Scientific Journal of Informatics
Vol. 10, No. 3, Aug 2023

p-ISSN 2407-7658 http://journal.unnes.ac.id/nju/index.php/sji e-ISSN 2460-0040

Capital Optical Character Recognition Using Neural Network Based on Gaussian Filter

Erna Zuni Astuti[1*], Christy Atika Sari[2], Mutiara Salsabila[3], Hendra Sutrisno[4], Eko Hari Rachmawanto[5], Mohamed Doheir[6]

[1,2,3,4,5]Department of Informatics Engineering, Faculty of Computer Science, Universitas Dian Nuswantoro, Indonesia
[6]Department of Computer Science, University Geomatrika, Malaysia

Abstract.
Purpose: As digital technology advances, society needs to convert physical text into digital text. There are now many methods available for doing this. One of them is OCR (Optical Character Recognition), which can scan images [1]–[4] containing writing and turn them into digital text, making it easier to copy written text from an image. Text recognition in images is complex due to variations in text size, color, font, orientation, background, and lighting conditions.
Methods: The technique of text recognition or optical character recognition (OCR) in images can be done using several methods, one of which is a neural network or artificial neural network. The artificial neural network method can help a computer make intelligent decisions with limited human assistance. Intelligent decisions can be made because the neural network can learn and model the relationship between nonlinear and complex input and output data. In this research, the scaled conjugated gradient is applied for optimization. SCG is very effective in finding the minimum value of a complex function, but it takes longer than some other optimization algorithms. Result/Findings: The dataset used is an image with a size of 28 x 28 which is changed in dimension to 784 x 1. This research uses 4000 epochs and obtained the best validation result at epoch 3506 with a value of 0.0087446.
Results: From the statistical test results, the effect of perceived usefulness on ease of use has the highest level of influence, obtaining a test value of 3.6. Furthermore, the effect of the attitude towards using on the behavioral intention to use has the lowest level of influence, which obtained a test value of 1.2.
Novelty: In this article, Gaussian filter is used as feature extraction to improve yield. Character detection results using a Gaussian filter are known to be almost 10% higher than those using only a neural network. The result with the Neural Network alone is 82.2%, while the Neural Network-Gaussian Filter produces 92.1%.

Keywords: Digital test, Neural network, Optical character recognition, Test recognition
Received March 2023 / **Revised** May 2023 / **Accepted** June 2023

INTRODUCTION

As digital technology advances, society needs printed text such as books, journals, and documents to be converted into digital form in order to increase efficiency in automatically converting documents into digital files. Text recognition in images is complex due to variations in text size, color, font, orientation, background, low image resolution, and lighting conditions [1]–[4]. In addition, similar image shapes greatly affect text recognition accuracy. The problem that arises is how to transfer the printed text to digital text form. A system to recognize text in images [5]–[10] is known as an optical character recognition (OCR) system. OCR was developed to address this problem. Optical character recognition (OCR) is a technology that allows computers to read text contained in images or printed documents [4], [11]. OCR is very useful for converting image documents or text images into editable text. OCR can be used to read text contained in images, such as in photos, books, or documents [11], [12]. OCR works by scanning an image or document and recognizing the letters contained in the image. After the letters are recognized, OCR uses complex algorithms to convert the letters into digital text that can be processed by a computer. This is done by comparing the recognized letters with a database of letters already recognized by OCR [1], [13]–[15]. If OCR cannot find a letter that matches its known dataset, it will try to guess the letter with a shape similar with its database [9], [16], [17].

