STUDENT SESSION

HANDWRITTEN DIGIT RECOGNITION USING CONVOLUTIONAL NEURAL NETWORKS AND BIG DATA PROCESSING

Pavle Matijašević*,
[0009-0007-4647-2555]
Miloš Mravik
[0000-0001-5442-3998]

Singidunum University,
Belgrade, Serbia

Abstract:
In this paper, we present a handwriting recognition system that combines neural networks and big data processing. We use a custom convolutional model, and optimized data augmentation techniques. To manage and process large volumes of data more efficiently, we relied on Apache Spark. Additionally, a user-friendly API has been created to enable real-time recognition of handwritten digits. Evaluation of the system on a custom dataset shows extremely high accuracy, with precision greater than 98% on the test data.

The main challenge now is to process large datasets, as well as manage different handwriting styles and ensure that the models perform accurately in real-world scenarios. The main contributions of this work are implementations of an efficient convolutional model for recognizing handwritten digits, a system for processing big data in a distributed environment, web API and user interface for real-time handwriting recognition, model error analysis and suggestions for further improvements.

Keywords:
Machine Learning, Big Data, Digit Recognition, Artificial Intelligence.

INTRODUCTION

Handwritten digit and symbol recognition is one of the most important areas in image processing and artificial intelligence itself, with wide applications in the areas of document digitization, automatic test scoring, and signature authentication. Traditional methods such as optical character recognition and hidden Markov models have shown limitations in recognizing handwritten digits and symbols due to large variability in writing styles. Handwritten digit recognition remains a challenge due to variations in handwriting, ambiguous symbols, and poor scan quality, and even a 1% recognition error may lead to serious issues in critical industries such as banking or healthcare. [1] With the development of deep neural networks, and with an emphasis on convolutional neural networks, we have seen significant performance improvements in handwritten digit recognition problems. Convolutional models can automatically recognize key features of handwritten characters, achieving high recognition accuracy. CNN architectures such as LeNet-5 have achieved state-of-the art performance on digit tasks, with error rates as low as 0.7% on the MNIST dataset when augmented data is used. [2]

Correspondence:
Pavle Matijašević

e-mail:
matijasevic.pavle99@gmail.com

DOI: 10.15308/Sinteza-2025-531-535

A breakthrough in the use of CNNs for large-scale image classification was achieved with the introduction of AlexNet [3], a deep CNN trained on the ImageNet dataset with over 1.2 million labelled images. This model drastically reduced the top 5 error rate from 26.2% (previous best) to 15.3%, demostrating the power of deep architectures in visual recognition.

2. METHODOLOGY AND SYSTEM IMPLEMENTATION

2.1. GENERAL SYSTEM OVERVIEW

The handwriting recognition system consists of four main components:

- Collection and augmentation: Generating a large dataset of handwritten digits using augmentation techniques.
- Training Convolutional Neural Networks (CNN): Developing and Optimizing a Model for Handwritten digit recognition.
- Distributed data processing using Apache Spark: Efficient processing of large datasets before model training.
- Real-time handwriting recognition API: Fast API implementation is a service that allows users to recognize digits in real time via a web interface.

2.2. DATA PREPARATION AND PROCESSING

To train the model, we created a custom dataset of handwritten digits that is like the MNIST dataset but has additional variations in handwriting. The dataset was manually generated, and further augmented with augmentation techniques:

- Rotation (15, 7, 5, 20, 10)
- Scaling (enlarging and reducing size)
- Translation in arbitrary directions
- Adding noise to simulate handwriting variability
- Blurring to simulate poorly scanned images
- Changing brightness and contrast

Augmentation was performed using the Pillow library, while big data processing was done through Apache Spark for scalable processing.

2.3. THE ARCHITECTURE OF THE CNN MODEL

To recognize handwritten digits, we used a convolutional neural network (CNN) with the following architecture:

- Input layer: 28x28 pixel grayscale image.
- Convolutional layers: Two convolutional layers with a ReLU activation function
- Pooling layer: Maximum pooling (2x2) for dimensionality reduction.
- Fully connected layers: Two-layer with Softmax output for digit classification
- Dropout layers: Regularization of the model to reduce overfitting.

