ATLANTIS PRESS
Advances in Computer Science Research (ACSR), volume 72
IV International Research Conference "Information Technologies in Science, Management, Social Sphere and Medicine"
(ITSMSSM 2017)

Patents Images Retrieval and Convolutional Neural Network Training Dataset Quality Improvement

Alla G. Kravets, Nikita Lebedev, Maxim Legenchenko
CAD Department
Volgograd State Technical University,
Volgograd, Russia
agk@gde.ru, asdnicki@rambler.ru

***Abstract* — The paper considers the problem of the analysis of patents' figures for formalization of subjective opinions of the patent office experts that reviews applications for inventions. Drawings omission may indicate an incomplete description of the invention and entail the rejection of patent applications and other problems. Since patent images, even if one considers images of the same type, class, etc., are unique, different from each other. Nowadays for image processing are applied neural networks with different architectures. Neural network, Convolutional neural network, Siamese neural network were considered in the research. 4 libraries (Theano, TensorFlow, Caffe, and Keras) were studied. The main contributions of the paper are the new classification of patents' imaged, training dataset formation and quality improvement approach, and the software implementation for CNN training.**

Keywords —patent image; neural network; formation dataset; training dataset quality; deep learning, patents images retrieval, convolutional neural network.

I. INTRODUCTION

The evaluation of cross-topical relationships between domains for technologies development at the global level requires the formalization of subjective opinions of the patent office experts that reviews applications for inventions. When processing patent application examiner must evaluate three parameters [5]:

- novelty;
- industrial applicability;
- the inventive step.

Assessing novelty, the expert compares the patent application with the existing patents for differences from each other. In this case, the International Patent Classification (IPC) is used to identify analogs in the world. If the difference is small or application replicates existing patents, it will be rejected at the stage of "consideration on the merits".

Industrial applicability is a factors' set that allows to establish mass production of goods and/or services using the invention or upgrade technological processes in the near future. At the same time, it's estimated the possibility of the invention implementation at the current level of scientific and technological development.

In assessing inventive step, the expert concludes that as an obvious and intuitive was alleged invention from the standpoint of technology and design complexity.

The automation of the patent examiner activities requires formalizing each of these three areas of work. Despite the fact that the expert estimation procedure has specialized guidance (prescribing criteria for the assessment [5]), this activity takes a long time [12], requires considerable intellectual work and, at the same time, its results have a greater share of subjectivity. Automation of this activity will significantly reduce the time [9, 15] and economic costs, and greatly reduce the influence of the human factor in the processing of patent applications [14, 16]. However, the patent examiner activity automation is required to implement a three-step approach [11, 21] to be able to formalize each of these parameters. Existing software in this area is based primarily on the analysis of the patent or patent application texts [10, 13]. Therefore, the aim of this article was to create more sophisticated methods using a different approach to this problem – the analysis of patents' figures.

Most of the patents contain pictures. They are necessary for full disclosure of the essence of the invention. Drawings omission may indicate an incomplete description of the invention and entail the rejection of patent applications and other problems [5, 28]. Patent documents present the association between pictures and texts. Drawings, as in most of the technical documents, numbered consecutively and referred to as "FIG. X "in the text. However, the reference number in the drawing (Figure 1) - part of the picture, so this designation is not always easy to extract automatically from the different fonts used [20].

Patents in chemistry, for example, are of key importance for the pharmaceutical and agrochemical industries. They contain information about the chemical structures (Table. 1) which may be represented in various ways in the patent, including their different coding methods in the test as well as in Figs. They are composed of a limited character set which reduces their volatility. Technical drawings or other objects in the image are easier to interpret than the chemical structures because they have more flexibility/variability [20].

A set of black and white drawings that are used in the patents poorly represent the complexity of many inventions.

Fig. 1. Different styles of figure captions in patents [20]

It is also distributed at the present time the system software patent license Search provides users with search capabilities based on a standard (attributive) forming request or the so-called search "substring" [18]. A majority of methods to search for the drawings are based on a comparison of images. Author of the article [1], argues that it would make sense for the patent offices accept electronic 3D models and other electronic supplementary material to make more clear information about the invention and, therefore, make the search process more efficient.