*Corresponding author.
Email addresses: erna.zuni.astuti@dsn.dinus.ac.id (Astuti)
DOI: [10.15294/sji.v10i3.43438](https://doi.org/10.15294/sji.v10i3.43438)

This research discusses an OCR-based system using the neural network or artificial neural network method. A neural network or artificial neural network is a set of algorithms used to imitate the workings of the human nervous system. Artificial neural networks [18] are one of the deep learning algorithms, which become more intelligent over time. Artificial neural networks [4], [13], [15], [19]–[22]] can learn on their own through experience and can become more accurate as more data is provided. A neural network is a system that processes information by imitating the workings of the human brain, allowing a computer to understand various things and make decisions like a human think [7], [8], [23], [24]. Neural networks can be used for a variety of tasks, such as pattern recognition [25], [26]. OCR (Optical Character Recognition) can use neural network or artificial neural network methods to improve accuracy in identifying letters in images or documents. Artificial neural networks can help OCR by processing the data given and finding patterns that can be used to identify those letters [10], [16], [27]. Using a neural network, OCR can be more accurate in identifying letters that are difficult to distinguish, such as the letters "D" and "O." The advantage of artificial neural networks is their ability to recognize patterns from previously taught images. In previous research, the same topic was discussed and can be used as a reference for comparison or to add further discussion about the topic being researched. In this research, several journals discussing the same topic, text recognition, were used as a comparison. The following are the journals by Khandokar [15] in 2021 using Convolutional Neural Network (CNN) to recognize the characters based on NIST dataset. This experiment yield 92.91% using 80:20 for training and testing data form 1000 image, while accuracy only 65.32% using 200 datasets. Using segmentation, clipping, pre-processing, and feature extraction, the results of this study might be optimized by combining or replacing extraction features. Another research by Susanto [28] in 2021 has Javanese alphabet classification using simple machine learning by K-Nearest Neighbor (KNN) and optimized using cropping, median filtering, otsu thresholding and HOG feature extraction. The highest accuracy is 98,5% in K=1.

The concept of artificial neural networks began in the 1940s and 1950s, when researchers began to understand how the brain processes information by studying the structure and function of the brain. In 1943, Warren McCulloch and Walter Pitts published a paper proposing that neurons in the brain could be represented as simple logic gates that accept inputs and produce outputs based on a set of rules [7], [18], [21], [24], [29]. Announced. This idea led Frank Rosenblatt to develop the first artificial neural network, the so-called perceptron, in the 1950s. A perceptron was a simple neural network consisting of a single layer of artificial neurons that could be trained to recognize patterns in data. However, perceptrons were limited in their ability to recognize only linearly separable patterns, limiting their ability to solve more complex problems. In the 1980s, the introduction of backpropagation, a learning algorithm that adjusts the weights and biases of neural networks to improve their performance, revived the development of artificial neural networks [30]. This has led to the creation of multi-layer neural networks (also known as deep learning networks) that can learn and recognize more complex patterns in data. Artificial neural networks are used today in a variety of applications, including image and speech recognition, natural language processing, and machine learning. It has also played a key role in the development of self-driving cars and other emerging technologies. The main purpose of this journal is to create a text recognition application that can facilitate the process of converting printed images or documents into editable digital text. The text recognition application can be useful for converting image documents or text images into editable text. The text recognition application can also be used to read text contained in images, such as book photos or documents. This can help facilitate document archiving.

METHODS

Research on text recognition in images or OCR (Optical Character Recognition) is very important because it can help in processing and accessing the information contained in photos by converting it into machine-readable text [14], [31]–[34]. OCR can be used in various fields such as education, business, and industry. In the education field, OCR can help in processing and storing documents contained in images such as textbooks, lecture notes, and other documents. OCR can also help in accessing documents written in unusual letters or in a foreign language that is not known to someone. In the business field, OCR can help in processing and storing documents contained in images such as invoices, financial reports, and other documents. OCR can also help in processing and storing data contained in photos such as names, addresses, and phone numbers, which can be used for marketing or data analysis purposes. In the industrial field, OCR can help in processing and storing documents contained in images, such as product specifications, quality reports, and other documents. OCR can also help in processing and storing data contained in images, such as serial numbers, production dates, and production locations, which can be used for product maintenance and care purposes. This research on text recognition in images is very important because it can help in processing and accessing the information contained in photos, which can be useful in various fields. Text recognition in photos is a very complex matter due to the many variations. For example, different colors, sizes, fonts, letter sizes, background images, and lighting conditions. Therefore, it is very difficult to achieve high accuracy even though reCAPTCHA is currently still collecting datasets to improve its program. Therefore, in this research, only capital text is used as shown in Figure 1. The objects used in this research are a document and an image containing capital letter text. The document and photos will be input into the system to read and recognize the text contained within it. The objects of this research were collected from various sources, such as books, letters, and personal documents.

