Development and Testing of a Pedestrian Traffic Monitoring System on a Convolutional Neural Network Platform

Aleksandr Mezhenin[1], Vladimir Polyakov[1], Vera Izvozchikova[2]

[1] *ITMO University, Kronverksky Ave. 49, St. Petersburg, 197101, Russia*
[2] *University Orenburg State University, Ave. Pobedy 13, Orenburg, 460018, Russia*

Abstract

Detecting, tracking and classifying objects in images and videos is an important task. His solution improves the safety of public places with large numbers of people. Many marketing departments and advertising agencies are interested in solving this problem. Various methods and algorithms of computer vision are used to analyze visual information in images and videos. There are ready-made implementations, both commercial and free. The paper discusses existing solutions for monitoring pedestrian traffic, their features. Open source solutions were tested and evaluated. The issues of using convolutional neural networks in monitoring systems are discussed in detail.

The issues of testing monitoring systems based on special data sets (Crowd Dataset) are considered separately. It is noted that such data are sometimes insufficient to conduct full-scale studies. It is proposed to use synthetic datasets (virtually constructed datasets) for testing software for detection, motion monitoring and object classification. It is proposed to create test videos using special 3D modeling and animation software, the so-called "animatics". Comparison of the two testing methods showed that test results on synthetic data correlate with test results on real data.

The author's algorithm is proposed, which made it possible to improve the accuracy of work, while maintaining an acceptable speed of work. The description of the algorithm is given and the results of its approbation are presented. Comparison with existing solutions was carried out on synthetic and real data. It was possible to improve the accuracy of recognition and further counting of pedestrians, ensuring an acceptable speed of work.

Keywords

Object tracking systems, pedestrian tracking, human detection, object detection, video surveillance, object movement monitoring, pattern recognition, convolutional neural networks, artificial intelligence, deep learning, YOLO, test suites, crowd datasets, synthetic datasets

1. Introduction

The proliferation of automatic tracking systems, including systems for monitoring or counting pedestrians, has become possible thanks to the development of image recognition technologies in the image. Such systems are used in various fields - counting visitors to trading floors, collecting statistics on visits to public events and cultural sites.

Typically, these systems use a webcam and firmware. However, systems are proposed that process the input video after shooting on separate personal computers. In such cases, it is important to achieve a reasonable compromise between detection and counting accuracy and program performance. This paper discusses existing solutions for monitoring pedestrian movement, their features, and proposes an approach based on the use of convolutional neural networks, which improves the accuracy of work, while maintaining an acceptable speed of work.

Russian Advances in Fuzzy Systems and Soft Computing: Selected Contributions to the 10th International Conference «Integrated Models and Soft Computing in Artificial Intelligence» (IMSC-2021), May 17–20, 2021, Kolomna, Russian Federation

EMAIL: mejenin@mail.ru (A. 1); v_i_polyakov@mail.ru (A. 2); viza-8.11@mail.ru (A. 3)
ORCID: 0000-0002-7150-9811 (A. 1); 0000-0002-2606-5385 (A. 2); 0000-0002-8707-9510 (A. 3)

CEUR Workshop Proceedings (CEUR-WS.org)

With an increase in the number of proposed monitoring systems, the issue of assessing their quality becomes important. Existing data sets do not always meet the required requirements. Creating test videos requires significant resources: providing the required number of people, choosing a shooting location, various camera configurations. This paper proposes an approach to testing pedestrian traffic monitoring systems and object classification based on the use of synthetic datasets (virtually constructed video datasets), which reduces the cost of creating test videos [19, 20].

2. Modern systems for monitoring the movement of pedestrians

Currently, a large number of applications have been implemented designed for monitoring and counting pedestrians on video.

Commercial solutions produced by manufacturers of specialized cameras. Macroscop is a system that allows you to determine the number of visitors entering and leaving using IP-surveillance cameras. The key features of this solution are the determination of the number of entered and exited visitors using IP cameras both through one and through several entrances and counting people in moving groups [1]. Domination is an application that performs statistical counting of visitors, peak load analysis. The system determines the number of people who entered and left shows a report in the form of graphs [2].