The network incorporated several innovations, including the use of ReLU activations, dropout to reduce overfitting, overlapping pooling, and parallel training on GPUs. These architectural and training choices made it possible to successfully train a network with 60 million parameters and achieve state-of-the-art results on complex visual tasks [3]. PyTorch was used to train the model, optimization was performed using the Adam optimizer, and the loss function was Cross-Entropy Loss.

2.4. DISTRIBUTED PROCESSING USING APACHE SPARK

Apache Spark was used to process large datasets before training the model, which allows for: Efficient loading and filtering of data from a large set of images, parallel augmentation using Spark RDD and PySpark functions, and faster data preprocessing before training the model.

This approach enables scalable management of large data sets and reduced processing time.

2.5. REAL-TIME HANDWRITING RECOGNITION API

The system allows users to recognize handwritten digits in real time via Fast API service that:

- Receives an image of handwritten digit via a web application.
- Converts the image to a format suitable for the CNN model.
- Runs a prediction using the trained CNN model and returns the result
- Allow visualization of predictions on the frontend application.

The user interface is implemented using Vue.js and Bootstrap, while the API allows for the fast and accurate request processing.

2.6. MODEL PERFORMANCE EVALUATION

The trained model was tested on a test dataset, with the key metrics being:

- Model accuracy
- Precision and responsiveness have high values on all digit classes
- Error analysis

3. EXPERIMENTS AND EVALUATION OF RESULTS

For training and learning the model, we used a dataset of 100,000 images of handwritten digits, including original and augmented images. To assess the quality of the model, we used standard classification metrics such as accuracy, precision, response, f1 value, and confusion matrix.

Similar evaluations conducted in recent studies have confirmed that CNN models consistently outperform other machine learning methods in terms of accuracy, precision, and robustness. In a 2024 study, the CNN model achieved 99.31% test accuracy, surpassing MLP and SVM models, which scored 96.89% and 95.31%, respectively. [1]

Table 1 shows the results of our CNN model on the test set:

Table 1. Results of the CNN model on the test set

	Precision	Recall	F1-score
0	1.0	1.0	1.0
1	0.89	1.0	0.94
2	1.0	0.9	0.95
3	0.95	1.0	0.98
4	1.0	1.0	1.0
5	0.89	1.0	0.94
6	1.0	1.0	1.0
7	1.0	1.0	1.0
8	1.0	1.0	1.0
9	1.0	0.84	0.91
Accuracy			0.98
Macro average	0.97	0.97	0.97
Weighted average	0.98	0.98	0.98

Figure 1 shows an example of a successfully classified digit:

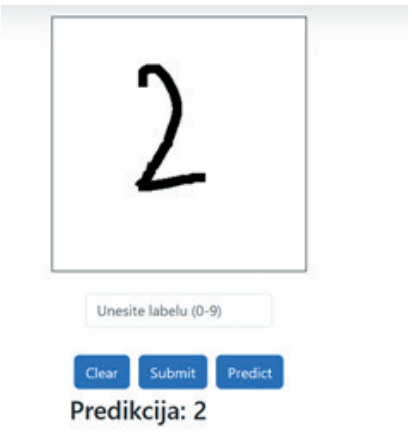

Figure 1. Successful classified digit

In Figure 2, we show example of unsuccessfully classified digit:

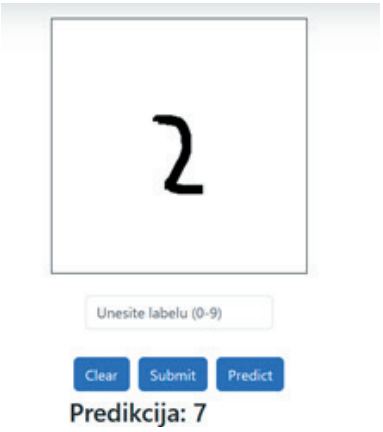

Figure 2. Unsuccessful classified digits

In Table 2, we show a comparison of our model with classical OCR methods and standard deep learning models:

Table 2. Model comparison results

Column 1	Column 2
Model	Accuracy
Tesseract OCR	86.5%
LaNet 5	97%
Our model	98.2%

By analysing the incorrect predictions, we identified several examples where the model had problems. Generally, the model fails in the following cases: Badly written digits, overlap between numbers, and too dark or bright images.

