

# Algorithm for detecting violations of traffic rules based on computer vision approaches

Samir Ibadov<sup>1</sup>, Ragim Ibadov<sup>2,\*</sup>, Boris Kalmukov<sup>1</sup>, Vladimir Krutov<sup>1</sup>

<sup>1</sup>Institute of Services Sector and Entrepreneurship (branch), Don State Technical University, 344010, Rostov-on-Don, Russia

<sup>2</sup>Institute Radioengineering Systems and Control, Southern Federal University, 344006, Rostov-on-Don, Russia

**Abstract.** We propose a new algorithm for automatic detect violations of traffic rules for improving the people safety on the unregulated pedestrian crossing. The algorithm uses multi-step proceedings. They are zebra detection, cars detection, and pedestrian detection. For car detection, we use faster R-CNN deep learning tool. The algorithm shows promising results in the detection violations of traffic rules.

## 1 Introduction

According to statistics, in the Russian Federation in 2015, there were 58,221 road accidents (RA) in which pedestrians suffered. Of this number of accidents, pedestrians hit 56886 times. As a result, 52,306 people were wounded, and 6,991 people were killed. Pedestrians who crossed the road on crosswalks have been hit by drivers of vehicles of 19779 times. At crosswalks, for one the year 1233 people have died, and 19576 people are wounded. How many of the wounded became disabled is not specified. The number of accidents of running-down pedestrians at crosswalks in 2015 on comparisons with 2014 has increased by 2%.

Analyzing the statistics of road traffic violations in the Russian Federation in 2015, it can be concluded that the creation of automatic detection systems for traffic violations on unregulated crosswalks is necessary to improve safety on the roads of the country. Therefore, the development of an algorithm for fixing traffic violations on unregulated crosswalks is urgent and will be in demand.

The algorithm works in two stages. At the first stage, it is necessary to detect three classes of objects in the video sequence. First class crosswalk (zebra), the second class moving vehicles and the third class the pedestrians going on the crosswalk. After all the object classes are found on each frame of the video sequence, the second stage begins. It is necessary to impose video sequence frames with the allocated objects at each other. If on the same frame of the video sequence the pedestrian and the vehicle are is at the crosswalk, the vehicle means violates Traffic regulations, having done not pass the pedestrian.

## 2 Zebra Detection

At the first stage, we need to detect a zebra; the video sequence is for this purpose loaded. Further, there is a splitting the video sequence into frame and allocation of an object and background using tags by the user [1]. Stages of the offered approach are presented in Figure 1.

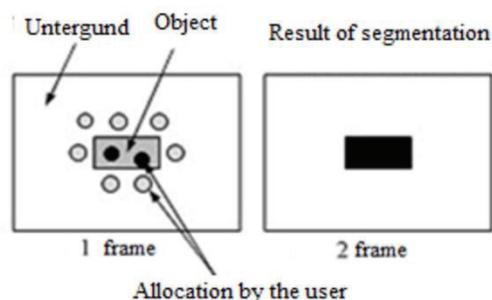


Fig. 1. Stages of segmentation.

To do this, first select the object frame is marked with a white background, the background is marked in black color the figure 2a, in the figure 2b an alpha the channel of the image received using a segmentation method on the received tags at the first stage is present. In the figure, 2c show the selected object on the original frame of the video sequence.

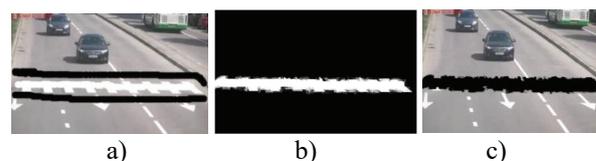


Fig. 2. Stages of segmentation of a zebra: a) marking of an object and background, b) alpha channel of the image, c) the allocated object.

Next, it is necessary to detect pedestrians on all frames of the video sequence.

\* Corresponding author: ragim\_ibadov@mail.ru

### 3 Car Detection Using Faster R-CNN Deep Learning

Faster R-CNN [2] is an extension of the R-CNN [3] and Fast R-CNN [4] object detection techniques. All three of these techniques use convolutional neural networks (CNN). The difference between them is how they select regions to process and how those regions are classified. R-CNN and Fast R-CNN use a region proposal algorithm as a pre-processing step before running the CNN. The proposed algorithms are typical techniques such as Edge Boxes [5] or Selective Search [6], which are independent of the CNN. In the case of Fast R-CNN, the use of these techniques becomes the processing bottleneck compared to running the CNN. Faster R-CNN addresses this issue by implementing the region proposal mechanism using the CNN and thereby making region proposal a part of the CNN training and prediction steps.

**Load Dataset.** This example uses a small vehicle data set that contains 295 images. Each image contains 1 to 2 labeled instances of a vehicle. A small dataset is useful for exploring the Faster R-CNN training procedure, but in practice, more labeled images are needed to train a robust detector.

