Comparison Of Algorithms For Recognition Of Distorted Wagon Inventory Numbers

Mukhamadieva Kibriyo[1]
[1]*Tashkent University of Applied Sciences, Gavhar Str. 1, Tashkent 100149, Uzbekistan.*
mkb78@mail.ru

https://doi.org/10.5281/zenodo.10471758

Keywords: Augmentation, segmentation, combined algorithm, recognition, optimality coefficient.

Abstract: The relevance of the research is due to the need to develop methods, algorithms, and software tools to improve the efficiency of semantic segmentation of wagon numbers from the video stream in real-time. Despite the intensive development of modern methods and algorithms, they often do not provide the required quality of work and reliability, so today there is a need to improve the quality and speed of the semantic segmentation of objects in the images. Based on the analysis, we have concluded that the most effective solution for the semantic segmentation is the convolutional neural network CNN with approximated hyperbolic tangent FastTanh as an activation function and the optimization algorithm ADAM. A convolutional neural network model with an original architecture consisting of six layers is developed. Software implementation of the algorithm is done; it allows us to segment more precisely wagon numbers from the video stream in real-time and to increase the stability and speed of the algorithm in cases of heavily contaminated, low-contrast, and non-standard wagon number markers. A comparison of the results of different learning algorithms for the developed neural network is presented.

1 INTRODUCTION

The task of automatic detection, segmentation, and classification of objects is one of the most interesting tasks of modern computer vision. If in the tasks of classification, it is necessary to determine only the type of the depicted object, in the tasks of detection - to construct a bounding rectangle (or to determine the coordinates) for all objects of a given type, in the task of semantic segmentation it is required not only to detect and classify objects but also to determine their boundaries. In other words, for each pixel of the image, it is necessary to determine the class of object to which it belongs. Thus, the task of semantic segmentation is the most difficult task of image processing. [17] The difficulty of processing is complemented by the high variability of objects within one class and the high similarity of elements of objects of different classes. Of particular interest is the possibility of solving the problem of semantic segmentation on computing systems in real-time.

This article aims to find the most effective way for semantic segmentation of images from the viewpoint of a compromise between speed and accuracy. Speed issues are extremely important for the application of real-time image analysis algorithms, in the case we are considering, the recognition of wagon numbers.

This article discusses the chosen method of the algorithm, training, and activation of neural networks designed for license plate segmentation. In the experimental part of the article, the numerical characteristics describing the results of the combined algorithms under study are analyzed.

Given the fact that the problem we are considering is not widely covered in publications, we have decided to compare the existing and newly developed algorithms in native conditions, comparing the quality and speed of their work. The conclusion contains a discussion of the results and main conclusions.

2. Materials and methods

The main approaches to the semantic segmentation of images include the combined use of three types of algorithms: detectors, descriptors, and classifiers, which determine the basic image parameters, select objects, and classify them. The basic image parameters can be brightness, color, texture, corners, and borders of objects in the image and the like.

Among the most popular and effective algorithms that include detectors and descriptors are SIFT, SURF, FAST, MSER, and HOG algorithms [23-28].

The SIFT (Scale Invariant Feature Transform) algorithm includes a detector and a descriptor. The SIFT detector is based on the use of scalable spaces - the set of all possible, smoothed by a particular filter, versions of the same image. Using a Gaussian filter, this scalable space becomes invariant to shifts, rotations, and scaling, which does not shift local extrema. Three parameters are used to determine the key points: the displacement from the exact extremum using the Taylor polynomial; the contrast value of the difference Gaussian; and finding the point on the object boundary using the Hesse matrix. Then the orientation of the key point is calculated based on the direction of gradients of neighboring points [23, 24].

The SURF (Speeded Up Robust Features) algorithm is an upgrade of the SIFT detector, but instead of the Gaussian function, it uses a rectangular 99 filter to approximate it, thus speeding up the result of the algorithm. In the SURF descriptor, a square area is built around the point of interest and divided into square sectors in which the responses to the Haar wavelets, directed vertically and horizontally, are computed. These responses are weighted and summed for each of the sectors [25].