Classification of patent images is a difficult task (Figure 2). Images of patents can be classified with high accuracy only basic classes of patent images (Table. 1). This classification can be made using a neural network based on the text or on the titles of graphics. However, such a classification will not give any information to the Patent Office. It can be used for primary image processing, and then apply the search.

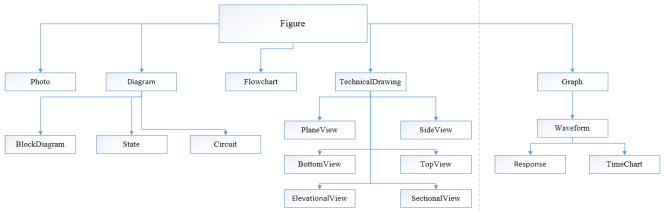
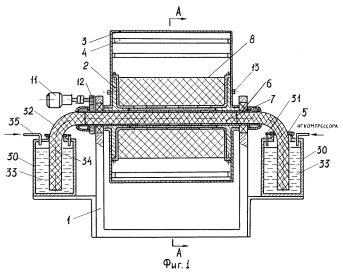
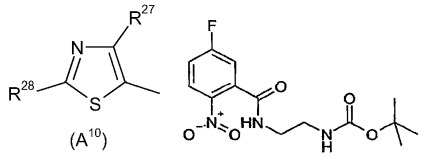
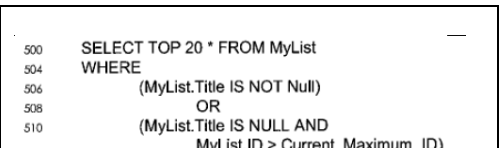
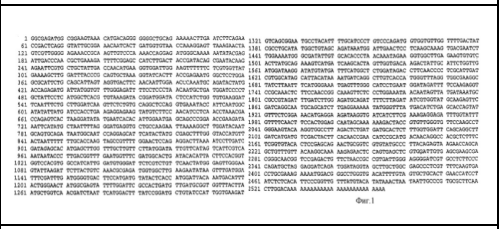
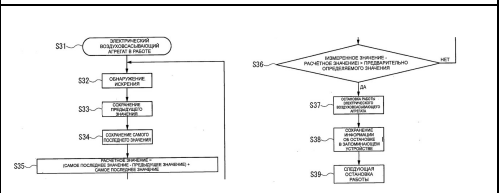
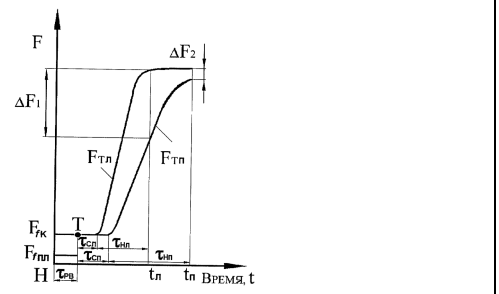
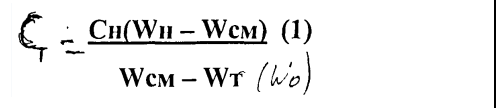
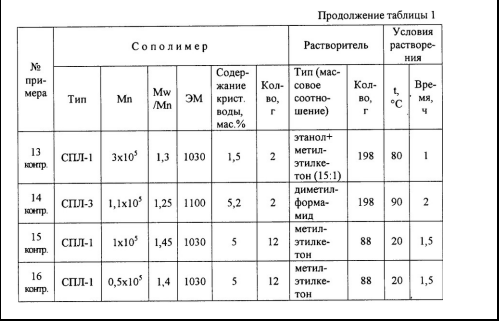


Fig. 2. Patent Classification of image [20]

TABLE I. CLASSES DOWNLOADED IMAGES

1	**Technical drawing**	
2	**Chemical structure**	

3	**Program code**	
4	**Sequence of gene**	
5	**Flowchart and UML diagram**	
6	**Diagram**	
7	**Mathematical Formula**	
8	**Table**	