Figure 1. Sample data collection

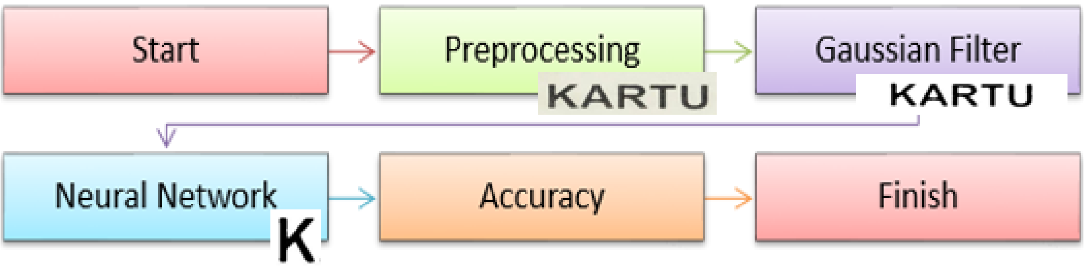

Figure 2. Flowchart proposed method

In this research, a Neural Network will be used. However, before training, preprocessing must be done to make the image clearer and cleaner. Then, segmentation is also carried out to focus on specific parts. Preprocessing and segmentation are important steps in the OCR process as they help improve the quality of the input photos and focus on the specific area of interest, which can help improve the accuracy of the OCR system [5], [23], [35], [36]. Preprocessing can involve steps such as image resizing, noise removal, and image enhancement, while segmentation involves dividing the image into smaller regions or segments. These steps can help the OCR system to enhanced accurately identify and recognize the text in the image. The dataset used in this research was taken from Kaggle, which contains 372038 characters image a set of 26 capital character (A to Z) printed in the System font. Each letter image is represented by 28 x 28 pixels and have a black background that has been resized to 784 x 1, where each pixel can be either on (represented by a '1') or off (represented by a '0'). The process involves scanning each pixel in the enhanced letter image from top to bottom and left to right to locate the capital letter printed on the paper. It is assumed that the letters are clearly separated from each other (capital letter). In this research, we use the neural network method. A neural network or artificial neural network is a system consisting of many units called "neurons" that work together to solve problems that the system will face [12], [37]. The way a neural network works is that it receives input data to be processed. This input is processed by the neurons in the network. Each neuron calculates its output value using a predetermined function [38]–[40].

The output of each neuron is summed and sent to the neurons in the next layer. This process continues until the data reaches the output layer, where the final output of the network is produced. During the learning process, neural networks can learn patterns in data. For example, there may be specific features on an object to be recognized. For example, the letter "A" has characteristics such as a sloping line, a horizontal line, and so on. During the training process, the neural network will try to identify these characteristics. On a neural network there is such a thing as input layer. The input layer of a neural network is the layer that receives the input data. It is the first layer of the network and does not perform any computations or transformations on the data. Its primary role is to pass the input data through the network to the subsequent layers for processing. The input layer is composed of units in Figure 3, also known as neurons or nodes, that represent the input data. The number of units in the input layer is determined by the dimensions of the input data. For example, if the input data is a 2D image with 28 x 28 pixels, then the input layer would have 28 x 28 = 784 units. If the input data is a 1D time series with 10 times steps, then the input layer would have 10 units. The input layer does not have any internal weights or biases, as it does not perform any computations. It simply serves as a conduit for the input data to flow through the network to the subsequent layers. And then there is also a hidden layer. A hidden layer in a neural network is a layer of neurons that is not visible to the input data or the output predictions of the network. Hidden layers are located between the input layer and the output layer in a neural network and are responsible for extracting features from the raw input data and transforming it into a more useful representation for the output layer to process. The number of hidden layers and the number of neurons in each hidden layer can vary greatly depending on the complexity of the problem being solved. In general, a deeper neural network with more hidden layers can learn more complex relationships in the data, but it also requires more computational resources and can be more prone to overfitting. Each hidden layer in a neural network consists of a set of neurons, each of which receives input from the neurons in the previous layer, applies a nonlinear activation function, and passes the output to the neurons in the next layer. The activation function is a mathematical function that determines the output of a neuron given its input. Common activation functions include sigmoid, tanh, and ReLU (Rectified Linear Unit).