The FAST (Features from Accelerated Segment Test) algorithm does not require the calculation of brightness derivatives but compares the brightness in a circle from the tested point. First, a quick test of four points from the tested one is carried out, and then the others are tested. The number of tests and their sequence is determined on the training sample [26].

The MSER (Maximally Stable Extremal Regions) algorithm is based on determining the pixel intensity of the image and comparing it with some threshold (if the pixel intensity is greater than the threshold, it is considered white, otherwise - black). Thus, we build a pyramid of images with white images at the beginning and black images at the end. Such a pyramid allows one to construct a set of coherent intensity components that are invariant to affine transformations [27].

The HOG (Histogram of Oriented Gradients) algorithm is a key point descriptor based on counting gradient directions in local image regions. The image is divided into small coherent regions, which are called cells, and for each cell, a histogram of gradient directions and edge directions for pixels within a cell is calculated. The output of the descriptor is a combination of these histograms [28].

The advantages of these algorithms include high stability to various geometric and photometric transformations and image scaling. The disadvantage of these algorithms is the low stability of operation when the registration angles, illumination conditions, and reflective surfaces change. Especially in cases of heavily contaminated, low contrast, and non-standard wagon number markers.

Among the classifiers for semantic segmentation, various variants of CNN are most actively used

Faster-RCNN [19] introduced the regional suggestion network (RPN) to replace selective search, which makes Faster-RCNN faster and gives higher accuracy. However, the region proposal stage is still a bottleneck due to the use of the selective search algorithm

FCN [20] fully convolutional encoder-decoder-based networks are widely used for dense image labeling tasks. "Encoder" networks are typically backbone CNNs that use cascading and convolutional levels to learn semantic information about objects. In contrast, the "decoder" parts are usually up sampling or deconvolution operations to recover the lost spatial resolution of encoded features. **SegNet** [29] has a similar design but uses pooled indexes to record and recover spatial information.

RefineNet [11] strengthens the decoder by multilevel function fusion of different levels. Multilevel function fusion is further enhanced in Exfuse [6] by using both pixel sum and concatenation operations. The connection between high-level and low-level functions is also introduced in DeepLabv3+ [12]. DenseASPP [13] and UNet++ [14]. One of the limitations of these decoder-encoder designs is that there is a significant loss of spatial detail at the encoding stage, and the decoders are still not powerful enough to recover all the lost information.

Of interest is the development of a segmentation algorithm that applies machine learning techniques to analyze a string image with little or no additional preprocessing or post-processing (end-to-end). Such approaches are distinguished by the fact that they do not require manual fine-tuning for a particular case but require a representative training sample of sufficiently large size. This makes it possible to simplify and accelerate the creation of segmentation algorithms for new types of recognized objects, as well as to increase the accuracy and robustness of various distortions arising in the imagery.

A special feature of our approach is the use of the convolutional neural network CNN with an error back propagation algorithm, L2-regularization, Mini-batch gradient descent method and as an activation function FASTTANH fast hyperbolic tangent approximation and ADAM optimization algorithm for semantic segmentation

2.1. DEVELOPMENT OF A CONVOLUTIONAL NEURAL NETWORK

In recent years, CNNs have shown high results in solving problems of object classification on images. The efficiency of this approach is explained by the fact that convolutional neural networks are flexible tools and allow adapting their structure and parameters to solve the task at hand.

Most approaches to building semantic segmentation algorithms involve the following steps:

1. Data preprocessing.
2. Pre-segmentation.
3. Feature description.

Classifier training and classification.

5. Context-aware post-processing.

It can be noted that the algorithms have a modular structure, which allows for choosing different methods at each stage and their combination.

To date, there are no clearly regulated rules for the implementation of CNN structure: the number and organization of layers, the number, and size of feature maps, the size of convolution matrices, and the choice of the learning algorithm. CNN is based on the principles of local perception and separable weights. Local perception implies that the input of one neuron does not receive all outputs of the previous layer, but only a certain part of them [3, 7].

Convolutional neural networks have a much smaller number of tunable parameters. Also, this type of neural network is very robust to scaling, shifting and rotating and other input data transformations [7-9].