Solutions triggered by various accessories. The CleverCamera system works with IP cameras from Axis, SpezVizion, Beward and other manufacturers, using ONVIF, RTSP protocols. The system also includes computing and processing servers on which the software is installed [3]. SimbirSoft offers a solution based on UpBoard [4]. The implemented algorithm includes: selection of moving objects from the general video stream; segmentation of objects and search for people among them; division of groups into separate people; tracking people; counting the passage of visitors through certain gates.

Open source solutions. Footfall is a C ++ system using RaspberryPi, PiCam and openFrameworks [5]. Overhead-camera-people-counter is a C ++ solution using the OpenCV library [6]. Open source solutions are People-detecting [7] and People_counter [8]. "People-detecting" application uses algorithms: *accumulateWeighted. Pedestrian, FollowMovement*. The first, the simplest of the three, has low accuracy. The *Pedestrian* method uses a pre-trained oriented gradient histogram (HOG) model and support vector machine (SVM) [9]. Shows more accurate results, but consumes more resources and is slow on regular CPUs. Can be used if processing is done on GPU. FollowMovement method. This solution implements the method from the article "PeopleCounterwithOpenCVPython" [10]. The algorithm includes background selection, filtering, outline search, the most important step is motion tracking. The result is quite accurate, the method has an acceptable performance. The People_counter application uses an oriented gradient histogram classifier to detect people.

Testing of existing solutions. The features of these solutions were selected for analysis: People-detecting includes operations for background selection, filtering, edge search and motion tracking based on OpenCV methods. The above solutions were tested on videos shot with different shooting conditions. The results of testing the considered methods are shown in Fig. 1 and 2.

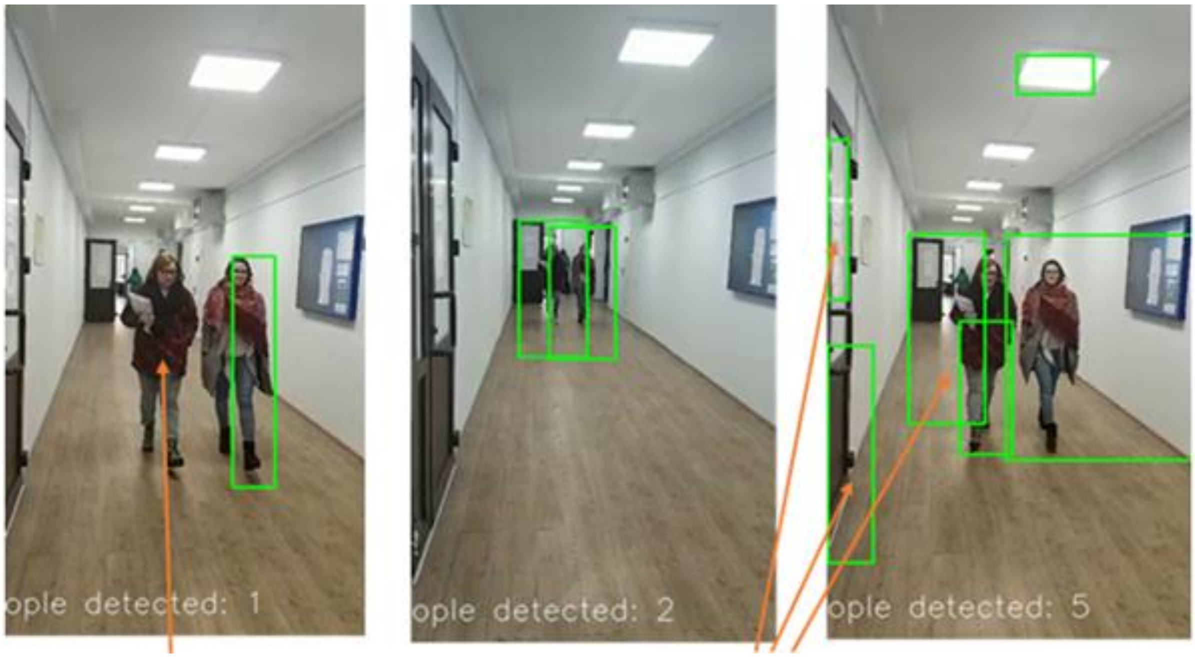

Figure 1: Monitoring results with a small number of people

Testing the three selected solutions allows us to draw the following conclusions: in most cases, the FollowMovement method detects and counts a large number of extra non-human fragments, the accumulateWeighted method detects fewer people than there are in the frame, even in the case of a small number of people (1-2) ... The most accurate results were obtained by the Pedestrian method using a pre-trained oriented gradient histogram (HOG) model and a support vector machine (SVM). However, the speed of this method is the lowest and it seems impossible to use this method without special powerful equipment.

