
ЗАСТОСУВАННЯ МЕТОДІВ ТА ЗАСОБІВ МОДЕЛЮВАННЯ

doi: <https://doi.org/10.15407/emodel.42.02.059>
UDC 004.932.2.021'416:004-052

D.O. Zubariev, Post-graduate

G.E. Pukhov Institute for Modelling in Energy Engineering
National Academy of Sciences of Ukraine
(15, General Naumov Str., 03164, Kiev, Ukraine,
tel. +380996810567; e-mail: dmytrozubariev@gmail.com),

I.S. Skarga-Bandurova, Doct. of Technical Sciences,

O.M. Sapytska, Cand. of Historical Sciences,
Volodymyr Dahl East Ukrainian National University
(59a, Tsentralny pr., 93400, Severodonetsk, Luhansk oblast, Ukraine,
tel. +380645228997; +380664838802;
e-mail: skarga-bandurova@snu.edu.ua; helensapitskaya@gmail.com)

Efficiency Analysis of the Image Optimization Algorithm for the Formation of a Database Database Not Confirmed by the User

The ultimate goal of any process optimization in a particular area is to save time and human resources. The article analyzes the effectiveness of the original algorithm image processing when sampling for training artificial neural network CNN Class based on User-Confirmed Image-Dataset for needs defining image elements within binary logic.

Key words: Deep learning, Image-Dataset, image optimization function, network learning algorithm.

Introduction. The brain of a living being has certain features in the field of mental activity, the most important for researchers of artificial neural networks are learning by insight, finding important information for the user in the general flow of data, as well as finding common features of certain phenomena or artifacts. Artificial neural networks, as well as biological ones, can change the course of their actions under the external influences. In the process of data processing, they can adjust and learn so that their feedback is correct. A trained neural network is more resistant to some distortions of the input data, which allows to correctly recognize images and patterns [1].

And what is a more important is that the network can recognize data that has some deviations, which allows it to identify the true image, regardless of

© Zubariev D.O., Skarga-Bandurova I.S., Skarga-Bandurova I.S., 2020

the accompanying distortion or interference. That is, by processing information, as a rule, selected by the user, networks learn, find distinctive features and similarities in the fragments of the received data. However, in most cases, these data must be prepared in advance by the user. And this is an essential aspect of the Deep Learning process, which requires improvement, taking into account the time and resources of the user.

Related works. In recent years, numerous scientific and practical developments and studies have appeared on artificial neural networks in general, the theory and practice of deep learning, the use of various programming languages, libraries, modules, interfaces, etc. Some of them are rapidly losing their relevance, but remain a prime example of the practical application of specific methods developed or the theoretical basis for further practical development and research.

Each year, the arrays of thematic work increases by several times, which indicates the extremely dynamic growth of this area and the vital role of artificial neural networks for the development of modern global information-digital civilization. Methods are being improved, new mathematical models and approaches are being developed for realizing the goal of maximum optimization of machine learning and deep learning, new paradigms of learning of artificial neural networks are derived for their maximum effective functioning. Given this, the issues of scientific definition of new methods of deep learning and practically experimental implementation of these developments remains a very urgent task.

Modern algorithms and methods of deep learning based on applied mathematics, as well as the current trends and applications in this area, are highlighted in [2]. Different options for using Keras as an open source deep learning library for creating efficient neural networks are provided in [3]. Also, certain practical implementations of the Keras methods are given in the book of the creator of the “Keras deep learning library” [4]. The theoretical approach to deep learning of neural networks is given in detail in [5], although it is worth noting that the described implementation of the fundamental machine learning methods is designed for beginners. In work [6], the variants of the application of deep learning algorithms for modeling and learning advanced neural networks for the implementation of various tasks of computer vision in the field of understanding and manipulating images were proposed. The book [7] provides the best, from our point of view, balance of the theory and practical implementation of Python for deep learning. One of the founders of deep learning [8] notes that deep networks study data in the same way children perceive the world, gradually acquiring the skills necessary to navigate in new environments: learning algorithms extract information from the original data; information can be used to create knowledge; knowledge is the basis of understanding; understanding leads to wisdom.

The contribution of this research are adding knowledge and experience about process initial targeted sampling for image optimization algorithm used in CNN without direct human involvement.

The overall goal is to obtain an efficient mechanism for sampling user-unconfirmed dataset; this means obtaining high-quality testing datasets without direct human involvement.

