Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Kirill Hazukov, Vladimir Shepelev, Tatiana Carpeta, Salavat Shabiev, Ivan Slobodin, Irakli Charbadze

South Ural State University, 454080 Chelyabinsk, Russia

Correspondence to Vladimir Shepelev email: shepelevvd@susu.ru

Abstract

This study deals with the problem of rea-time obtaining quality data on the road traffic parameters based on the surveillance camera data. The purpose of the paper is to develop a system to collect data on the traffic flow structure and to determine the traffic flow speed and direction in real-time. Our solution is based on the use of the YOLOv3 neural network architecture and open-source tracker SORT. To increase the accuracy of detection and classification, we used multi-scale prediction with an increased number of anchors. To determine the speed, we used a matrix of the perspective transformation of the source image to geographical coordinates. To train the neural network, we marked over 6,000 images and performed augmentation, which allowed us to increase the dataset to 60,000 images. Checking the system at night and in the day showed an absolute percentage accuracy of counting vehicles of no less than 92%. The error in determining the vehicle speed by the projection method, taking into account the camera calibration, did not exceed 2.74 m/s. The presented study allows us to generate big data for the intelligent transport systems decision-making system in real-time and to lower the requirements for peripheral equipment.

Keywords: YOLO v3 neural network, Data for training the neural network (Dataset), Traffic flow assessment, Vehicle detection, Vehicle classification, Vehicle speed, Traffic monitoring, Big data.

Introduction

In the conditions of infrastructural restrictions of cities and budget deficits, road network capacity can be effectively increased by introducing intelligent transport systems (ITS). ITS allow us to analyze and improve the efficiency of the road transport infrastructure, transport and performance indicators of traffic. The introduction of ITS influences the improvement of road safety while reducing traffic congestion and environmental impact.

The basis of ITS is big data, which should be obtained, interpreted, and used in real-time. In this context, the most important task in studying transport systems, as a science, is the development of information collection technologies and standards through the use of existing communication networks based on the application of modern data analytics.

We propose a method based on the use of the deeply trained YOLOv3 neural network for the real-time detection and calculation of traffic flow parameters.

The paper is structured as follows. First, related works regarding the road traffic parameters based on the surveillance camera data are reviewed. Then a training dataset, the architecture of the neural network, the results of the neural network (recognition and classification of vehicles), as well as the necessary metrics are described. Also the calculated indicators: speed and counting of vehicles in all directions of a road junction is obtained. Finally, a packaged solution is provided.

Related work

Artificial neural networks have proven themselves in the development of methods and technologies for monitoring road traffic and diagnosing the effectiveness of using the road transport infrastructure. Currently, many cities have video surveillance systems, which include cameras with different resolutions and fixed frame rates [1]. Such systems operate on a 24/7 basis and generate a massive amount of information. Among other applications, this data can serve as a database for an automated traffic monitoring system. Some of the representative works [2] and [3] use low-resolution video surveillance system data and deep neural networks to count vehicles on the road and estimated traffic density. Examples of using conventional machine vision methods are systems developed in [4, 5], which analyzed the problems of freight traffic. To detect a vehicle, most modern works discuss the adaptation and improvement of modern detection systems, such as R-CNN [6], YOLO [7], and SSD [8]. This includes architectural innovations solving the problems of scale sensitivity [9], vehicle classification [10, 11, 12], and increasing the speed and accuracy of the detection methods [13, 14]. Improving the detection rate [15], temporary information is also used for joint detection and tracking of objects [16, 17, 18].

The existing solutions in the problems of real-time vehicle detection and classification require large computing capabilities and place strict requirements for the installation location and camera performance.

Methodology

In this article, we describe a system for real-time determining the intensity and structure of traffic flows, i.e., the system should classify vehicles and calculate their movement directions and speeds. To achieve this objective, we divide the problem into four sub-tasks: vehicle detection and classification, vehicle tracking, determining the movement direction and calculating the vehicle speed. This approach leads to an easy-to-test modular architecture. In the following sections, we will describe each module in detail, together with the data collected for training.

Detection and classification of vehicles

Neural network architecture

We chose the YOLOv3 neural network based on CNN for object detection and classification. An important feature of this architecture is that ultra-precise layers are applied to the image once, as opposed to such architectures as R-CNN [19, 20, 21] and Fast R-CNN [22], which gives a multiple increase in the image processing speed without significant accuracy losses: one image is processed through the use of YOLO 1000 times faster than through the use of R-CNN, and 100 times faster than through the use of Fast R-CNN [23]. The architecture of the YOLOv3 neural network is shown in Fig. 1.