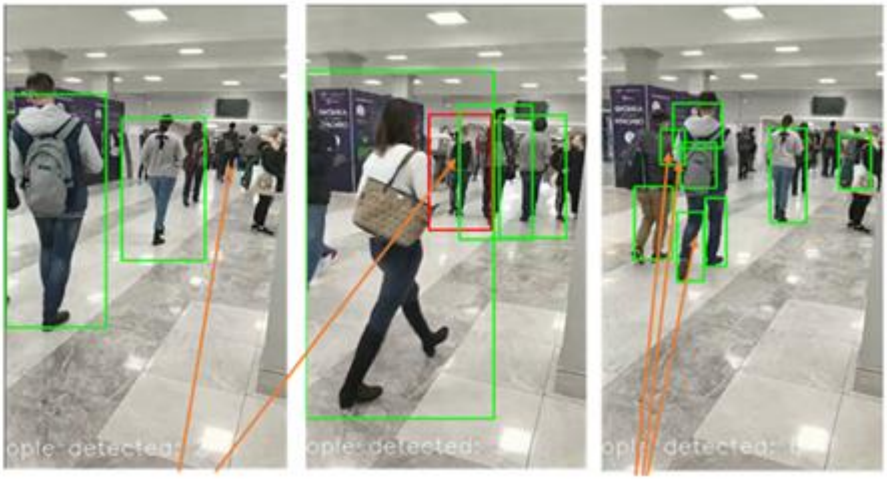

Figure 2: Monitoring results with a large number of people

Thus, all the considered solutions have certain disadvantages - a large number of errors in recognition or extremely slow operation speed in the absence of specialized equipment.

3. Convolutional neural networks in monitoring systems

There are many software tools that perform pattern recognition on an image. These include, first of all, various methods of machine learning, including classification, clustering, neural networks, and combinations of several methods [11]. Recently, the use of neural networks has become more and more successful. Convolutional neural networks are primarily intended for image analysis. The main feature of convolutional neural networks is the alternation of convolutional layers and subsampling layers (poolinglayers, or pooling layer), by analogy with the alternation of the so-called simple and complex cells of the visual cortex [12]. The essence of the convolution operation is the step-by-step multiplication of image fragments by some matrix - the convolution kernel.

Convolution is the first layer aimed at extracting features from the input image. Convolution preserves the connection between pixels by highlighting features in small squares of the input image. Fragments of the image and the convolution kernel or filter are fed to the input of the convolution operation [13]. In general, the convolution operation can be represented as the following formula:

$$(f * g)[m, n] = \sum_{k,l} f[m - k, n - l] * g[k, l] \tag{1}$$

where f is the original image matrix, g is the convolution kernel. By applying the convolution operation with various filters, you can perform such operations as detecting the edges of objects, blurring, and sharpening. The pooling layer compresses the fragments of the image, in which each disjoint group of pixels is compressed to one pixel. There are several selection algorithms - MaxPooling (maximum function), AveragePooling (average function), SumPooling (sum function). The point of pooling is to reduce redundant information and save only important features for further processing, as this also avoids overfitting. The alternation of convolutional and pooling layers is repeated until the desired result is achieved.

