Economic Efficiency of Innovative Projects of CNN Modified Architecture Application

Viktor Khavalko[1[0000-0002-9585-3078]], Andriana Mazur[1[0000-0002-5985-5674]],
Vladyslav Mykhailyshyn[1[0000-0003-1889-9053]], Roman Zhelizniak[1[0000-0001-9033-4490]],
Iryna Kovtyk[2[0000-0001-8876-0538]]

[1] Lviv Polytechnic National University, S. Bandery Str., 12, Lviv 79013, Ukraine
viktor.m.khavalko@lpnu.ua, andriana.v.mazur@lpnu.ua,
vladyslavmykhailyshyn@gmail.com, roman.y.zhelizniak@lpnu.ua
[2] Separated structural unit - College of telecommunications and computer technologies of Lviv Polytechnic National University, Volodymyra Velykogo Str., 12, Lviv, 79000, Ukraine
iryna.zhovnir@gmail.com

Abstract. The paper deals with involves the use of a modified architecture of a convolutional neural network to solve the problem of recognizing the Cyrillic alphabet letters in real time and with high accuracy. The analysis of the existing approaches and methods of handwriting recognition is carried out, features and the basic difficulties which arise at the decision of a recognition problem are considered. To effectively solve this problem, it was decided to form a dataset of letters of the Cyrillic alphabet. In order to cover the widest possible range of letter spelling options, a dataset has been formed, which includes seventy classes of letters of the Ukrainian alphabet. The conducted research of basic algorithms allowed to reveal bottlenecks and shortcomings of the existing approaches, and also to develop the modified architecture of a convolutional neural network which showed on the formed dataset accuracy of recognition within 97-98%. At the same time, a significant economic effect was obtained from the implementation of this solution.

Keywords: Convolution Neural Network, Domain Adaptation, Cyrillic letters recognition, ML Model, Handwriting Text, MNIST.

1 Introduction

Recently, handwriting recognition systems have become popular. Because such systems collect information other than the actual image of the text, the accuracy of the work is greatly increased. The system can also be adapted to the handwriting of a particular person. With offline recognition, when static documents with different people's handwriting are processed, this is not possible.

That is why the problem of handwriting recognition both online and offline is quite relevant. Even in the case of handwritten text, when the letters are written separately from each other (without joints), without unnecessary artifacts (spots, scan inaccura-

 CybHyg-2019: International Workshop on Cyber Hygiene, Kyiv, Ukraine, November 30, 2019.

cies, lighting, background, etc.), the recognition accuracy usually reaches 80-90%. This is a rather low figure, as each page of such text has several dozen errors. If we talk about a full-fledged handwritten text, the problem is not solved at all.

The development of deep neural networks [1] for image recognition contributes to the development of already known research areas in machine learning. One such area is domain adaptation (DA) [2]. The essence of this adaptation is to train the model on data from the source domain so that it shows the appropriate quality of recognition on the target domain. For example, the source can be synthetic data that can be generated, and the target domain can be photos of users. Then the task of DA is to train the model on synthetic data, which will work well with "real" objects.

There are many application tasks, which are characterized by a small amount of training data. In these cases, the generation of synthetic data and the adaptation of the model trained on them can be very helpful.

2 Analysis of existing solutions

The biggest problems in handwriting recognition are those that make it difficult for people to read even their own handwriting [3]. First, the fact that most characters can be written differently. It is also rare to meet two people with the same handwriting. This problem is due to the difference in fonts in the classic problem of text recognition. But unlike fonts, each letter in a person's text may have a different style depending on the context in which the surrounding letters are written, and many other factors. To deal with this problem, many systems contain a component that itself learns the resulting handwriting, distinguishes users, and uses this data in decision-making. That is, the task of forming a dataset of possible options for writing the letters of the alphabet immediately arises.

The logistic regression algorithm is one of the basic algorithms for recognition problems. It is widely used when the task of classification arises. This algorithm allows you to distribute a set of objects according to certain of their characteristics and classify them accordingly.