To achieve this goal, we create the image processing algorithm and test its efficiency in sampling for training CNN based on the unverified Image-Dataset when the `optimize_image` function completely replaces the user. Trained in this way, CNN could be used to image identification within the bit logic.

Research methodology and dataset generation. At this date, the most effective artificial neuron network for image recognition is the convolutional neural network or CNN. The CNN architecture uses a variation of multilayer perceptrons designed to require the use of a minimum amount of pre-processing.

A properly trained network, ideally at the output, should demonstrate a fairly clear understanding of a more abstract level of perception and definition of target elements. The training process itself consists of setting up internal parameters of CNN for a specific task. It is carried out on a training set. The training set includes the input values and the corresponding original dataset values. During the training, the neural network finds certain dependencies of the output fields on the input ones. The effectiveness of training largely depends on the correctness of the initial setup of the network itself.

The general methodology of the study is as follows:

1. Input required parameters
2. Uploading image for dataset from Google
3. Deleting crashed images
4. Optimizing images for dataset
5. Initialize CNN
6. Training CNN
7. Build charts of loss and accuracy
8. Save trained model

Final testing and analysis that goes beyond this methodology is presented in a separate section.

Experimental software and tools. Computer parameters: Intel Core i5-4210U CPU 1.70GHz × 4, 12gb RAM, Ubuntu 16.04 OS. For this study, we used the Python programming language and the following modules and tools: `os` – module for interacting with the operating system [9]; `cv2` – library for computer vision, image processing and many algorithms used for image processing and optimization [10]; `google_images_download` – a Google image

module that is used to retrieve images from the Internet and create target datasets [11]; subprocess – module responsible for executing Python software processes used at Popen and PIPE startup to identify corrupted images [12]; keras – a library for building, learning, and testing a neural network [13]; matplotlib – a library for rendering 3D graphics [14]; argparse – a library for processing command-line parameters or keys that is used to enter a keyword to create a dataset and train a neural network [15].

Dataset creation. An obligatory parameter for creating an Image-Dataset (I-D) is the keyword, by which the Image-Dataset is created and CNN is trained. I-D is formed by downloading targeted images with a keyword from the Internet using the `google_images_download` module.

Loading was performed in two stages. Within each of them a request was sent to download 100 images. At the first stage all images corresponded to keyword in jpg format were loaded, at the second stage all images corresponded to keyword in png format were loaded. After that, we corrected the obtained images by adding the exact file format, as some files had the wrong extension, which can cause errors in further image optimization. The next step is to write the path to the array with the images. Each image loaded into the array was tested for integrity using the `is_chashed_image` function. Later, each image was optimized using the `optimize_image` function below.

1. Read image
2. Convert to gray
3. Adding threshold using `THRESH_BINARY_INV`
4. Add morphological transformation
5. Remove noise
6. Find the max-area contour
7. Crop image
8. Rewrite image.

The result of applying the function `optimize_image` represented as line (Fig. 1) illustrating the optimization of a selected image. where, and E - final result. Transferred images are converted using the `cv2.imread` command to a NumPy array containing representations of data from the image. The resulting image array is configured with a grayscale for further processing. Next, we apply the fixed-level threshold value function for a single-channel array. In our case, this is image in grayscale, and the function for `THRESH_BINARY_INV` is as follows [16]:

$$\text{dst}(x, y) = \begin{cases} 0 & \text{if } \text{src}(x, y) > \text{thresh}, \\ \text{max val} & \text{otherwise.} \end{cases}$$

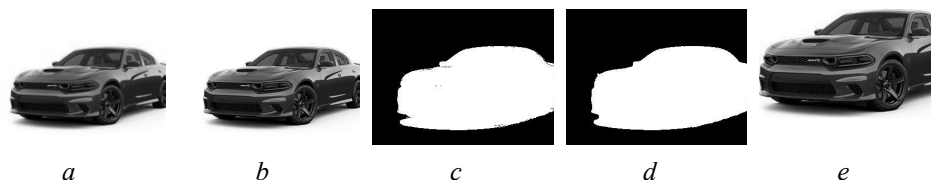


Fig. 1. Result of image transformation: *a* — is the original image; *b* — conversion to gray; *c* — adding a threshold; *d* — adding morphological transformation; *e* — final image, after resize

To the resulting element of a size of 11×11 the following form MORPH_ELLIPSE an elliptical structuring element, that is, a filled ellipse inscribed in the rectangle $\text{Rect}(0, 0, \text{esize.width}, \text{esize.height})$ is applied. Then a complex morphological transformation is performed with MORPH_CLOSE, for the closing operation.

