

Handwritten Ukrainian Character Recognition using a Convolutional Neural Networks and Synthetic Dataset

Yevhen Chychkarov, Olha Zinchenko

State University of Telecommunications, Solomenska street, 7, Kyiv, 03110, Ukraine

Abstract

Abstract text. This paper considers several options for the architecture of convolutional neural networks for the recognition of isolated handwritten Ukrainian characters and numbers, which were trained using a synthetic dataset built on the basis of a set of handwritten and cursive fonts. Comparison of the results of recognition of several variants of images containing handwritten letters and numbers using neural networks with different architectures showed that an increase in the number of convolutional layers leads to a decrease in the frequency of erroneous character recognition. The size of the training dataset significantly affects the reliability of character recognition. The data sets used in the work contained from 192 to 2304 samples per class. The upper limit of the number of samples per class is close to the limit that provides acceptable recognition accuracy. Reducing the sample size by reducing the number of samples per class leads to a significant decrease in recognition accuracy (from 90% recognition accuracy of elements of real inscriptions to 40-60% with a 4-fold decrease in sample size).

Keywords 1

handwriting recognition; recognition of Ukrainian characters; convolutional neural networks (CNN); digit recognition; deep learning; image processing

1. Introduction

Optical character recognition (OCR) is a technology that is widely used today. The basis of this technology is the process of classifying images of symbols, which are selected on the original digital image, according to the corresponding samples [1].

Information technologies based on optical recognition allow solving a wide range of various practical tasks: identification of vehicle registration numbers from images of license plates, which helps control traffic [2], conversion of printed academic records into text for storage in an electronic database, decoding ancient inscriptions and texts, automatic data entry by optical scanning of cards or bank checks.

In most cases, modern optical recognition systems are based on deep learning neural networks [3, 4]. Convolutional neural networks (CNN) are widely used for image processing. It is one of the most popular types of deep neural networks and can be used to effectively recognize characters present in an image [5].

2. Literature review

Convolutional neural networks are widely used to solve optical recognition problems. They are able to automatically highlight the conditional features of the input data. The properties of such networks make them a very convenient tool for solving computer vision problems, in particular, for recognizing images of letters or numbers.

MoMLeT+DS 2023: 5th International Workshop on Modern Machine Learning Technologies and Data Science, June 3, 2023, Lviv, Ukraine
EMAIL: chychksrovia@gmail.com (Y.Chychkarov); zinchenkoov@gmail.com (O.Zinchenko)

ORCID: 0000-0002-4362-5129 (Y.Chychkarov); 0000-0002-3973-7814 (O.Zinchenko);



© 2023 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



CEUR Workshop Proceedings (CEUR-WS.org) Proceedings

Initially, most of the research was focused on the recognition of the Latin alphabet letters, but in recent years other alphabets began to attract attention - Arabic, Russian, Kazakh, Chinese, Indian, etc. [6-11]. For research on handwritten Latin alphabet recognition technologies, the EMNIST dataset became the de facto standard [12]. Many different variants of neural network architectures have been proposed to classify the images of this set.

One of the first successful attempts to use deep learning for character recognition was the creation of the LeNet-5 architecture [13]. This architecture showed the highest accuracy of classification of handwritten digits among the solutions available at that time (1998).

Similar solutions are widely used in relatively recent works on computers with low computing power. For example, the ConvNet architecture proposed in [14] consists of two convolutional layers with kernels of size 5x5 each, and using non-linearity activation (ReLU) functions, a MaxPooling layer after each convolutional layer, and two fully connected layers that contain 500 neurons and the last layer with a selection of 26 classes. Such a neural network has only 60,000 learning parameters. This number of parameters is much smaller than the AlexNet network (60 million training parameters and 650,000 neurons) [15] or the GoogleNet network (6.8 million training parameters) [16].

The best results for training handwritten digit recognition models using the EMNIST Digits (or MNIST) datasets were achieved using convolutional neural networks (see [17] for a review).

One way to improve the accuracy of letter or number image recognition is to use models with a more complex architecture than AlexNet or LeNet. For example, good results in recognition accuracy were achieved due to the use of capsular layers [18]. The authors of [19] proposed a convolutional neural network that contains 14 convolutional layers to represent character characteristics, two MaxPooling layers to reduce the number of features or highlight strong features, one softmax layer, and one classification layer for isolated character recognition.

Pre-training using ImageNet accelerates convergence, especially at the beginning of training. However, for models with random initialization, the results achieved do not differ from models with pretraining for a comparable number of epochs [20].

According to [21], models created from scratch, as a rule, give better results compared to pre-trained models in the recognition of handwritten characters of the Arabic language. Regarding the complexity of the CNN architectures used, according to [21], less complex CNN models are less accurate, but have higher classification and learning rates (and vice versa). The authors of [21], based on the obtained results, suggested that learning from scratch all used models can improve the accuracy of model classification and the speed of obtaining results, regardless of the complexity of the model.

In numerous studies devoted to the recognition of handwritten symbols, there is experience in using sufficiently complex architectures of neural networks. For example, in [22], modern pre-trained CNN architectures were used to classify 231 different Bangla handwritten characters based on the CMATERdb dataset. The images were first converted to black and white form with a white foreground color. Images were resized to 28x28 pixels. These images were used as input for the CNN architectures that were used. The learning rate was set to 0.001. Categorical cross-entropy was used as the error function. After 50 epochs, InceptionResNetV2 achieved the best accuracy (96.99%). DenseNet121 and InceptionNetV3 also demonstrated excellent recognition accuracy (96.55 and 96.20%, respectively). The authors [22] also considered a combination of pre-trained architectures InceptionResNetV2, InceptionNetV3 and DenseNet121, which provided even better recognition accuracy (97.69%) compared to other individual CNN architectures, but concluded that its practical use requires large computing power and memory and therefore hard for practical use. The models were tested in cases where character recognition appears difficult to a human, but all architectures showed the same ability to reliably recognize such images. According to [22], the InceptionResNetV2 architecture can be called the most efficient model, taking into account the computational complexity, the amount of memory and the ability to recognize distorted symbols.