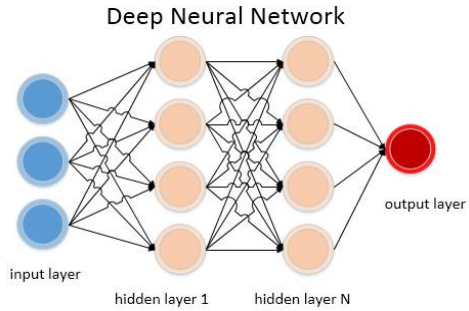
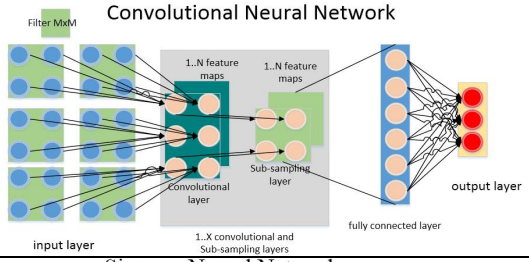
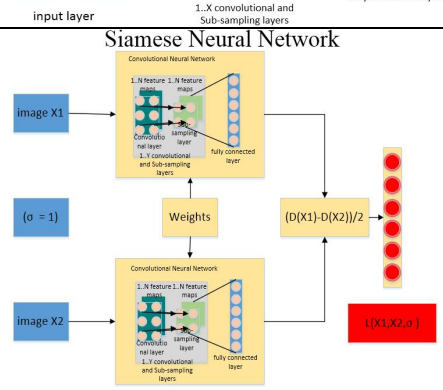
II. Architecture of Neural Networks for Image Processing

Nowadays for image processing are applied neural networks with different architectures (Table 2).

A. Neural Network (NN)/ Deep Neural Network

For working with images it is possible to use an ordinary neural network. It has been successfully applied to recognize

TABLE II. DIFFERENT NEURAL NETWORKS' ARCHITECTURES FOR DEEP LEARNING

Architecture	Description	Advantages / Disadvantages
	-usually, is used to classify images -consists of the many hidden layers (more than 2)	Advantages: - successful usage in many areas Disadvantages: - the training process can be very slow
	- appropriate for 2D data, such as images - based on the neurobiological model of the visual zone of the cerebral cortex	Advantages: - requires fewer neuron connections relative to a typical neural network -a lot of different variants of architectures (GoogLeNet [25] AlexNet [17]) Disadvantages: - each convolution layer requires its own, non-trivial convolution kernel to find the visual features of the image -usually, requires a large set of labeled image data
	-usually used to find similarities or relationships between two comparable things -contains two identical subnets, most of the convolution neural networks	Advantages: -fulfill task similarity and relationship between images Disadvantages: - are the same as the convolution neural network - libraries do not provide a standard feature of creating this architecture

handwritten numbers images with dimension 28x28 pixels [24]. In [24] network had two layers, an input layer with 800 neurons and an output layer with 10 neurons. The input of each neuron receives values from all 784 (28 * 28) pixels in the image. At the output layer 10 neurons, each neuron for the digit. The problem with this architecture is that it needs a large quantity of training data and image magnification in the dimension of each input layer neuron receives an even greater number of values (a large number of weights for training). Also, the image appears as a flat array - the loss of information about the topology.

B. Convolutional Neural Network (CNN)/ Deep Convolutional Neural Network

Name of the network is derived from the convolution operation, which is an easy way to perform a complicated operation using a convolution kernel. CNN does not use predefined convolution kernels, as this task is not trivial, but instead, they are determined as a result of training [19]. For example, if the image is 200 × 200, CNN will not process immediately 40 thousands of pixels. Instead, the network considers the square of $n \times n$ size (usually from the upper-left corner), then shifted by one pixel, and finds a new square, etc. These input data are then passed through convolutional layers in which not all the nodes are interconnected. These layers have a tendency to shrink with depth, with frequently used powers of two: 32, 16, 8, 4, 2, 1. In practice, the fully connected neural layer has attached the end of CNN for further processing [19, 24].

The principles of convolutional neural networks:

- local perceptions;
- shared weight;
- reduction of dimension.

C. Siamese Neural Network (SNN)

SNN is the class of a neural network architecture which contains two or more identical subnet. Identical here means that they have the same configuration with the same parameters and weights. The updated parameter is reflected through both the subnets. Siamese neural networks are appropriate for popular tasks that involve similarity or relationship between the two comparable things. [8] Some examples:

- a comparison of narration, where the original data - two proposals, and the result is - by how similar they are;
- the signature verification on the image, whether the two signatures belong to one person.