Further, the result of sequential convolution and pooling is fed to the input of a fully connected neural network, which can also consist of several layers - FullyConnectedlayers. Finally, for the final classification, the neuron activation function is used - some non-linear function that determines the output signal from a set of input signals. Very often the sigmoidal function is used as an activation function:

$$f(s) = \frac{1}{1 + e^{-s}}. \tag{2}$$

YOLO (You Only Look Once) is a real-time object detection system. It is a convolutional neural network belonging to the SingleShot class of detectors. With YOLO, you can recognize 80 classes of objects, including people, cars, furniture, etc. Due to its high operating speed, this architecture is suitable for processing and detecting objects in video [14]. YOLOv3 is an enhanced version of the YOLO architecture. It consists of one hundred six-fold layers and detects small objects better than its previous version - YOLOv2. The main feature of YOLOv3 is that the output has three layers, each of which is designed to detect objects of different sizes - the idea of the FeaturePyramidNetworks architecture is used. Also, the number of reference rectangles is increased to 9, and a more accurate classification function is used [15]. In the developed application, the computer vision library OpenCV was used, in the latest versions of which support for YOLOv3 was added, which ensured the convenience of their integration [16].

4. Monitoring algorithm implementation

Let us describe the main stages of the implementation of the proposed pedestrian monitoring algorithm based on the selected technologies - YOLOv3 and OpenCV. At the first stage, preliminary actions are performed to initialize the libraries and input data. Loading the configuration and YOLO weights is performed using the *OpenCVdnn.readNetFromDarknet* function from the *dnn - DeepNeuralNetwork* module. The result of executing this function is an object representing a neural network read from the input configuration files; initialization of input and output video files using the *OpenCVVideoCapture* function. The result of executing this method is an object of the input video stream. The height and width of frames of the input video are read using the *get ()* method of the input video stream. The variables used below are determined. Sets the location of the counting line, the color for marking the bounding boxes in the frame. In addition, the arrays are initialized to store the coordinates of the found objects in the current and previous frames and the counter required for the total count of people. Further in the cycle, frame-by-frame processing is performed. The frame is read using the *read ()* function called on the input video stream object. If the next block could not be read, processing ends and the cycle is exited. From the read source frame, a binary object is formed using the *dnn.blobFromImage* method and fed to the input of the neural network using the *setInput* and forward methods to process and obtain bounding boxes and their corresponding probabilities. Further, for all found objects, filtering by class is performed - only objects of the Person class remain, and objects with a probability of falling into the Person class for which are less than 0.5 are discarded. Based on the calculated coordinates, bounding rectangles are plotted on the output frame.

In the previously considered solutions for counting people, the main errors occurred at the detection stage - unnecessary objects were found that are stationary or are not people. Parts of a person were counted as several separate people or, on the contrary, the moving person was not detected, and therefore not counted. Finally, in solutions with a sufficiently high accuracy, errors are most likely if there are several people walking close to each other in the frame. In the developed solution, it is proposed to increase the accuracy of the final counting of people, first of all, through the use of tools that allow more accurate recognition of people in each separate frame.

5. Testing the developed system

Testing of the developed application was carried out using "live" test videos and synthetic data sets obtained in computer programs for 3D modeling and character animation [21]. The comparison was carried out with two previously considered open source solutions - "People-detecting" and "People_counter".

Testing on the evidence. A set of test videos of people moving along the same trajectory was filmed, in which various parameters were used: the height of the camera and the distance between people walking (Fig. 3). A vertical counting line is placed on the frame of the processed video. The detected people are circled in yellow, the probability of falling into the Person class is displayed for each object above the bounding rectangle, in the frame in the lower right corner there is an inscription with the current value of the counter of people who crossed the line. The proposed solutions made it possible to obtain greater clarity, and also helped to concretize the moments in which mistakes were made.

