for determining the appropriate type of garbage) for navigating user to the nearest recycling plants for different types of household waste [1].

Sowmya Kini M. came up with a solution for determining traffic congestions. An intelligent transport system (ITS) is needed to manage the congestion in traffic and to give smooth planning for drivers. The implementation was done using OpenCV Python as a tool [2].

In certain works, for example, [8], was researched algorithms that can be used to detect objects of different nature and are based on different approaches: detection of color non-uniformities, frame difference, and feature detection. As the input data, we use a video stream that is obtained from a video camera or from an mp4 video file.

Similar investigation was done as well for mobile platform [4]. The method of functional adaptation of the algorithm of search and recognition of objects to features of video is offered, which consists in processing of video image by smoothing and minimization filter, was designed and presented.

IV. FORMULATION OF THE TASK

The goal of the project is to automate the traffic rules violation detection system and make it easy for the traffic police department to monitor the traffic and take action against the violated vehicle owner in a fast and efficient way. Detecting and tracking the vehicle and their activities accurately with small number of frames per second (5-6) is the main priority of the system. The second requirement for the system is the price. It should be built on free/open-source components in order to reduce the price and increase chances of deploying it. The goal of this article is to study and describe methods of vehicle recognition and detect violations of traffic rules.

V.TASK IMPLEMENTATION

The software system consists of several main modules which are responsible for the various stages of processing the input data and subsequent actions. The system includes a module for processing device's camera data, analytics module [1], [2]. Vehicle detection and tracking flow are shown in Fig. 1.

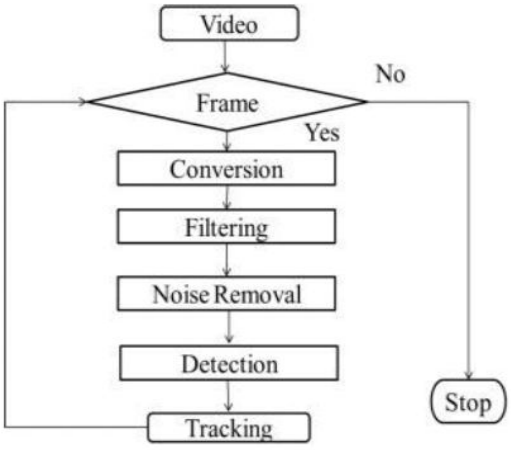

Fig. 1. Detection and tracking flow.

A. RASPBERRY PI

We used Raspberry Pi as a hardware platform to implement prototype described in the paper.

Raspberry Pi is a series of small single-board computers (SBCs) developed in the United Kingdom by the Raspberry Pi Foundation in association with Broadcom. The Raspberry Pi project originally leaned towards the promotion of teaching basic computer science at schools and in developing countries. Original model became more popular than the anticipated one selling outside its target market for such application as robotics. It is widely used in many areas, such as for weather monitoring, because of its low cost, modularity, and open design. It is typically used by computer and electronic hobbyists, due to its adoption of HDMI and USB devices.

B. OPENCV

OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms.

These algorithms can be used to detect and recognize faces, identify objects (Fig. 2), classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high-resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc [3].

Fig. 2. Detection and tracking flow.

C. YOLO

We'll use the YOLOv3 model with OpenCV-python. Open-CV is a real-time computer vision library of Python. We can use YOLO directly with OpenCV [4], [5].

YOLO, short for You Only Look Once is a convolutional neural network architecture designed for the purpose of object detection. There are 3 versions of YOLO, namely version 1, version 2 and version 3. The latter two versions are improvements of the first one.

Object detection is the problem of localization and classifying a specific object in an image which consists of multiple objects. Prior to YOLO, image classifiers were used to carry out the task of detecting an object by scanning the entire image to locate the object. The process of

scanning the entire image begins with a pre-defined window which produces a boolean result that is true if the specified object is present in the scanned section of the image and false if it is not. After scanning the entire image with the window, the size of the window is increased which is used for scanning the image again. Such system as deformable part model (DPM) uses this technique which is called Sliding Window. Other detection methods like R-CNN and Fast R-CNN are primarily image classifier networks which are used for object detection with the following steps.