Typically in such problems, two identical subnets used to process two original images, and the other module will produce a result and images' final assessment [8].

III. An Overview of the Libraries for Deep Learning

A. Keras

Keras [3] is a high-level API for the development of neural networks, is written in Python and is able to run on top of TensorFlow, MS Computational Network Toolkit (CNTK) or Theano. The library has been designed with an emphasis on the possibility of rapid experiments that is the key to conducting good research.

Keras allows for easy and rapid prototyping (due to the convenience, modularity, and extensibility), it supports both CNN and recurrent neural networks (RNN), and combinations thereof. Also, the library operates with both the CPU and the GPU.

The library contains many implementations commonly used building blocks of a neural network, such as layers, objects, activate functions, optimizers and a plurality of tools to facilitate the work with images and text data.

B. Caffe

Caffe Library [2] - developers are doing the special emphasis - in contrast to its predecessors, is fully focused on commercial use. The entire code is open, is written in C ++, and the product fully supports writing custom algorithms on Python / NumPy and is compatible with MATLAB.

Caffe offers a tool for the creation and use of modern deep learning algorithms. Among other things, Caffe created a good stepping stone for the future - and at the moment it is successfully used to solve the problems of image and speech recognition, including in such important fields as astronomy and robotics.

C. Theano

Theano [27] is an extension of the Python language, which allows to effectively evaluate mathematical expressions containing multidimensional arrays. Theano developed at LISA laboratory to support the rapid development of machine learning algorithms.

The library is implemented in Python, it is supported on the operating systems Windows, Linux and Mac OS. Theano structure includes a compiler, which translates mathematical expressions written in Python to efficient code in C or CUDA.

Theano provides a basic set of tools for the configuration of neural networks and their training. Possible to implement multi-layer fully connected networks (Multi-Layer Perceptron), CNN, RNN, auto-encoders (AE) and restricted Boltzmann machines. Also, there are different activation functions, in particular, sigmoid, softmax-function, cross-entropy. The batch gradient descent is used during training.

Since this is a low-level library, the process of creating the model and its parameters definition requires voluminous and noisy code writing.

However, Theano's advantage is its flexibility, as well as the availability of feasibility and use of its own components. Also, advantages of the library are the tight integration with NumPy, transparent use of the GPU, effective variables differentiation, fast and stable optimization, dynamic code generation in C, enhanced unit testing and self-tests.

Theano is widely used in high-intensity computing researches which need more flexibility.

D. TensorFlow

TensorFlow [26] is a library of open source software for numerical modeling using a flow graph. Nodes in the graph represent mathematical operations, and the graph edges represent the multidimensional data array transmitted between nodes. The flexible architecture allows expanding computing both in CPU, GPU and on a personal computer, a server computer cluster and mobile device using a single API. On GPUs of graphics cards, calculations are possible in applications for general computing using the optional CUDA extension for GPGPU.

TensorFlow runs on 64-bit servers and desktop computers, Linux, Windows and Mac OS X, as well as on mobile platforms, including Android and iOS. TensorFlow was originally developed by researchers and engineers working in Google Brain Research Organization team in Google Machine Intelligence for machine learning and research of deep neural networks, however, this system can be applied in other areas. TensorFlow provides an API for the Python language, as well as an API for C ++, Haskell, Java, Go and Rust. Furthermore, there is a third party package R.

TensorFlow calculations are expressed as data flow graphs with persistence (stateful). Google's algorithms library instructs the neural network to receive the information and reason like a man so that new applications originally possess such "human" qualities. The task TensorFlow is the training of the neural network to detect and identify patterns and correlations in the data arrays.

E. A General Comparison of the Functional Capabilities of Libraries

We consider 4 libraries for deep learning in Python programming language: Theano, TensorFlow, Caffe, and Keras (Table 3).

For comparison purposes, the following criteria were identified: the name of the creator, the operating system, programming language, enabling the creation of fully-connected neural networks (FC NN), CNN, AE and RNN, OpenMP support technology and the ability of cloud computing.

According to analysis results, the Keras library was selected for software implementation of CNN model.