The most universal approach to solving the problem of handwriting recognition is neural network [3-8]. The main advantages of neural networks are the ability to learn independently and automatically based on sampling, to be productive on noisy data, the possibility of parallel implementation and the ability to be effective tools for processing large databases. There are many different methods in this approach[3,5,6]. The most popular are fuzzy neural networks [8], Heming's network [1,3], Hopfield's network [1], Kohonen's self-organizing map [3] and many others [4,9,10].

Today, there are many approaches to solving the problem of character recognition in the image [5,7,10,11], but most of them provide the results of low probability with a high percentage of recognition errors, which requires further research and improvements in algorithms.

The latest research in domain adaptation [2] touches on the use of previous experience gained by the neural network in the new task. In addition, domain adaptation can help solve one of the fundamental problems of deep learning: training large networks with high recognition quality requires a very large amount of data, which in practice is not always available. One solution may be to use DA methods on synthetic data that can be generated in almost unlimited quantities.

3 Problem statement

The purpose of the work is to develop and investigate algorithmic and software tools for recognition of handwritten Cyrillic characters, to demonstrate examples of implementation of the above algorithms and to provide results that show the quality of recognition. At the same time, on the basis of the received personal data from school competitions to form an open dataset of Ukrainian-language symbols and to develop the convolutional neural network architecture, which will ensure the accuracy of Cyrillic symbols recognition within 95-98%.

4 Main benchmarks

As in any field of machine learning, domain adaptation accumulates over time a number of studies that need to be compared. To do this, the community produces datasets, on the training part of which the models are trained, and on the test - are compared. Despite the fact that the field of deep domain adaptation research is still relatively young, there is already a large number of articles and databases used in many articles [1-3, 7,8].

4.1 Numbers dataset

In computer vision, the simplest datasets are associated with handwritten numbers or letters [3, 10-12]. There are several data sets with numbers that first appeared for experiments with image recognition models. In works on domain adaptation it is possible to meet the most various their combinations in pair source - target domain. Among these datasets:

- MNIST - handwritten numbers that do not require additional presentation (Fig. 1);
- USPS - handwritten numbers in low resolution;
- SVHN - house numbers from Google Street View;
- Synth Numbers - synthetic numbers, as the name implies.

From the point of view of the learning task from synthetic data for use in the "real" world, the greatest interest are pairs:

- Source: MNIST, Target: SVHN;
- Source: USPS, Target: MNIST;
- Source: Synth Numbers, Target: SVHN.

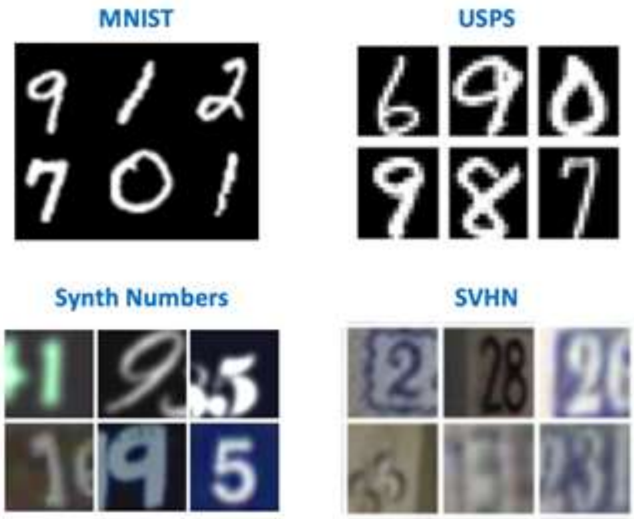


Fig. 1. Data sets for learning

Most methods have benchmarks on "digital" datasets. But other types of domains can be found not in all works. Datasets for learning and recognition of the Cyrillic text in general, and for the Ukrainian language in particular, are of considerable interest. Most of the research conducted with this dataset is concerned with improving recognition accuracy, but does not address the scope of expanding recognition language. More recently, the Comnist project [13], a Cyrillic-oriented MNIST, has emerged. The disadvantage of this solution is the large amount of technical work to achieve an acceptable result. Another problem was that the dataset used was no different from other synthetic fonts, but with different slopes or writing styles. However, when people write in the program interface, they never write as they would with a pen or pencil on paper (Fig. 2). That is, most of the letters used in the Comnist project's dataset have an ideal or close character, which cannot be said of traditional letter writing.