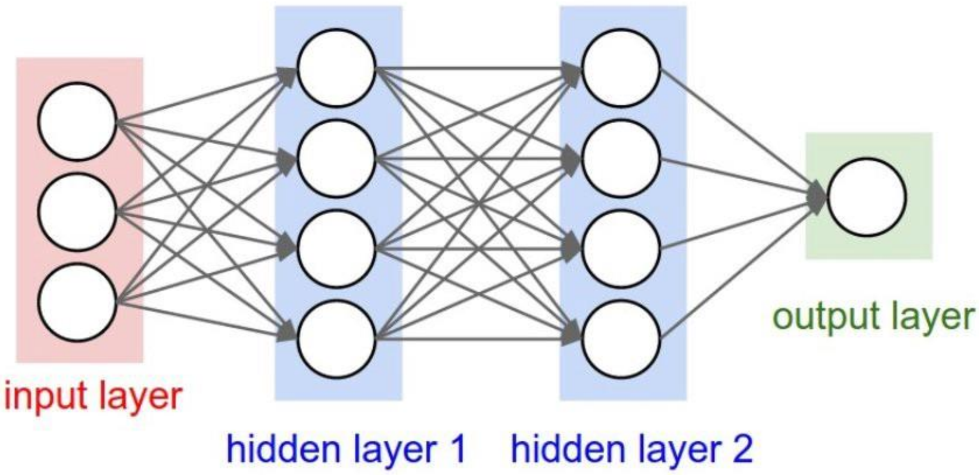

Figure 3. Illustration of neural network [4][8][30]

The weights and biases of the connections between the neurons in the hidden layers are learned during the training process of the neural network using an optimization algorithm, such as stochastic gradient descent. The learning process involves adjusting the weights and biases in order to minimize the error between the predicted output and the ground truth label for a given input. In this research we used 65 hidden layers. Having 65 hidden layers in a neural network would be considered a very deep neural network. Deep neural networks, which have many hidden layers able to learn and model very complex relationships in the data. However, they also require a lot of computational resources to train and can be prone to overfitting, which means that they perform well on the training data but do not generalize well to unseen data. There are several factors to consider when deciding how many hidden layers to include in a neural network

One factor is the complexity of the problem being solved. A deeper neural network may be necessary for more complex tasks, but it may not be necessary for simpler tasks. Another factor is the amount of data available for training. A deeper neural network may be able to learn more complex relationships in the data, but it also requires more data to learn from in order to avoid overfitting. In general, it is a good idea to start with a relatively shallow neural network and gradually increase the depth as needed, rather than starting with a very deep neural network. This allows you to determine the optimal number of hidden layers for your specific problem, while minimizing the risk of overfitting. and the last is output layer. In a neural network, the output layer is the final layer of neurons in the network. It is responsible for producing the output of the neural network based on the inputs it receives from the previous layers. The output layer receives input from the other layers in the network, processes this input using the weights and biases of the neurons, and produces the final output of the network. The output of the output layer is often used to make a prediction or classification based on the input data. The number of neurons in the output layer is typically determined by the number of classes that the network is trying to predict. For example, if the neural network is trying to classify images into one of 10 different categories, the output layer would have 10 neurons, each corresponding to a different class. The output of each neuron in the output layer would be a probability, indicating the likelihood that the input image belongs to that class. The class with the highest probability would be the one that the network predicts. In some cases, the output layer may also have a continuous output, such as a real number, rather than a probability. This can be useful for tasks such as regression, where the goal is to predict a continuous output rather than a discrete class. Overall, the output layer is an important part of a neural network, as it produces the final output of the network based on the input it receives from the other layers. This measures how well your regression model to predict the actual output for new data that was not used in the modeling. If your regression model is good, then it will predict the actual output with a low level of error. Figure 6 shows the best validation result, which is 0.0087446 at epoch 3506. Mean squared error (MSE) is one matrix that can be used to evaluate the performance of a neural network validation. MSE is calculated by taking the difference between the desired output value and the actual output value produced by the model, then multiplying the difference by 2, and then summing all the results. Then, the sum is divided by the number of data used for validation. MSE is used to measure the difference between the predicted output and the actual output, and a lower MSE indicates a better model. To calculate MSE, you first need to calculate the difference between the predicted value and the actual value for each data point, square the difference, and then sum all the squared differences. Finally, divide the sum by the number of data points to get the average squared error. The lower the MSE, the better the model is at predicting the actual output. The smaller the MSE value, the better the model is considered at predicting the desired output. However, it is important to note that MSE is not always the most appropriate metric for every case. If the data used has very large values, MSE may not give an accurate picture of the model's performance as in (1), where n is the number of data points used for validation, $\sum$ is the sum symbol, indicating that the values should be added together, Predicted value is the value predicted by model, Actual value is the true value of the output.