Use Region Proposal method to generate potential bounding boxes in an image. The next step is to run the classifier on these boxes. After classification, perform post processing to tighten the boundaries of the bounding boxes, remove duplicates.

YOLO works using mainly these techniques (Fig. 3):

1) Residual Blocks – Basically, it divides an image into NxN grids.

2) Bounding Box regression – Each grid cell is sent to the model. Then YOLO determines the probability of the cell which contains a certain class and the class with the maximum probability is chosen.

3) Intersection Over Union (IOU) – IOU is a metric that evaluates intersection between the predicted bounding box and the ground truth bounding box. A Non-max suppression technique is applied to eliminate the bounding boxes that are very close by performing the IoU with the one having the highest-class probability among them.

YOLO is implemented as a convolution neural network and has been evaluated on the PASCAL VOC detection dataset. It consists of a total of 24 convolutional layers followed by 2 fully connected layers [6], [7].

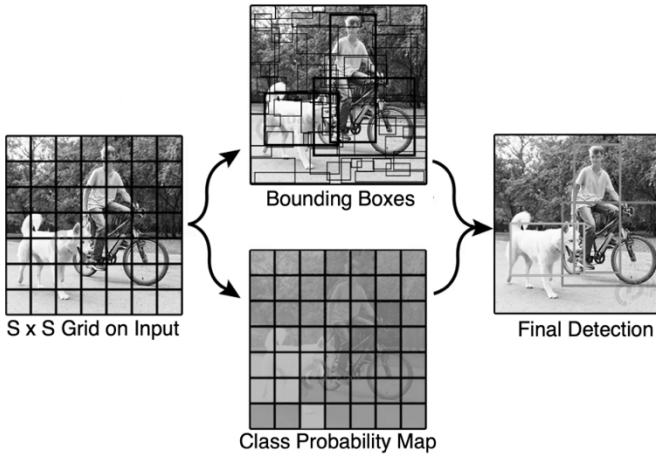

Fig. 3. Combination of 3 YOLO techniques.

The layers are separated by their functionality in the following manner (Fig. 4):

First 20 convolutional layers followed by an average pooling layer and a fully connected layer is pre-trained on the ImageNet 1000-class classification dataset.

The layers comprise of 1x1 reduction layers and 3x3 convolutional layers.

Last 4 convolutional layers followed by 2 fully connected layers are added to train the network for object detection.

Object detection requires more granular detail hence the resolution of the dataset – 448x448.

The final layer predicts the class probabilities and bounding boxes.

