Evgeniy V. Ershov*[1]*, Lyudmila N. Vinogradova*[2]*, M.I. Chevychelov*[3]*,

***Andrey A. Buturlakin*[4]*, Egor O. Volkov*[5]*, Anton A. Vasyaev*[6]**

FSBEI of Higher Education «Cherepovets State University», Cherepovets, Russia

[1]eve@chsu.ru, [2]lnvinogradova@bk.ru, [3]chevychelovmatt@gmail.com, [4]andrey.wheer@gmail.com, [5]egortech@live.ru, [6]nyacat96@yandex.ru

DOI: 10.25045/jpit.v10.i2.07

SOFTWARE VIDEO DETECTOR FOR THE DETECTION, CLASSIFICATION AND COUNTING OF VEHICLES FROM CCTV CAMERAS

Received: 21.12.2018 Revised: 27.02.2019 Accepted: 10.03.2019

The article describes approaches, methods and software implementation of algorithms for detection, classification and counting of vehicles based on convolutional neural networks of indepth learning.

Keywords: *convolutional neural networks, network input, frame, training, test, sample.*

Introduction

The detection, classification and counting of vehicles through images of video cameras is an actual task for:

- analysis of traffic congestion for traffic management optimization;
- detection of violations of the weight regime;
- analysis of the road deteriorations or objects of transport infrastructure .

High complexity of automated identification is one of the problems of transport classification in streaming mode [1].

The use of manual classification is routine and costly from human factor standpoint. Automated classification requires the implementation of machine learning algorithms and the formation of training samples. Algorithm analysis for recognizing transport types showed the neural network approach could be advanced [2], and convolutional neural networks (CNN) and groups of several deep neural networks were presented as the most effective tools [3–6].

Convolutional neural network, or convolution network, is a multi-layer perceptron contrived for recognizing two-dimensional surfaces with a high degree of invariance to transformations, scaling, distortions and other kinds of information [7].

The use of classical neural network architectures (Hopfield models, Kohonen’s self-organizing maps, Elman’s recursive networks) for the problem-solving in the conditions of the video stream is not effective because of external factors affect sensitivity (changing the camera angle, scale, speed of movement of cars). A convolutional neural network doesn’t have this disadvantage, so its implementation would be effective for a vehicle class recognition system.

Method of solution

The YOLO was selected as an CNN architecture [8] that operates in real time and outperforms other solutions such as Faster RCN, ResNet and SSD [9].

The structure of the convolutional network is shown in figure 1.

Darknet was chosen as the Interface for the operation description of a convolutional neural network based on the YOLO architecture [10]. This interface was implemented as a configuration file, which describes the architecture of the neural network [11].

The configuration file has the .cfg extension and represents the following structure: description of the sizes of the neural network input layer and the instructions for learning setting up and the description of the neural network layers.

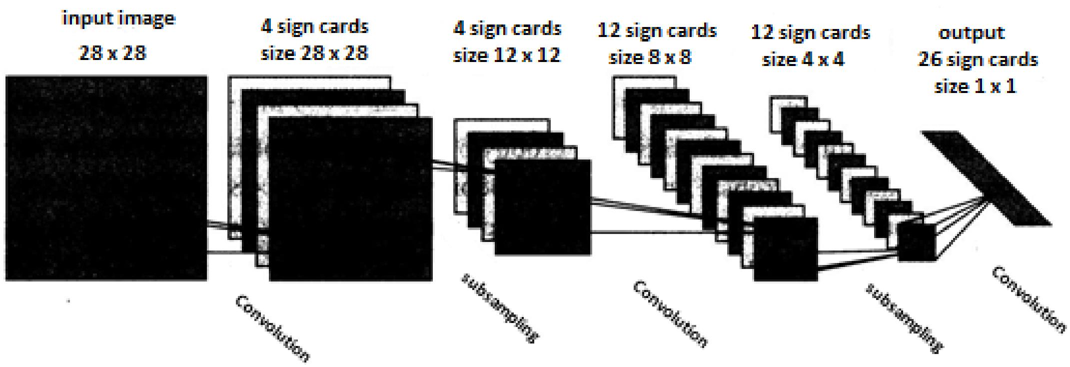


Figure 1. Structure of the convolutional neural network VGG

The characteristics of the neural network were determined to compile the configuration file:

1. Number of classes – 3.
2. Size of the subsample – 64.
3. Division of the subsample – 16.
4. Size of the neuron input matrix – 416 by 416.
5. Size of the gradient step – 0.0005
6. Size of the gradient step acceleration – 0.9.

