ISSN: 0976-3104
ISSUE: Multidisciplinary Science & Technology

ARTICLE

THE COMPARATIVE ANALYSIS OF METHODS FOR DETECTING PEDESTRIANS FOR UNMANNED VEHICLES

Rustem R. Ziyatdinov*, Ravil A. Biktimirov, Ksenia V. Klochkova

Naberezhnye Chelny Institute of Kazan Federal University, Naberezhnye Chelny, RUSSIA

ABSTRACT

Advanced driver assistance systems and unmanned vehicles require an environmental recognition function to make management decisions. To solve such problems, for example recognition of pedestrians, one of the methods of detecting dynamic objects should be used. To date, there are quite a lot of similar methods that differ in the quality and speed of recognition. The use of classical neural network architectures in such problems is of low efficiency. Therefore, to solve the pedestrian recognition problem, the convolutional neural networks were chosen as the main algorithm, since they provide invariance to small changes in input images, such as changing the survey angle, scale, rotation, displacement and other distortions. This paper presents a comparative analysis of methods for detecting objects in pedestrian recognition: SSD, YOLO, HOG + SVM methods. Comparison of these algorithms for the speed of work when recognizing pedestrians in the video stream was made in two modes of operation: using a central processor (CPU) and a graphics processor (GPU). The results of the work showed that the most promising method of detection in pedestrian recognition problems is the Tiny YOLO convolutional neural network method. Good results on the speed of work were also demonstrated by HOG + SVM. The use of GPU allowed to significantly increase the speed of object recognition.

KEY WORDS
pattern recognition, convolutional neural networks, pedestrian detection, ADAS systems, unmanned vehicles.

INTRODUCTION

Recognition of objects is quite a difficult task due to the diversity of both the recognized objects, and the methods used. The need for such systems exists in different areas - from recognition of malfunctions in automated diagnostic systems to the management of unmanned vehicles. Especially, this task is relevant in the development of automated systems for the recognition of the environment for use in advanced driver assistance systems (ADAS) and unmanned vehicles. In this case, the detection methods used must provide high recognition accuracy and real-time operation [1-4].

The object of the study are methods for detecting objects in pedestrian recognition, based on artificial neural networks and providing the ability to perform calculations by the graphics (GPU) or central (CPU) processors. The main advantage of using neural networks for detecting pedestrians is their learning ability based on previously obtained samples of images.

METHODS

The problems of image recognition most often use classical neural networks (a network of radial-basis functions, a multilayer perceptron, etc.). However, the analysis of experimental data obtained using such networks shows that the use of classical neural network architectures in such problems is of little effectiveness for the following reasons:

- recognizable images often have large dimensions, which leads to an increase in the structure of the neural network;
- a large variety of parameters increases the size of the system, which leads to an increase in the need for a larger training sample and, consequently, an increase in the complexity of computations and the time required for training the system;
- provision of high efficiency of the recognition system requires the use of several neural networks trained with a different order of providing the initial images and the initial values of the synaptic coefficients. This leads to an increase in the complexity of the solution to the problem and the time of its performance;
- high sensitivity to various changes in the geometry of input images, such as changing the shooting angle, image scale and other distortions [5].

Received: 21 Oct 2018
Accepted: 18 Dec 2018
Published: 6 Jan 2019

Subject to the above facts, to solve this problem of recognition of pedestrians, the convolutional neural networks were used as the main algorithm. This is due to the fact that they have some resistance to small scale changes, angle changes, shifts, turns, and other distortions.

***Corresponding Author**
Email:
rust.kfu@gmail.com
Tel.: 9534874499

The structure of classical convolutional neural networks contains many layers. These layers, as a rule, are of two types: convolutional and sub-sampling, alternating with each other [Fig. 1].

Neurons located within the layer are organized in the form of a plane. Each of the layers has a set of a corresponding number of planes. In this case, neurons in the same plane have the same synaptic coefficients, connecting them with all the local sections of the previous layer. Each neuron of the layer is connected to the area of the previous layer so that the input image of the previous layer is scanned through a small window. Then it is passed through the system of synaptic coefficients and fixed by the corresponding neuron of the current layer. The plane sets are feature maps, and each plane finds the corresponding parts of the image on the previous layer. The size of the scanning window is determined at the stage of development of the neural network.

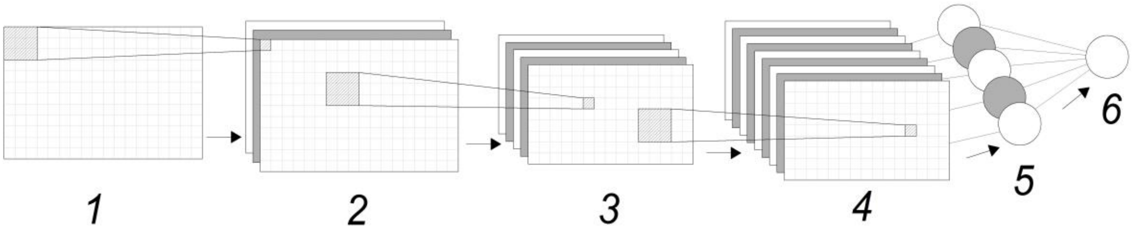

Fig. 1: The architecture of convolutional neural networks.
1 - input; 2,4 - convolutional layers; 3 - sub-sample layer; 5 - fully connected layer; 6 - neural network output.