Fig. 2. Writing letters in Comnist and real life

4.2 Office dataset

This dataset contains 31 categories of different objects (Fig. 3), each of which is presented in 3 domains: an image from Amazon, a photo from a webcam and a photo from a digital camera.

Fig. 3. Office dataset

It is useful for testing how the model will respond to background additions and image quality in the target domain.

4.3 Road signs datasheet

Another pair of datasets (Fig. 4) for learning the model on synthetic data and applying it to "real" data:

Source: Synth Signs - road signs images generated so that they look like real signs on the street;

Target: GTSRB is a well-known recognition base that contains signs from German roads.

Fig. 4. Road signs datasheet

The peculiarity of this databases pair is that the data from Synth Signs are generated quite similar to "real" data, so the domains are quite close.

4.4 From the car window

Dataset for segmentation. Quite an interesting couple, closest to real conditions. The source data is obtained using the game engine (GTA 5), and the target - from real life. Similar approaches are used to train models used in autonomous vehicles.

SYNTHIA or GTA 5 engine - pictures with a view of the city from the car window, generated by the game engine;

Cityscapes - car photos taken in 50 different cities.

5 Dataset formation

When creating a dataset, you should recognize uppercase and lowercase letters, as well as the possibility of different spellings of the same letter. In general, we found that there are more than 70 classes that form a dataset of Ukrainian-language symbols (Fig. 5).

Fig. 5. Dataset formation for letter classification

Response forms of school competitions were used in the dataset formation. On the basis of the received forms two sets (fig. 6) were formed - the first is a text record of certain letters, and the second set - the image which corresponds to the given text record.

Fig. 6. Data collection from competition forms

After that, the next step is to create a dataset. It is necessary to take a text file and the corresponding form - the first letter in the text file is the letter "Г", according to the form of the image we select the first letter and we move the image of this letter to the necessary folder (fig. 7). Of course, when creating a dataset, it should be borne in mind that anyone who filled out the forms when participating in the competition could make mistakes, namely:

- write the letter in more than one cell;
- the letter could not be printed, but written;
- there could be letter corrections.

Therefore, as a result of such actions in each of the folders could be many images of letters that do not carry useful information, ie are garbage. That is, it is necessary to filter the data and remove all debris.

a) b)

Fig. 7. Dataset formation: a) a set of folders that correspond to the letters; b) different images of the corresponding letter

Filtering took place in two stages: the first was manual filtering, the essence of which was to remove all visible garbage from each of the folders, which corresponded to the letter of the Ukrainian alphabet. The second is automated filtering using machine learning (Fig. 8). A model was built that implements the task of letter recognition with a certain accuracy. At the entrance of this model submitted all manually filtered letters for training. After that, new selected letters are fed to the input of the model for further recognition. As a result of the model, we obtain letters that the model has classified and not classified. All classified letters are sent to the appropriate folder, and letters that have not been classified are classified again. If the "garbage" got into the folder with unclassified letters, we delete it. That is, the built model helps to filter the data when forming a dataset.

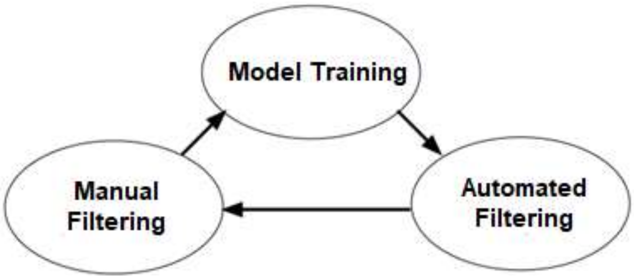


Fig. 8. Automated filtering principle

6 Algorithmic bases of character recognition

Once the dataset is formed, it is necessary to select the algorithmic base that will be used to recognize the Cyrillic alphabet letters. In general, it should be understood that there is an accuracy of letter recognition, and there is, accordingly, the accuracy of word recognition, which in turn depends on the word length and the accuracy that we achieve in recognition.

Three algorithms were used in the work, which allowed comparing their accuracy and choosing the best one, both from the point of view of calculations and from the point of view of recognition.