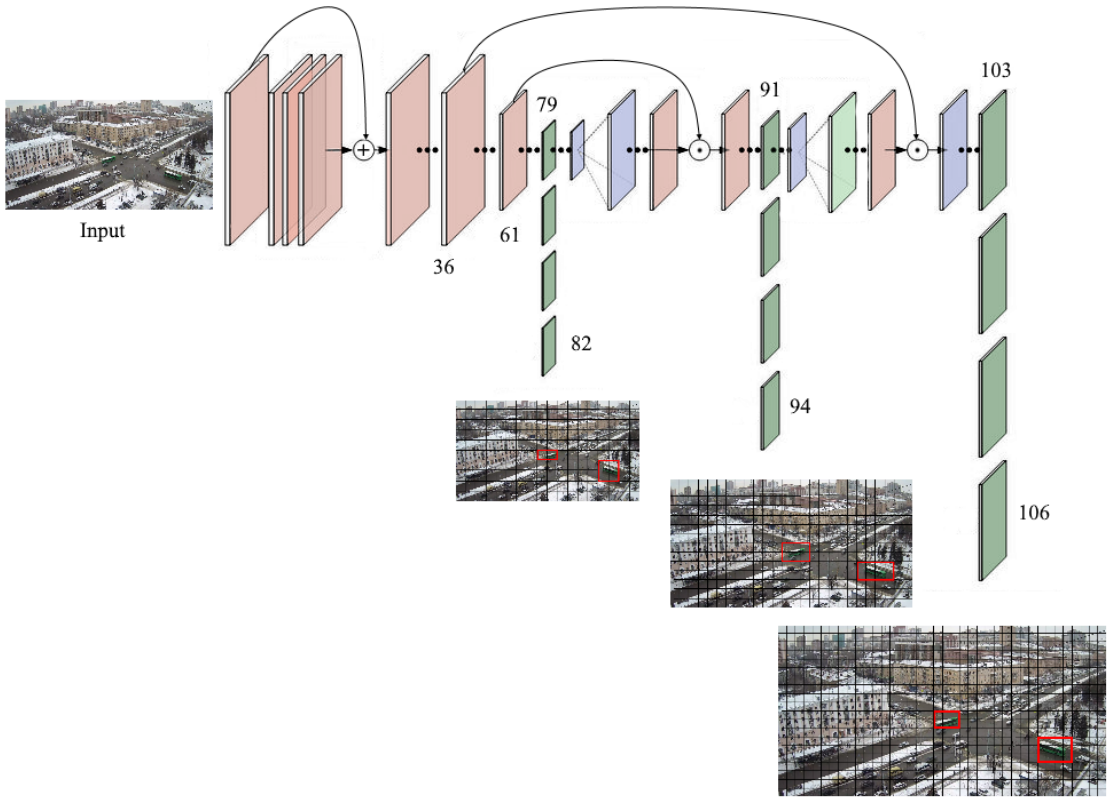

Fig.1 The YOLO v3 network architecture

This neural network consists of 106 layers. In addition to the use of ultra-precise layers, its architecture also contains residual levels [24], layers with increased discretization and passed connections. CNN takes the image as input data and returns a tensor (Fig. 2), which represents:

- coordinates and positions of the predicted bounding boxes, which should contain the objects;
- the probability that each bounding box contains an object;
- the probability that each object within its bounding box belongs to a certain class.

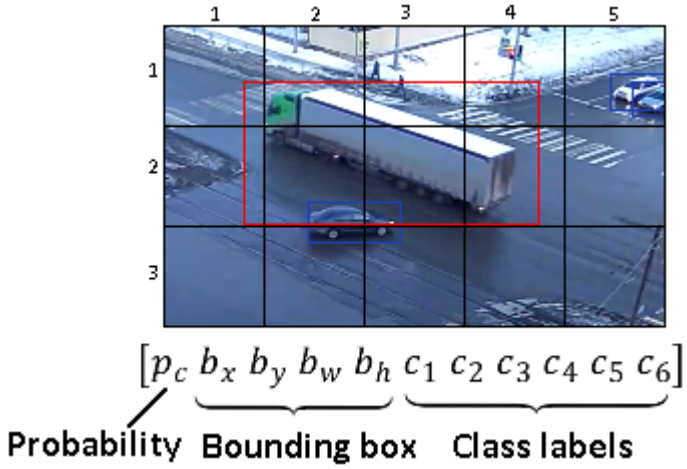

Fig. 2 Output tensor

Data preparation and training

We have access to several Intersvyaz cameras in the city of Chelyabinsk, as well as to the camera of the administration of the city of Tyumen. Most of the cameras are mounted high above main intersection to provide a general overview of the traffic situation. These cameras ensure steady 25 frames per second and support a resolution of 1920×1080 pixels. However, the video stream is not perfect because of compression artifacts, erosion, bad weather conditions, and hardware errors, which prevent the detection and classification of vehicles, as well as the determination of speed indicators through the use of the existing methods [25].

We collected and tagged frames of video streams from 7 cameras of various road junctions as the data for training the neural network. As a result, we obtained about 430,000 vehicle objects. The indexation of the classes and their corresponding colors further used to display the detection results are presented in Table 1.

The input data are presented as follows: a JPG or PNG image and a text file with marking:

$$
\begin{array}{ccccc}
C_1 & X_1 & Y_1 & W_1 & H_1 \\
C_2 & X_2 & Y_2 & W_2 & H_2 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
C_i & X_i & Y_i & W_i & H_i \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
C_n & X_n & Y_n & W_n & H_n
\end{array}
$$

Fig. 3 The input data

In Fig 3. i is the object number; n is the number of objects in the image; C_i is the index of the class of the i-th object; X_i, Y_i are the coordinates of the center of the rectangle containing the object; W_i, H_i are the width and height of the rectangle containing the object.