**Create a Convolutional Neural Network (CNN).** A CNN is the basis of the Faster R-CNN object detector. Create the CNN layer by layer using Neural Network Toolbox™ functionality. Begin with the image input layer function, which defines the type and size of the input layer. For classification tasks, the input size is typically the size of the training images. For detection tasks, the CNN needs to analyze smaller sections of the image, so the input size must be similar in size to the smallest object in the data set. In this data set all the objects are larger than [16 16], so select an input size of [32 32]. This input size is a balance between processing time and the amount of spatial detail the CNN needs to resolve.

Next, define the middle layers of the network. The middle layers are made up of repeated blocks of convolutional, ReLU (rectified linear units), and pooling layers. These layers form the core building blocks of convolutional neural networks. Combine the input, middle, and final layers.

**Configure Training Options.** Trains the detector in four steps. The first two steps train the region proposal and detection networks used in Faster R-CNN. The final two steps combine the networks from the first two steps such that a single network is created for detection [2]. Each training step can have different convergence rates, so it is beneficial to specify independent training options for each step.

**Train Faster R-CNN.** Now that the CNN and training options are defined train the detector using `trainFasterRCNNObjectDetector`.

During training, image patches are extracted from the training data. The 'PositiveOverlapRange' and 'NegativeOverlapRange' name-value pairs control which image patches are used for training. Positive training samples are those that overlap with the ground truth boxes by 0.6 to 1.0, as measured by the bounding box intersection over union metric. Negative training samples are those that overlap by 0 to 0.3. The best values for these parameters should be chosen by testing the trained detector on a validation set. To choose the best values for these name-value pairs, test the trained detector on a validation set.

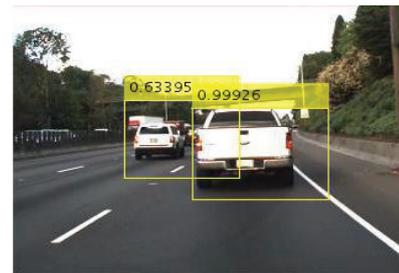


Fig. 3. Vehicle detection result.

After the network is trained, it shows reliable results.

### 4 Pedestrian Detection

We detect pedestrians using Motion-Based Multiple Object Tracking of a method.

Detecting of moving objects and tracking of their movement is an important element of computer sight [7].

Detection of moving objects uses the algorithm of subtraction of a background based on a Gaussian mixture of the model. Morphological operations are applied to a resultant mask of the foreground the figure 4a to the elimination of noise. Finally, the analysis reveals a group of related pixels that are likely to correspond to a moving object, the figure 4b.

The association of detection besides to an object is based only on the movement. The movement of each traced object is estimated using Kallman's filter. The filter is used for forecasting of tracking of the provision of objects in each shot.

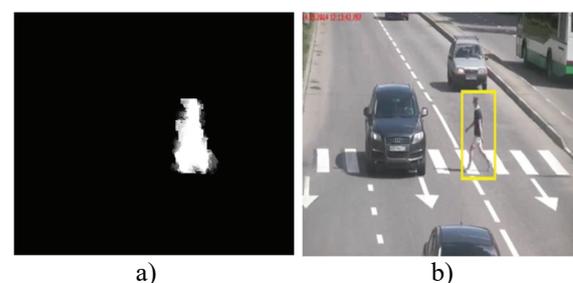


Fig. 4. Detecting of the pedestrian:  
a) the foreground, b) the detected pedestrian.

The detect objects function returns average points and limits to a framework the found objects. It also

returns a binary mask which has the same size, as an entrance shot. Pixels with value 1 correspond to the foreground, and pixels with value 0 corresponds to a background.

Function carries out segmentation of the movement, using the foreground detector. Then it carries out morphological operations over the received binary mask to remove noisy pixels and to fill the remained holes.

Kalman filter is used to predict the midpoint of each of the following structural frame and updates the bounding rectangle respectively.

### 5 Block diagram of the proposed algorithm

The block diagram of the proposed algorithm shows on figure 5. Algorithm for fixing a violation of the Traffic Rules does not allow pedestrians to pass at a pedestrian crossing by a vehicle in a video sequence.

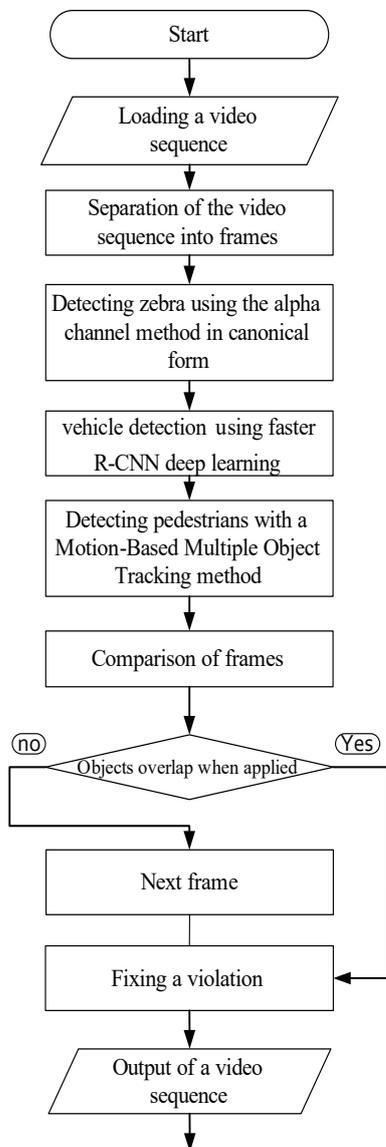


Fig. 5. Block diagram of the proposed algorithm.