KNN (k-nearest neighbors) algorithm is a metric algorithm for automatic objects classification. The main principle of the nearest neighbors method is that the object is assigned to the class that is most common among the neighbors of this element. Neighbors are taken based on the set of objects whose classes are already known, and based on the key value of k for this method, it is calculated which class is the most numerous among them. Each object has a finite number of attributes (dimensions). It is assumed that there is a certain set of objects with an existing classification.

In our case, each letter image is converted into a vector of pixels, which take the grayscale value from 0 to 255. We take two samples and plot the difference between them. Similarly, a vector is taken that corresponds to an unrecognized instance and a graph is also constructed. Next, you need to determine k, ie how many neighbors to consider to determine to which class the recognized object belongs. In the work used k = 3. The research results showed that the MNIST algorithm gives an accuracy of 80-83%, but on the generated dataset, the algorithm showed 35-37%.

To improve the result, PCA (Principal Component Analysis) was used, which makes it possible to reduce the dimension of the problem and select three matrix components from the image, one of which plays the most important role. For letters images from the formed dataset, the dimension was reduced by 5-6 times, i.e. from 784 pixels received 50-60. The use of the PCA algorithm increased the accuracy by 4-6%.

The analysis results showed why such low recognition accuracy - there are many letters that in different spellings look similar or differ only by a few pixels (for example, the letters "Б" and "В").

The next algorithm that was considered was XGBOOST [14]. This algorithm builds a decision tree based on the key pixels values (Fig. 9). Of course, in reality, in practice, more than one tree is built and the depth of such trees is also significantly greater than in Fig. 9.

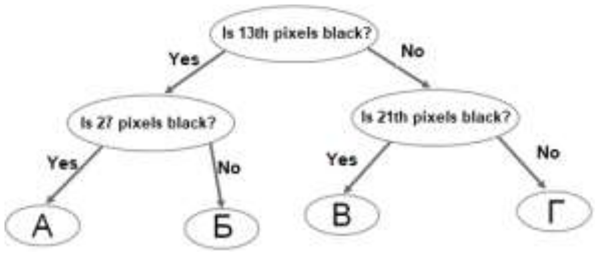


Fig. 9. Decision tree

The algorithm showed significantly better accuracy than the previous two algorithms - 86-88%. However, if you take the name recognition, then the accuracy will drop accordingly. For example, if you take a last name that consists of five letters, the accuracy will be equal to 0.88 * 0.88 * 0.88 * 0.88 *, 88 = 0.527. That is, in fact, every second last name will be recognized incorrectly!!!

7 CNN application for letters classification

In order to improve the accuracy of Cyrillic text recognition, consider use of convolutional neural networks. We use the classic CNN architecture [5.10], which consists of three main layers, namely - the convolution layer, the activation function and the MaxPooling layer, which allows to highlight the part of the image that responded most to the filters and activation function of our network. As an activation function, we use the ReLu function, i.e. a function that will discard all negative values obtained from the previous CNN layer. If we consider the CNN use on MNIST, than the most common is the following architecture (Fig. 10), which gives a recognition accuracy of 92%. On the formed dataset of Cyrillic letters such network showed accuracy of 84%. Which is a good result, but not exactly what we would like.

Fig. 10. CNN architecture

To significantly improve the accuracy of forecasting, an improved CNN architecture is proposed, the structure of which is shown in Fig. 11. This neural network consists of three layers: the convolutional layer (3x3 filter), the ReLu activation function, and the MaxPooling layer, which are repeated three times. After that, the result is submitted for prediction and we get the probability of Cyrillic character recognizing.