The parameters X_i, Y_i, W_i, H_i are recorded in relative values of the image size ($X_i Y_i W_i H_i \in [0;1]$).

Table 1 Indexation of the classes and their corresponding colors

Index	**Class**	**Color**
0	car	yellow
1	mini bus	blue
2	bus	brown
3	truck	red
4	tram	pink
5	trolleybus	green

For better training of the neural network, we expanded the dataset by applying augmentation, which increased the dataset by 10 times. For augmentation, we applied the following transformations in various combinations: horizontal display; affine and perspective transformations; noise overlay; color distortion. The final

dataset amounted to 4.3 million objects. The distribution of objects of each class in the training sample is presented in Table 2.

Table 2 Distribution of vehicle classes in the training sample

Class	Number of objects	Relation to the total number
car	3,518,370	81.8%
mini bus	228,810	5.3%
bus	163,430	3.8%
truck	184,520	4.2%
tram	91,540	2.1%
trolleybus	112,690	2.6%

We divided the dataset into training and validation samples in the ratio of 80/20% and started training for 20,000 iterations with an increment of 0.001. The batch size per one iteration was 64 images, which was divided into 16 units during training to run several images at once.

The results of training the neural network

Assessment of the neural network accuracy

One of the methods for assessing the accuracy of the classifier is the assessment through the use of precision and recall metrics [26]. Precision is the proportion of objects belonging to this class for all the objects assigned to this class by the neural networks (NN).

Recall is the proportion of objects belonging to the class found by the classifier for all the objects of this class in the test sample.

Table 3 contains the information on how many times the system made true and false decisions on the objects of this class. Namely:

- TP - true-positive decision;
- TN - true-negative decision;
- FP - false-positive decision;
- FN - false-negative decision.

Table 3 Confusion matrix

		True values	
		True	False
NN response	True	True Positive	False Positive
	False	False Negative	True Negative

Precision and recall are calculated as follows:

$$precision = \frac{TP}{TP + FP} \quad (1)$$

$$recall = \frac{TP}{TP + FN} \quad (2)$$

Based on the aforesaid formulas (1-2), we compiled a confusion matrix (Fig. 4) and calculated the precision and recall of determining six classes of vehicles (Table 4).

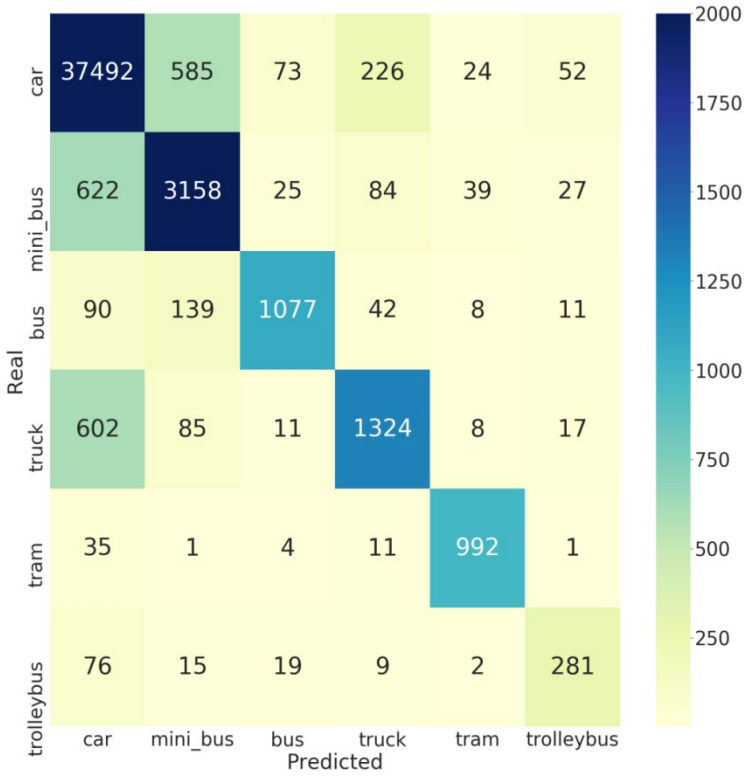

Fig. 4 Confusion matrix obtained as a result of testing the trained model

Table 4 Precision and recall indicators of the trained neural network

Class	Precision	Recall
car	0.96	0.98
mini bus	0.79	0.8
bus	0.89	0.79
truck	0.78	0.65
tram	0.92	0.95
trolleybus	0.72	0.7

F-measure

F-measure is a harmonic mean between precision and recall. The higher the recall and precision, the better; however, in reality, these metrics cannot simultaneously reach maximum performance indicators; therefore, we should search for a balance. To this end, the F-measure is used.

$$F = 2 \times \frac{\mathrm{Pr}ecision \times \mathrm{Re}call}{\mathrm{Pr}ecision + \mathrm{Re}call} \quad (3)$$

Formula (3) gives the same weight to the precision and recall and reduces them to one number, so the F-measure will fall equally with a decrease of the precision and recall (Table 5).