The activation function of neurons on all convolutional layers, except the last one is leakyReLU. The activation function of the neurons on the last convolutional layer and on the layers that are added through the contraction layers is a linear function [12].

A convolutional neural network, to perform the assigned task, must be trained to use a set of data that would produce a training sample. It is also necessary to prepare a test sample. And these samples should be different.

The teaching selection consists of image files submitted to the neural network input during training, as well as metadata files for each of them, which indicates the class and coordinates of the object that the neural network needs to recognize. The training sample consists of image files submitted to the neural network input during training and metadata files for each of image file, which indicate the class and coordinates of the object for the recognition of the convolutional neural network.

Data used for training and verification of neural network for training are presented in three files. The first file contains a list of names of all classes, second one stores the paths to the images included in the training sample, and the last one contains paths to the images included in the test sample. Elements of these lists are separated by a new line. The paths to these files are specified in the neural network configuration file during training. A metadata file with a txt extension is placed around each selection element. The image file and the metadata file have similar names. There are descriptions of each selected object from the new line of file with txt extension in the format: *cxywh*, where

c– class ID of object that starts from zero and indicates the line number in the file of class list

x, y – coordinates of the frame center where the object is located

w, h – width and height of frame

All four parameters, defining the frame with the object, are relatively rationed to the image dimensions, i.e. $x, y \in [0; 1]$ and $w, h \in [0; 0,5]$.

Figure 2 shows the algorithm for preparing the training (L) and the test (C) sample. The backpropagation algorithm is used to train a convolutional neural network. This algorithm allows to determine which weights have had a greater impact on the losses and finds ways to adjust to reduce losses.

After training the neural network, an algorithm for detection and classification was developed (figure3), which can be divided into 4 stages:

1. Acquisition

At this stage, the frame is fed to the network input and analyzed for vehicles (TU). YOLO divides the image into a S x S lattice and searches for suitable TU in each cell. At the output, each detected TU is determined by a rectangle around, which is characterized by five values:

1. X coordinate of the cell center
2. Y coordinate of the cell center
3. the width of the framing rectangle applied to the entire source image
4. height of the framing rectangle
5. the probability that the rectangle correctly found the TU

2. Classification

All TU founded by the network must be classified. The network attaches a series of values to the TC, in addition to the five values, when the TU passes through the neural network. Attached values are the probabilities of TU belonging to each of the classes. Subsequently, the TU is assigned to the class where it has the maximum likelihood of belonging.

3. Tracking

Due to the fact that the network initially considers two identical TU on different frames as unique when moving to a new frame, it is necessary to use the tracking algorithm (detection). Using the received information on the TU frameworks detected on two consecutive video frames (current and previous) allows to define frames that are geometrically the most similar and close to each other, and also belong to the same class. This comparison allows to assume that these frames belong to the same TU.

4. Number-crunching

Checking the passage of detected TU through the specified control zones occurs at the counting stage. If the video editor finds out that the same TU was first on one side of the control zone and on the opposite one after, then, according to the direction of traffic specified for the given zone, TU passage through the control zone would be recoded. Upon completion of the detection and classification algorithm the TU counting algorithm starts to work using the data presented in figure 4. Red rectangles allocate the detected objects, and green dotted lanes show the counting lines for the calculating the TU.

The video detector is software implemented as a set of modules: preparation of the sample in the neural network format, detection and counting of vehicles, datastorage and UI.

The software used for development are .NET Framework 4.7.1, Microsoft SQL Server Compact 4.0, Microsoft Visual C ++ Redistributable Package 2017 x64, NVidiaDriver 387.128, compatible with CUDA 9.1.