Research on the use of various architectures of neural networks without prior training based on ImageNet is also known. For example, in [23], two variants of convolutional neural networks with different architectures, varying the depth, width, and number of network parameters, were tested for recognition of Devanagari characters.

The first model consisted of three convolutional layers and one fully connected layer. The second model came from the LeNet family, and consisted of two convolutional layers followed by two fully

connected layers. The best result in terms of recognition accuracy (over 98%) was obtained by the authors with a model with more convolutional layers.

A similar result was obtained in [24]. The authors investigated three variants of the architecture of CNN networks: LeNet-5, a modified variant of LeNet and AlexNet CNN. Using the latest version of the neural network architecture, Devanagari character recognition accuracy of 99% was achieved.

Numerous experiments with several convolutional neural networks (basic CNN, VGG-16, and ResNet) were conducted in [25] using regularization approaches such as filtering and data augmentation. The VGG and Resnet architectures gave close results in recognition accuracy: using the Resnet architecture, it was possible to achieve the best result with a recognition rate of 98.57%, for the VGG-16 architecture, a result of 97.14% was achieved.

The work [26] also noted the higher achieved recognition accuracy when using the deeper architecture of the CNN neural network. But increasing recognition accuracy is achieved only by using input data augmentation. In [27], different CNN architectures were investigated for recognizing the EMNIST dataset. According to [27], using the GoogleNet architecture always gives higher accuracy compared to ResNet18, but requires 2.5-2.9 times more time to train the model.

Neural network architectures that use prior learning have been created to classify color images of different sizes. Therefore, for many datasets (e.g., EMNIST Letters 28×28), single-channel images must be converted to three-channel to use existing libraries and pre-training capabilities [27]. In particular, the ResNet module from the tensorflow package requires an input image with a size of at least 32×32×3.

When using a modified CNN architecture and training models without loading the weights of the pre-trained model, the input data may contain single-channel images. When comparing variants of color and monochrome image recognition [27], it is indicated that variants with an input image size of 40×40 pixels (for the resized EMNIST data set) in monochrome versions with rotation and shift augmentation have the highest results in the models studied by the authors (ResNet18 and GoogleNet).

For the recognition of Cyrillic characters, similar studies are quite few. There is experience in using the MobileNet architecture, which included 30 layers [28] for character recognition of the Kazakh and Russian languages.

Some results of Cyrillic character recognition are also presented in [29-30].

Regarding the data set for the recognition of Ukrainian letters, individual works in this direction are known. According to [31], when creating a data set for model training, it is necessary to distinguish between uppercase and lowercase letters, as well as take into account the possibility of different spellings of the same letter. The authors [31] identified more than 70 classes that form a complete set of symbols of the Ukrainian language (for example, different spellings of the lowercase letter "a" were taken into account).

3. Experimental setup and proposed approach

There are quite a few studies of handwriting recognition technologies that are based on the use of the EMNIST data set [17] (at least for English). There is known experience of using various classifiers and neural network technologies to recognize Cyrillic alphabet symbols, but comparative studies of recognition technologies for them are fragmentary. Also, there are no EMNIST-like datasets for the Ukrainian alphabet.

This article is devoted to researching the possibilities of recognizing Cyrillic (mainly Ukrainian) handwritten letters using convolutional neural networks and analyzing the influence of the selected neural network architecture on the accuracy and reliability of recognition. In addition, the possibility of using a synthetic data set and the effect of augmentation of the original data set on the recognition results were investigated.

The goals of this study:

- Analysis of the influence of the architecture of convolutional neural networks on the accuracy of recognition of handwritten numbers and letters of the Ukrainian alphabet.
- Analysis of the peculiarities of the recognition of Ukrainian symbols under the conditions of learning convolutional neural networks using a synthetic data set with various options for increasing the training sample.

3.1. Building a dataset for model training

The dataset used for training the models was built using a set of handwritten and italic fonts (a total of 48 font variants with Ukrainian glyphs were selected). All images of letters and numbers were divided into 76 classes (33 lowercase, 33 uppercase letters and 10 numbers) or 43 classes (33 letters and 10 numbers). All images of the data set were centered and a dataset was created from them with the size of each image 28x28, or 32x32, or 64x64, or 128x128 pixels. The Pillow library was used to create or transform images with letters or numbers (from one-channel to three-channel).

The test data set was built using the same fonts. The selection of specific fonts and augmentation options was chosen randomly. The volume of the test dataset was about 10% of the volume of the training one.

The presence of only a small number of suitable fonts with Ukrainian glyphs required the use of augmentation to form the necessary data set. We used the capabilities of the Image Data Generator from the tensorflow package to perform three options for transforming character images: random rotation, shift transformation, scaling transformation.

The number of generated images varied from 2 to 48 for each symbol. For 32 images per each symbol the total volume of the dataset was 116,736 samples (32 images per character of one font). This sample volume is quite comparable to the EMNIST Letters dataset [12, 17], which contains mixed lowercase and uppercase letters (26 classes and a total of 145,600 samples).