The main goal of the experiments was to build the configuration of the neural network with the smallest number of parameters. In the process of experimental studies, CNN's of different architectures were implemented, including different numbers of parameters. Experiments showed that neural networks with simplified architecture and a small number of parameters showed the worst results. By sequentially complicating the CNN architecture, we managed to find the optimal architecture which ensured high classification results (Figure 1). Further experiments on complicating the architecture and increasing the number of CNN parameters did not improve the quality of classification, but the network operation and learning time increased significantly.

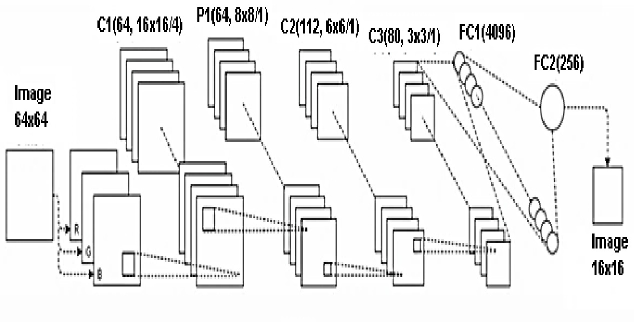

Figure 1 The architecture of the developed convolutional network

The experimental neural network was built using the Caffe framework [2]. This neural network consists of 6 layers and includes 3 convolutional layers, 1 subsample layer, and 2 fully connected layers. Color images are used as input data. The input layer has a size of 64*64 neurons. This layer does not perform any transformations and is only intended to feed it with input data.

After the input layer, the first hidden layer C1 is located. This layer is convolutional and contains 64 feature maps, each of which has the size of 16*16 neurons. The convolution matrix has a size of 44 neurons. The displacement is performed by 4 neurons.

The second hidden layer P1 is a subsampling layer, it consists of 64 feature maps, each of which has the size of 88 neurons. The convolution matrix has a size of 22 neurons. The shift is performed by 1 neuron. This layer reduces the size of the previous layer by half.

The third hidden layer C2 is convolutional and consists of 112 feature maps, each of which has the size of 66 neurons. The convolution matrix has a size of 22 neurons. The displacement is performed by 1 neuron.

The fourth hidden layer C3 is also convolutional and consists of 80 feature maps of size 33 neurons. The convolution matrix has a size of 33 neurons. The displacement is performed by 1 neuron.

The fifth hidden layer FC1 is fully convolutional. This layer consists of 4096 neurons and has a structure in the form of a one-dimensional vector.

The sixth hidden layer FC2 consists of 256 neurons and also has a structure as a one-dimensional vector.

The first four layers of the network have a two-dimensional structure and are designed to extract features from the image. The last two layers have a one-dimensional vector structure and are designed to classify the features extracted from the previous layers. At the output, the neural network generates a vector of 256 values, which is converted into a two-dimensional matrix of 16*16 pixels in grayscale. The values of each pixel of the output image range from 0 to 255. Initialization of synaptic coefficients of the network was set randomly in the range from 0 to 1.

When developing the neural network structure, it is also necessary to select an activation function that is designed to calculate the output signal of the artificial neuron.

The hyperbolic tangent function was chosen to solve this problem because it has several advantages [4], which are as follows:

- it is symmetric about the origin and provides faster convergence compared to the logistic function;
- it has a simple derivative;
- it is easily differentiable, which simplifies training of the network by the backward error propagation method
- has a maximum of the second derivative at = 1.

The hyperbolic tangent function has a range of values from -1 to 1. This allows the dynamic range of the sigmoid to be used twice in training to give negative values to the output signals in the classification layers. The hyperbolic tangent is given by the formula [4]:

$$f\ (x) = \ a\ tanh(bx\ ) = a\frac{(e^{bx}-1)}{(e^{bx}+1)}\ ; \qquad (1)$$

where a and b – constants.

