Convolutional Neuro Network Architecures Analyzis for Digits Recognition

Anna E. Kiryushyna, Igor P. Tishchenko
The Program Systems Institute of Russian Academy of Science
Pereslavl-Zalessky, Russia
{annkiryushina, igor.p.tishchenko}@gmail.com

***Abstract*—This paper is devoted to digits recognition with some types of convolutional neural networks. Photos taken from a UAV are processed, centered and normalized, to be given on the neural network input. Fully connected net, LeNet-5, Deep learning net were tested on different data sets for result comparison.**

I. Introduction

Photos taken from a UAV (Unmanned Air Vehicle) [1] contain different types of information. One of them is text information such as signs, documents, tabloids and etc. Recognized text information can be identified like a signal for a moving UAV or for a customer. After a photo has been taken, text boxes should be detected. Then isolate symbols (letters, digits, punctuators) should be recognized.

Detection and clusterization problems are mainly solved with any metric method. Chosen method depends of the type of a problem. For example, if an object can be describe with a set of features which quantity is huge and everyone of them is essential, a curse of dimensionality appears. So in this particular case both classification and its explanation are needed. The metric which can be learnt should be applied. It is a convolutional neural network [2].

In the article [3] a method of digits recognition is demonstrated. The authors created a data base consisting of house numbers. They divided the base into train and test sets and took them for HOG, Binary Features, K-Means and Stacked Sparse Auto-Encoders learning methods. The best accuracy, 90,6%, of test is on K-Means method.

A method of digits recognition using Convolutional Neural Network is described in the article [4]. This method includes fields extraction, segmentation and recognition. The authors uses different CNN architectures and recognition tools to compare results. After testing, Bosted LeNet-4 performed best, achieving a score of 0,7%, closely followed LeNet-5 at 0,8%.

The article [5] shows a full end-to-end recognition method in natural images. The method combines highly-accurate text detector and character recognizer modules. The authors use multi-layer CNN, two convolutional layers and two average-pooling layers. As data sets ICDAR 2003 and SVT are used. The classifier accuracies are 83,9% and 70% for ICDAR 2003 and SVT respectively.

II. Convolutional neural network

As it was mentioned before, a convolutional neural network has been chosen as a method of classification. Later on some neural network architectures and problems they solve will be described.

CNNs allow to reduce the dimension of input data by rotating of convolutional layers, sub-sample layers and fully connected output layers. The CNN's main feature is local processing of an image (the input is a part of it) which lets keep connection of images from different layers. CNN offers little or no invariance to shifting, scaling, and other forms of distortion. The topology of the input data is completely ignored, yielding similar training results for all permutations of the input vector.

CNN uses shared weights that lets reduce a quantity of them. CNN generalizes image's features that helps to find invariants in pictures with noise and low quality.

III. Data bases and sets

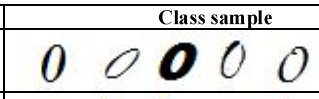
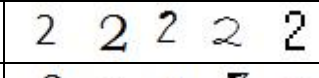
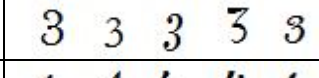
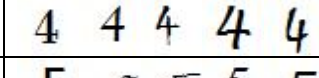
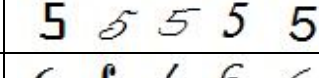
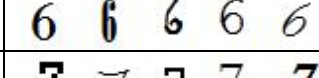
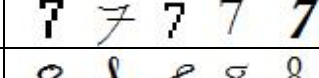
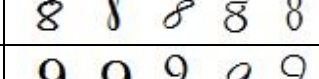
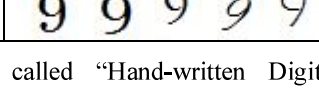
Further in the paper some scanned digits data bases and sets used in metric research were demonstrated.

The first one is MNIST data base [6] made of images with handwritten digits. It's part for classification consists of 60 000 objects of 28 × 28 pixel size and 10 000 images of the same size for recognition. All of the images are normalized and centered. 500 different persons took part in creation of the base. Fig. 3 shows MNIST examples of images.