3.2. Preprocessing of images for recognition

Tools from the OpenCV library were used to select image regions containing letters or numbers, which were then recognized. The findContours function or the algorithm for extracting the most stable extreme region (mser function) was used to select the contours of recognized symbols.

The algorithm for preprocessing the image and selection of the area containing letters or numbers included the following stages:

1. Image filtering to reduce the noise level (the Gaussian filter was used - function cv2.GaussianBlur);
2. Binarization of the image to cut off noise (the cv2.threshold function was used, its parameters were chosen for reliable selection of character contours);
3. Morphological transformation (dilation – function cv2.dilate, several iterations were used);
4. Selection of contours and their sorting (selection of contours was carried out using the function cv2.findContours);
5. Image segmentation, ie. selection of recognition areas as a set of rectangles containing contours of letters and numbers (cv2.boundingRect functions were used).

Directly for recognition, selected regions of interest were cut from the original image, binarization was again applied to them, after which the obtained images of individual symbols (without dilation or other distortions) were scaled to the size of the image in the dataset. Each pixel value of the images was in the range 0 to 255, so normalization of these pixel values was performed by dividing by 255 so that all values in the array describing the image were in the range 0 to 1.

3.3. Proposed CNN Architectures

At the first stage of the research, the models were trained using single-channel images sized 28x28 pixels. The simplest variants of the architecture of LeNet-type convolutional neural networks for character image recognition are presented in **Table 1**.

More complex neural network architectures are presented in **Table 2**. Architecture 4 and architecture 5 are implementations of the AlexNet architecture for single-channel images. Architecture 6 included thirteen convolutional layers and three dense layers, as well as MaxPooling and Dropout layers. This version of the architecture is the most complex and repeats the VGG-16 architecture with respect to single-channel images. However, it turned out to be the best in terms of accuracy and reliability of recognizing the test sample and real inscriptions.

Table 1

The simplest variants of convolutional neural network architecture

Architecture 1	Architecture 2	Architecture 3
Input (28x28x1)	Input (28x28x1)	Input (28x28x1)
conv2d, 128 filters	conv2d, 64 filters	conv2d, 128 filters
conv2d, 128 filters	conv2d, 64 filters	conv2d, 128 filters
MaxPooling2D	MaxPooling2D	MaxPooling2D
Dropout	Dropout	Dropout
Flatten	conv2d, 128 filters	conv2d, 256 filters
Dense, 256 filters	conv2d, 128 filters	conv2d, 256 filters
Dropout	MaxPooling2D	MaxPooling2D
Dense (output - 76 classes)	Dropout	Dropout
	Flatten	conv2d, 512 filters
	Dense, 256 filters	conv2d, 512 filters
	Dropout	MaxPooling2D
	Dense (output - 76 classes)	Dropout
		Flatten
		Dense, 1024 filters
		Dropout
		Dense (output - 76 classes)

Table 2

Variants of the architecture of convolutional neural networks such as AlexNet and VGG 16

Architecture 4	Architecture 5	Architecture 6
Input (28x28x1)	Input (28x28x1)	Input (28x28x1)
conv2d, 128 filters	conv2d, 64 filters	conv2d, 128 filters
conv2d, 128 filters	conv2d, 64 filters	conv2d, 128 filters
MaxPooling2D	MaxPooling2D	MaxPooling2D
Dropout	Dropout	Dropout
Flatten	conv2d, 128 filters	conv2d, 256 filters
Dense, 256 filters	conv2d, 128 filters	conv2d, 256 filters
Dropout	MaxPooling2D	MaxPooling2D
Dense	Dropout	Dropout
(output - 76 classes)	Flatten	conv2d, 512 filters
	Dense, 256 filters	conv2d, 512 filters
	Dropout	MaxPooling2D
	Dense (output – 76 classes)	Dropout
		Flatten
		Dense, 1024 filters
		Dropout
		Dense (output – 76 classes)

Architecture 1 included an input layer, one convolution block of two layers, a MaxPooling subsampling layer, a Dropout regularization layer, a dense layer, a Flatten dimensionality transformation layer, another regularization layer, and an output layer.

Two more variants of the architecture of convolutional neural networks with an increased number of convolutional blocks are also presented in **Table 1** (architecture 2 and architecture 3). They differ from the simplest version by an increase in the number of convolutional blocks from two layers (two blocks in the architecture 2 or three blocks in the architecture 3).

At the second stage of research, taking into account the presence of recognition errors even when using the best models, several variants for more complex neural networks architectures were

considered. Research has been done with VGG16 and VGG19 [32], ResNet[33] or ResNetV2[34], MobileNet or MobileNetV2 [35, 36], InceptionResNetV2 [37] architectures.

Several variants of implemented architectures for ResNetV2 family are shown in **Figure 1**.