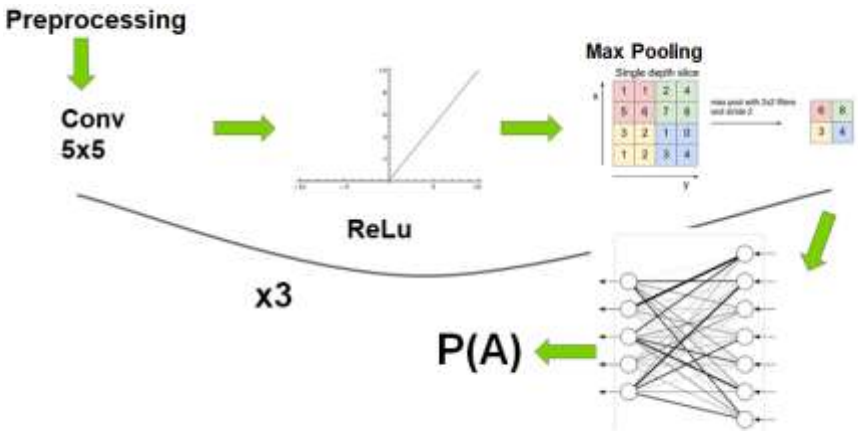
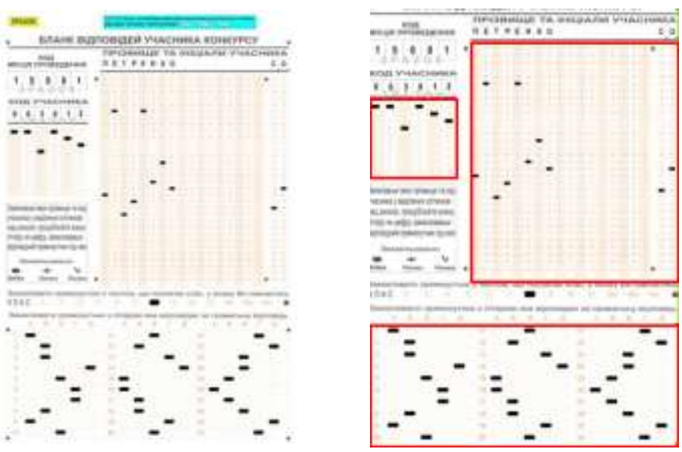


Fig. 11. Improved CNN architecture

Thus, it was possible to achieve recognition accuracy in the range of 97-98%. That is, for a surname consisting of five letters, we will already have an accuracy of about 88%.

8 Economic effect from the proposed solution implementation

Since the dataset was formed from questionnaires filled out by the competition participants (Fig. 12), The question arises what will give the implementation of the developed improved CNC architecture. All questionnaire forms are printed in B5 format. If the proposed CNN is responsible for the recognition of surnames and other areas, then this form can be reduced to A5 format.



a) b)

Fig. 12. Competition participant's questionnaire: a) general view, b) important areas of the questionnaire that are recognized

If we compare the area of these formats, we see that the A5 format is 30% smaller. Given that the competition involves more than 0.5 million participants, we get significant savings in paper for printing forms (about 200-250 packs of paper).

Another type of saving is saving the operator's time, and hence saving money. That is, when organizing such competitions, the bottleneck of the system is the scanner, which then scans the answer sheets (this applies to any such activities - external evaluation, passing exams, etc.). The scan time directly depends on the size of the form (Fig. 13). Therefore, the smaller the form, then shorter the scanning time and the greater the operator's time savings. If you place the forms of participants as shown in Fig., you will get a saving of 40% time. This means that with 30 days of work we get savings (250 - 148) / 250 · 30 days = 12.24 days.

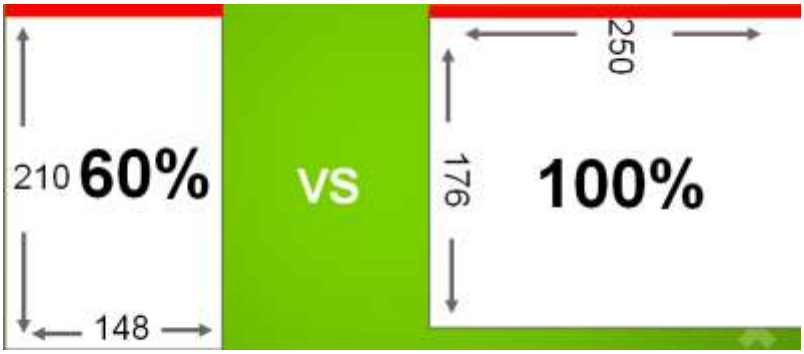


Fig. 13. Form scanning options

That is, the implementation of the proposed improved architecture of CNN will give a significant economic effect.