TABLE III. NN LIBRARIES COMPARATIVE EVALUATION

Name	Creator	OS	Language	FC NN	CNN	AE	RNN	OpenMP Support	Cloud computing
Theano	University of Montreal	Cross platform	Python	+	+	+	+	+	+
Caffe	University Berkeley	Linux, Windows, macOS, Android	C++, Python, Matlab	+	+	–	+	–	+
TensorFlow	Google	Linux, Windows, macOS	C++, Python	+	+	+	+	–	+
Keras	François Scholle	Linux, Windows, macOS	Python	+	+	+	+	–	+

IV. TRAINING DATASET FORMATION AND QUALITY IMPROVEMENT

A. Data Sources for Patents' Images Collection

Two websites for data collection have been considered: Freepatent [6] and Findpatent [4]. In Findpatent there are uncomfortable indexed pages (cross-references within the categories). There are some reasons to choose the Freepatent:

- all patents are separated into IPC categories with levels of hierarchy;
- figures for each patent are in a separate directory.

Also, this site includes patents registered in Russia since 1994. The total number of the main sections is corresponding with IPC [30].

Each section contains subsections, etc. For the training dataset we used the main sections:

A Human necessities
B Performing operations; Transporting
C Chemistry; Metallurgy
D Textiles
E Fixed constructions
F Mechanical engineering; Lighting, Heating, Weapons, Blasting
G Physics
H Electricity

B. Images Downloading

To collect images, a scrapper was written that recursively traverses all subcategories and downloads all the images in the patent to the created directory for this category of patents (Fig. 3). We used Python programming language, Grab library [22] to retrieve data from a Website (site scraping), and Awesome-slugify library [29] to convert the category name in Unicode to the valid string. We downloaded 328562 images in 8 main sections.

C. Dataset Pre-processing

Step 1. Subdirectories hierarchy forming for dataset labeling. For training the neural network on external data, Keras library requires the formation of sub-directories (Fig. 4, left) with the training and testing dataset filled .png or .jpg images. Written Python script moves images from the lower level of the hierarchy and pulls out all the files in directories at a given level for the formation of the labeled classes of images (Fig. 4, right).

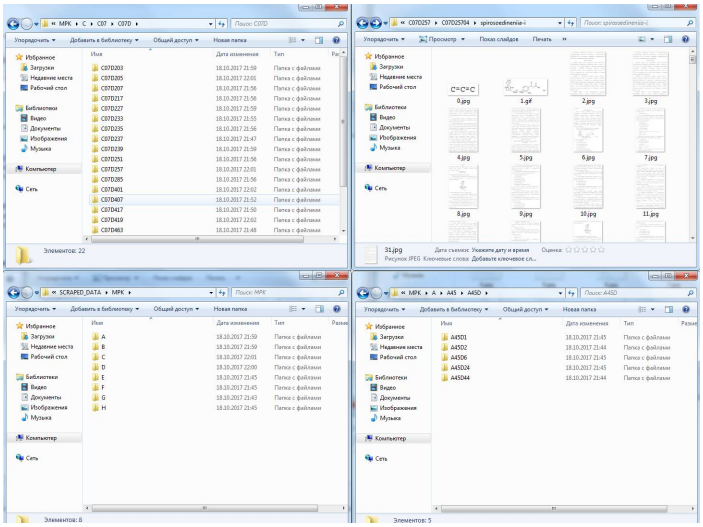


Fig. 3. Subdirectories with downloaded patents and images

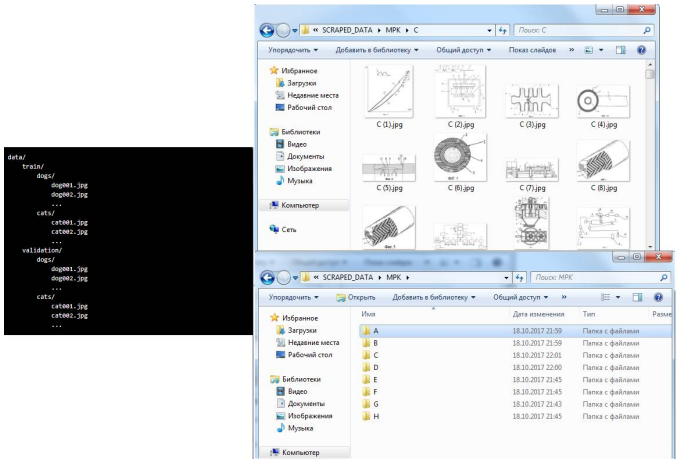


Fig. 4. Directories structure for training

Step 2. Definition of incorrect images. Since a large number of images consisted entirely of text (Fig. 5, left), contained blocks of text (Fig. 5, middle) or were "broken" (Fig. 5, right), they had to be removed so that they did not interfere with training.