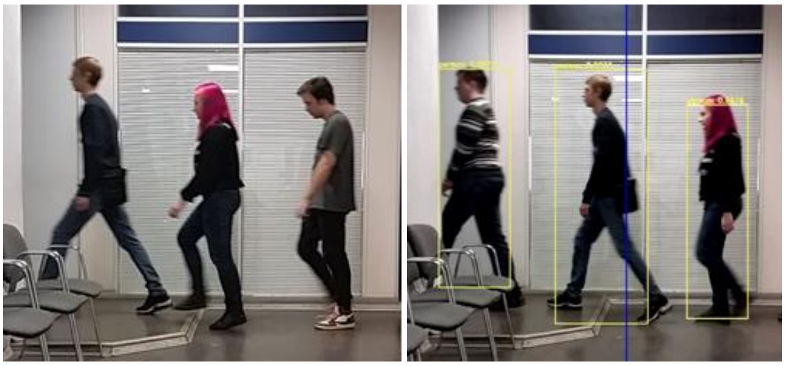

Figure3: Fragments of the test video and test results

Testing using synthetic data. Next, we consider the approach proposed by the authors - the use of synthetic datasets (virtually constructed datasets) to test software for detection, motion monitoring and object classification. It is proposed to create test videos by means of special software for character computer 3D modeling and animation, the so-called "animatics". In the studies conducted, we used test videos created in the Antics3D AnticsTechnologies program [17]. Antics3D was developed to create 3D character animation and event visualization in cinematography, forensic animation, education. Currently, the program is no longer supported, but it is freely available. The software employs drag, hover and click paradigms for animation, allowing users to create virtual scenes, direct characters to perform actions, and use virtual cameras to record scenes that can be edited on a timeline [18]. The result of processing test videos is shown in Fig. four.

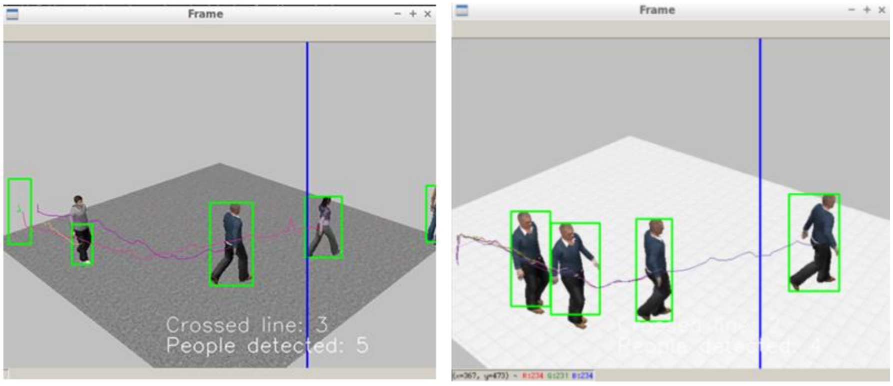

Figure 4: Test video processing results

For testing, videos were created with different spacing between pedestrians, different camera positions and different viewing angles. The results of one of the tests (shooting at an angle of more than 90 degrees) are shown in the diagram in Fig. 6. The diagram shows the obtained values of the number of people in comparison with the real value. The data is grouped by the height of the camera above the ground.

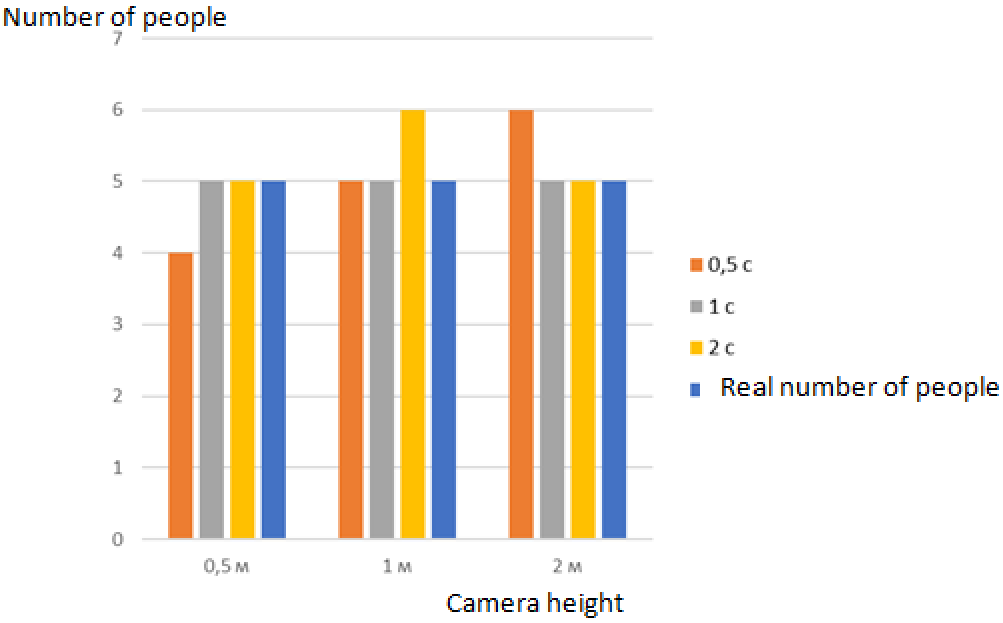

Figure 5: People counting results

Based on the data obtained, the following conclusions can be drawn: the developed algorithm is characterized by a small number of errors in counting, regardless of the location of the camera and the considered intervals between people. The reduction in the number of errors that occurs when people intersect in the frame is obtained through the use of the YOLOv3 architecture.