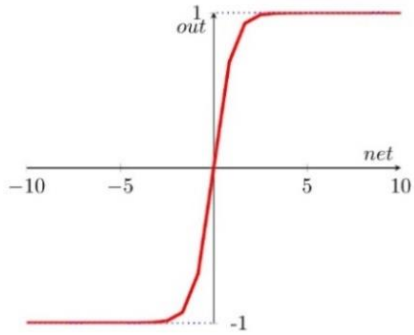

Figure 2 Hyperbolic tangent

However, the use of the hyperbolic tangent in the network with a large number of neurons leads to a slow-down of the calculation and learning process, this is because it is required to calculate the exponential function which affects the CPU time.

To solve this problem the algorithm FASTTANH [5] based on POSIT arithmetic was developed.

It is known that the sigmoidal function is

$$sigmoid(x) = \frac{1}{(e^{-x}+1)} \qquad (2)$$

then the hyperbolic tangent of $tanh(x)$ can be expressed as:

$$tanh(x) = \frac{(e^{2x}-1)}{(e^{2x}+1)} = 2 * sigmoid\ (2 * x) - 1 \qquad (3)$$

From this formulation, an equivalent is constructed which uses only L1 operators to construct an approximated hyperbolic tangent. Since we are dealing with 0 exponent bits of POSIT, all the computation is just a matter of manipulating the bits, thus efficiently and quickly computable [5].

Thus, going from sigmoid to a fast version called FastSigmoid, the approximated hyperbolic tangent looks like this:

$$FastTanh(x) = -\left(1 - 2 * FastSigmoid(2 * x)\right) \quad (4)$$

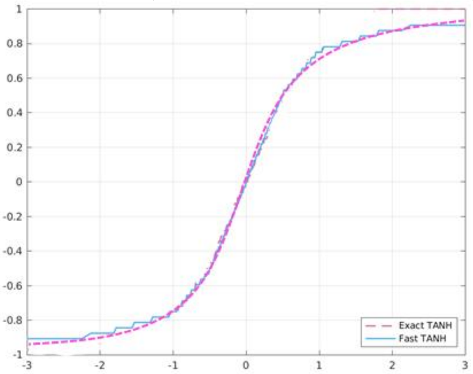

Figure3 Comparison of approximated and real hyperbolic tangent

Such approximation of the hyperbolic tangent in contrast to *k-tanh* gives an insignificant loss in accuracy of 0.3%, with a gain in calculation speed of 1.5-2 times [5].

2.2.1CHOICE OF THE TRAINING ALGORITHM

Neural network training is the sequential correction of synaptic weights between neurons. One of the most common and effective learning algorithms for neural networks is the error back propagation algorithm [10, 14]. The algorithm gets its name from the fact that the error calculated at each iteration propagates through the ANN from the output to the input to reconfigure the synaptic weights. In the process of training the network, when the input vector is fed, the network output is compared with the output from the training sample, forming the error [14]. The correction of synaptic weights is performed by the following formula [14]:

$$E_k = \frac{1}{2}\sum_{j=0}^{N}(t_{kj} - x_{kj})^2 \quad (5)$$

where t_{kj}- learning rate coefficient; x_{kj} - neuron input value;

Value of the network neuron error is defined by the formula [14]:

$$\Delta w_{ij} = -\eta \delta_{kj} x_{kj} \quad \mathrm{I}(6)$$

where η-learning rate factor; x_{kj} - neuron input values; δ_{kj}-neuron error;

$$\delta_j^{(q)} = (f_i^{(q)}(S))' \sum_j w_{ij} \delta_j^{(q+1)} \quad (7)$$

$\delta_j^{(q)}$- value of the error of the *i*-th neuron in the layer q; $\delta_j^{(q+1)}$ - value of the error of the *j*-th neuron in the layer *q+1*; w_{ij} - weight of the connection connecting two neurons; $(f_i^{(q)}(S))'$ - value of derivative activation function of the *i*-th neuron in the layer q.

To regularize the network, we used L2 regularization, which is a large penalty for a too high value of the weight, and a small one for a low value, which is expressed in the use of the regularization coefficient.

We add to the error function a component that is proportional to the square of the weight's values

$$C = E + \frac{\lambda}{2n}\sum_{i=1}^{n}\omega_i^2 \quad (8)$$

$$\frac{\partial C}{\partial \omega_i} = \frac{\partial E}{\partial \omega_i} + \lambda \omega_i \quad (9)$$

This forces the weights to be small, except when the error gradient is large

The advantages of this learning algorithm are:

- ease of implementation,
- ability to use many loss functions,
- ability to apply large amounts of data.

