https://doi.org/10.15407/csc.2024.01.050
UDC 004.032.26

M.D. SNITKO, Student, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Peremohy Ave, 37, Kyiv, 03056, Ukraine, marsnitko17@gmail.com
IA.V. KHITSKO, PhD, Senior Lecturer, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Peremohy Ave 37, Kyiv, 03056, Ukraine, ORCID: https://orcid.org/0000-0002-6455-8498, khitsko@pzks.fpm.kpi.ua
N.A. RYBACHOK, PhD, Asoc. Professor, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Peremohy Ave 37, Kyiv, 03056, Ukraine, ORCID: https://orcid.org/0000-0002-8133-1148, rybachok@pzks.fpm.kpi.ua

RECOGNITION OF HANDWRITTEN TEXTS ON IMAGES USING DEEP MACHINE LEARNING

The article is devoted to the aspects of using deep machine learning to recognize handwritten text containing letters of the Latin alphabet and numbers. Software has been developed that recognizes handwritten text. A convolutional neural network consisting of 13 layers was trained for 50 epochs on images of 814255 characters taken from the EMNIST dataset. The prediction accuracy was 0.9468, the response rate was 0,9673, the F1-index reached 0.9429, and the average processing time of one image was 1.15 seconds.

Keywords: *neural network, deep machine learning, handwriting recognition, OCR, CNN.*

Introduction and Statement of the Problem

In an age where information technology is rapidly developing, the ability to convert images to text using optical character recognition (OCR) is becoming an important tool in the digitization process. OCR allows computers to convert various types of images that contain text into editable and searchable formats such as TXT or DOC.

The purpose of this work is to develop a neural network and appropriate software for recognizing handwritten texts on images.

For this, it is necessary to solve the following tasks:

- determine the requirements for the neural network and the developed software;
- develop a recognition algorithm, a neural network and the corresponding software;
- analyze the neural network and software.

An Application for Recognizing Handwritten Texts on Iimages Bbased on Deep Machine Learning

Neural Network Requirements. The neural network must recognize handwritten text written in any language that uses the Latin alphabet and numbers.

Цитування: Snitko M.D., Khitsko Ya.V., Rybachok N.A. Recognition of Handwritten Texts on Images Using Deep Machine Learning. *Control Systems and Computers*, 2024, 1, 50—56. https://doi.org/10.15407/csc.2024.01.050

Software Requirements

The software must allow the user to:

- upload images;
- recognize handwritten text on the image;
- save the recognized text.

Algorithm for Recognition of Handwritten Test on Images

This algorithm includes steps such as loading an input image, transposing it, inverting it, character detection and segmentation, neural network analysis, and then text recognition and composition.

Design and Implementation of Neural Networks and Software

To design and implement a neural network and software, it is necessary to perform the following steps:

1. choose a dataset for neural network training (dataset);
2. carry out preliminary image processing (preprocessing);
3. design a neural network (methods);
4. programmatically implement the recognition algorithm with a neural network and the corresponding software (software implementation);
5. conduct training and network testing (results).

The specifics of each step are described below.

1. ***Dataset (Dataset)***

For training the neural network, data was taken from the EMNIST (Extended Modified National Institute of Standards and Technology) dataset [1], which is widely used in machine learning and computer vision. Designed as an extension of the classical MNIST, EMNIST includes not only numbers but also handwritten Latin letters, including both upper and lowercase letters. This makes EMNIST more comprehensive and suitable for tasks that require recognition of not only numbers, but also the entire alphabet. A special feature of EMNIST is its extended variety and number of samples, which provide a high level of challenges for classification and pattern recognition algorithms. The images in EMNIST, like MNIST, are 28×28 pixels in size, but include a much larger number of writing styles, allowing the model to be trained to recognize a wider range of handwritten characters.

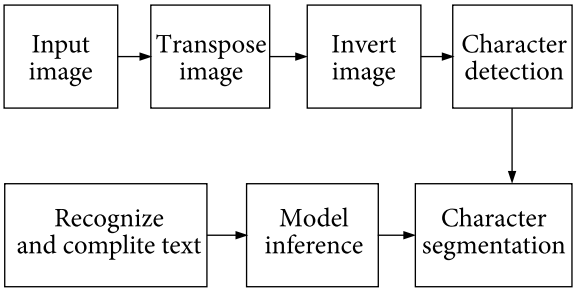

Fig. 1. The sequence of steps of the text recognition algorithm on images

The EMNIST ByClass dataset contains 814.255 character samples that are distributed among 62 unbalanced classes [1]. Each class represents a unique symbol or letter assembled from a variety of sources, including numbers, uppercase and lowercase letters of the Latin alphabet. Class imbalance in this context means that the number of samples in each class is not the same, which creates additional challenges for machine learning algorithms in character classification and recognition, as the models need to learn to identify characters even from underrepresented categories.