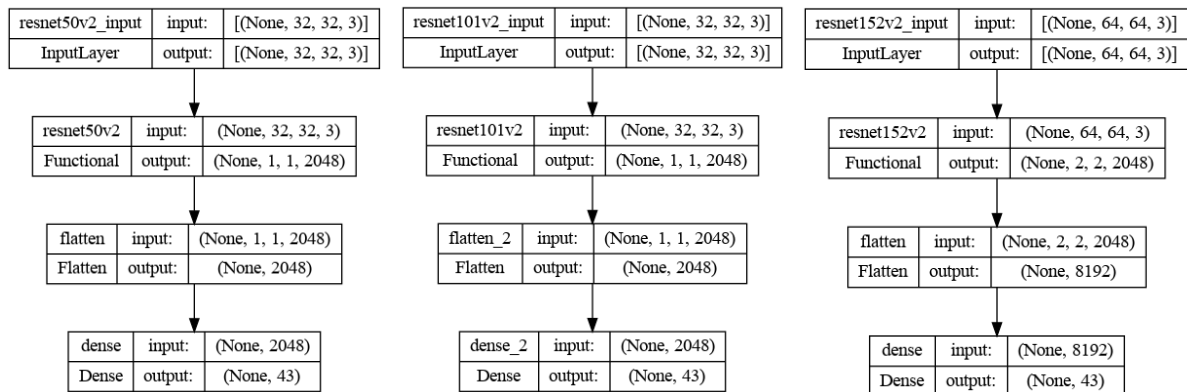


Figure 1: Examples of implementation of models with architectures ResNetV2 family

4. Experimental results and discussion

An example of the learning results of model with architecture 1 is presented in **Figure 2**. An example of the results of learning a model with architecture 6 (recognition accuracy and amount of loss depending on the number of learning epochs) is presented in **Figure 3**.

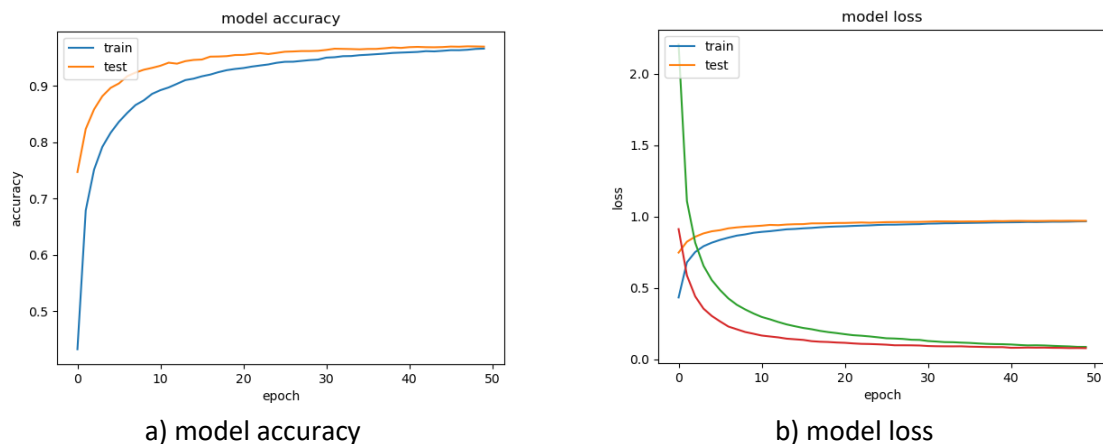


Figure 2: Results of learning the simplest LeNet-type model

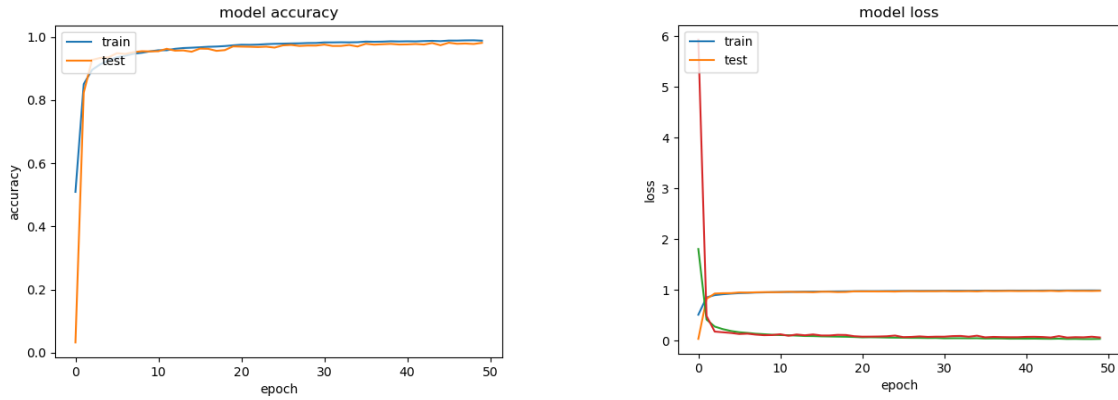
All the architectures used during training on the maximum size sample provided a recognition accuracy of 95-99% of the sample items.

An increase in the number of neural network parameters due to the use of a deeper architecture leads to an increase in recognition accuracy. The calculation time during neural network training increases with the increase in the number of adjustable parameters (when comparing architectures 1 and 6 - approximately an order of magnitude).

However, when trying to recognize images with real inscriptions that do not belong to the training or test sample, a significant difference in the behavior of the studied architecture variants was found regarding the possibility of reliable character recognition.

A typical example of recognizing an inscription containing letters is given in **Table 3**. As can be seen from the obtained results, 100% recognition accuracy is provided only by the most complex variant of the architecture (variant 6).

An attempt to recognize an inscription containing only numbers gave an even more pronounced result of the accuracy of recognizing an image with isolated numbers, see **Table 4**.



a) model accuracy b) model loss

Figure 3: Results of learning the VGG-16-type model

Table 3

A sample of the results of an inscription with letters recognition





Inscription on the image	CNN Architecture	Recognized (in Ukrainian)	Accuracy score
 Inscription ("АБІІВ" in Ukrainian)	Architecture 1	ДБІВ	50%
	Architecture 2	АБІєв	100%
	Architecture 3	ДБіїв	80%
	Architecture 4	ДБіїв	80%
	Architecture 5	АБІєв	100%
	Architecture 6	АБІєв	100%
 Selection of recognition areas			

Table 4

A sample of the results of an inscription with numbers recognition

Inscription on the image	CNN Architecture	Recognized (in Ukrainian)	Accuracy score
 Inscription ("12345" in Ukrainian)	Architecture 1	1r5ц5	20%
	Architecture 2	1r3ц5	40%
	Architecture 3	1r5ц5	40%
	Architecture 4	123ц5	80%
	Architecture 5	123ц5	80%
	Architecture 6	12345	100%
 Selection of recognition areas			

Similar results were obtained for many other variants of inscriptions, including those with letters and numbers at the same time: acceptable results in terms of recognition accuracy were obtained for more complex variants of architecture.