Table 5 F-measure of the trained model

Class	F-measure
car	0.97
mini bus	0.79
bus	0.84
truck	0.71
tram	0.93
trolleybus	0.71

Calculated indicators

The average vehicle speed

Elimination of the camera distortion

Modern cameras are imperfect - they distort the image, changing the size, shape, and distances of objects. In our case, the image transmitted from the camera is subject to distortion. To determine accurately the coordinates of objects, we should eliminate the distortion by calibrating the camera. The easiest method of calibration is to use a spatial test object, such as a checkerboard [27], as shown in Fig. 5.

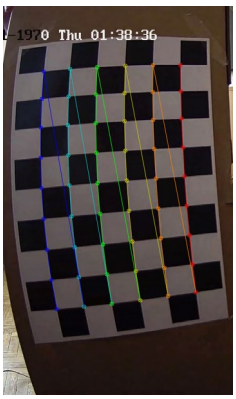

Fig. 5 Demonstration of correcting the image distortion through the use of a checkerboard

Fig. 6 shows the source images and images after the calibration was applied.

Source images Images after the calibration

Fig. 6 Source and corrected camera images

Calculation of the distance

To calculate distance traveled, we must find the change in the latitude and longitude of the vehicle's location over a certain time interval using the change of coordinates

in the camera image. To solve this problem, we calculated the perspective transformation matrix (4) by selecting four reference points in the map and comparing the corresponding points in the image (Fig. 7).

Fig. 7 Reference points in the image

$$A \times \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \begin{pmatrix} x_i' \\ y_i' \\ t_i \end{pmatrix} \tag{4}$$

where A is the transformation matrix; x_i, y_i are the pixel coordinates in the image; x'_i, y'_i are the latitude and longitude of the point.

To calculate the distance between two points, we used the inverse haversine formula, which is explicitly expressed through the arcsine [28]:

$$dist = 2r \times \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_2)\cos(\varphi_1)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \tag{5}$$

where $dist$ is the measured distance; φ_i, λ_i are the latitude and longitude of the i-th point; r is the radius of the earth (r= 6371 km).

Now, to calculate the average speed, we apply the following formula:

$$v = \frac{dist}{t_2 - t_1} \tag{6}$$

where t_1, t_2 are the time of the beginning and end of movement at a distance.

Let us theoretically assess the error in calculating the distance in this way. There is an error in calculating the real coordinates using the projection of the pixel map because the region transmitted by one pixel on the frame has some nonzero area (Fig. 8).

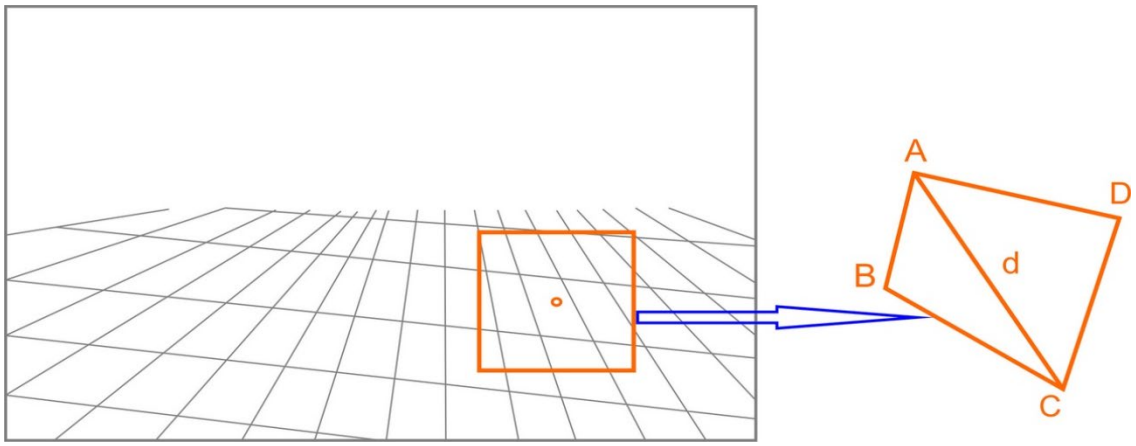

Fig. 8 The region transmitted by one pixel

To get an upper estimate of this area, taking into account the perspective, we took the region of the pixels used to transmit the farthest part of the road (Fig. 9). The actual size of this distance is 88.5m; in the image, this segment is transmitted by 91px.