Comparison of the results of the developed algorithm with existing solutions. Comparison of the calculation accuracy in the considered solutions was made on the basis of the value of the relative error. For each application, the total number of errors made in all tests was calculated, and then the relative error was calculated, the final formula is:

$$\delta = \frac{\Delta n}{n_д} \tag{3}$$

where n is the number of people counted. For all cases, n_d is the actual value of the value, equal to 45, as the total number of people passed in the test videos. The results of calculating the relative error in determining the number of pedestrians, for real and synthetic data, are presented in the form of a bar chart (Fig. 6). The columns are grouped by the application under test.

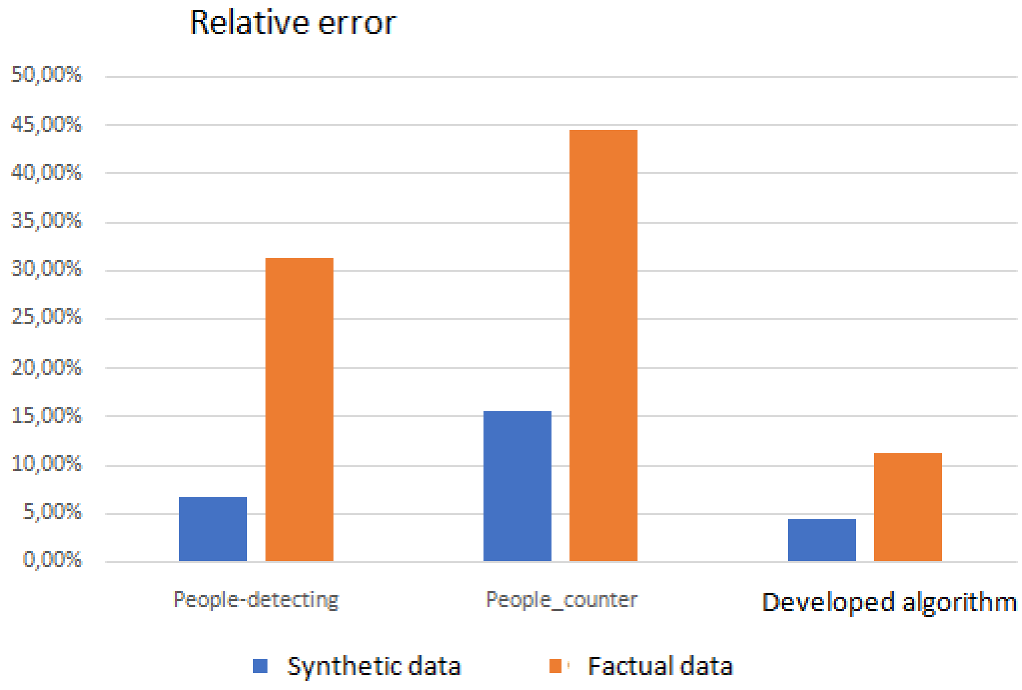

Figure 6: Comparison of relative error for synthetic and actual data

Based on the diagram obtained, the following conclusions can be drawn. When testing on actual data, there were more errors in recognition and counting compared to testing on virtual scenes. This is primarily due to the peculiarities of the background, clothing and gait of various people when shooting in real conditions. The developed algorithm showed the best results when compared with the considered open source solutions, when tested on real and synthetic data. There is a correlation between the number of errors made during testing in two different ways. The worse the result shown by the application when testing on synthetic data, the worse the result when testing on video with real people. Thus, despite the fact that when testing in two ways the number of errors does not coincide, the presence of a qualitative relationship between the results allows using the proposed testing method as less costly, which allows simulating various situations and shooting conditions.