2. ***Preprocessing of images***

Image preprocessing is an important step in preparing images for analysis. It includes normalization (such as brightness and contrast correction), Otsu binarization or adaptive binarization, and denoising (application of filters such as median or Gaussian). These methods allow you to correct or reduce the influence of variable factors that can negatively affect character recognition.

Pixel intensity normalization is implemented by converting the original pixel values, which range from 0 to 255, to a range from 0 to 1. Mathematically, this is expressed using the formula:

$$Xnorm = X/255.$$

This approach ensures homogeneity of data and contributes to more effective training of the model [2].

The next step is to center the data, where the average value of each pixel is shifted to zero. This is done using a formula:

$$X = Xcenter - \mu,$$

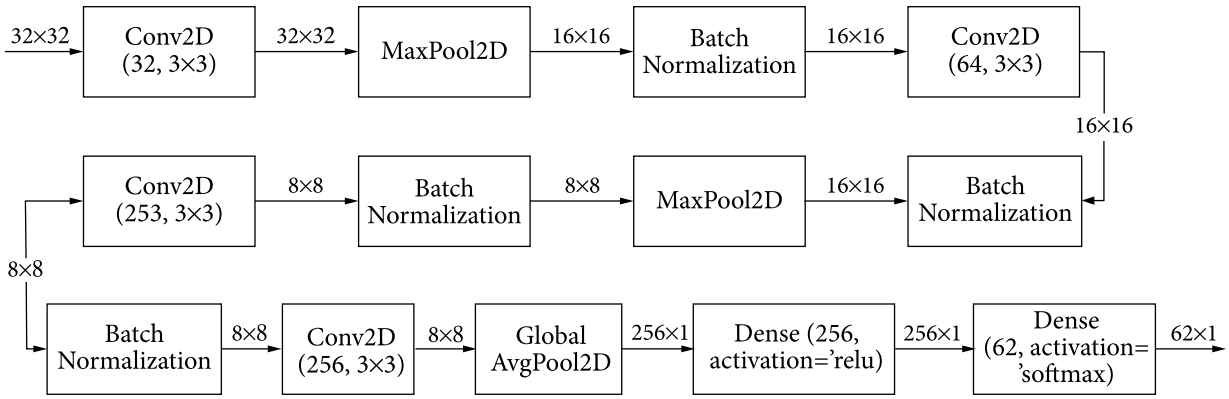

Fig. 2. Convolutional network architecture

where μ is the average pixel value of the entire dataset. Centering the data helps to reduce the internal bias of the model during training.

Data standardization is done by scaling the pixel values so that they have a standard deviation of about 1. This process can be expressed through the formula:

$$Xstand = (X-\mu)/\sigma,$$

where σ is the standard deviation of pixel values.

Image binarization is an important part of the processing process because it converts images to black and white, making them easier to analyze. This step is particularly effective in cases where the images have different levels of illumination or contrast [3].

Noise removal using median and Gaussian filters is necessary to improve the quality of images, which directly affects the accuracy of character recognition. This approach allows the model to focus on the key features of the image.

Contrast enhancement is also an important step because it improves the distinction between text and background, which is important for accurate recognition.

3. *Methods.* CNNs are noted for their high accuracy in recognition and classification tasks, outperforming traditional deep learning approaches in many areas [4—11]. From automatic face recognition to social media image classification, CNNs demonstrate their ability to adapt and learn from complex data sets, providing accurate and reliable results.

The advantages of using CNN are as follows:

- feature extraction: CNNs naturally identify visual patterns and extract local features important for handwritten character recognition, even with changes in size, style and noise;
- scalability: CNNs are effectively trained on large data sets, which contributes to increasing accuracy as the amount of training data increases;
- robustness: CNNs can be trained to be robust against distortions, blurs and noises commonly found in scanned documents.

The developed architecture of the neural network is presented in Fig. 2.

Characteristics of a Neural Network:

1. Conv2D (32 filters with kernel size 3). This first layer is convolutional. It uses 32 3×3 filters (or kernels). This layer is designed to extract low-level features such as edges and corners from input images. Mathematically, this process can be described as:

$$f\,ij = \sum\sum I(i+u, j+v)Kuv\ V-1\ v=0\ U-1\ u=0,$$

where I is the input image, K is the filter kernel, and f is the output characteristic.