Next, we look for contours in the binary image obtained from the morphological transformation, using only the restoration of the outer contour. Recovery sets the following hierarchy for all contours:

$$\text{hierarchy}[i][2] = \text{hierarchy}[i][3] = -1.$$

With the contour approximation method CV_CHAIN_APPROX_SIMPLE, horizontal, vertical and diagonal segments are compressed, only their endpoints remain. After completing all the listed sequential steps to form a training I-D, an operation is performed to search for contours on the received binary images using the CV_RETR_EXTERNAL constant, which, in turn, extracts only the outermost contours, and the CV_CHAIN_APPROX_NONE constant, which stores absolutely all contour points. That is, any two following points (x_1, y_1) and (x_2, y_2) contours will be neighbors horizontally, vertically or diagonally. The formula for the constant CV_CHAIN_APPROX_NONE is as follows:

$$\max(\text{abs}(x_1 - x_2), \text{abs}(y_2 - y_1)) = 1.$$

The final step in the optimize_image function is cropping images and re-writing current images into the same training I-D.

After executing the dataset generation algorithm described above (searching by a keyword, downloading an unconfirmed raw data, and deleting damaged files), we obtained an array of 191 images with a total size of 23.5 MB. After optimizing I-D while reducing the physical dimensions of each image, the total weight of the entire array increased to 25.3 MB due to the improved image quality. Image loading time for the training I-D generation was 155.43 seconds. Their optimization time was 7.3 seconds. The total formation time of the training array of images was 162.73 seconds.

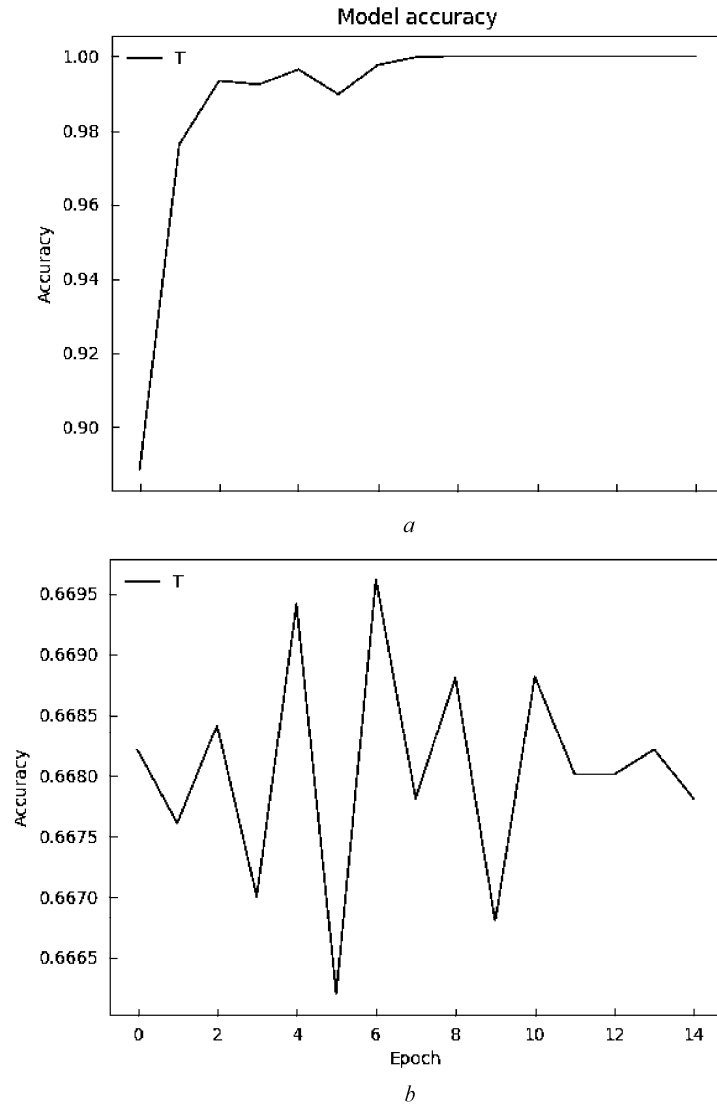


Fig. 2. The accuracy of recognition of this object for each learning era: *a* — CNN-sd, *b* — CNN-sx

The final testing and analysis. For the experiment, we used one I-D and two CNN with the same architecture [1] but with different activation functions, namely CNN with the sigmoid activation function (CNN-sd) and CNN with the softmax activation function (CNN-sx).