Recognition errors were obtained on some samples of inscriptions and when using deep architectures.

Neural networks for all architectures were trained using the Adam optimizer, the learning rate was chosen to be 0.0001, the number of training epochs was chosen to be 50.

The size of the training sample strongly affects the reliability of character recognition. The generation of 1,536 images per letter or number (32 images for each character for 48 font types) is actually the limit for acceptable recognition accuracy. Reducing the sample size leads to a significant decrease in recognition accuracy (from 100% accuracy to 40-60% when the sample size is reduced by 4 times). An increase in the size of the sample leads to a noticeable increase in the time spent on training the model.

The use of ResNet or MobileNet architectures required a transition to the formation of a training dataset from three-channel images. It has been established that reliable recognition of various alphanumeric inscriptions for all variants of the model architecture was achieved using a training set of sufficient size.

Training a model using three-channel images, especially as the resolution of the training sample increases, is a very resource-intensive process. Therefore, the authors were forced to reduce the number of recognizable classes to 43, abandoning the difference between lowercase and uppercase letters.

An example of the recognition result for alphabetic and digital inscriptions is shown in **Figure 4**. The figure shows the selected areas of interest and recognition results.

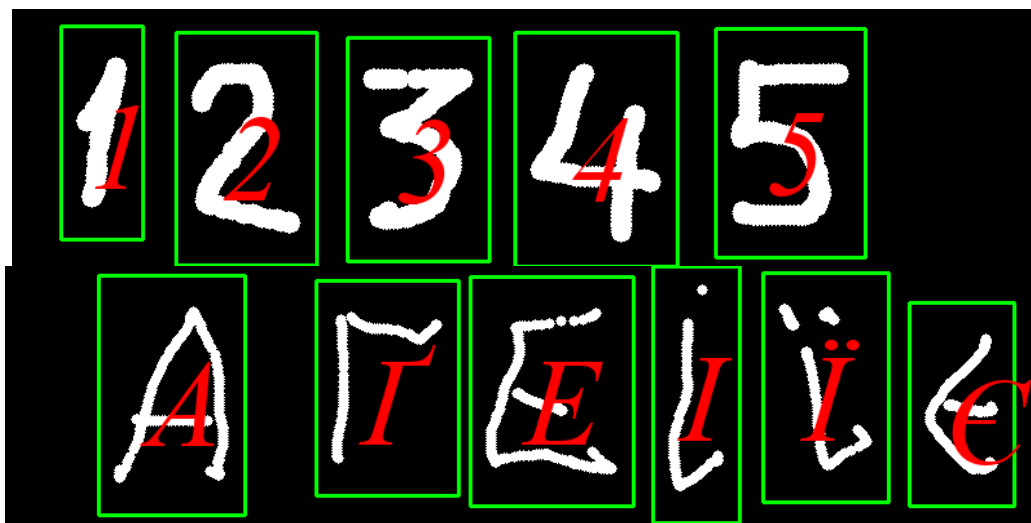


Figure 4: An example of recognition results using the VGG16 neural network (in this case, all letters and numbers are recognized accurately)

Comparing different model architectures, all the options considered showed the recognition accuracy of the test set in the range of 99.2-99.6% when trained on a dataset of sufficient volume. An increase in the number of samples in the training data set for all the considered architectures led to an increase in recognition accuracy. An example of the experimental results for the model with the MobileNet architecture is shown in **Figure 5**.

The recognition accuracy of real inscriptions with an accuracy of 80-90% was achieved with a training sample size of at least 700, and preferably more than 1500 images per class. An example of the experimental results for the model with the MobileNetV2 architecture is shown in **Figure 6**.

Variation of the parameters of the transformations that were used for augmentation also has a noticeable effect on the recognition results: deformation or rotation of the image by more than 10-15% increases the frequency of errors.

Increasing the resolution of the training sample images had little effect on the results due to saturation.

For example, when training a model with the MobileNetV2 architecture on a dataset with a resolution of 32x32 data, the recognition accuracy of the test dataset was 98%, on a dataset with a resolution of 64x64, respectively, 99%, and on a dataset with a resolution of 128x128 - 99.5% (example is shown in **Figure 7**: An example of the influence of the training dataset images resolution on the achieved recognition accuracy (MobileNetV2 and ResNet152v2 architecture).**Figure 7(a)**). However, for other architectures, the result of resolution increase was much less pronounced.

The number of errors in recognition of elements of real inscriptions has changed little: for the model with the ResNet152V2 architecture, an increase in the resolution of images of the training sample led to a decrease in the proportion of erroneous recognition from 18.0% to 11.4% (**Figure 7 (b)**), for models with the MobileNet or MobileNetV2 architecture, it has not practically changed. However, with an increase in the resolution of the training sample, the time spent on training increased quite significantly (by more than an order of magnitude).