2. MaxPool2D. This is the maximum convolution layer that follows the first convolutional layer. It reduces the spatial dimensions (i.e., height and width) of the input feature maps, effectively reducing the size of the input representation and reducing the computational burden.

3. BatchNormalization. This layer normalizes the activations of the previous layer on each batch,

that is, it applies a transformation that keeps the mean activation close to 0 and the standard deviation of the activation close to 1. Batch normalization helps to speed up the training process and reduce sensitivity to network initialization.

4. Conv2D (64 filters with kernel size 3). This is the second convolutional layer with 64 filters, each also 3×3 in size. This layer further processes the features extracted by previous layers.

5. BatchNormalization. Another batchwise normalization layer that follows the second convolutional layer.

6. MaxPool2D. A second layer of maximum collapse to further reduce feature maps.

7. BatchNormalization. Another layer of normalization by parties.

8. Conv2D (256 filters with kernel size 3). This is a deeper convolutional layer with 256 filters, indicating the increasing complexity of the features extracted as we move deeper into the network.

9. BatchNormalization. A batchwise normalization layer that follows the third convolutional layer.

10. Conv2D (256 filters with kernel size 3). Another convolutional layer with 256 filters, similar to the previous one, for further feature extraction.

11. GlobalAvgPool2D. This layer performs global mean convolution on the feature maps. Instead of aligning feature maps and passing them through fully connected layers, global average convolution directly computes the average of each feature map, reducing the total number of parameters and computations in the network [7].

12. Dense (256 units with ReLU activation). This is a fully connected layer with 256 units. A ReLU (Rectified Linear Unit) activation function is used, which introduces nonlinearity into the network, allowing it to learn more complex patterns.

13. Dense (62 units with softmax activation). The last layer is a fully connected layer with 62 units where the softmax activation function is used. This layer is used for multiclass classification tasks where 62 units correspond to 62 different classes. The softmax function outputs the probability distribution for these 62 classes.

Neural Network Training:

The model is trained using the stochastic gradient descent method [4]. This method is a fundamental algorithm for training neural networks, including convolutional neural networks. SGD optimizes the model weights by adapting them in such a way as to minimize the loss function. The basic idea of SGD is to gradually adjust the model weights based on the gradients of the loss function.

The model was trained for 50 epochs, indicating a balance between needing sufficient time to train the network and avoiding overtraining, which could lead to overtraining. Using a batch size of 100 samples provides an efficient use of computing resources, allowing the model to process a sufficient amount of data in one step while maintaining variability in the training process. Applying a 10% percentage of the data for validation ensures that the model is regularly evaluated on previously unseen data, which is important to test its ability to generalize and prevent overtraining. Additionally, learning optimization techniques such as early termination, monitoring the learning process through callbacks, and dynamically reducing the learning rate were used in the model.

Evaluation of the Efficiency of the neural network:

The following metrics are used to evaluate the effectiveness of the model:

- accuracy;
- feedback;
- F1 measure;
- average processing time of one image.

Accuracy is defined as the ratio of the number of correctly classified samples to the total number of samples [8].

Feedback indicates the ability of the model to detect all relevant cases in the dataset, regardless of whether they were correctly classified [8]. A high response means that the model effectively identifies most or all of the handwritten characters present in the test data

The F1-measure is a harmonic mean between accuracy and response that provides a balanced assessment of these two important metrics [8].

The average processing time of one image is calculated based on the processing time for each input sample.

4. *Software implementation.* The web application has a modular architecture. According to the

application requirements described above, it contains the following modules:

- The data processing module, which prepares the data for further processing by the machine learning module, transforming and standardizing it to the appropriate format.
- A machine learning module responsible for building, training and evaluating deep learning models.
- A GUI module that provides an interface for entering handwriting and viewing recognition results.
- Handwritten character recognition module, which passes data to the machine learning module and returns recognition results.
- A visualization module used to analyze and evaluate the effectiveness of the machine learning module.
- Test modules for testing application functions.

The Python programming language was used to implement the convolutional neural network and software due to its easy and intuitive syntax and access to libraries for working with large volumes of data and for image processing. The TensorFlow framework, Keras, NumPy, Pillow, OpenCV, idx-2numpy, Scikit-learn, Matplotlib, Tkinter libraries were used.