The sub-sample layer reduces the dimensionality of the generated feature maps by locally averaging the values of the neuron outputs. As a result of this operation, the network becomes more invariant to the size of the input image. The map on each subsequent layer decreases in size, but the number of feature cards increases. This leads to the ability to recognize complex feature hierarchies [5].

Gradually, the neural network learns to distinguish the key characteristics of the necessary patterns in the incoming images. The responses of the neural network form maxima in the locations of these images [Fig. 2] [5, 6].

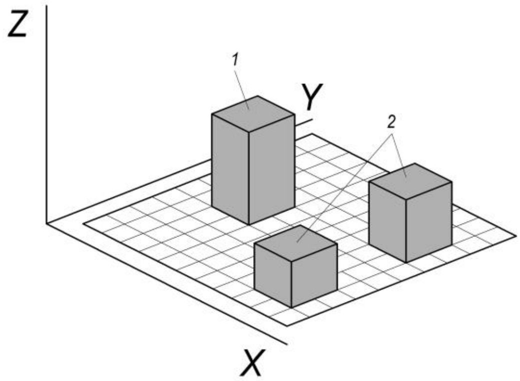

Fig. 2: Feedback from a convolutional neural network.
1 - the most probable position of the source pattern on the image; 2 - the least likely pattern finding on the image (noise).

The most significant limitation in the development of neural network algorithms is the extremely high computational cost of implementing these methods. The main and traditional ways to address this problem include the organization of distributed and parallel computing using specialized hardware, such as systolic neural processors, neural chips, distributed cluster systems, programmable logic integrated circuits (PLD), and GRID technologies.

A Compute Unified Device Architecture (CUDA) was designed to use graphics processors (GPU) to accelerate scientific and engineering calculations and perform various computations comparable in performance to cluster systems [7, 8]. The main difference between these architectures is the following: the execution of commands on the cluster takes place according to the MIMD architecture (each processor independently executes a different set of instructions, processing different data sets), and in the CUDA environment it is characterized by the SIMD architecture (a number of processors execute the same set of commands over different data, for example, array elements).

For convolutional neural networks, NVIDIA developed the CUDA Deep Neural Network (cuDNN) library. Unlike NVIDIA CUDA, which provides the ability to perform various calculations on the GPU, the cuDNN library is designed specifically for training deep neural networks. This library contains the optimized for the graphics processors implementation of convolutional and recurrent networks, various activation functions (sigmoidal, semilinear, hyperbolic tangent), back propagation algorithm, etc. cuDNN allows training neural networks on graphics processors several times faster than just CUDA.

RESULTS AND DISCUSSION

As the main models of convolutional neural networks for pattern recognition in the image, the following were used:

1. SSD: Single Shot Multi Box Detector. An object detection system with a single network based on Caffe – the framework for deep learning. This approach is based on dividing the output space of the bounding rectangles into a set of standard fields with different proportions and scales for each feature card. In forecasting, this method calculates the probability of having each category of objects in each such field and makes its corrections to better match the shape of the object [9].
2. YOLO: You Only Look Once. An advanced online object detection system, which uses a multiscale method of learning convolutional neural networks. YOLO can work with images of different sizes, offering a compromise between speed and accuracy [10].
3. HOG + SVM. To compare the speed of convolutional neural networks with classical classification algorithms, a pedestrian recognition system consisting of the HOG (Histogram of Oriented Gradients) algorithm, necessary for preliminary search for a pattern in an image and SVM (Support Vector Machine) classifier, is used to recognize the found assumptions [11].

For training, the COCO Image Dataset (for the Tiny YOLO convolutional neural network) and INRIA Person Dataset (for the SVM classifier) were used.

The results of using the pedestrian detection method are shown in [Fig. 3].

Fig. 3: shows the results of using pedestrian detection methods (from top to bottom): SSD, YOLO, HOG+SVM.

Comparison of these algorithms for the speed of work when recognizing pedestrians in the video stream was made in two modes of operation: using a central processor (CPU) and a graphics processor (GPU). To test the algorithms, various pre-trained models of convolutional neural networks and the SVM classifier were used. Compilation of the source code of programs was carried out in Visual Studio 2013 using Microsoft Visual C++ 2013.