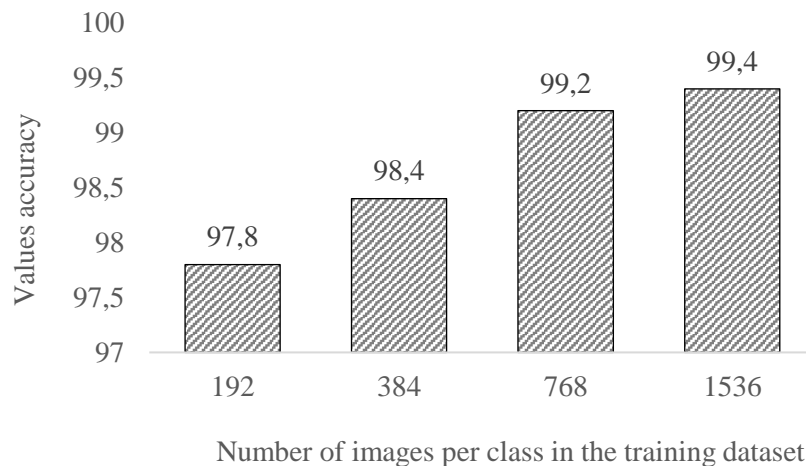


Figure 5: An example of the influence of the size of the training dataset on the achieved recognition accuracy (MobileNet architecture).

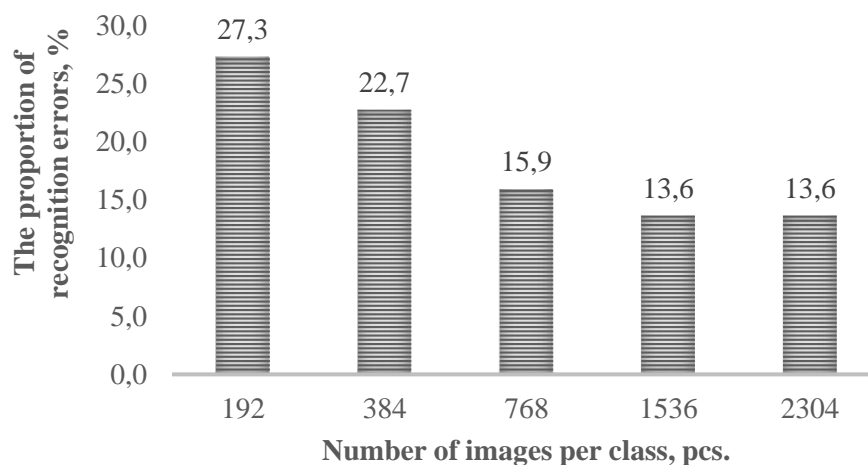


Figure 6: Recognition errors of real inscriptions depending on the size of the training dataset (MobileNetV2 architecture, 32x32x3 dataset images).

When using deep neural networks to recognize letters or numbers, the reliability of recognition of elements of real inscriptions depended primarily on the size of the training dataset.

The recognition accuracy of the test dataset after training all variants of the models was quite high - 97-98% and higher. However, the generation of training datasets of a small size - 300-500 images per class - practically did not provide any reliable recognition.

The use of a model with the InceptionResNetV2 architecture, which requires an image resolution in the training set of at least 75x75x3 (actually, the model was trained on 128x128x3 images), did not lead to a noticeable increase in recognition accuracy.

In general, when comparing the achieved accuracy of recognition of real images and the speed of training the model, the best performance was provided by models of the ResNetV2 or MobileNetV2 family.

Experiments with changing the optimization algorithm from those available in the Tensorflow/Keras package did not give any improvement in the accuracy and reliability of recognition of real samples. Increasing the number of model training epochs above the marked one also did not lead to a change in the results.

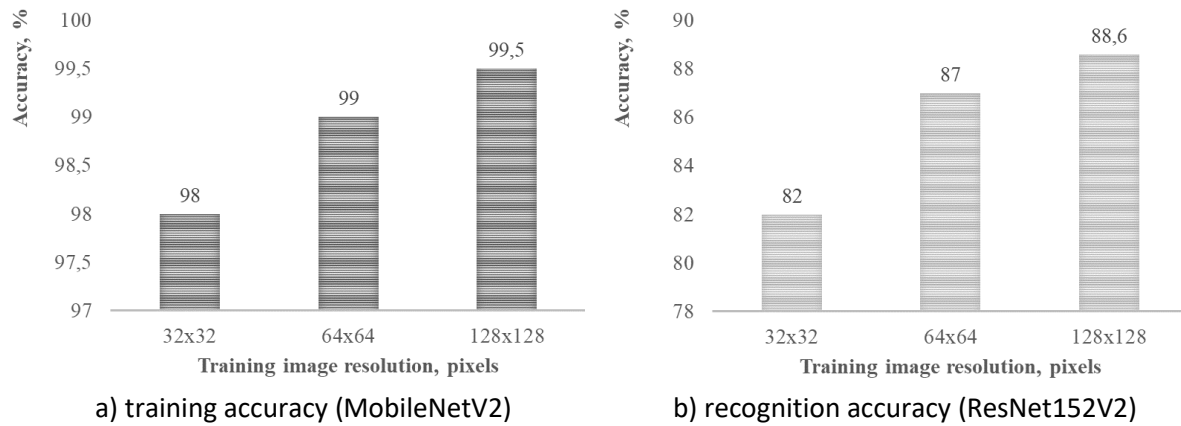


Figure 7: An example of the influence of the training dataset images resolution on the achieved recognition accuracy (MobileNetV2 and ResNet152v2 architecture).