Step 3. Dataset quality improvement. For the classification of images with text (Fig. 6, top) and without (Fig. 6, bottom), CNN was trained for the recognition of two classes. CNN program implementation and architecture is presented in Fig. 7.

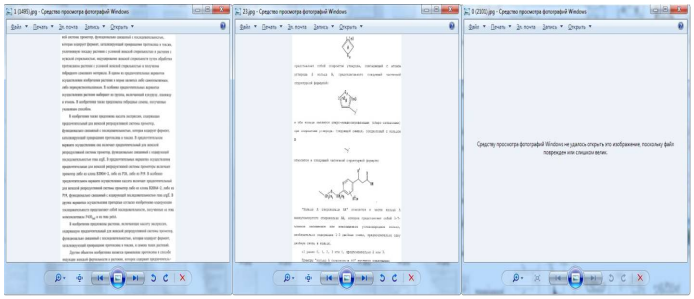


Fig. 5. Examples of images to remove

For training and testing datasets used 3000 and 800 patent images, respectively. Recognition accuracy on the test data was 98.75%. Also, in the class of images without the text, diagrams, tables, gene sequence (Fig. 6, bottom-right) and technical drawings have been added that the network did not recognize them as an image with text and excluded from deletion.

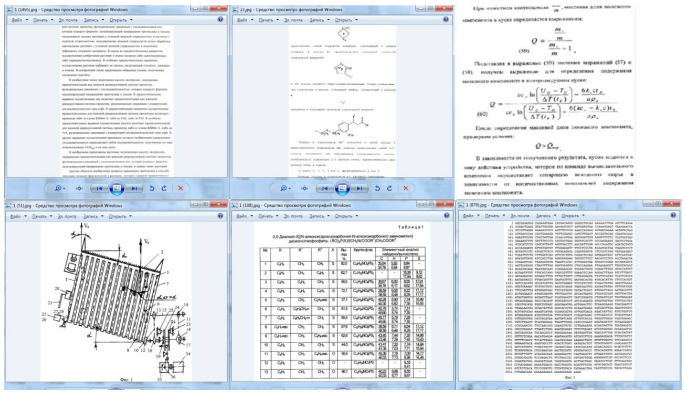


Fig. 6. Images for deletion (top) and exceptions (bottom) examples

Pre-processing results. The Python script was written to delete images with text and "broken" images. It used a trained CNN (Step 3). Specifically designed for image analysis Imghdr module [23] checked the validity of images. This module gives the image format when the file is read successfully. A number of downloaded and remaining after removal images, labeled for major categories, is presented in Table 4. As a result, the dataset of 45168 images was formed, 86,25% of images from Freepatent Website were defined as inappropriate for CNN training.

TABLE IV. LABELED DATASET

Category	Number of downloaded images	Number of remaining images after removal
A Human necessities	66421	8348
B Performing operations; Transporting	44506	8506
C Chemistry; Metallurgy	105174	8222
D Textiles	4976	1192
E Fixed constructions	14222	2872
F Mechanical engineering; Lighting, Heating, Weapons, Blasting	36501	7174
G Physics	45540	6742
H Electricity	11222	2112

V. SOFTWARE IMPLEMENTATION FOR CNN TRAINING

In the course of the study, a CNN was trained to classify patent images using the Python programming language, the Keras library for the neural network modeling and the Theano library for the computational backend.

The developed software receives at the input a training and test dataset of patents' images separated into labeled classes of images. The output is a file with the neural network in JSON format and trained weights in H5 format.

CNN architecture used for training (Fig. 7) consisting of three repeating layers convolution and sub-sampling for feature extraction image classifier and mesh of 64 neurons and an output layer of neurons 8. Activation function ReLU shows good results in training the neural network and is responsible for cutting off unnecessary parts in the channel (with a negative output).