The testing of the received programs was carried out on a computer with a 2GB video card NVIDIA GeForce GTX 950M, with an Intel Core i5-7200U processor and 8GB RAM. The original resolution of the test video: 640x360 pixels. This comparison does not take into account the accuracy of recognition, and the key emphasis is on the speed of the algorithms and the maximum number of frames per second. For each method, the detected image is considered positive at the threshold of 0.5.

The results of testing the methods are given in [Table 1].

Table 1: Results of comparison of the pedestrian detection methods

Method	The model used	Operating mode	Method operating time, msec	Number of frames per second
HOG+SVM	OpenCV	CPU	105	9.5
		GPU	33	29.4.
YOLO	Tiny YOLO	CPU	525	1.9.
		GPU	22	43
	YOLOv2 416x416	CPU	2250	0.43.
		GPU	63	15.8
	YOLOv3-320	CPU	2605	0.38.
		GPU	95	10.2.
	YOLOv3-416	CPU	4942	0.20.
		GPU	137	7.2.
SSD	PASCAL VOC COCO	CPU	2269	0.43.
		GPU	83	10.6.
	PASCAL VOC 07+12	CPU	2215	0.45.
		GPU	82	10.7.
	PASCAL VOC 07++12	CPU	2317	0.42.
		GPU	103	9.2.
	ILSVRC2016	CPU	2833	0.35.
		GPU	115	8
	COCO	CPU	2453	0.40
		GPU	98	9

The columns in [Table 1] show:

- Method: methods used;
- The model used: the used pre-trained model;
- Operating mode: the mode of operation of the compiled program (on a graphics processor (GPU) or using CPU resources);
- Method operating time: operation time of the method in milliseconds (not subject to the time of additional actions for preliminary and post-processing of images);
- Number of frames per second: the maximum number of frames per second for the video stream.

CONCLUSION

The results of testing these methods for detecting pedestrians in a video stream shows that the use of the GPU improves the speed of operation by an average of more than 25 times.
The best model without considering the accuracy of speed recognition is the Tiny YOLO convolutional neural network algorithm with 43 frames per second. The classic SVM classification algorithm also showed a high result with 29 frames per second. The low speed of the other models of convolutional networks is due to a large pre-trained model and a large number of layers embedded in their architecture. Using a more powerful graphics card in conjunction with the processor will allow these algorithms to be used in real-time operation.

The Tiny YOLO and HOG + SVM methods of the convolutional neural network can be used to detect various objects (cars, pedestrians, etc.) in the implementation of automated systems for the recognition of the environment for unmanned vehicles and advanced driver assistance systems. This will improve the safety of driving and reduce the number of road accidents.

CONFLICT OF INTEREST
There is no conflict of interest.

ACKNOWLEDGEMENTS
The work is performed according to the Russian Government Program of Competitive Growth of Kazan Federal University.

FINANCIAL DISCLOSURE
None

REFERENCES

[1] Ziyatdinov RR, Biktimirov RA, Klochkova KV. [2017] The comparative analysis of classification of algorithms in the problems of pattern recognition. Journal of Fundamental and Applied Sciences, [S.l.]. 9(1S):1790-97,1112-9867.

[2] Ziyatdinov RR, Mukhametzyanov VR, Nabiullina GI. [2015] Application of neural networks for the diagnosis of depth sucker rod pumps International Journal of Applied Engineering Research. 10-24(1):45022-26.

[3] Sharafieva LN, Ziyatdinov RR. [2016] Application of artificial neural network in the problems of diagnosis of borehole pumps Information Technology. Automation. Actualization and solving the problems of training highly qualified personnel: Proceedings

of the international conference Naberezhnye Chelny. 83-87. (in Russian)

[4] Ziyatdinov R, Shigabiev R, Talipov D. [2017] Automated road marking recognition system. IOP Conf. Series: Materials Science and Engineering. 240. 012070 doi:10.1088/1757-899X/240/1/012070.

[5] Yu N, Notkin BS, Sedov VA. [2009] Neuro iterative algorithm of tomographic reconstruction of the distributed physical fields in the fibre optic measuring systems Computer optics. 33(4):446–455.

[6] Wilson DR, Martinez TR. [2004] The general inefficiency of batch training for gradient descent learning, Neural Networks. 16(4):1429–1451.

[7] Fatica M. [2009] CUDA for High Performance Computing: materials of HPC-NA Workshop 3.

[8] FASTRA: Belgian researchers develop desktop supercomputer. Available at: http://fastra.ua.ac.be/en/index.html (accessed 15 April 2018).

[9] SSD: Single Shot Multi Box Detector. Available at: https://github.com/weiliu89/caffe/tree/ssd (accessed 17 April 2018).

[10] YOLO: Real-Time Object Detection. Available at: https://pjreddie.com/darknet/yolo/ (accessed 24 April 2018).

[11] Open CV: Open Source Computer Vision Library. Available at: https://opencv.org/ (accessed 01 April 2018).