Conclusions

1. A handwritten dataset for letter recognition of the Cyrillic alphabet has been created, which has 70 classes of 1000 copies for each letter.
2. The comparative analysis of symbols recognition methods on an example of a dataset of MNIST and the generated dataset of Cyrilic letters is carried out. The expediency of using neural networks, namely convolutional NM, is substantiated.
3. Developed the architecture of the convolutional neural network, which has a high accuracy of recognition (97-98%) of the Cyrillic alphabet letters.
4. A significant economic effect has been achieved on the example of organization and processing of competition results among schoolchildren (over 0.5 million participants), both in terms of saving paper and in terms of operating time.

References

1. Zhang G.P.: Neural networks for classification: a survey. In: IEEE Transactions on Systems, Man, and Cybernetics., Part C: Applications and Reviews. Vol.30(4), p. 451–462. (2000).
2. Mei Wang, Weihong Deng: Deep Visual Domain Adaptation: A Survey. In: Neurocomputing. Vol. 312, p.135-153. (2018).
3. Yavorska T., Prihunov O., Syerov Yu. Efficiency of Using Social Networks in the Period of Library Activity in Remote Mode. CEUR Workshop Proceedings. Vol 2616: Proceedings of the 2nd International Workshop on Control, Optimisation and Analytical Processing of Social Networks (COAPSN-2020), Lviv, Ukraine, May 21, 2020. p. 214-226. http://ceur-ws.org/Vol-2616/paper18.pdf
4. Patel K., Gandhi M.: Offline Handwritten Character Recognition: A Review. In: International Journal of Scientific & Engineering Research. Vol. 5, p.193–196. (2016).
5. Tsmots I., Skorokhoda O., Tsymbal Yu., Tesliuk T., Khavalko V.: Neural-Like Means for Data Streams Encryption and Decryption in Real Time. In: IEEE Second International Conference on Data Stream Mining & Processing. pp.438-443 (2018).
6. Gatys L.A., Ecker A.S., Bethge M.: Image style transfer using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, p. 2414–2423. (2016).
7. Fedushko S., Syerov Yu., Skybinskyi O., Shakhovska N., Kunch Z. (2020) Efficiency of Using Utility for Username Verification in Online Community Management. Proceedings of the International Workshop on Conflict Management in Global Information Networks (CMiGIN 2019), Lviv, Ukraine, November 29, 2019. CEUR-WS.org, Vol-2588. pp. 265-275. http://ceur-ws.org/Vol-2588/paper22.pdf
8. Boyko N., Pylypiv O., Peleshchak Y., Kryvenchuk Y., Campos J.: Automated document analysis for quick personal health record creation. In: 2nd International Workshop on Informatics and Data-Driven Medicine. IDDM 2019. Lviv. p. 208-221. (2019).
9. Bengio Y.: Learning deep architectures for ai. Foundations and Trends. In: Machine Learning. Vol. 2(1), P.1–127. (2009).
10. Borde S., Shah E., Rawat P., Patil V.: Fuzzy Based Handwritten Character Recognition System. In: International Journal of Engineering Research and Applications. p.151–154. (2012).
11. Kryvenchuk Y., Vovk O., Chushak-Holoborodko A., Khavalko V., Danel R.: Research of servers and protocols as means of accumulation, processing and operational transmission of measured information. Advances in Intelligent Systems and Computing. Vol.1080. p.920-934. (2020).
12. Dongare S., Kshirsagar D., Waghchaure S. Handwritten Devanagari Character Recognition using Neural Network. IOSR Journal of Computer Engineering. Vol.14. p.74–79. (2014).
13. Xi Z., Panoutsos G.: Interpretable Machine Learning: Convolutional Neural Networks with RBF Fuzzy Logic Classification Rules. International Conference on Intelligent Systems. p.448–454. (2018).
14. Kaur M., Kumar S.: A recognition system for handwritten gurmukhi characters. International Journal of Engineering Research & Technology. №1. p. 1–5. (2012).
15. Cyrillic-oriented MNIST. A dataset of Latin and Cyrillic letter images for text recognition. URL: https://github.com/GregVial/CoMNIST.
16. Tianqi Chen, Carlos Guestrin: XGBoost: A Scalable Tree Boosting System. In: Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p.785–794. (2016) https://doi.org/10.1145/2939672.2939785