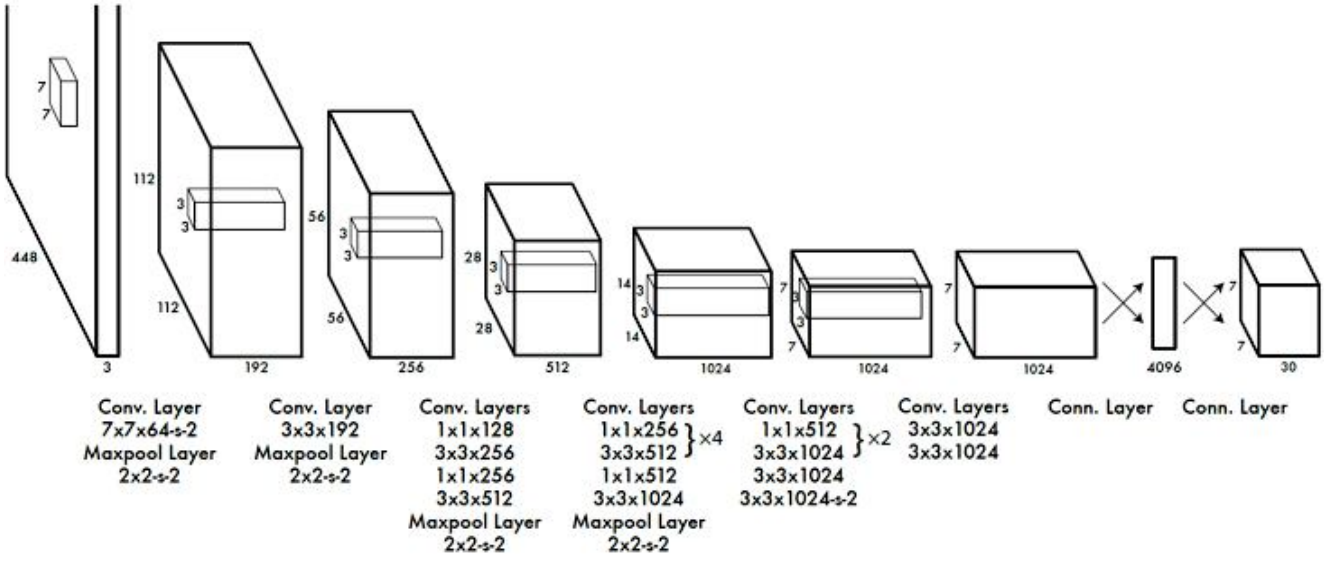

Fig. 4. Example of architecture. Detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1x1 convolutional layers reduce the feature space from the preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half of the resolution (224x224 input image) and then double the resolution for detection. The pretraining for classification is performed on dataset with resolution 224x224.

D. VEHICLE DETECTION

Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class in digital images and videos.

From the given video footage, moving objects are detected. An object detection model YOLOv3 is used to classify those moving objects into respective classes. YOLOv3 is the third object detection algorithm in YOLO (You Only Look Once) family. It improved the accuracy with many tricks and is more capable of detecting objects.

The classifier model is built with Darknet-53 architecture. Fig. 5 shows how the neural network architecture is designed. A couple of computations have been introduced for the conditions; some of them are executed in OpenCV, for instance, Background Subtraction MOG. The background subtraction uses 2 to 4 distributions in clearing small artifacts. Other method for removing separating foreground and background is Background Subtractor GMG in OpenCV which relies upon and unites the establishment picture estimation methodology with Bayesian division.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Fig. 5. Darknet-53 architecture adopted by YOLOv3.

The count used in the proposed structure is called establishment Subtractor MOG2. It relies upon two assessments and by Zikovic. One of the huge features of this count is that not under any condition like where amount of disseminations for the modelling of establishment methods are portrayed, Background SubtractorMOG2 uses a robotized prospect and picks a relevant amount of the Gaussian mixes for pixel.

Thus, this methodology is good, if there are any issues with contrast and brightness in any frame. This method also provides good visibility on the shadow of the objects, ability in defining the shadow and also helps weather shadow to be detected or not in particular scene. In default settings are set to detect shadow of an object.

1) Extraction of Contour.

These contours are the binary representation of an object. The shape and the co-ordinates of an object are considered for object recognition. The finding of contour can be increased with the help of canny edge detection technique which is the best method in defining the object boundaries. OpenCV has inbuilt package for detecting these counters.

2) Vehicle Count

The count of objects will be taken when the contour areas centroid touches the ROI. This ROI is an imaginary line drawn diagonally across the road touching the two ends. It is also noted that when the centroid nears this imaginary line, the counter value will increase indicating an object moment observed.

3) Tracking (Virtual Object Detector)

For counting, moving objects form a lane, a method known as “Virtual Object Detector” is proposed here. Usually, the objects on the road can be detected by using the old theories of induction. It was a sensor-based technique in which sensors were kept below road. Here, the objects will be detected when the vehicle passes on the road depending on its induced loop. The calculation of object nearness has to be calculated. By using this method, it was found out that practically it is challenging due to considerable costs, position (it is difficult to place under road), lifetime of these systems will be short etc. Here, we propose new computer vision-based technique to detect objects and also to track them by counting number of vehicles on road. In this method of virtual detector, a rectangle box will be drawn and that will be considered as ROI (Region of interest). The objects within that box will be identified based on their area and motion vectors. The colors will be specified to track the object – this will be a histogram- oriented technique.