The disadvantages of the algorithm include small correction of weights, which leads to a long learning process. This raises the problem of selecting the optimal step size. Too small step size leads to slow convergence of the algorithm, too large step size can lead to loss of stability of the learning process [14].

To solve these problems, there are various optimization methods for this algorithm. Out of many existing optimization methods for training and subsequent comparison of their performance, Adam (Adaptive moment estimation), an optimization algorithm that combines the principles of momentum accumulation and gradient frequency conservation, was chosen. This method has advantages of Nesterov accelerated gradient [1], and AdaGrad [4]. The algorithm, unlike others, does not fall into the traps of local minima. Adam optimization can improve the performance of a wide and deep neural network [21][22].

Also, during training, the minimization of the loss function was performed using the Mini-batch gradient descent method [16].

2.2.2 TRAINING AND TESTING THE DEVELOPED ALGORITHM

For training and testing the developed CNN, a database of images consisting of several thousand images of wagons was used. The size of each image is 1920*1080 pixels.

To improve stability, artificial expansion (augmentation) of the training sample was used using data transformation. Synthesis of each sample was carried out by applying a random set of transformations, simulating the transformation of the real field image.

To expand the training sample, the following transformations were applied (modeling errors of the system in real conditions): addition of Gaussian noise distortion, projective distortion to simulate non-ideal finding, Gaussian blur to simulate defocusing, image stretching in height and width, vertical and horizontal shifts and mirror reflections. The following are illustrations of the described transformations.

Conversion	Illustration
Original image	
Gaussian noise	
Projective distortions	
Gaussian Blur	
Shifts	

Reflections

Stretch

Transformations Combination

Figure 4. Augmentation of training samples

All images are grouped into training, test, and validation samples in the ratio of 0.7/0.2/0.1. As seen in Figure 2, the images contain different classes of objects. The main objects of interest for the task at hand are wagon numbers. Figure 3 shows images of segmented objects. These images correspond to the original images from the training sample and are intended for CNN training. In the process, CNN processes small portions of the input images according to the size of the input layer (64*64 pixels). Thus, the input image is sequentially scanned by a window of 64*64*64 pixels in size. At each location of this window, the neural network performs image feature segmentation, forming a map of 16*16 pixels at the output. This difference in size is since when sampling a small portion of the image, it is often difficult to know what is depicted on it (Figure 4). The increased size of the input image area allows for saving some data for more effective classification (Figure 4). To avoid the problem of overtraining, in the fifth full-link layer is implemented method DropOut [15], which is that during the training from the overall network is repeatedly and randomly allocated to a certain subnet, and update weights occur only within this subnet. Neurons fall into a subnetwork with a probability of 0.5.

The following neural network parameters were used in training and testing:

- 0.0005 learning coefficient;
- frequency of learning coefficient change 104;
- the training coefficient variation value is 0,1;
- attenuation for L2 regularization 0,0005.

The configuration of the network remained unchanged. The number of training epochs for each case was 400.

3. COMPARISON WITH OTHER IMPLEMENTATIONS OF CONVOLUTIONAL NETWORKS

When developing segmentation methods, as in the development of any algorithms, it is necessary to fix a way to assess the quality of their performance. This method should allow for the comparison of the developed method with other algorithms. Let us describe the quality indicators used in this paper to evaluate methods of wagon number segmentation.

The purpose of text segmentation into symbols is its subsequent recognition, which determines the popularity of using the final recognition quality as an evaluation of segmentation algorithm quality. An estimate of the quality of the recognition algorithm can be both the accuracy of recognition of individual characters or words, and the average Levenshtein distance. The indicators of quality of the wagon number recognition system in this work were the accuracy of full recognition within a symbol because of the high cost of a single error in a single field - an error of even a single digit is critical.

In all experiments as the recognition algorithm was used Tesseract, with default settings. All experiments were done with the following hardware: Intel Core i5-6400 processor, 8GB RAM, NVIDIA Quadro K5200 graphics card.