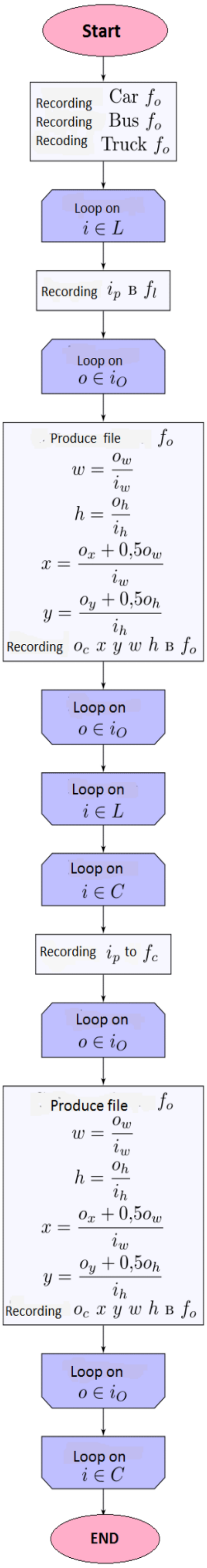


Figure 2. Algorithm flowchart for sample preparation

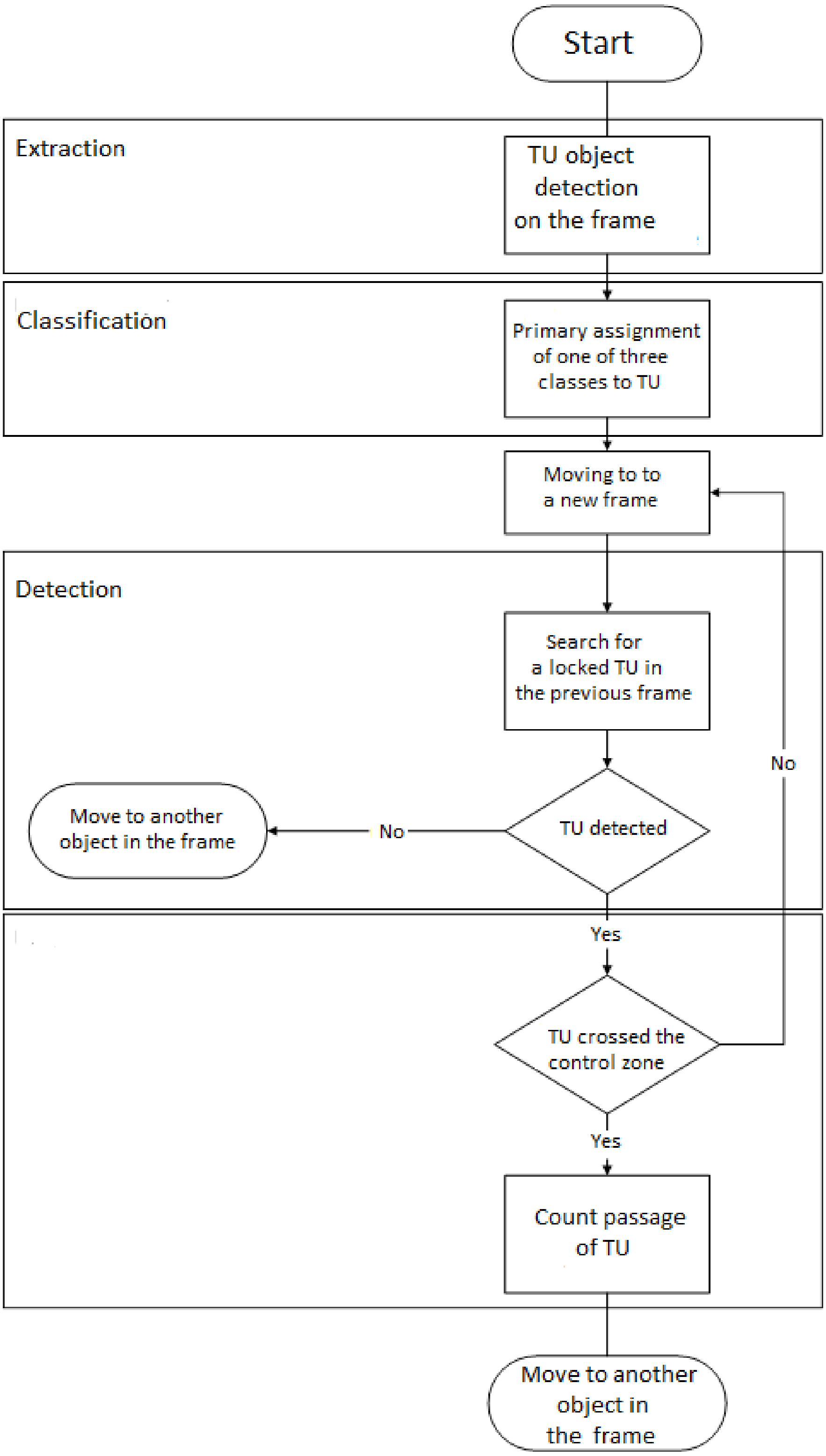


Figure 3. Algorithm flowchart of detection and classification

Results

The results of the tests confirmed the reliability of the software detector. The total percent of recognition in the daytime (figure 5, a) is 95% for cars, 90% for buses and 99% for freight vehicles; at night (figure 5, b) is 93% for cars, 69% for buses and 85 for freight