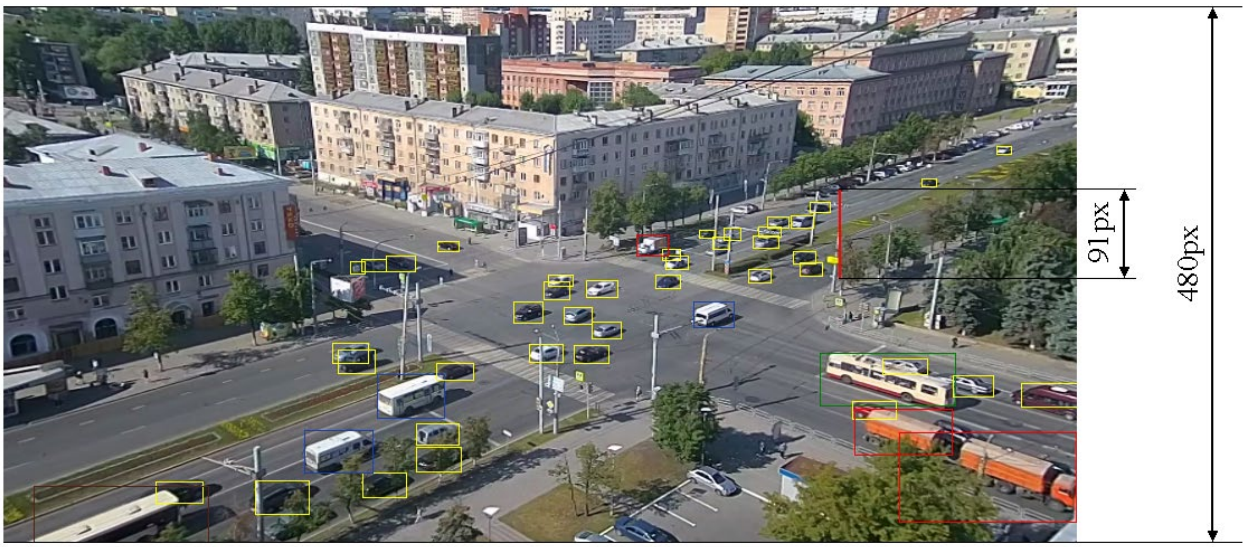

Fig. 9 The distance transmitted by pixels

Thus, one pixel covers a region 0.97 m long; taking this region as a square, we can estimate the maximum point projection error within this region of 1.37m. Given this error (Fig. 10), the error in determining the distance does not exceed 2.74 m:

$$dist - r \le 2e = 2 \times 1.37\text{m} = 2.74\,\text{m}$$

where r is the real distance; e is the error of calculating the coordinates of one point.

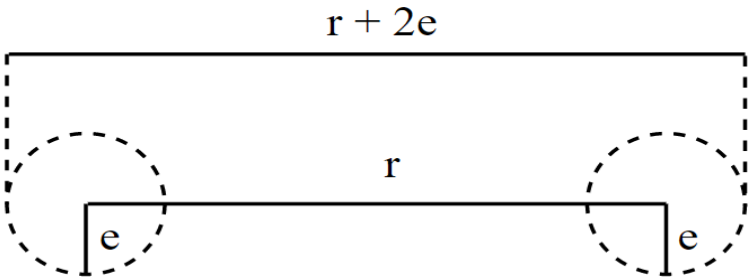

Fig. 10. Distance measuring error

Counting the vehicles

To count vehicles at a given intersection, we identified the zones in which the vehicles are counted (Fig. 11). We determined the point of origin of the vehicles, their direction of travel, and the vehicle class.

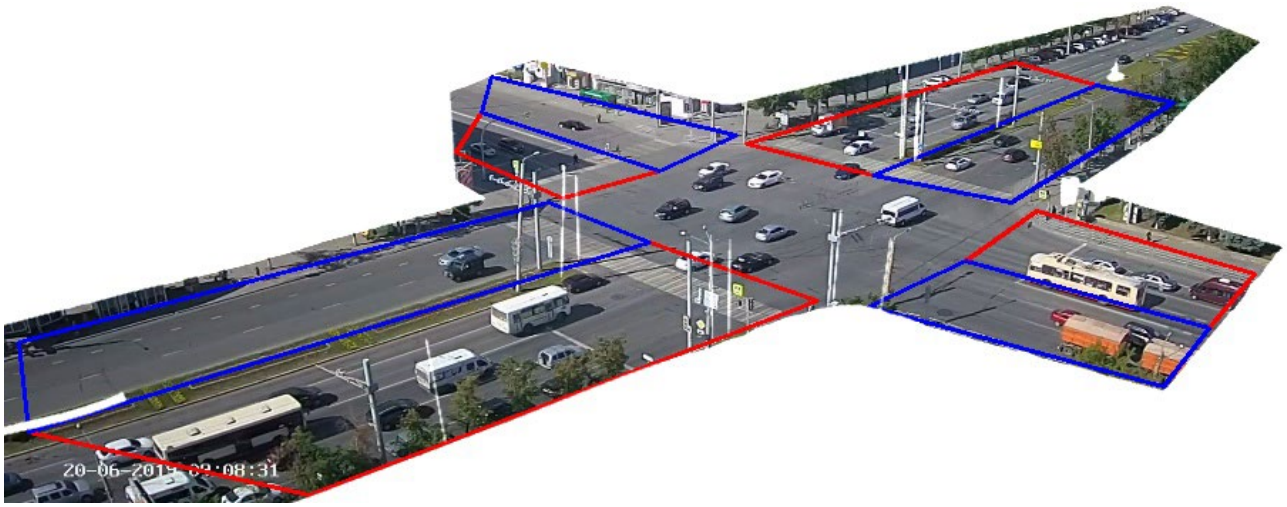

Fig. 11 An image with an applied mask and visualization of the areas of the vehicles' entry and exit from the intersection indicated red and blue, respectively

So that the neural network did not recognize vehicles outside the road, we apply a mask to each frame, painting the unnecessary areas white. To assess the accuracy of counting the vehicles, we created 4 15-minute video fragments at different times of the day and compared the real amount of vehicles with the result of our software for each direction.