Fig. 3. MNIST examples

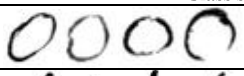
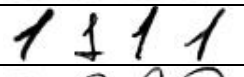
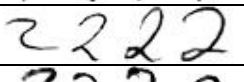
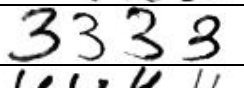
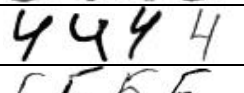
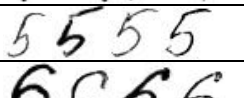
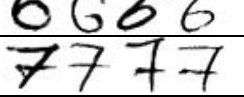
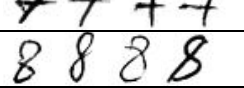
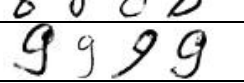
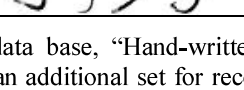
The next data base, “Digits”, created by the auther consists of images with digits examples taken from MS Word fonts. The images of 3×32 pixels do a set of 203 objects (TABLE I).

TABLE I. “DIGITS” DATA BASE EXAMPLES

Class name	Class sample
“0”	
“1”	
“2”	
“3”	
“4”	
“5”	
“6”	
“7”	
“8”	
“9”	

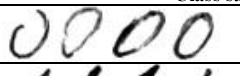
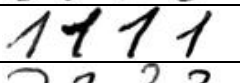
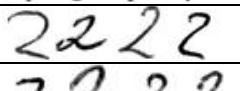
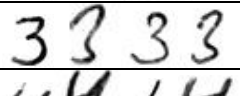
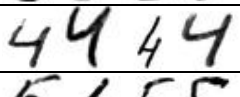
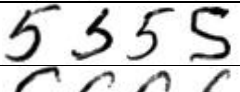
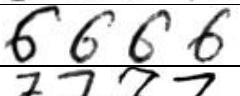
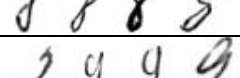
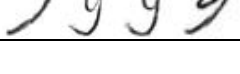
The third set called “Hand-written Digits” (“HD”) (TABLE II) was made of 6236 grayscale images with hand-written digits. They were isolated from digital documents written by 22 humans. The images have the same size as “Digits” images keeping their scale.

TABLE III. “HD” DATA BASE EXAMPLES

Class name	Class sample
“0”	
“1”	
“2”	
“3”	
“4”	
“5”	
“6”	
“7”	
“8”	
“9”	

The fourth data base, “Hand-written Digits 2” (“HD2”) (TABLE IV) is an additional set for recognition. It consists of 1498 images made by six humans who are different from the humans made “HD” base. “HD2” objects have the same size and scale characteristics.

TABLE IV. “HD2” DATA BASE EXAMPLES

Class name	Class sample
“0”	
“1”	
“2”	
“3”	
“4”	
“5”	
“6”	
“7”	
“8”	
“9”	

IV. IMAGE PREPROCESSING

When an image (Fig. 1) is taken from a UAV, it goes to preprocessing to prepare a set of images with isolate objects.

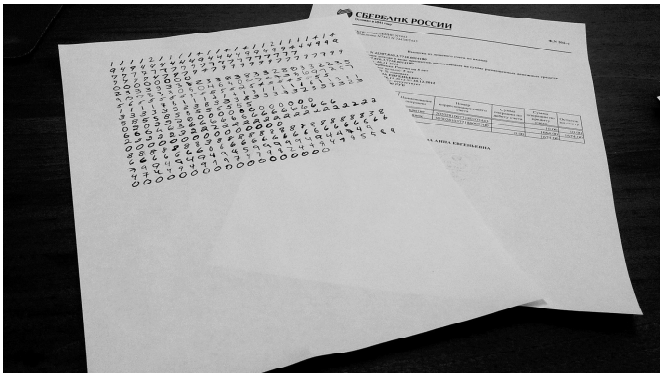


Fig.1. The image taken from the UAV

There is an Algorithm 1 making image processing:

Algorithm 1

1: To take an image from a UAV.
2: To find text regions, break the output image into separates images.
3: Geometric normalization of the images received on Step 2.
4: To find connected areas that would be symbols (digits, letters, punctuators, mathematic signs).
5: To save isolated symbols as images.
6: To scale images to a size of 32×32 or 28×28 pixels that is suitable for a CNN input. Moreover, all of the pictures keep their proportions.
7: To train the CNN with a set made up beforehand or found on Internet.
8: To test isolated symbols on the trained CNN.

This algorithm is shown in Fig.2.

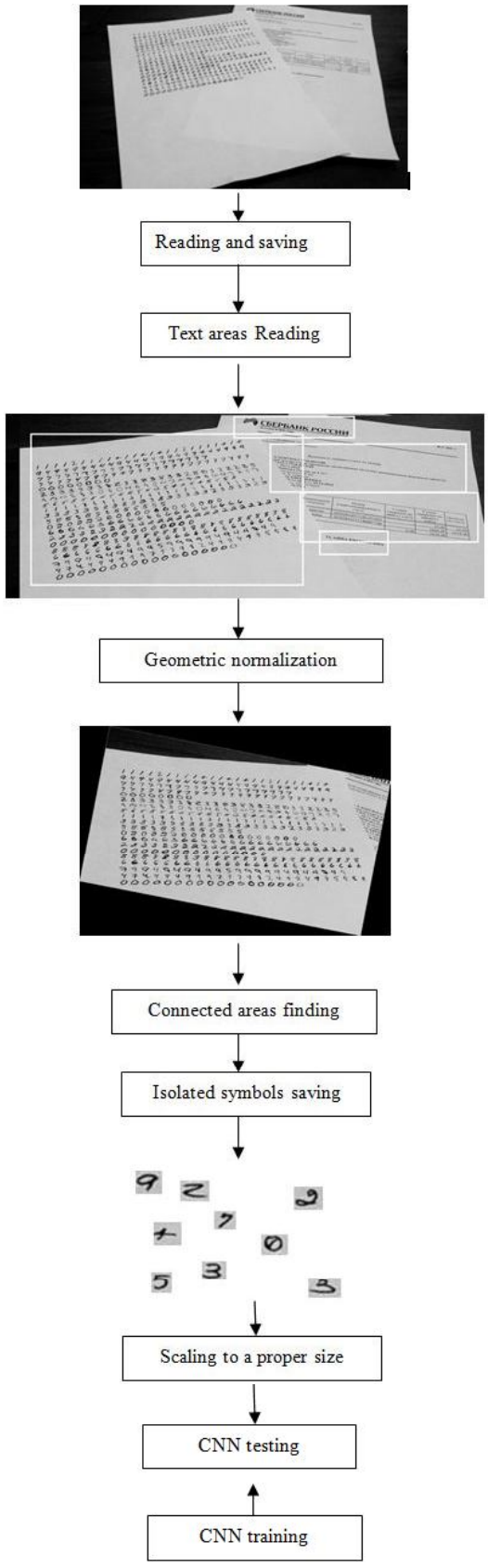


Fig. 2. Image preprocessing algorithm

We'll focus on the problem of classifying individual digits. We do it because the segmentation problem is rather easy to solve. There are many approaches to fix this problem.

V. CNN TRAINING AND TESTING SCHEME

Any neural network, in particular CNN, is trained following the algorithm in Fig. 4.

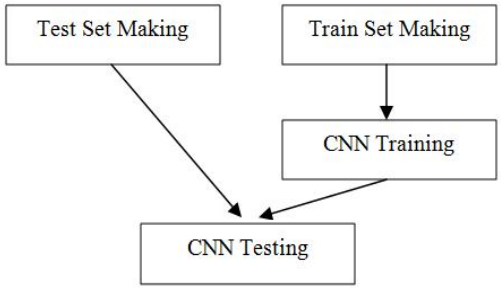


Fig. 4. A general scheme of training and testing processes

Test Set Making and *Train Set Making* make up data sets of images for testing and training respectively. PNG images are used as the data format. Images are required to be preprocessed according to the algorithm in Fig. 2. They should be gray scale and has a proper size of 32 × 32. Images for training should be sorted out by classes.