vehicles. At night, the result is worsened by headlights and low lighting.

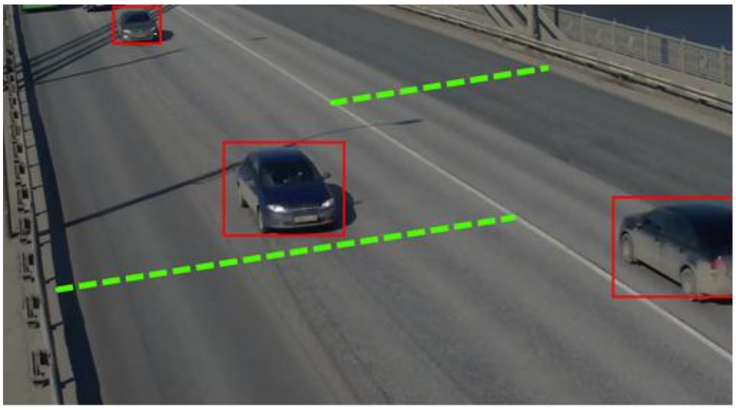
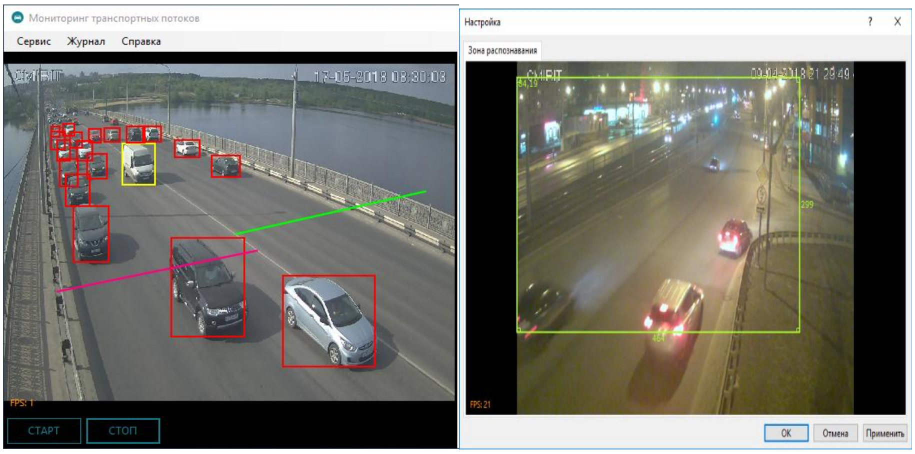


Figure 4. TU Calculating through counting lines



a)

b)

Figure 5. Detection of vehicles

a) Daytime; b) Nighttime

The work was carried out by a team of teachers and students of the Department of Mathematical and Computer Software of the Cherepovets State University, commissioned by LLC "Mullenom Systems" (Cherepovets).