4) Blob Tracking

In moving vehicles tracking refers to detecting the motion of a particular object by tracing its line of activity. In our research, we have considered a pre-recorded traffic video and applied blob-based tracking method so that we can track all the moving objects in the given area by subtracting the background.

This procedure of tracking using blob methodology has some stages: detecting the foreground, detecting new blob, tracing the blob module, generation of new path module, detection of the flow direction module [8] (Fig. 6,

Fig. 7, Fig. 8, Fig. 9, Fig. 10, Fig. 11, Fig. 12). Note: Fig. 8 – Fig. 12 colors are inverted.

5) Detecting the Foreground

Identification of pixels from the image and categorizing weather object is in foreground or in the background. The result of this step will be continued as input to step 2.

6) Detecting new blob

After getting the foreground information from the previous step the following is to find out the object entering the ROI.

7) Tracing the blob module

Here tracing of new objects and those blobs will be monitored and the tracing will be done for already present objects.

8) Generation of new path module

Checking the traced moving objects and saving their new path at the end of every frame.

9) Detection of flow direction module

The path will be of soft trajectory after one lane is saved.

Fig. 6. Grayed background.

Fig. 7. Grayed current input frame.

Fig. 8. Extracted foreground.

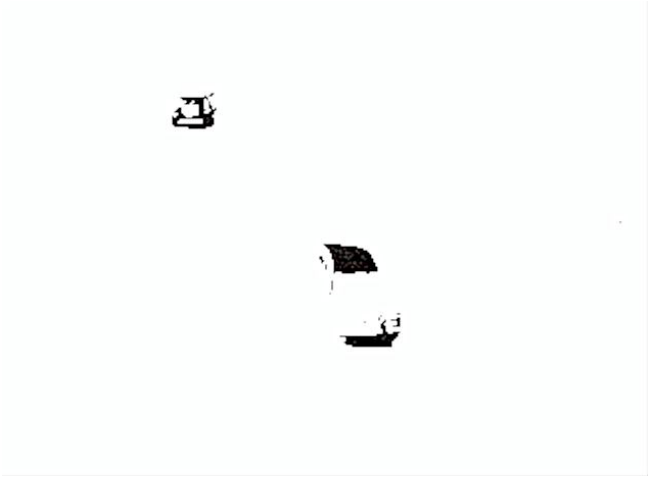

Fig. 9. Binary image.

Fig. 10. Edge of current frame.

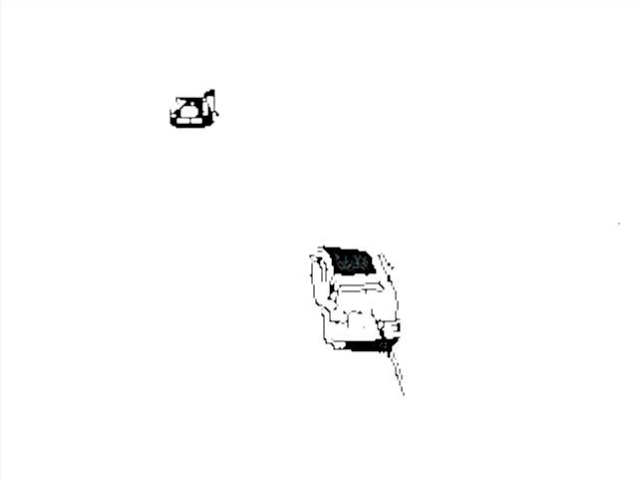

Fig. 11. Combined image.