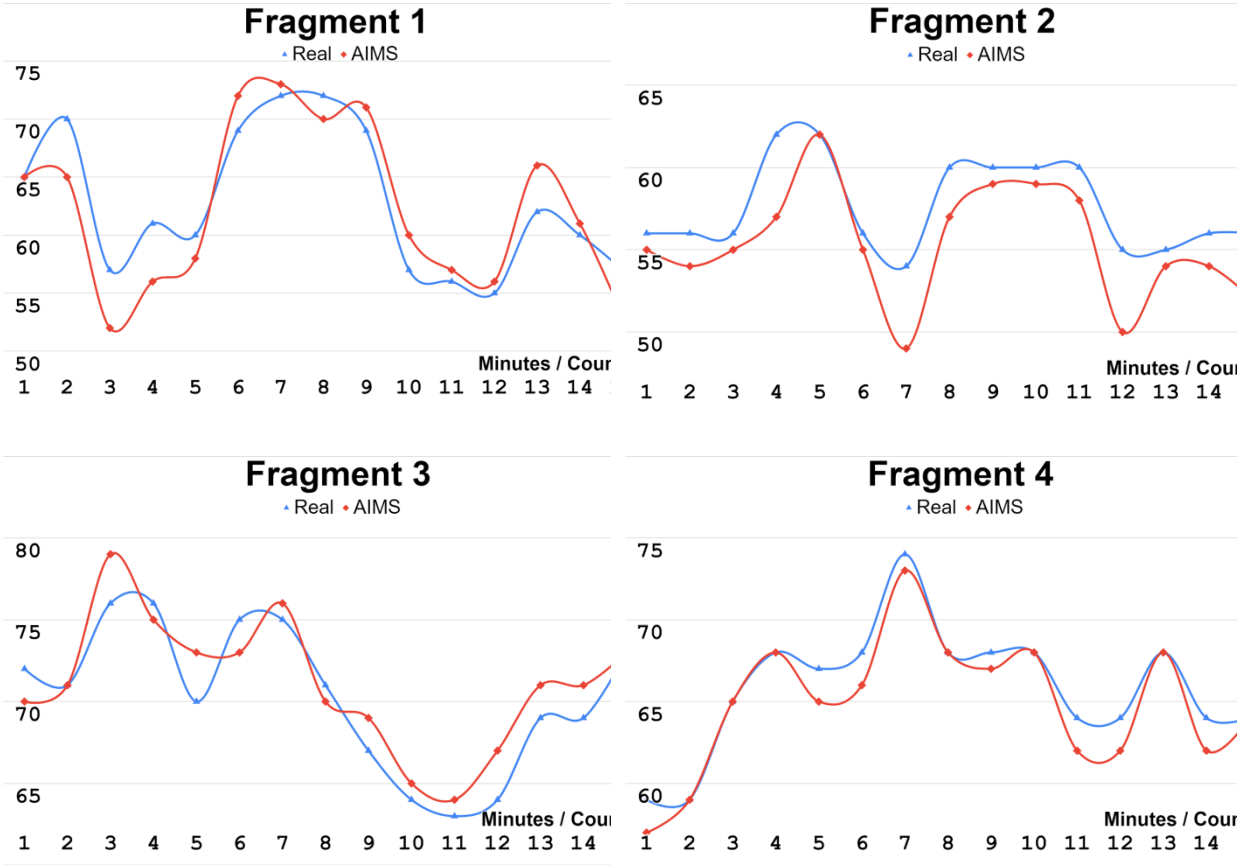

Fig. 12 A comparison of the system's counting the vehicles

Fig. 12 contains the following graphs: blue is the real number of vehicles, red is the result of counting the vehicles by our system. As we can see, the maximum counting error is five vehicles per minute at the total number of vehicles averaging 64. Thus, the absolute error percentage does not exceed 0.08%.

Results and discussion

Software solution

Fig. 13 shows a diagram of our developed system. The system includes the following sequence of processes:

- frames reading (Process 1);
- detection and classification of vehicles from the current frame (Process 2);
- vehicle tracking and counting in all directions of the road junction (Process 3);
- calculation of the latitude and longitude of the vehicle location (Process 4);
- calculation of vehicle speeds (Process 5);
- calculation of metrics related to the amount of harmful substances emitted by each vehicle (Process 6).