$$MSE = (1/n) \sum(Yi - Yi')^2 \qquad (1)$$

RESULTS AND DISCUSSIONS

Before creating the neural network, we import the existing dataset. In this case, we create 65 hidden layers in the neural network. Before training the input from the dataset, we first normalize it using a custom function. This function will change the background of the images in the dataset to white. Then we normalize the target to be trained, resulting in the transpose of the normalized target. Here, we use 4000 epochs or iterations. The training process will take approximately 2-4 hours and will result in the following performance. Based on Figure 4 and Figure 5, the value of R for Training, Validation, and Test is 0.89136 or 89.1%, 0.87411 or 87.4%, and 0.86548, respectively.

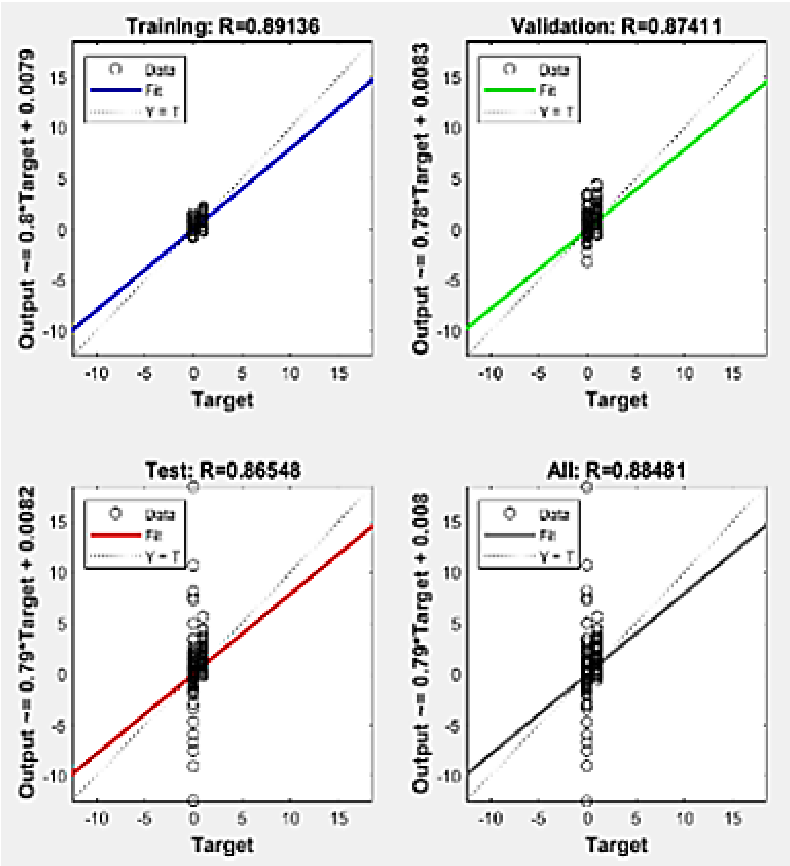

Figure 4. Regression training neural network result

Inputting an image is the first process carried out. In this process, all input images are read by the system. The image is then processed in the next step, which is pre-processing. After creating an artificial neural network, the next step is to pre-process the input image. Pre-processing is the process of cleaning and preparing the data before it is used in training. Pre-processing is the process of preparing the image for further analysis by cleaning, enhancing, and adjusting the image so that it can be accurately processed by the OCR system.

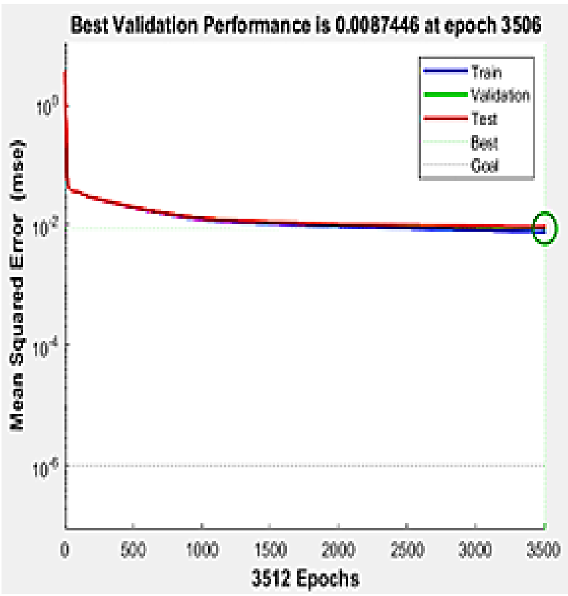

Figure 5. Validation performance

This may involve tasks such as removing noise or blur, correcting distortions, and adjusting the image contrast. Pre-processing is an important step in the OCR process because it helps to improve the accuracy and efficiency of the OCR system. The first step is to add a Gaussian filter, which is used to smooth the indentations in the