Reference

1. Titov I.O. Contour Detection of Moving Image / I. O. Titov, G. M. Emelianov // Federal State Budgetary Educational Institution of Higher Education “Yaroslav-the-Wise Novgorod State University” bulletin, 2010, 27 p.
2. Mikheeva T.N. Application of Neural Network Methods for Analyzing Spatial Data Proceedings of Seven International Research and Practical Conference St. Petersburg, Saint Petersburg State University of Architecture and Civil Engineering , 2006, pp.81–84.
3. Barsky A.B. Path Choice in Step-by-step Alternative Routing in Wireless Telecommunication Network // World of Transport and Transportation, 2011, no.2. pp.4–11.
4. Barsky A.B. Path Choice in Step-by-step Alternative Routing in Wireless Telecommunication // World of Transport and Transportation, 2011, no.3. pp.14–19.
5. Yudin D. A. Recognition of Events on Images with Using Tweak of a Deep Neural Network // Optico-Electronic Devices and Devices in Recognition Systems Image,Image Processing and Symbol Information. Recognition - Digest of XIII International Sientific and Technical Conference, Kursk, 2017, pp.379–381.

6. Goodfellow I. Deep Learning, DMK-press, 2018, 652 p.
7. Graupe D. Principles of Artificial Neural Networks (Advanced Series in Circuits and Systems) / D. Graupe // Second Edition, World Scientific Pub CoInc, 2007, 238 p.
8. Umiarov N.K. Logical and Physical Interpretation of the Convolutional Neural Network Architecture / II All-Ukrainian Scientific and Technical Conference of Students, Postgraduates and Young Scientists, 11–13 April 2011, Donetsk, 2011, pp.81–85.
9. YOLO. Unified, Real-Time Object Detection. URL: https://pjreddie.com/media/files/papers/
10. YOLO. Real-Time Object Detection. URL: https://pjreddie.com/darknet/yolo/
11. Ershov E.V. Software Video Detector for the Detection, Classification and Counting of Vehicles from CCTV Cameras // Optoelectron Processor and Devices in Pattern Recognition System, Image Processing and Symbolic Information Systems. Identification / Digest of XIV International Sientific and Technical Conference, Kursk, 2018, pp.113–115.
12. Jim-Shih L. Dynamic synapse for signal processing in neural networks, United States Patent. No.: US 6,643.627 B2. Nov. 4, 2003.

UOT 004.94

Yerşov Yevgeniy V.[1], Vinoqradova Lyudmila N.[2], Çevıçelov Matvey İ.[3], Buturlakin Andrey A.[4], Volkov Yeqor O.[5], Vasyayev Anton A.[6]
Çerepovets Dövlət Universiteti, Çerepovets, Rusiya
[1]eve@chsu.ru, [2]lnvinogradova@bk.ru, [3]chevychelovmatt@gmail.com, [4]andrey.wheer@gmail.com, [5]egortech@live.ru, [6]nyacat96@yandex.ru

Videomüşahidə kameralarından nəqliyyat vasitələrinin aşkar edilməsi, klassifikasiyası və hesablanması üçün proqram videodetektoru

Məqalədə dərin təlimin konvensional neyron şəbəkələri əsasında nəqliyyat vasitələrinin aşkarlanması, klassifikasiyası və hesablanmasına dair yanaşmalar, metodlar və proqram reallaşdırılması alqoritmlərinə baxılmışdır.

Açar sözlər: *konvensional neyron şəbəkəsi, şəbəkə girişi, kadr, təhsil, seçim, yoxlama.*

УДК 004.94

Ершов Евгений В.[1], Виноградова Людмила Н.[2], Чевычелов Матвей И.[3], Бутурлакин Андрей А.[4], Волков Егор О.[5], Васяев Антон А.[6]
Череповецкий государственный университет, Череповец, Россия
[1]eve@chsu.ru, [2]lnvinogradova@bk.ru, [3]chevychelovmatt@gmail.com, [4]andrey.wheer@gmail.com, [5]egortech@live.ru, [6]nyacat96@yandex.ru

Программный видеодетектор для обнаружения, классификации и подсчета транспортных средств с камер видеонаблюдения

В статье рассмотрены подходы, методы и программная реализация алгоритмов обнаружения, классификации и подсчета транспортных средств на основе сверточных нейронных сетей глубокого обучения.

Ключевые слова: *сверточная нейронная сеть, сетевой вход, кадр, обучение, выборка, проверка.*