These results are consistent with previous studies, where LaNet-5 also outperformed classical methods, achieving 99.3% accuracy when trained with augmented data [2].

Tesseract OCR, although widely used due to its open-source nature and historical significance, achieves a significantly lower accuracy compared to CNN-based methods. The original Tesseract v2.0 achieved character-level error rates ranging from 1.61% to 2.22% on standard test sets, which is notably worse than CNN-based models like LeNet-5 or the proposed system in this work [4].

Techniques such as ridge regression-based feature selection have also shown great potential in predictive modelling tasks by improving generalization and reducing overfitting. [5] In addition, CNN-based systems have demonstrated remarkable success in healthcare applications, where accurate recognition of medical patterns is critical. [6]

The achieved model accuracy is 98.2%, which far surpasses classic OCR system like Tesseract. The model was trained on a dataset of 100,000 images, including augmented versions, which allowed it to better generalize different users handwriting. The most common errors occurred with visually similar digits (3-5, 1-7), while the other digits had the lowest error rates. Apache Spark significantly accelerated data processing and enabled efficient management of large datasets. Fast API enabled fast image processing and real-time prediction retrieval.

Although our model achieves high accuracy, we have identified certain challenges that may be the subject of future research:

- Visual similarity of digits: Some digits are very similar to each other (3 and 5), which can cause errors even in more advanced CNN models.
- Different handwriting styles: The model is trained on a set of handwritten digits but may also have difficulty with highly stylized or illegible digits.
- Extreme lighting conditions: Images that are too bright or dark can make recognition difficult, despite the normalization techniques applied.

Future research will focus on extending the dataset to include letters and symbols, using advanced neural networks such as the Transformer architecture for improved analysis of handwriting sequences, as well as combining convolutional neural networks (CNNs) with recurrent neural networks (RNNs). Additionally, efforts will be directed toward real-time optimizations and the development of a mobile application. In addition to Transformer-based models, deep residual networks offer a powerful and scalable approach to improving handwriting recognition. ResNets alleviate the degradation problem that occurs with increasing network depth by introducing identity shortcut connections, enabling the training of models with over 100 layers and achieving a top 5 error of 3.57% on the ImageNet test set [7].

4. CONCLUSION

The model based on convolutional neural networks achieved very good results and proved to be reliable in practical use. Most of the errors happened when the digits looked like each other, for example, 3 and 5, or 1 and 7. This setup also includes an API that responds in real time, and several data augmentation methods were applied to help the model perform better and generalize across different handwriting styles.

Similar techniques that focus on generalization and robustness, such as regularized feature selection, have shown success in reducing overfitting and improving model performance, especially when compared to traditional approaches like SVM or kNN.

Moreover, architectures like ours have been successfully applied in real-world domains, including the medical field, where convolutional models are used for precise pattern recognition and decision support based on visual data.

All of this shows that combining deep learning with scalable processing tools can produce fast and accurate systems, even for tasks that involve a lot of variation, like recognizing handwritten digits.

REFERENCES

[1] N. F. A. M. Raheem, "Handwritten digit recognition using machine learning," *Journal of Theoretical and Applied Information Technology*, 2024.

[2] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol.86, no. 11, pp. 2278-2324, 1998.

[3] A. Krizhevsky, I. Sutskever, G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, 2012.

[4] R. Smith, "An Overview of the Tesseract OCR Engine," in *Proceedings of the Ninth International Conference on Document Analysis and Recognition,* 2007.

[5] M. Mravik, T. Vetriselvi, K. Vankatachalam, M. Sarac, N. Bacanin, S. Adamovic, "Diabetes Prediction Algorithm Using Recursive Ridge Regression L2," *Computers, Materials* & *Continua*, 2022.

[6] M. Sarac, M. Mravik, D. Jovanovic, I. Strumberger, M. Zivkovic, N. Bacanin, "Intelligent diagnosis of coronavirus with computed tomography images using a deep learning model," *Journal of Electronic Imaging*, vol. 32, no. 2, 2022.

[7] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.