By increasing the number of training examples, the network can learn more about handwriting, and so improve its accuracy.

CNN Training uses a train set for learning, the training inputs are matched with corresponding desired outputs. The process of training may be interrupted and the weights of learning will be written down to a file. Then training may be continued and the weights received on the lowest error rate in the file will be used. In other words, the neural network uses the examples to automatically infer rules for recognizing handwritten digits

CNN Testing is the next step after training. It uses a test set and weights after training written down in a file. The test set will be used that must be differ from the train set. Results of an image matches a class will be put down to a file.

VI. TYPES OF CONVOLUTIONAL NEURAL NETWORKS

The architecture of neural networks consist of the input layer, and the neurons within the layer are called input neurons, the output layer contains the output neurons, the middle layer is called a hidden layer, since the neurons in this layer are neither inputs nor outputs.

A. LeNet-5

LeNet-5 [7] was created by Yann LeCun for hand-written digits recognition. LeNet-5 is resistant to noise, rotation that makes it suitable for object recognition. However LeNet-5 doesn't promise good recognition accuracy in comparison with other tools of recognition. This type of CNN goal is to minimize time spending for a process of recognition.

LeNet-5's architecture consists of an input layer, four hidden layers and an output layer (Fig. 5).

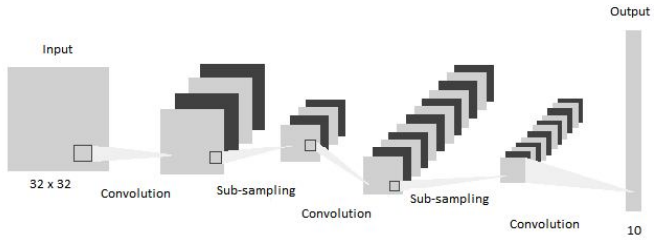


Fig. 5. LeNet-5's architecture

The input layer N_0 is an image of 28×28 or 32×32 pixels. It contains neurons encoding the values of the input pixels with a value of 0 representing white, a value of 1 representing black, and in between values representing gradually darkening shades of grey.

Second layer N_1 and fourth one N_3 are convolutional layers reducing data dimension. Third layer N_2 and fifth layer N_4 are sub-sampling layers.

The output layer N_5 is sixth layer which quantity of neurons is equal to quantity of recognized 10 classes.

Leaky ReLU (rectified linear unit) (1) [8] uses as an activation function. It woldn't let a neuron value become "0", if previous value of the neuron were negative.

$$f(x)=\begin{cases}x, x>0\\ 0.01x, otherwize\end{cases}, \text{where x} - \text{neuron value} \qquad (1)$$

1) LeNet-5 Training and Performance

As a special form of the multilayer perceptron, convolutional neural networks are trained through backpropagation.

The network was tested with MNIST database, all normalized and centered in the input image. An error rate of about 0.95% was achieved after 20 iterations. A lager training set could improve the performance of LeNet-5.

B. Fully connected Net

FullyConnectedNet [9] is classical neural network with backpropogation which is used for solving of image processing problems. They are filtering, optimization problems, content addressable storage (building up an image). One of the examples is to fix a damaged image. Although FullyConnectedNet solves such difficult tasks, it uses little memory that takes much time. Sometimes it might come to a stable state which is seldom a right result.

FullyConnectedNet's architecture is used in this case for digit recognition so it has ten neuron output layer. The net consists of an input layer N_0, two hidden fully connected layers N_1 and N_2 and an output layer N_3 (Fig. 6).