Let's look at examples of images of numbers of wagons taken by us in the working railway station:

a)

b)

Figure. 5. Groups of Wagon Numbers

The images of the wagon numbers were divided into several groups

a) Poor quality wagon numbers, with high contamination and low contrast

d) images having inscriptions close in size to the wagon numbers, as well as in some cases being placed on the same line.

In the course of our experiments with the segmentation of wagon numbers with the above-mentioned algorithms, we made a selection of 100 images for each group. We encountered the problem of low segmentation accuracy in images of groups (a) and (b). The problems of segmentation in the group (b) were partially compensated by comparing the coordinates of the resulting segments.

Table. 1-4 shows the results of image segmentation by convolutional neural networks.

Table 1 shows the results of the experiments:

Segmentation algorithm	a	b
SIFT	15,2	77,8
SURF	35,3	71,5
FAST	26,4	81,7
MSER	12,7	74,9
HOG	47,2	72,3

Given that the frequency distribution of the groups in the sample of 5000 images was the following ratio:

Table №2. The ratio of parameters of the frequency distribution of groups in a sample of 5000 images

Parameter	Image group (accuracy %)	
	a	**b**
Quantity	2500	600
Frequency	0,5	0,12

The final accuracy of the algorithms is calculated according to probability theory, and we get the result shown in Table 3

Table №3. The result of the final accuracy of algorithms calculated by probability theory,

Segmentation algorithm	Total accuracy %
SIFT	50,9
SURF	59,4
FAST	56,4
MSER	46,2
HOG	64,2

As can be seen from Table 3, the result of the existing algorithms is unsatisfactory in real conditions. Due to this result, it was decided to develop a segmentation algorithm based on a different approach.

Table №4. Performance of convolutional network optimization algorithms

Name of algorithm	**Time of training**		**Accuracy %**
	h	**min**	
Nesterov accelerated gradient	10	46	78,13
AdaGrad		27	77,97
Adam		24	85,31

CONCLUSION

As can be seen from Table 4, our trained convolutional network provides with the Adam optimization algorithm the best results relative to the others and high enough segmentation efficiency. The training time was 10 h 24 min, and the accuracy of classification was 85,31%. Almost all wagon numbers on the images were accurately singled out, however, there are errors, mainly since in some areas of the image numbers have a weak contrast to the rest of the background and are poorly distinguishable. Thus, in the future, it is planned to conduct experiments with algorithms to improve image quality, contrast, and application of various filters.

REFERENCES

[1] Botev A., Lever G., Barber D. Nesterov's Accelerated Gradient and Momentum as approximations to Regularised Update Des_ cent // Machine Learning. – 2016. – V. 1. – P. 1–7.

[2] Caffe deep learning framework. URL: http://caffe.berkeleyvision.org.

[3] Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition / K. He, X. Zhang, S. Ren, J. Sun // Transactions on Pattern Analysis and Machine Intelligence (TPAMI). –2015. – V. 4. – P. 534–542.

[4] Olgac, A & Karlik, Bekir. (2011). Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks. International Journal of Artificial Intelligence And Expert Systems. 1. 111-122.

[5] Cococcioni, Marco & Rossi, Federico & Ruffaldi, Emanuele & Saponara, Sergio.. A Fast Approximation of the Hyperbolic Tangent when Using Posit Numbers and its Application to Deep Neural Networks. doi 10.1007/978-3-030-37277-4_25. Lecture Notes in Electrical Engineering, vol 627. Springer, Cham (2020)

[6] Zhang, Z.; Zhang, H.; Peng, K.; Xue, H.; Sun, D. Exfuse: Feature synthesis enhancement for semantic segmentation. In Proceedings of the European Conference on Computer Vision, Munich, Germany, September 8-14, 2018; pp. 269-284.

[7] Efficient BackProp / Y. LeCun, L. Bottou, G.B. Orr, K.R. Muller //Neural Networks: Tricks of the trade. – Berlin: Springer, 1998. –44 p.

[8] Krizhevsky A., Sutskever I., Hinton G. ImageNet Classification with Deep Convolutional Neural Networks // Conference on Neural Information Processing Systems (NIPS). – Nevada, USA, 2012. – P. 27–35.