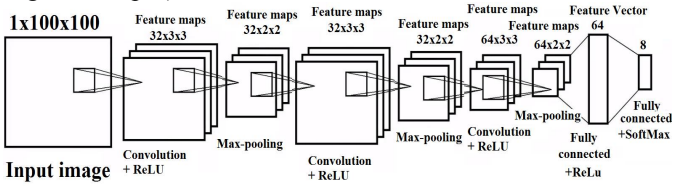


Fig. 7. CNN architecture developed

VI. RESULTS OF CNN TRAINING

For training the neural network used 2400 patent images and 800 images in the testing dataset. For optimization, the gradient descent method with the size of the mini-sample 32 is used, that is, the first 32 images are taken, the direction of the gradient is determined and, in accordance with this direction, we perform the change of weights, etc. It is also used to teach 40 epochs - the number of times the training with the help of a dataset. The last stage of training was 871 seconds, the error function was 2.464. The accuracy of the program during the training was 51%, and the accuracy of the test dataset was 30%. The accuracy of the test data is less than that of the training. And the accuracy on the test dataset was still changing. Hence, the retraining of the neural network did not happen and we can continue training by increasing the number of epochs.

The CNN accuracy estimation was made, depending on the size of the image and the number of epochs. Data for training/validation/testing did not change. Training took place on a personal computer (Intel Core i3-4160, 8 GB RAM). Retraining of the neural network, in all cases, did not happen.

TABLE V. DEPENDENCE OF THE ACCURACY OF THE NEURAL NETWORK ON THE SIZE OF IMAGES AND TIME OF TRAINING

Image size	Number of epochs	Time of study, s	Accuracy of training data,%	Accuracy of test data,%
100x100	8	744	26.46	20.15
100x100	16	1211	35	22.88
200x200	8	2137	28	21.25
200x200	16	4748	38	25
200x200	32	9848	50	28
400x400	16	14231	37	19
400x400	40	31925	51	30

VII. Discussion and Conclusions

Since the re-training of the neural network did not happen, there was not a maximum from the training dataset used. Potential, for the used learning dataset, is. Results of neural network accuracy estimation show that a classification based on the IPC [30] has perspectives for improvement rather than based on the main classes of patent images (Table 1). To improve accuracy, one can use all the remaining images after removal (Table 4) in the dataset, but for this, it is necessary to train CNN on multiprocessor GPUs. Also, to create a larger training dataset, it is possible to supply the formed dataset using other sources of patent images.

Since patent images, even if one considers images of the same type, class, etc., are unique, different from each other. So we need to define the similar images using other architectures of neural networks, such as the Siamese neural network.

We also discuss the possibility to increase the classification accuracy of the convolution neural network:

1. To train the CNN on a larger number of images;
2. To change layers parameters, the number of neurons on the classifier to find the optimal structure for patent images;
3. To use other CNN architecture (GoogLeNet, AlexNet);
4. To label the images in the dataset into a sufficient number of classes. Images should be as similar as possible to each other within the same class.
5. To use the pre-trained model before the training. In Keras have several trained models (Xception, VGG16, VGG19 etc.) [7].

The main contributions of the paper are the new classification of patents' imaged (Table 1), training dataset formation and quality improvement approach (part IV), and the software implementation for CNN training.

VIII. Acknowledgments

This research was partially supported by the Russian Foundation of Basic Research (grant No. 15-07-06254 A).

References

[1] Adams S., "Electronic non-text material in patent applications-some questions for patent offices, applicants and searchers," in World Patent Information, vol. 27, 2005, pp. 99-103.

[2] Caffe 2017. Berkeley Center, [Online]. Available: http://caffe.berkeleyvision.org/

[3] Chollet Francois, "Keras: The Python Deep Learning library" 2017. [Online]. Available: https://keras.io/

[4] Findpatent 2017. [Online]. Available: http://www.findpatent.ru/

[5] FIPS, "Library of documents" 2017. [Online]. Available: http://www1.fips.ru/wps/wcm/connect/content_en/en

[6] Freepatent 2017. [Online]. Available: http://www.freepatent.ru/

[7] Keras Documentation, "Keras: Available models" 2017. [Online]. Available: https://keras.io/applications/

[8] Koch G., Zemel R. and Salakhutdinov R., "Siamese Neural Networks for One-shot Image Recognition", CA:Department of Computer Science, 2015.