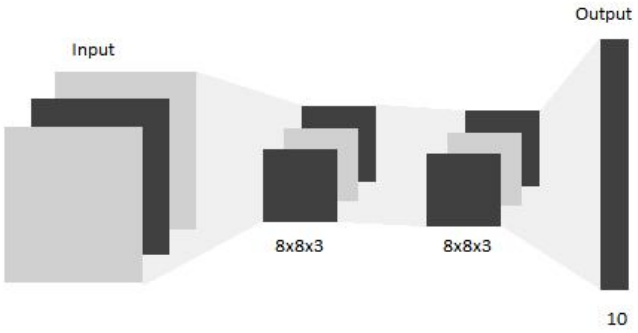


Fig. 6. FullyConnectedNet's architecture

Second layer N_1 and third layer N_2 can be described with some settings such as a quantity of input image channels, field size, field step and activation function Leaky ReLU (rectified linear unit) [8].

Fourth output layer N_3 looks like previous ones, the softmax function (2) [10] is used as an activation function that normalized neuron values setting them from "0" to "1".

$$y_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_i}}, \text{for neuron } i \qquad (2)$$

TABLE IV. NEURON QUANTITY OF EACH FULLYCONNECTEDNET LAYER

Layer number	Connection quantity for N_{i+1} layer, i is a number of a current layer	Neurons
N_0	$32\times32\times8\times8\times3=196608$	$32\times32\times3=3072$
N_1	$8\times8\times8\times8\times3=12288$	$8\times8\times3=192$
N_2	$8\times8\times3\times10=1920$	$8\times8\times3=192$
N_3	-	10

1) FullyConnectedNet Traing and Perfrmans

After been trained with using MNIST data set, FullyConnectedNet gave an error rate of 0.1406% after 1064 epochs (99,86% of right classified images). Then "Digits" data set was recognized with 87% right classified images.

Fully connected net has the only problem of increasing of neuron connections from one layer to another. It takes far much time for training.

C. Deep Learning

A neural network with deep learning is usually used for text information analyzing and speech recognition. Deep learning has two main ideas [11]:

1) Learning with many levels of features to create complex data connections.

2) Learning based on unmarked data (supervised learning) or partly marked data.

Receptive fields of 3 × 3, 5 × 5, 11 × 11 pixels are suitable for Deep learning CNN with architecture in Fig. 7.

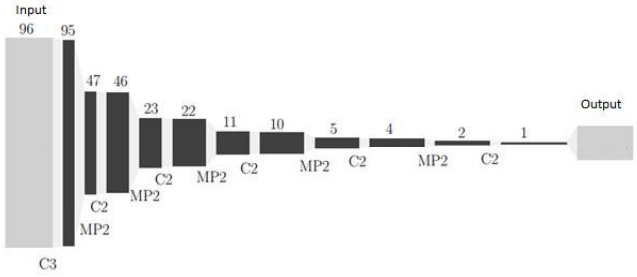


Fig. 7. Deep learning CNN's architecture

The main idea of this type of CNN is max-pooling [12], a pair of convolution and max-pooling layers. A convolutional layer turn to max-pooling one by using a receptive field of 2 × 2 pixels.

1) Deep learning CNN Traing and Perfrmans

Deep learning CNN was training on MNIST data set. The CNN gave an error rate of 0.6% after 955 iterations (99,4 % of right classified images).

Then an extension data base was create by adding "HD" to MNIST. After been trained with using "HD+MNIST", we got an error rate of 0.15% on 998 epoch (99,85% of right classified images).

"HD2" data set was specially prepared for recognition by Deep learning CNN trained with "HD+MNIST".

D. CNN

The next type of CNN is a mixture of Fully connected net and deep learning. Let's call it CNN for short (Fig. 8).

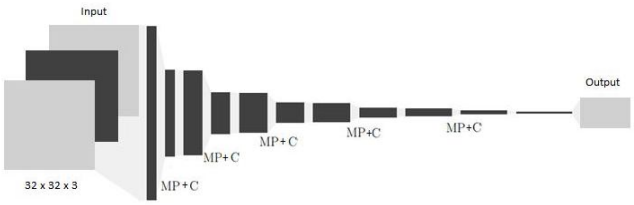


Fig. 8. CNN's architecture

It consists of an input layer, 11 hidden layers and an output layer. The input is a three channel image of 32×32 pixels.