Training CNN was performed using prepared I-D without visual assessment of its elements. The number of optimized and rewritten I-D images in the .png and .jpg formats was 191 out of 200 for which the request was sent, and

the number of images that were used for direct training was 163 files. Remaining 28 files were used for validation. Training was performed during 15 epochs for both cases.

Results of training CNN-sd: for 15 learning epochs of the CNN-sd, the speed ranged from 128 to 143 seconds, with a speed of one step of 782-878 milliseconds; the amount of errors, calculated according to [17], ranged from -0.2631 on the first in $-9.2243e-05$ after the last; recognition accuracy ranged from 93.41% in the first cycle to 100% in the last cycle. A graphical representation of the CNN-sd learning outcomes is presented in Fig. 2, *a*.

Results of training CNN-sx are worse than CNN-sd, as can be seen from Fig. 2, *b*, maximum recognition accuracy was 67.15% versus 100% in the case of CNN-sd.

Testing for the adequacy of CNN-sd and CNN-sx. The final stage of the study was testing trained on the basis of automatically optimized using the `optimize_image` function I-D CNN. To test the learning outcomes, 10 images of various formats were used, where only 2 had a target element. CNN-sd identified 5 out of 10 images as "True", 2 images from these 5 had a target element. Thus, the error was in 3 out of 10.

CNN with softmax activation function showed an unacceptable recognition result. All images from the test sample were classified as a target element. We cannot say that this is a 100% negative result, because the two target images were also identified as "True".

Conclusions

The developed algorithm allowed to confirm that CNN with the same architecture and various activation functions, trained on the basis of an array of images previously processed by the `optimize_image` function without direct human involvement, show positive results for the search for key elements. This algorithm allows without human intervention to use automatic sampling from the Internet for training a neural network, but the final results of training a neural network based on the proposed algorithm may not be acceptable for the user. The best quality of transformation was shown by images that do not contain many details and are located on a mono-colored background. However, the capacity of the image that we get at the output is greater than the original.

In future, this approach can be used to create a dataset that will combine primary image processing using the `optimize_image` function and final image processing of the training array by the user, to train CNN with other activation functions.

REFERENCES

1. Zubarev, D.O. and Skarga-Bandurova, I.S. (2018), “Analiz efekty`vnosti navchannya CNN za pry`ncy`pom "vchy`tel`-uchen`" z vy`kory`stannyam nepidgotovlenogo Image-Dataset, Visny`k Nacional`nogo texnichnogo universy`tetu «XPI»”, *Seriya «Informaty`ka i modelyuvannya»*, available at: <https://doi.org/10.20998/2411-0558.2018.42.10> (Accessed 8 Jan 2019).
2. Goodfellow, I., Bengio, Y., Courville, A. and Bach F. (2017), *Deep learning*, Cambridge, MA: MIT Press.
3. Atienza, R. (2018), *Advanced Deep Learning with Keras*, Packt Publishing, Birmingham, UK.
4. Nielsen, M. (2018), “Neural Networks and Deep Learning”, available at: <http://neural-networksanddeeplearning.com> (Accessed 15 Jan 2019).
5. Chollet, F. (2017), *Deep Learning with Python*, Manning Publications, New York, USA.
6. Rosebrock, A. (2017), *Deep Learning for Computer Vision c Python*, PyImageSearch.
7. Shanmugamani, R. (2018), *Deep Learning for Computer Vision* Packt, PacktPublishing, Birmingham, UK.
8. Sejnowski, T. J. (2018), *The Deep Learning Revolution*, MA: MIT Press.
9. Pejić-Bach, M. (2007), “Developing system dynamics models with «step-by-step» approach”, available at: https://www.researchgate.net/publication/28811323_Developing_system_dynamics_models_with_step-by-step_approach (Accessed 24 Dec 2018).
10. Sewak, M., Rezaul K. and Pujari, P. (2018), *Practical Convolutional Neural Networks*, Packt Publishing.
11. “OpenCV 2.4.13.7 documentation. Miscellaneous Image Transformations. Adaptive Threshold”, available at: https://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html?highlight=threshold#threshold (Accessed 18 Jan 2019).
12. (2018), “Python Script to download hundreds of images from Google Images”, available at: <https://github.com/hardikvasa/google-images-download> (Accessed 24 Jan 2019).
13. Langtangen, H.P. (2015), “Doing operating system tasks in Python”, available at: <https://hplgit.github.io/edu/ostasks/ostasks.pdf>. (Accessed 26 Jan 2019).
14. Howse, J. (2013), *OpenCV Computer Vision with Python*, Packt Publishing, Birmingham, UK.
15. “The Python Standart Library. Subprocess management”, available at: <https://docs.python.org/2/library/subprocess.html> (Accessed 26 Jan 2019).
16. Gulli, A. and Pal, S. (2017), *Deep Learning with Keras*, Packt Publishing.
17. Tosi, S. (2009), *Matplotlib for Python Developers*, Packt Publishing, Birmingham, UK.
18. “The Python Standart Library. Parser for command-line options, arguments and sub-commands”, available at: <https://docs.python.org/3.7/library/argparse.html> (Accessed 26 Jan 2019).
19. Changhau, I. (2017), “Loss Functions in Neural Networks”, available at: https://isaac-changhau.github.io/post/loss_functions/ (Accessed 26 Jan 2019).