The Flask-based web interface allows users to upload images and instantly receive recognition results. The results are displayed to users in an understandable format, after which they can be used for further processing or analysis.

5. *Results.* For the experimental comparison, the method of optical character recognition using the LeNet-5 neural network, which is traditionally a standard in the classification of images of handwritten digits MNIST, was chosen. During the experiment, it was found that the average time of processing one image by our system is 1.15 seconds, compared to 2.40 seconds for LeNet-5. The model's recognition accuracy was 94.68%, while LeNet-5 showed an accuracy of 93,5% on the same data set.

The following metric values were obtained during neural network testing:

- the accuracy of the neural network was determined at the level of 0.9468,
- the response rate was 0.9673,
- F1-measure was 0.9429,
- the average time of processing one image was 1.15 seconds.

Conclusions

The developed handwriting recognition system, which is based on a neural network with CNN architecture, was implemented using TensorFlow 2.3 and trained on the MNIST dataset. The system is able to recognize individual Latin letters or words in images.

Once trained, the neural network can be saved and used for handwriting recognition in a variety of applications, including mobile and web applications.

The architecture of the software system is flexible and can be expanded by adding new modules that will expand the system's functions.

The materials of the article will be useful in solving the problems of classification of images using neural networks.

REFERENCES

1. LeCun, Y., Cortes, C., Burges, C.J.C. "The MNIST Database of Handwritten Digits" [online]. Available at: <http://yann.lecun.com/exdb/mnist/> [Accessed 06 Dec. 2023].
2. Digital Image Processing / Rafael C. González; Richard Eugene Woods (ed.). Prentice Hall, 2007, pp. 85, ISBN 978-0-13-168728-8.
3. Chauhan, S., Sharma, E., Doegar, A. (2016). "Binarization Techniques for Degraded Document Images. A Review". 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO).
4. Krizhevsky, A., Sutskever, I., Hinton, G.E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems 25 (NIPS 2012), pp. 1097—1105.

5. Mouton, C., Myburgh, J.C., & Davel, M.H. (2020). "Stride and Translation Invariance in CNNS". In Southern African Conference for Artificial Intelligence Research, Vol. 1342, Cham: Springer International Publishing, pp. 267—281. arXiv:2103.10097. doi:10.1007/978-3-030-66151- 9_17.
6. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Rabinovich, A. (2015). "Going deeper with convolutions". In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1—9.
7. Nikitha, A., Geetha, J., JayaLakshmi, D.S. (2020). "Handwritten text recognition using deep learning". Int. Conf. on Recent Trends on Electronics, Information, Communication & Technology (RTEICT). pp. 388—392. https://doi.org/10.1109/RTEICT49044.2020.9315679.
8. Nurseitov, D., Bostanbekov, K., Kanatov, M., Alimova A., Abdallah, A., Abdimanap, G. (2020). "Classification of handwritten names of cities and Handwritten text recognition using various deep learn". Advances in Science, Technology and Engineering Systems Journal. Vol. 5, No. 2, XX—YY, https://doi.org/10.25046/aj0505114.
9. Khitsko, Ya.V., Snitko, M.D. (2023). "Sposib ta prohramne zabezpechennya dlya rozpiznavannya tekstu na zobrazhennyakh". Zbirnyk tez XVI konferentsiyi molodykh vchenykh «Prykladna matematyka ta komp'yutynh», Kyiv, pp. 627—631 (In Ukrainian).
10. Powers, D.M.W. (2011). "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation". Journal of Machine Learning Technologies, 2 (1), pp. 37—63.
11. Manokhyn, A.V., Rybachok, N.A. (2021). "English Accent Recognition Using Deep Machine Learning". Control Systems and Computers. № 4. pp. 52—59. https://doi.org/10.15407/csc.2021.04.028