Very Leaky ReLu (3) [13] is an activation function:

$$f(x) = \begin{cases} x, x > 0 \\ \frac{x}{a}, otherwize \end{cases}, \quad (3)$$

where x – neuron value, $a = 5.5$.

1) CNN Traing and Preformance

The CNN was trained with MNIST and gave an error rate of 0.16% after 672 iterations which means 99.84% of images were classified correctly. Then three experiments were run. On the first one a random text from Internet was chosen (Fig. 9).

Fig. 9 – Random image from Internet

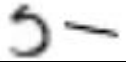
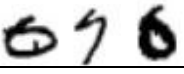
It was a grey scale image with hand written digits. The preprocessing was rather easy, the image was divided into isolate objects with one digit on it. There are 439 symbols on Fig.10 that makes "Text" data base. After "Text" had been recognized, the CNN showed recognition results (TABLE V). The CNN had been running for approximately 35 hours, before it spent a minute for recognition.

TABLE V. RESULTS OF "TEXT" AND "DIGITS" RECOGNITION

Error rate "Digits"	Error rate "Text"	Epoch's number	Error rate of training
86%	93,82%	672	0.16%

Some badly written digits weren't recognized (TABLE VI).

TABLE VI. RECOGNITION RESULTS

Symbol	Comment
	An input disconnected symbol
	An illegibly written symbol
	A turned symbol

Then MNIST was used as train set for the CNN, an error rate after 672 iteration went down to 0.16% (99.84% of right classified objects).

As a train set "HD" was taken, the CNN shown 75% of right recognized images.

Third experiment were performed on “HD+MNIST” data set and an error rate of 0.22% after 538 iteration was given (99.78% of correctly classified images). “HD2” was recognized with a score of 99.11%.

VII. Conclusion

For a sake of comparison, used types of CNNs trained on different sets are shown in Fig. 10.

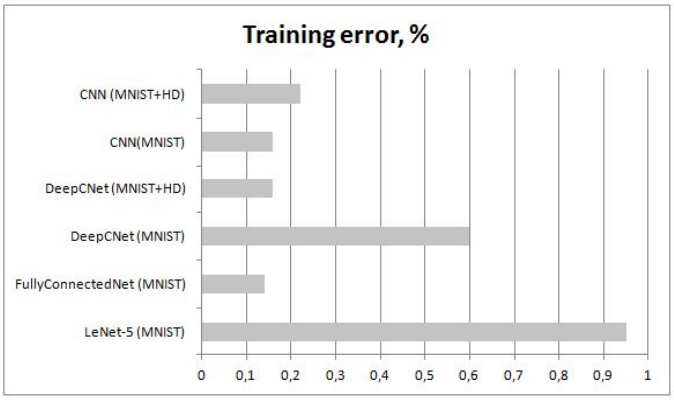


Fig. 10. Training errors of different CNN types

Fig. 10 gives a simple improvement that Fully connected net trained on MNIST and Deep learning net trained on “MNIST+HD” let receive the lowest error rate. Some images, such as disconnected, illegibly written or written like another digit (sometimes written “1” looks like “2”, “0” like “6”), were difficult to recognize even for humans. If a training data has much enough various images the lower error rate is received. DeepCNet trained on “MNIST+HD” gave lower error rate than it was trained on MNIST only.

A summary of the performance testing is shown in Fig. 11.