The input is a video sequence. Next, the video is a separation of the video sequence into frames. After that, we need to detect three classes of objects. First class crosswalk (zebra), the second class moving vehicles and the third class the pedestrians going on the crosswalk. Detecting zebra occurs using the alpha channel method in canonical form. The vehicle is detected using Vehicle detection using faster R-CNN deep learning. Then, pedestrians are detected using the Motion-Based Multiple Object Tracking method.

The second stage of an estimated method of definition not of the admission by the vehicle of the pedestrian on a zebra consists in the comparison of the frame with the allocated objects. To reveal violation, not of the admission of the pedestrian on a zebra, it is necessary to impose video sequence frame with the allocated objects at each other. If on the same frame of the video sequence the pedestrian and the vehicle cross a zebra, the vehicle means violated traffic regulations, namely not the admission of the pedestrian on a zebra.

### 6 Comparison

For example, when comparing the 130th frame of the video sequence with the allocated vehicle the figure 5a the 1st frame of the video sequence with the allocated zebra the figure 6b we see that the vehicle crossed a zebra the figure 6c.

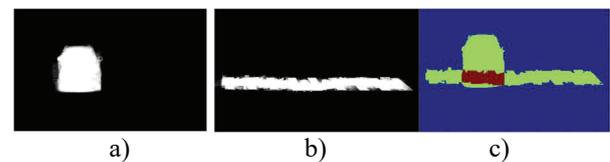


Fig. 6. Zebra crossing by the vehicle  
 a) the allocated vehicle on the 130th frame of the video sequence, b) the allocated zebra on the 1st frame of the video sequence, c) zebra crossing by the vehicle on the 130th frame of the video sequence.

Further, we check for crossing the 130th frame of the video sequence with the allocated pedestrian the figure 6a, with the 1st frame of the video sequence with the allocated zebra the figure 7b we see that the pedestrian crossed a zebra the figure 7c.

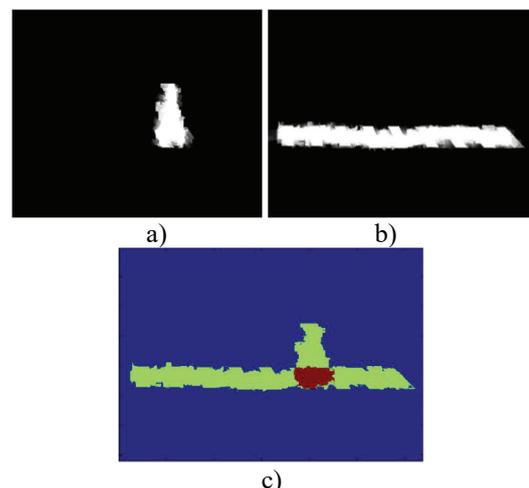


Fig. 7. Crossing by the zebra pedestrian:

a) the allocated pedestrian on the 130th frame of the video sequence, b) the allocated zebra on the 1st frame of the video sequence, c) crossing of a zebra by the pedestrian on the 130th frame of the video sequence.

Therefore, the vehicle violated traffic regulations, without having missed the pedestrian on the 130th frame of the video sequence.

## 7 Conclusion

We proposed the algorithm for automatic detect violations of traffic rules for improving the people safety on the unregulated pedestrian crossing. The algorithm uses multi-step proceedings. They are zebra detection, cars detection, and pedestrian detection. For car detection, we use faster R-CNN deep learning tool. The effectiveness of the new approach is shown in the examples demonstrating the effectiveness of the algorithm on test frames.

The reported study was supported by the Russian Foundation for Basic research (RFBR), research projects №15-07-99685 and №16-37-00386.

## References

1. Voronin V.V., Gapon N.V., Sizyakin R.A., Ibadov R.R., Ibadov S.R., Semenishchev E.A. *In Proc. XI International scientific-technical forum / DSTU* (2014). [in Russian]
2. Ren, Shaoqing, et al. *Advances in Neural Information Processing Systems* (2015)
3. Girshick, Ross, et al. *In Proc. IEEE Conference on Computer Vision and Pattern Recognition* (2014)
4. Girshick, Ross. *In Proc. IEEE International Conference on Computer Vision* (2015)
5. Zitnick, C. Lawrence, and Piotr Dollar. *In Proc. ECCV*, 391-405 (2014).
6. Uijlings, Jasper RR, et al. *In Proc. IJCV* (2013): 154-171.
7. Motion-Based Multiple Object Tracking, available at: [http://www.mathworks.com/help/vision/example\\_s/motion-based-multiple-object-tracking.html/](http://www.mathworks.com/help/vision/example_s/motion-based-multiple-object-tracking.html/)