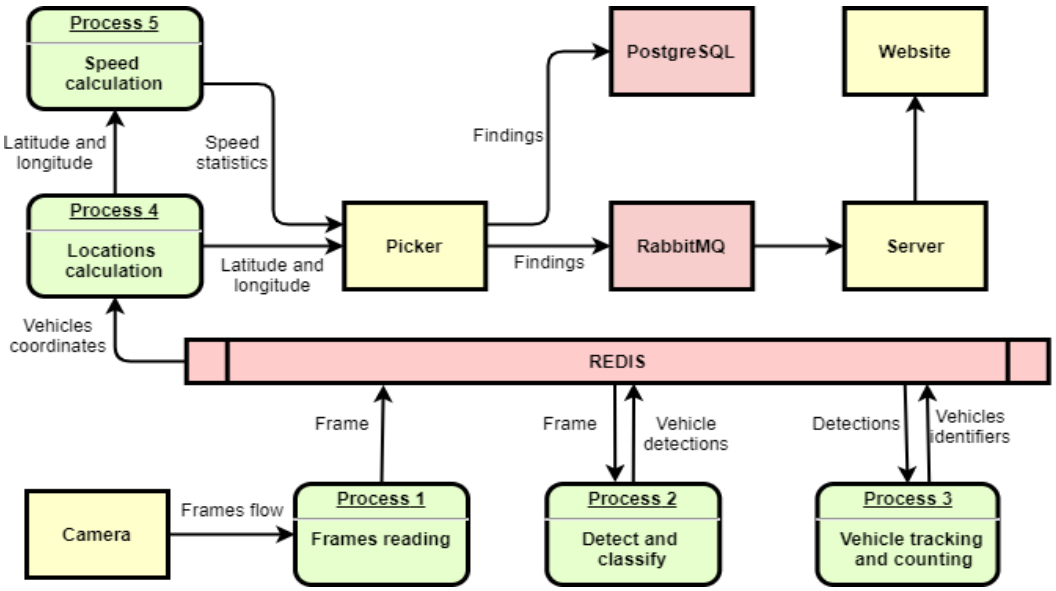

Fig. 13 System workflow

Used technologies

We used the following technologies for the software implementation of the presented architecture:

1. OpenCV is an open-source library designed to work with computer vision algorithms, image processing and general-purpose numerical algorithms. We used this library to perform the following tasks:

a. Resizing an image and applying a mask to it;

b. Setting and displaying of entry and exit areas, as well as determining the presence of vehicles in said areas;

c. Camera calibration and elimination of distortion;

d. Use of the perspective transformation matrix and determining the length of the distance;

e. Data visualization.

2. Sort is an open-source library for 2D tracking of several objects in video sequences based on the elementary data association and state estimation methods. We used it to track vehicles in the video stream.

3. Redis is a resident open-source NoSQL-class database management system. We used it to store intermediate results of the modules.

4. RabbitMQ is a software message broker based on the AMQP standard. We used it to organize a data queue for transferring to a web page.

5. PostgreSQL is a free object-relational database management system. To compile statistics and calculate various metrics, such as KPI and daily flow structure, we aggregate and save the received data in a database every hour.

Conclusion

In this study, we focused on the problem of obtaining high-quality data on road traffic based on the video stream from closed-circuit television (CCTV)

cameras. This task is made more difficult by the presence of different viewing angles, the long distance between the intersection and the camera, and overlapping of objects. We used multi-scale prediction in the architecture of the YOLOv3 neural network to improve the accuracy of detecting and classifying objects of different sizes and modified the SORT tracker to improve the quality of object tracking. The method based on the use of the matrix of the perspective transformation of the source image to geographical coordinates allowed us to determine the speed with an absolute error of no more than 2.74m/s.

To train the neural network, we performed augmentation of 6,000 source images, which allowed us to form a dataset of 4.3 million vehicles. The dataset was collected at 7 intersections, which allows us to use the trained neural network in various road sections with a comparable viewing angle and camera installation height.

The proposed system was tested during the day and at night, showing the absolute accuracy of counting vehicles of no less than 92%. The error in determining the vehicle speed by the projection method, taking into account the calibration of the camera at the test intersection, did not exceed 2.74m/s. This solution can generate big data, which can be used in real-time decision-making systems. Within the framework of this study, we did not consider the solution of many problems such as overlapping of objects, more detailed classification of the vehicles, identification of accidents, and blocking objects. We consider our solution as the basis for further studies aimed at solving these problems.

Abbreviations

ITS: intelligent transport systems

CNN: convolutional neural networks

NN: neural networks

CCTV: closed-circuit television

Declarations

Availability of data and materials

https://github.com/Readix/TrafficMonitoring

Acknowledgements

Not applicable.

Funding

The work was supported by Act 211 Government of the Russian Federation, contract No. 02.A03.21.0011.

Author information

Affiliations

South Ural State University, 454080, Chelyabinsk, Russia

Vladimir Shepelev, Tatiana Carpeta, Kirill Hazukov, Salavat Shabiev, Ivan Slobodin, Irakli Charbadze

Contributions

VS and KH designed research, performed research, analyzed the data, and wrote the paper. IS and TC designed research and was a major contributor in writing the manuscript. IC and SS gathered data and wrote the paper. All authors suggested related works, discussed the structure of the paper and results. All authors read and approved the final manuscript.

Corresponding author

Correspondence to Vladimir Shepelev.

Ethics declarations

Competing interests

The authors declare that they have no competing interests.