5. Conclusions

In this work, several variants of the architecture of convolutional neural networks for the recognition of isolated handwritten digits and Ukrainian letters are considered.

The results of recognition of various images containing letters and numbers were compared on models with different architectures. It has been established that when training a model on a set of one-dimensional images 28x28, an increase in the number of convolutional layers of a neural network in most cases leads to an increase in the reliability of recognition. Among the options considered, the best accuracy and reliability of recognition was provided by a model with an architecture of the VGG16 type, which included 13 convolutional and three dense layers.

The possibility of learning convolutional neural networks using a synthetic data set built on the basis of handwritten or cursive fonts is shown. The size of the training dataset significantly affects the reliability of character recognition. The data sets used in the work contained from 192 to 2304 samples per class.

The lower limit of the sample size, which provides acceptable recognition accuracy, was 1536 characters per class. Reducing the sample size by reducing the number of samples per class leads to a significant decrease in recognition accuracy (from 90% recognition accuracy of elements of real inscriptions to 40-60% with a 4-fold decrease in sample size). An increase in the volume of the training data set did not provide an increase in the accuracy and reliability of recognition, but led to a significant increase in the training time of the model

An increase in image resolution from 32x32x3 to 128x128x3 of the training dataset in most cases did not lead to an increase in the reliability of real image recognition.

6. References

- [1] A. Chaudhuri, K. Mandaviya, S. K. Ghosh, P. Badelia, Optical Character Recognition Systems for Different Languages with Soft Computing, volume 352 of Studies in fuzziness and soft computing, Springer, 2017. doi: 10.1007/978-3-319-50252-6.
- [2] H. Li, P. Wang, C. Shen, Toward end-to-end car license plate detection and recognition with deep neural networks, IEEE Transactions on Intelligent Transportation Systems 20(3) (2018) 1126-1136. doi:10.1109/TITS.2018.2847291.

- [3] A.Rajavelu, M. T. Musavi, M. V. Shirvaikar, A neural network approach to character recognition, *Neural Networks* 2(5) (1989) 387–393. doi:10.1016/0893-6080(89)90023-3
- [4] J. Bai, Z. Chen, B. Feng, B. Xu, Image character recognition using deep convolutional neural network learned from different languages, in: *IEEE International Conference on Image Processing (ICIP)*, Paris, France, 2014, pp. 2560-2564. doi:10.1109/ICIP.2014.7025518.
- [5] D. S. Maitra, U. Bhattacharya, S. K. Parui, CNN based common approach to handwritten character recognition of multiple scripts, in: *3th International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 1021-1025. doi:10.1109/ICDAR.2015.7333916.
- [6] E. F. Bilgin Taşdemir, Online Turkish Handwriting Recognition Using Synthetic Data, *Avrupa Bilim ve Teknoloji Dergisi* 32 (2021) 649-656. doi:10.31590/ejosat.1039846.
- [7] D. Nurseitov, K. Bostanbekov, D. Kurmankhojayev, A. Alimova, A. Abdallah, R. Tolegenov, Handwritten Kazakh and Russian (HKR) database for text recognition, *Multimedia Tools Appl.* 80 21–23 (2021) 33075–33097. doi:10.1007/s11042-021-11399-6.
- [8] A. Abdelrahman, M. Hamada, D. Nurseitov, Attention-Based Fully Gated CNN-BGRU for Russian Handwritten Text, *Journal of Imaging* 6(12) (2020) 141. doi:10.3390/jimaging6120141.
- [9] Z. Ullah, M. Jamjoom, An intelligent approach for Arabic handwritten letter recognition using convolutional neural network, *PeerJ Computer Science* (2022) 8:e995. doi:10.7717/peerj-cs.995.
- [10] D. Jeevitha, S. Muthu, I. Nila, V. Santhoshi, Handwritten Letter Recognition using Artificial Intelligence. *International Journal for Research in Applied Science and Engineering Technology*, 10 (2022) 2752-2758. doi:10.22214/ijraset.2022.42949.
- [11] L. Gannetion, K. Y. Wong, P. Y. Lim, K. H. Chang, A.F.L. Abdullah, An exploratory study on the handwritten allographic features of multi-ethnic population with different educational backgrounds, *PloS one*, 17(10) (2022) e0268756. doi:10.1371/journal.pone.0268756.
- [12] G. Cohen, S. Afshar, J. Tapson, A. Van Schaik, EMNIST: Extending MNIST to handwritten letters, in: *2017 international joint conference on neural networks (IJCNN)*, 2017, pp. 2921-2926. doi:10.48550/arxiv.1702.05373.
- [13] Y. LeCun, B.E. Boser, J.S. Denker, D. Henderson, R.E. Howard, W.E. Hubbard, L.D. Jackel, Backpropagation Applied to Handwritten Zip Code Recognition, *Neural Computation* 1 (1989) 541-551. doi: 10.1162/neco.1989.1.4.541.
- [14] D. Núñez Fernández, S. Hosseini, Real-Time Handwritten Letters Recognition on an Embedded Computer Using ConvNets, in: *2018 IEEE Sciences and Humanities International Research Conference (SHIRCON)*, Lima, Peru, 2018, pp. 1-4, doi: 10.1109/SHIRCON.2018.8592981.
- [15] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, ImageNet classification with deep convolutional neural networks. *Commun, ACM* 60 6 (2017) 84–90. doi:10.1145/3065386.
- [16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S.E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp.1-9. doi:10.1109/CVPR.2015.7298594.
- [17] A. Baldominos, Y.Saez, P. Isasi, A Survey of Handwritten Character Recognition with MNIST and EMNIST, *Appl. Sci.* 9(15) (2019) 3169. doi:10.3390/app9153169.
- [18] B. Mandal, S. Dubey, S. Ghosh, R. Sarkhel, N. Das, Handwritten Indic Character Recognition using Capsule Networks, in: *2018 IEEE Applied Signal Processing Conference (ASPCON)*, 2018, pp. 304-308, doi:10.1109/ASPCON.2018.8748550.
- [19] K.S. Yadav, K. Anish Monsley, S.A. Barlaskar, N. Ahmad, R.H. Laskar, M.K. Bhuyan, Recognition of isolated characters across different input interfaces using 2D DCNN, in: *TENCON 2021-2021 IEEE Region 10 Conference (TENCON)*, Auckland, New Zealand, 2021, pp. 504-509, doi: 10.1109/TENCON54134.2021.9707451.
- [20] K. He, R.B. Girshick, P. Dollár, Rethinking ImageNet Pre-Training, in: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2018, pp. 4917-4926, doi:10.1109/ICCV.2019.00502.