[9] Howard A.G. Some Improvements on Deep Convolutional Neural Network Based Image Classification // International Conference on Learning Representations (ICLR). – Banff, Canada, 2014. – V. 10. – № 4. – P. 652–659.

[10] Kecman V., Melki G. Fast online algorithms for Support Vector Machines // IEEE South East Conference (SoutheastCon 2016). –Virginia, USA, 2016. – P. 26–31.

[11] Lin, G.; Milan, A.; Shen, K.; Reed, ID RefineNet: Multipath Refinement Networks for High-Resolution Semantic Segmentation. Cpr 2017, 1, 5.

[12] Chen, L.K.; Zhu, Yu; Papandreou, G.; Shroff, F.; Adam, G. DeepLabv3+ Atron Separable Convolution Encoder-Decoder for Semantic Image Segmentation. Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, September 8-14, 2018; With. 801-818.

[13] Jan, M.; Yu, K.; Zhang, K.; Lee, Z.; Yang, K. DenseASPP for Semantic Segmentation in Street Scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, June 18-22, 2018; pp. 3684-3692.

[14] Zhou, Z.; Siddiqui, M.M.R.; Tajbakhsh, N.; Liang, J. Unet++: Nested u-net architecture for medical image segmentation. In deep learning of medical image analysis and multimodal learning for clinical decision support; Springer: New York, NY, USA, 2018; pp. 3-11.

[15] Wager S., Wang S., Liang P. S. Dropout training as adaptive regularization //Advances in neural information processing systems. – 2013. – T. 26.

[16] Haji S. H., Abdulazeez A. M. Comparison of optimization techniques based on gradient descent algorithm: A review //PalArch's Journal of Archaeology of Egypt/Egyptology. – 2021. – T. 18. – №. 4. – C. 2715-2743.

[17] Mukhamadieva K.B. Image processing method for recognizing the desired objects/ Conference: Innovative ways to solve current problems, 2020

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-CNN: Towards real-time object detection with region proposal networks," in Neural Information Processing Systems (NIPS), 2015.

[20] Jonathan, L.; Shelhamer, E.; Durrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, Massachusetts, USA, June 7-12, 2015; With. 3431-3440.

[21] Mohd Jais I., Ritahani A., Qamrun S. Adam Optimization Algorithm for Wide and Deep Neural // Network Knowledge Engineering and Data Science (KEDS) pISSN 2597-4602 Vol 2, No 1, Juni 2019, pp. 41–46

[22] Rasmus B., Rohl´en A. A comparison of training algorithms when training a Convolutional Neural Network for classifying road signs // 142X EECS/KTH 2019

[23] Ivashechkin A.P., Vasilenko A.Yu., Goncharov B.D. Methods of finding the singular points on image and their descriptors. Young cientist, 2016, no. 15, pp. 138–140. In Rus.

[24] Park S., Yoo J.H. Realtime face recognition with SIFT based local feature points for mobile devices. The 1st International Conference on Artificial Intelligence, Modelling and Simulation (AIMS 13). Malaysia, 2013. pp. 304–308.

[25] Tawfiq A., Ahmed J. Object detection and recognition by using enhanced Speeded Up Robust Feature. International Journal of Computer Science and Network Security, 2016, vol. 16, no. 4, pp. 66–71.

[26] Tore V., Chawan P.M. FAST Clustering Based Feature Subset Selection Algorithm for High Dimensional Data. International Journal of Computer Science and Mobile Computing, 2016, vol. 5, no. 7, pp. 234–238.

[27] Mammeri A., Boukerche A., Khiari E. MSER based text detection and communication algorithm for autonomous vehicles. IEEE Symposium of Computers and Communication. Messina, Italy, 2016. pp. 456–460.

[28] Dalal N., Triggs B. Histograms of Oriented Gradients for Human Detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). San Diego, USA, 2005. Vol. 1, pp. 886–893.

[29] Badrinarayanan, V.; Kendall, A.; Sipolla, R. SegNet: Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Trans. Model analog. Car. Intelligence. 2017, 39, 2481–2495. [CrossRef] [PubMed]