Received 15.02.2024

ЛІТЕРАТУРА

1. LeCun Y., Cortes C., Burges C.J.C. The MNIST Database of Handwritten Digits. URL: http://yann.lecun.com/exdb/mnist/ [дата звернення: 06.12.2023].
2. Digital Image Processing / Rafael C. González; Richard Eugene Woods (ed.). Prentice Hall, 2007, pp. 85.
3. Chauhan, S., Sharma E., Doegar A. "Binarization Techniques for Degraded Document Image. A Review". 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). 2016.
4. Krizhevsky A., Sutskever I., Hinton G.E. ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems 25 (NIPS 2012). 2012. pp. 1097—1105.
5. Mouton C., Myburgh J.C., Davel M.H. Stride and Translation Invariance in CNNS. In Southern African Conference for Artificial Intelligence Research, Vol. 1342, Cham: Springer International Publishing. 2020. pp. 267—281. arXiv:2103.10097. doi:10.1007/978-3-030-66151- 9_17.
6. Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Rabinovich A. Going deeper with convolutions". In Proceedings of the IEEE conference on computer vision and pattern recognition. 2015. pp. 1—9.
7. Nikitha A., Geetha J., JayaLakshmi D.S "Handwritten text recognition using deep learning". Int. Conf. on Recent Trends on Electronics, Information, Communication & Technology (RTEICT). 2020. pp. 388—392. https://doi.org/10.1109/RTEICT49044.2020.9315679.
8. Nurseitov D., Bostanbekov K., Kanatov M., Alimova A., Abdallah A., Abdimanap G. "Classification of handwritten names of cities and Handwritten text recognition using various deep learn". Advances in Science, Technology and Engineering Systems Journal. 2020. Vol. 5, No. 2, pp. XX—YY. https://doi.org/10.25046/aj0505114.
9. Хіцко Я.В., Снітко М.Д. Спосіб та програмне забезпечення для розпізнавання тексту на зображеннях. Збірник тез XVI конференції молодих вчених «Прикладна математика та комп'ютинг», Київ. 2023. С. 627—631.
10. Powers D.M.W. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. Journal of Machine Learning Technologies, 2011, № 2 (1). pp. 37—63.
11. Manokhin A.V., Rybachok N.A. English Accent Recognition Using Deep Machine Learning. Control Systems and Computers. 2021. № 4. С. 52—59. https://doi.org/10.15407/csc.2021.04.028.

Надійшла 15.02.2024

М.Д. Снітко, студентка, Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
03056, м, Київ, просп. Перемоги, 37, Україна,
marsnitko17@gmail.com
Я.В. Хіцко, кандидат технічних наук, стрший викладач, Національний технічний
університет України «Київський політехнічний інститут імені Ігоря Сікорського»
ORCID: https://orcid.org/0000-0002-6455-8498,
03056, м, Київ, просп. Перемоги, 37, Україна,
khitsko@pzks.fpm.kpi.ua
Н.А. Рибачок, кандидат технічних наук, доцент, Національний технічний університет
України «Київський політехнічний інститут імені Ігоря Сікорського»
ORCID: 0000-0002-8133-1148, 03056, м, Київ, просп. Перемоги, 37, Україна,
rybachok@pzks.fpm.kpi.ua

РОЗПІЗНАВАННЯ РУКОПИСНИХ ТЕКСТІВ НА ЗОБРАЖЕННЯХ ІЗ ВИКОРИСТАННЯМ ГЛИБИННОГО МАШИННОГО НАВЧАННЯ

Вступ. Розпізнавання рукописного тексту є актуальною задачею, рішення якої сприяє створенню нових програмних систем автоматизації та оптимізації багатьох процесів у різних сферах людського життя. У статті висвітлено аспекти використання глибокого машинного навчання для розпізнавання рукописного тексту, що містить літери латинського алфавіту та цифри.

Мета статті. Метою даної роботи є розроблення нейронної мережі та відповідного програмного забезпечення для розпізнавання рукописного тексту з латинськими літерми та цифрами.

Методи. Обрано датасет для навчання нейронної мережі. Здійснено початкове оброблення даних, яке полягає у нормалізації, бінаризації за методом Оцу видаленні шуму. Спроєктовано згорткову нейромережу, що складається із 13 шарів. Нейромережа та відповідне ПЗ реалізовано програмно. Здійснено тренування мережі протягом 50 епох на множині 814255 символів, взятих із датасету *EMNIST*.

Результат. Досягнуто точність прогнозування 0,9468, частота відповідей склала 0,9673, показник *F*1 досяг 0,9429, середній час обробки одного зображення становить 1,15 секунди. Розроблено програмне забезпечення для розпізнаванння рукописного тексту із використанням латинського алфавіту та цифр.

Висновки. Нейромережа може бути використана для розпізнавання рукописного тексту в інших додатках, включаючи мобільні та вебзастосунки. Архітектура програмної системи є гнучкою і її можна розширювати додаванням нових модулів, що розширять функції системи. Матеріали статті будуть корисними при вирішенні задач класифікації зображень із використанням нейронних мереж.

Ключові слова: *нейронна мережа, глибоке машинне навчання, розпізнавання рукописного тексту, OCR, CNN.*