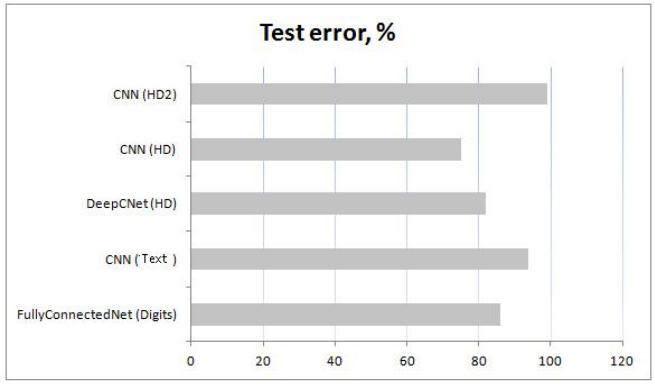


Fig. 11. Test errors

It’s clear that CNN gives the best recognition result on “HD2” set. It is worth mentioning that CNN and DeepCNet tested on “HD” have been trained only on MNIST.

After making recognition tests using different types of CNN, the CNN that is a mixture of fully connected net and deep learning, gave a high error rate on “HD” set. So two bases, “HD” and MNIST, were combined. Then the CNN was trained with a new data set. The lowest error rate was reached on “HD2”.

For better training and testing results some problems need to be solved:

- Analysis of disconnected objects.
- Analysis of multiple characters (consist of several digits)
- Extending of a data base with various ways of digits handwriting.

Future work is suggested to fix these problems to approve “HD” data set.

Acknowledgment

This work was supported by the Ministry of Education and Science of the Russian Federation, agreement № 14.607.21.0012 for a grant on “Conducting applied research for the development of intelligent technology and software systems, navigation and control of mobile technical equipment using machine vision techniques and high-performance distributed computing”. Unique identifier: RFMEFI60714X0012.

The types of CNNs presented in the paper are modules used in Distributed Data Processing System [14].

References

[1] UAV explanation, Web: http://www.theuav.com/.

[2] Convolutional Neural Network, Web: http://andrew.gibiansky.com/blog/machine-learning/convolutional-neural-networks/.

[3] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, Andrew Y. Ng, “Reading Digits in Natural Images with Unsupervised Feature Learning”, *NIPS Workshop on Deep Learning and Unsupervised Feature Learning,* 2011, pp. 1-9.

[4] Yann LeCun, Leon Bottou, Yoshua Bengio, Patrick Haffner, “Gradient-Based Learning Applied to Document Recognition”, *in Proc. IEEE,* November 1998, pp. 2278-2324.

[5] Tao Wang, David J. Wu, Adam Coates, Andrew Y. Ng, “End-to-End Text Recognition with Convolutional Neural Networks”, *in Proc. of the Twenty-First International Conference on Pattern Recognition (ICPR 2012).*

[6] MNIST Data Base, Web: http://yann.lecun.com/exdb/mnist/.

[7] LeNet-5, Web: http://yann.lecun.com/exdb/lenet/.

[8] Xavier Glorot, Antoine Bordes, Yoshua Bengio, “Deep Sparse Rectifier Neural Networks”, *in Proc. of the Fourteenth International Conference on Artificial Intelligence and Statistics*, Apr. 2011, pp. 315-323.

[9] Fully connected net, Web: http://geektimes.ru/post/74326/.

[10] Softmax function, Web: http://www.faqs.org/faqs/ai-faq/neural-nets/part2/section-12.html.

[11] Deep learning, Web: http://www.faqs.org/faqs/ai-faq/neural-nets/part2/section-12.html.

[12] Jawad Nagi, Frederick Ducatelle, Gianni A. Di Caro, Dan Ciresan, Ueli Meier, Alessandro Giusti, Farrukh Nagi, Jurgen Schmidhuber, Luca Maria Gambardella, “Max-Pooling Convolutional Neural Networks for Vision-based Hand Gesture RecognitionMax-Pooling Convolutional Neural Networks for Vision-based Hand Gesture Recognition”, *in Proc. IEEE International Conference on Signal and Image Processing Applications,* 2011, pp. 342-347.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”, *CoRR,* 2015, pp. 1-11.

[14] Aleksey Kondratyev, Igor Tishchenko, “Distributed Processing System of Images Flow”, *in Proc. RiTA2015*, in press.