References

1. Zhang S, Wu G, Costeira JP, Moura JM. Table of contents. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). 2017. pp. v–liii. IEEE.doi.org/10.1109/CVPR.2017.4,
2. Peppa MV, Bell D, Komar T, Xiao W. Urban traffic flow analysis based on deep learning car detection from cctv image series. Int Arch Photogramm Remote Sens Spat Inf Sci. 2018;42(4):565–72. doi.org/10.5194/isprs-archives-XLII-4-499-2018.
3. Zhang S, Wu G, Costeira JP, Moura JM. FCN-rLSTM: Deep spatio-temporal neural networks for vehicle counting in city cameras. In: Proceedings of the IEEE international conference on computer vision. 2017. pp. 3687–96. doi.org/10.1109/ICCV.2017.396.
4. Rathore MM, Son H, Ahmad A, Paul A. Real-time video processing for traffic control in smart city using Hadoop ecosystem with GPUs. Soft Comput. 2018;22(5):1533–44. doi.org/10.1007/s00500-017-2942-7.
5. Sun X, Ding J, Dalla Chiara G, Cheah L, Cheung NM. A generic framework for monitoring local freight traffic movements using computer vision-based techniques. In: 5th IEEE international conference on models and technologies for intelligent transportation systems (MT-ITS). 2017. pp. 63–8. doi.org/10.1109/MTITS.2017.8005592.
6. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell. 2017;39(6):1137–49. doi.org/10.1109/TPAMI.2016.2577031.
7. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR). 2016, pp. 779–88. doi.org/10.1109/CVPR.2016.91.
8. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC. SSD: single shot multibox detector. In: Lecture notes in computer science. 2016; 9905: 21–37. doi.org/10.1007/978-3-319-46448-0_2.
9. Hu X, Xu X, Xiao Y, Chen H, He S, Qin J, Heng PA. SINet: A scale-insensitive convolutional neural network for fast vehicle detection. In: IEEE transactions on intelligent transportation systems. 2019; 20(3):1010. doi.org/10.1109/TITS.2018.2838132,
10. Jung H, Choi MK, Jung J, Lee JH, Kwon S, Jung WY. ResNet-based vehicle classification and localization in traffic surveillance systems. In: 2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW). 2017, pp. 934–40. doi.org/10.1109/CVPRW.2017.129,
11. Li S, Lin J, Li G, Bai T, Wang H, Pang Y. Vehicle type detection based on deep learning in traffic scene. Procedia Comput Sci. 2018;131:564–72.

doi.org/10.1016/j.procs.2018.04.281.
12. Sommer L, Acatay O, Schumann A, Beyerer J. Ensemble of two-stage regression based detectors for accurate vehicle detection in traffic surveillance data. 2019, p. 1–6. https://doi.org/10.1109/avss.2018.8639149.
13. Wang L, Lu Y, Wang H, Zheng Y, Ye H, Xue X. Evolving boxes for fast vehicle detection. In: 2017 IEEE international conference on multimedia and Expo (IC-ME). 2017; p. 1135–40. doi.org/10.1109/ICME.2017.8019461.
14. Zhu F, Lu Y, Ying N, Giakos G. Fast vehicle detection based on evolving convolutional neural network. In: 2017 IEEE international conference on imaging systems and techniques (IST). 2017; pp. 1–4. doi.org/10.1109/IST.2017.8261505.
15. Anisimov D, Khanova T. Towards lightweight convolutional neural networks for object detection. In: 2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS). 2017; pp. 1–8.doi.org/10.1109/AVSS.2017.8078500,
16. Li C, Dobler G, Feng X, Wang Y. TrackNet: simultaneous object detection and tracking and its application in traffic video analysis. 2019; pp. 1–10, arXiv :1902.01466.
17. Li S. 3D-DETNet: a single stage video-based vehicle detector. 2018. arXiv:1801.01769 .
18. Luo W, Yang B, Urtasun R. Fast and furious: real time end-to-end 3D detection, tracking and motion forecasting with a single convolutional net. In: 2018 IEEE/CVF conference on computer vision and pattern recognition. 2018; pp. 3569–77. doi.org/10.1109/CVPR.2018.00376 ,
19. Girshick R. Fast R-CNN. In: Proceedings of the IEEE international conference on computer vision 2015 Inter. 2015. pp. 1440–8. https://doi.org/10.1109/ICCV.2015.169.
20. Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2012. pp. 580–7, http://arxiv.org/pdf/1311.2524v3.pdf.
21. He K, Gkioxari G, Dollar P, Girshick R. Mask R-CNN. In: 2017 IEEE international conference on computer vision (ICCV). vol. 2017 Oct. 2017. pp. 2980–8. doi.org/10.1109/ICCV.2017.322, http://ieeexplore.ieee.org/document/8237584/.
22. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell. 2017;39(6):1137–49. doi.org/10.1109/TPAMI.2016.2577031.
23. Shreyas Dixit KG, Chadaga MG, Savalgimath SS, Ragavendra Rakshith G, Naveen Kumar MR. Evaluation and evolution of object detection techniques YOLO and R-CNN. International Journal of Recent Technology and Engineering. 2019; 8: 824-9. doi: 10.35940/ijrte.B1154.0782S319.
24. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. pp. 770-8.

25. Javadi S, Dahl M, Pettersson MI. Vehicle speed measurement model for video-based systems. Computers & Electrical Engineering. 2019. 76: 238-48 doi.org/10.1016/j.compeleceng.2019.04.001.
26. Manning CD, Raghavan P, Schütze H. Evaluation in information retrieval. In: Introduction to Information Retrieval. Cambridge: Cambridge University Press; 2009.
27. Wu Y, Jiang S, Xu Z, Zhu S, Cao D. Lens distortion correction based on one chessboard pattern image. Frontiers of Optoelectronics. 2015; 8 (3); 319-28. doi: 10.1007/s12200-015-0453-7.
28. Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables. Abramowitz M and Stegun IA editors. New York: Dover Publications; 1965.