Received 04.11.19;
after revision 02.03.20

Д.А. Зубарев, И.С. Скарга-Бандурова, Е.М. Сапицкая

АНАЛИЗ ЭФФЕКТИВНОСТИ АЛГОРИТМА
ОПТИМИЗАЦИИ ИЗОБРАЖЕНИЙ ПРИ ФОРМИРОВАНИИ
НЕПОДТВЕРЖДЕННОГО ПОЛЬЗОВАТЕЛЕМ НАБОРА ДАННЫХ

Конечной целью любой оптимизации процессов в определенной сфере является сохранение времени и ресурсов человека. В статье проанализирована эффективность оригинального алгоритма обработки изображений при формировании выборки для обучения искусственной нейронной сети класса CNN на основе неподтвержденного пользователем Image-Dataset, которая будет использована для пользовательских нужд определения элементов изображений в пределах бинарной логики.

К л ю ч е в ы е с л о в а: глубокое обучение, набор изображений, функция оптимизации изображения, алгоритм обучения сети.

Д.О. Зубарев, І.С. Скарга-Бандурова, О.М. Сапицька

АНАЛІЗ ЕФЕКТИВНОСТІ АЛГОРИТМУ ОПТИМІЗАЦІЇ
ЗОБРАЖЕНЬ ПРИ ФОРМУВАННІ НЕПІДТВЕРДЖЕНОЇ
КОРИСТУВАЧЕМ НАВЧАЛЬНОЇ ВИБІРКИ

Кінцевою метою будь-якої оптимізації процесів у певній сфері є збереження часу та ресурсів людини. У статті проаналізовано ефективність оригінального алгоритму обробки зображень при формуванні вибірки для навчання штучної нейронної мережі класу CNN на основі неподтвердженого користувачем Image-Dataset, яку буде використано для потреб визначення елементів зображень в межах бінарної логіки.

К л ю ч о в і с л о в а: глибоке навчання, набір зображень, функція оптимізації зображення, алгоритм навчання мережі.

ZUBARIEV Dmytro Oleksandrovich, Post-graduate G.E. Pukhov Institute for Modelling in Energy Engineering National Academy of Sciences of Ukraine. Graduated from Vladimir Dahl East Ukrainian National University in 2018. Field of scientific research: neural networks, programming, cybersecurity, IT technologies.

SKARGA-BANDUROVA Inna Serhiivna, Doctor of Technical Sciences, Professor, Head of Department of Computer Science and Engineering Volodymyr Dahl East Ukrainian National University. Graduated from the Severodonetsk Institute of Technology of East Ukrainian National University in 1996. Doctor of Technical Sciences since 2015 (Kherson National Technical University). Fields of scientific research: artificial intelligence, decision theory, formal methods of data analysis, medical science, environmental science.

SAPYTSKA Olena Mikhailivna, Cand. of Historical Sciences, Associate Professor, Department of History and Archeology, Volodymyr Dahl East Ukrainian National University. Graduated from Volodymyr Dahl East Ukrainian National University in 2003. Candidate of Historical Sciences since 2009 (Volodymyr Dahl East Ukrainian National University). Field of scientific research: using of IT technology in historical science, creation of interactive historical digital projects, history of IT.