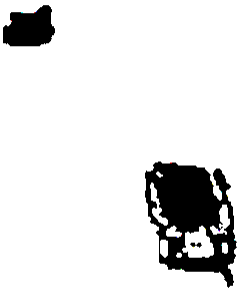

Fig. 12. Binary image after filling.

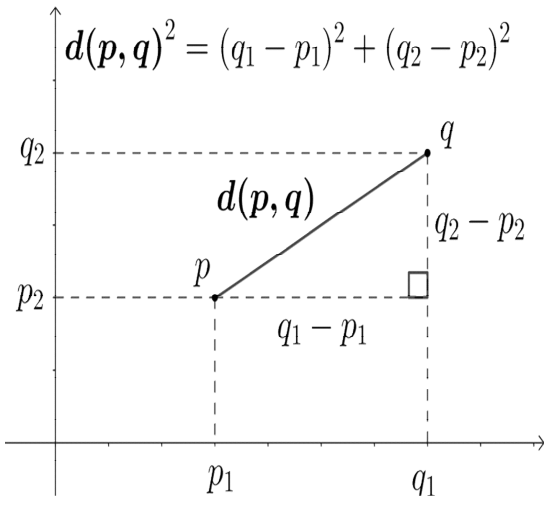

Fig. 13. Euclidean distance.

E. TRACKING

The tracker basically uses the Euclidean distance concept to keep track of an object (Fig. 13). It calculates the difference between two center points of an object in the current frame vs the previous frame, and if the distance is less than the threshold distance then it confirms that the object is the same object of the previous frame [9]. To be able to use Euclidean distance concept we need to determine object centroids.

F. VEHICLE SPEED

Let's review vehicle speed of our algorithm at a high level:

- Our speed formula is speed = distance / time.
- We have a known distance constant measured by a tape at the roadside.
- Meters per pixel are calculated by dividing the distance constant by the frame width in pixels (1).
- Distance in pixels is calculated as the difference between the centroids as they pass by the columns for the zone (2). Distance in meters is then calculated for the particular zone (3).
- Four timestamps (t) will be collected as the car moves through the FOV past four waypoint columns of the video frame.
- Three pairs of four timestamps will be used to determine three delta t values.
- We will calculate three speed values for each of the pairs of timestamps and estimated distances.
- Three speed estimates will be averaged for an overall speed (4).

Using this algorithm, we can calculate the speed of the vehicle running the road [10].

The following equations represent our algorithm: (1), (2), (3), (4).

$$meters\ per\ pixel = mpp = \frac{distance\ constant}{frame\ width} \quad (1)$$

$$distance\ in\ pixel = p_{ab} = |col_B - col_A| \quad (2)$$

$$distance\ in\ meters\ zone_{ab} = d_{ab} = p_{ab} * mpp \quad (3)$$

$$average\ speed = \frac{\frac{d_{ab}}{\Delta t_{ab}} + \frac{d_{bc}}{\Delta t_{bc}} + \frac{d_{cd}}{\Delta t_{cd}}}{3} \quad (4)$$

VI. PROSPECTS FOR FURTHER RESEARCH

Even with the current implementation there are drawbacks and space for potential improvements. A short list of possible improvements in the future:

1) Currently, detection of traffic rule violation system can recognize only speed violations, in the future it is good to have an ability to detect other types of violations.

2) In addition, in the current implementation of the learning it is implemented on a more powerful computer and the mobile application only uses the results of that training. So, it would be better to turn it all into a cloud solution where a single model is trained and distributed to different devices and all devices will use it.

3) Investigate the performance of the system on very low-cost devices such as Raspberry Pi Zero, Onion Omega 2 series, Orange/Banana/and other Pi(s).

4) Investigate system behavior in different weather conditions, at day and night. Currently, system supports only one type of vehicle – car, but there are a lot of others. So, as a potential improvement it would be good to train the system to be able to recognize other types of vehicles.

VII. CONCLUSION

This paper proposes an innovative approach to recognize vehicles on the road and detection of traffic rule violations. To use this approach, no special expensive equipment is required.