[9] Korobkin, D., Fomenkov, S., Kravets, A., Kolesnikov, S. Methods of statistical and semantic patent analysis (2017) Communications in Computer and Information Science, 754, pp. 48-61.

[10] Korobkin, D., Fomenkov, S., Kravets, A., Golovanchikov, A. Patent data analysis system for information extraction tasks (2016) 13 International Conference on Applied Computing 2016, pp. 215-219.

[11] Korobkin, D., Fomenkov, S., Kravets, A., Kolesnikov, S., Dykov, M. Three-steps methodology for patents prior-art retrieval and structured physical knowledge extracting (2015) Communications in Computer and Information Science, 535, pp. 124-136.

[12] Kravets, A.G., Kravets, A.D., Rogachev V.A., Medintseva I.P. Cross-thematic modeling of the world prior-art state: rejected patent applications analysis Journal of Fundamental and Applied Sciences. - 2016. - Vol. 8, No. 3S. - pp. 2442-2452.

[13] Kravets, A.G., Korobkin, D.M., Dykov, M.A. E-patent examiner: Two-steps approach for patents prior-art retrieval (2016) IISA 2015 - 6th International Conference on Information, Intelligence, Systems and Applications, art. no. 7388074 .

[14] Kravets, A., Kozunova, S. The risk management model of design department's PDM information system (2017) Communications in Computer and Information Science, 754, pp. 490-500.

[15] Kravets, A., Shumeiko, N., Lempert, B., Salnikova, N., Shcherbakova, N. "Smart Queue" Approach for new technical solutions discovery in patent applications (2017) Communications in Computer and Information Science, 754, pp. 37-47.

[16] Kravets, A.G., Mironenko, A.G., Nazarov, S.S., Kravets, A.D. Patent application text pre-processing for patent examination procedure (2015) Communications in Computer and Information Science, 535, pp. 105-114.

[17] Krizhevsky A., Sutskever I., and Hinton G. E., "Imagenet classification with deep convolutional neural networks", 2012, pp. 1097-1105.

[18] Kuchuganov A.V. and Mokrousov M.N., "Recognition of Images and Semantic Analysis of Text in Patent-License Search" Intellectual systems in the industry, no. 1, 2010, pp. 292-299 (in Russian).

[19] Lecun Y., Bottou L., Bengio Y. and Haffner P., "Gradient Based Learning Applied to Document Recognition" in Proc. Of the IEEE, 1998.

[20] Lupu Mihai and Hanbury Allan, "Patent Retrieval" Foundations and Trends in Information Retrieval, vol. 7, no. 1, 2013, pp. 1-97.

[21] Mironenko, A.G., Kravets, A.G. Automated methods of patent array analysis (2016) IISA 2016 - 7th International Conference on Information, Intelligence, Systems and Applications, art. no. 7785341, .

[22] Petukhov G., "Grab" 2017. [Online]. Available: http://grablib.org/en/latest/

[23] Python Documentation, "imghdr – Determine the type of an image" 2017. [Online]. Available: https://docs.python.org/2/library/imghdr.html

[24] Simard P.Y., Steinkraus D. and Platt J.C., "Best practices for convolutional neural networks applied to visual document analysis" IEEE Conference Publications, 2003, pp. 958-963.

[25] Szegedy C. et al., "Going deeper with convolutions" in Proc. Conf. Comput., 2015, pp. 1-9.

[26] Tensorflow 2017. Google, [Online]. Available: https://www.tensorflow.org/

[27] Theano 2017. Universite de Montreal, [Online]. Available: http://deeplearning.net/software/theano/

[28] USPTO, "Manual of patent examination procedure" 2017. [Online]. Available: https://www.uspto.gov/patents-getting-started/patent-basics/types-patent-applications/nonprovisional-utility-patent

[29] Voronin D., "Awesome-slugify" 2017. [Online]. Available: https://pypi.python.org/pypi/awesome-slugify

[30] WIPO, "International patent classification" 2017. [Online]. Available: http://www.wipo.int/classifications/ipc/en/