- [21] W. Albattah, S. Albahli, Intelligent Arabic Handwriting Recognition Using Different Standalone and Hybrid CNN Architectures, *Appl. Sci.* 12 (2022) 10155. doi:10.3390/app121910155.
- [22] Tapotosh Ghosh, Min-Ha-Zul Abedin, Hasan Al Banna, Nasirul Mumenin, Mohammad Abu Yousuf, Performance Analysis of State of the Art Convolutional Neural Network Architectures in Bangla Handwritten Character Recognition, *Pattern Recognit. Image Anal.* 31(1) (2021) 60–71. doi:10.1134/S1054661821010089.
- [23] A. Bhardwaj, R. Singh, Handwritten devanagari character recognition using deep learning - convolutional neural network (cnn) model, *PalArch's Journal of Archaeology of Egypt/Egyptology*, 17(6) (2020) 7965-7984. URL: <https://archives.palarch.nl/index.php/jae/article/view/2203>.
- [24] Duddela Sai Prashanth, R. Vasanth Kumar Mehta, Kadiyala Ramana, Vidhyacharan Bhaskar, Handwritten Devanagari Character Recognition Using Modified Lenet and Alexnet Convolution Neural Networks, *Wirel. Pers. Commun.* 122(1) (2022) 349–378. doi:10.1007/s11277-021-08903-4.
- [25] Aicha Korichi, Slatnia Sihem, Tagougui Najiba, Zouari Ramzi, Aiadi Oussama, Recognizing Arabic Handwritten Literal Amount Using Convolutional Neural Networks. In: *Artificial Intelligence and Its Applications*, 2022, pp. 153-165. doi:10.1007/978-3-030-96311-8_15.
- [26] Hossam Magdy Balaha, Hesham Arafat Ali, Mohamed Saraya, Mahmoud Badawy, A new Arabic handwritten character recognition deep learning system (AHCR-DLS), *Neural Comput. Appl.* 33(11) (2021) 6325–6367, doi:10.1007/s00521-020-05397-2
- [27] Gibrael Al Amin Abo Samra, Hadi Oqaibi, An Optimized Deep Residual Network with a Depth Concatenated Block for Handwritten Characters Classification, *Computers, Materials & Continua* 680 (2021) 1-28. doi:10.32604/cmc.2021.015318.
- [28] D.B. Nurseitov, K. Bostanbekov, M. Kanatov, A. Alimova, A. Abdallah, G. Abdimanap, Classification of Handwritten Names of Cities and Handwritten Text Recognition using Various Deep Learning Models, *arXiv preprint arXiv:2102.04816*, 2021. doi:10.25046/aj0505114.
- [29] O. Vovchuk, M. Kyrychenko, Recognition of Handwritten Cyrillic Letters using PCA, 2019. URL: https://www.researchgate.net/publication/336987544_Recognition_of_Handwritten_Cyrillic_Letters_using_PCA.
- [30] Cyrillic-oriented MNIST. URL: <https://github.com/GregVial/CoMNIST>.
- [31] V. Khavalko, V. Mykhailyshyn, R. Zhelizniak, I. Kovtyk, A. Mazur, Economic efficiency of innovative projects of CNN modified architecture application, in: *CEUR Workshop Proceedings. – 2020. – Vol. 2654 : Proceedings of the International workshop on cyber hygiene (CybHyg-2019) co-located with 1st International conference on cyber hygiene and conflict management in global information networks (CyberConf 2019)*. Kyiv, Ukraine; November 30, 2019, pp. 182–193. URL: <https://ceur-ws.org/Vol-2654/paper14.pdf>.
- [32] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, *CoRR abs/1409.1556* (2014). doi:10.48550/arXiv.1409.1556.
- [33] K. He, X. Zhang, S. Ren, J. Sun, (2015). Deep Residual Learning for Image Recognition, in: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 770-778. doi:10.1109/cvpr.2016.90.
- [34] Kaiming He, X. Zhang, Shaoqing Ren, Jian Sun, Identity Mappings in Deep Residual Networks, in: *European Conference on Computer Vision-2016*, Springer, 2016, pp. 630-645. doi:10.1007/978-3-319-46493-0_38.
- [35] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, *ArXiv abs/1704.04861* (2017), doi:10.48550/arXiv.1704.04861.
- [36] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks, in: *2018 IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510-4520, doi:10.1109/CVPR.2018.00474.

- [37] C. Szegedy, S. Ioffe, V. Vanhoucke, A. A. Alemi, Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, ArXiv abs/1602.07261 (2016), doi:10.1609/aaai.v31i1.11231