In this article, we explore how computer vision works by creating a deep-learning-based traffic rule violation detection system using OpenCV. In general, we should never use Raspberry Pi to train a neural network, but only use it to deploy a pre-trained deep learning network. The Raspberry Pi does not have enough memory or CPU power to train these types of deep, complex neural networks from scratch.

The simple Linux development board is capable to run trained network and able to recognize vehicles and traffic rule violations. The software was used to develop algorithms is an open-source. That means it can be used freely without any additional costs. The production system based on those components will be cheap.

The prototype device was based on Raspberry Pi Zero that costs around $5 in 2021. Overall system performance with a running application on that hardware is satisfactory. Vehicle tracking frame rate is 5-6 frames per second. It is very low rate for human eyes, but for automation it is more than enough. The system with such tracking frame rate is capable to detect traffic rule violation.

The use of systems on the road will much decrease rule violations, improve safety on the road and will help to save a lot of human lives.

References

[1] Bachynskyy, R., Chaku, O., and Huzynets, N. (2017) A Software Service for the Garbage Type Recognition Based on the Mobile Computing Devices With Graphical Data Input. *Advances in Cyber-Physical Systems*, 5(1), pp.1-7. DOI: 10.23939/acps2020.01.001.

[2] Sowmya, K.M., Rekha, B., Praveen, S.K. (2021) Real Time Moving Vehicle Congestion Detection and Tracking using OpenCV. *Turkish Journal of Computer and Mathematics Education*, 12(10), pp. 273-279. Available at: https://www.turcomat.org/index.php/turkbilmat/article/view/4139. [Accessed 22 November 2021].

[3] OpenCV (2021) Home - OpenCV. [online] Available at: http://opencv.org/ [Accessed 22 November 2021].

[4] Kushnir, D., Paramud, Y. (2019) Methods for real-time object searching and recognizing in video images on ios mobile platform. *Computer systems and network*, 1(1), pp.24-34, DOI: 10.23939/csn2019.01.024.

[5] TechVidvan (2021) Vehicle Counting, Classification & Detection using OpenCV & Python. [online] Available at: https://techvidvan.com/tutorials/opencv-vehicle-detection-classification-counting/ [Accessed 21 November 2021].

[6] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016) You Only Look Once: Unified, Real-Time Object Detection. CoRR, abs/1506.02640. Available at: http://arxiv.org/abs/1506.02640. [Accessed22 November 2021].

[7] Medium (2021) YOLO v1 : Part 1. [online] Available at: https://medium.com/adventures-with-deep-learning/yolo-v1-part-1-cfb47135f81f/ [Accessed 22 November 2021].

[8] Puyda, V., Stoian, A. (2020) On Methods of Object Detection in Video Streams. *Computer Systems and Networks*, 2(1), pp. 80-87, DOI: 10.23939/csn2020.01.080.

[9] Wiki (2021) Euclidean distance. [online] Available at: https://en.wikipedia.org/wiki/Euclidean_distance/ [Accessed 22 November 2021].

[10] Pyimagesearch (2021) OpenCV Vehicle Detection, Tracking, and Speed Estimation. [online] Available at: https://www.pyimagesearch.com/2019/12/02/opencv-vehicle-detection-tracking-and-speed-estimation/ [Accessed 21 November 2021].

Yevhen Fastiuk received the B.S. degree in Computer Engineering at Lviv Polytechnic National University, Lviv, Ukraine in 2018.

He is currently pursuing the M.S. degree in System Programming at Lviv Polytechnic National University, Lviv, Ukraine.

Ruslan Bachynskyy obtained his Ph.D. degree in Computer Sciences at Lviv Polytechnic National University, Lviv, Ukraine in 2008.

Nataliia Huzynets, an engineer at Lviv Polytechnic National University. She graduated from Lviv Polytechnic National University in 2002 with specialist degree. Areas of interest: programming, High Performance computing, Digital signal processing.