6. Conclusion

In the course of the work, an algorithm for monitoring the movement of pedestrians was developed and tested. It was possible to improve the accuracy of recognition and further counting of pedestrians, ensuring an acceptable speed of work. The developed algorithm showed results comparable in accuracy in cases where pedestrian crossing does not occur in the frame. In the case of a small interval between people and their intersection in the frame, the developed algorithm demonstrates significantly fewer errors compared to the two tested applications - for the tests performed, the number of errors decreased several times.

Comparison of two testing methods - on synthetic and real data - showed that despite the presence of quantitative discrepancies in the accuracy of pedestrian counting, there is a connection between the quality of the application, demonstrated during testing in different ways. So test results on synthetic data correlate with test results on real data. The presence of such a relationship allows us to conclude that the use of synthetic data can be used for automatic monitoring of pedestrian movement, since this approach can reduce testing costs, while creating test data with shooting conditions that would be difficult to reproduce in real conditions. ... In the future, a more detailed study of the effect of shooting conditions on the coincidence of the results when testing in two different ways is possible.

7. References

[1] MacroscopUltra 3.0. Podschet posetitelej - URL: https://macroscop.com/assets/documentation/ultra-3.0/analytics/counting/counting.htm

[2] Modul' analitiki «Podschet posetitelej» Domination - URL: https://domination.one/products/analitycs/modul-analitiki-podschet-posetiteley-domination-n0000131235/

[3] CleverCamera. Instrukciya po ispol'zovaniyu - URL: http://www.clevcam.ru/assets/files/how_to_use.pdf

[4] Detektirovanie i podschet posetitelej v rezhime real'nogo vremeni na odnoplatnom komp'yutere «Up-board» - URL: https://habr.com/company/simbirsoft/blog/359304/

[5] Footfall. Application that allows you to monitor the traffic in and out of your building, using the RPi Camera and openFrameworks – URL: https://github.com/WatershedArts/Footfall

[6] Overhead-camera-people-counter. People counting algorithm using an overhead video camera – URL: https://github.com/koba/overhead-camera-people-counter

[7] People-detecting. Detect people from video/camera using OpenCV. FollowMovement – URL: https://github.com/ndaidong/people-detecting/tree/FollowMovement

[8] Counting people in a video stream – URL: https://github.com/shashgupta/people_counter

[9] Pedestrian Detection OpenCV by Adrian Rosebrock – URL: https://www.pyimagesearch.com/2015/11/09/pedestrian-detection-opencv/

[10] OpenCVPeopleCounter – [2018] – URL: https://www.pyimagesearch.com/2018/08/13/opencv-people-counter/

[11] Howard, W.R. Pattern Recognition and Machine Learning. Kybernetes, Vol. 36 No. 2 – [2007]

[12] Understanding of Convolutional Neural Network (CNN) – URL: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148

[13] Svertochnaya nejronnaya set', chast' 1: struktura, topologiya, funkcii aktivacii i obuchayushchee mnozhestvo» – URL: https://habr.com/ru/post/348000/

[14] YOLO: Real-Time Object Detection // https://pjreddie.com/ – [2018] – URL: https://pjreddie.com/darknet/yolo/

[15] What's new in YOLO v3? – URL: https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b

[16] Darknet Yolo v2 is added to the OpenCV – URL: https://github.com/pjreddie/darknet/issues/242

[17] Antics3D – URL: https://en.wikipedia.org/wiki/Antics3D

[18] Antics' new premium content and free 3D animation software – URL: http://www.studiodaily.com/2008/03/antics-technologies-opens-3d-animation-to-all/

[19] Mezhenin, A., Izvozchikova, V., Shardakov, V., Korotkikh, A.Technologies of reconstruction and procedural generation of three-dimensional content //Journal of Physics: Conference Series, 2019, 1399(3), 033018

[20] Mezhenin, A., Izvozchikova, V., Ivanova, V. Use of point clouds for video surveillance system cover zone imitation //CEUR Workshop Proceedings, 2019, 2344

[21] Izvozchikova, V.V., Shardakov, V.M., Mezhenin, A.V. 3D Modeling Surveying Tasks in Photogrammetry //Proceedings - 2020 International Russian Automation Conference, RusAutoCon 2020, 2020, стр. 220–224, 9208090