character and improve the accuracy of the output. Then, the image is converted from grayscale to binary. Next, a cropping step is performed on the input image. Cropping is done to discard unnecessary parts by finding rows and columns that only contain a value of 0. The next step is segmentation, which is used to separate each character or letter by storing two indexes for the beginning and end in an array. Segmenting the parts or each letter is useful to make the text recognition more focused and get more specific information from the object. Then, cropping of the image containing the word into each letter will be performed using the two initial indexes and the saved end index previously. After this step, the following will be produced. After marking the parts of the letters that will be matched with the trained network, cropping of the image containing the word will be performed into individual letters using the previously stored initial and final indexes. Here, will use images specified in the object section. This study uses 15 testing data to test the model that has been created. This study will compare the results of testing using a gaussian filter during preprocessing and without using a gaussian filter.

Table 1. Comparison accuracy

Original Character	Neural Network		Gussian Filter + Neural Network	
	Recognized	Accuracy	Recognized	Accuracy
PEREMPUAN	JJAWXH	0%	PEREMOUAN	88.8%
KARTU	KARTU	100%	KARTU	100%
MAHASISWA	UAHASRSWA	77.7%	MAHASBSWA	88.8%
SEMENTARA	SEMENTARA	100%	SEMENTARA	100%
CITRA	CRTRA	75%	CITRA	100%
DIGITAL	DRGRTAL	71.4%	DIGBTAL	85.7%
PENGOLAHAN	PENGOLAHAN	100%	PENGOLAHAN	100%
KESEHATAN	KESEHAAN	88.8%	KESEHAAN	88.8%
PESERTA	PESERTA	100%	PESERTA	100%
NEGERI	NEGERB	83,3%	NEGERB	83.3%
SEKOLAH	SEKOLAH	100%	SEKOLAH	100%
KEBUDAYAAN	KEBUDAYAAR	90%	KEBUDAYAAN	100%
REPUBLIK	REPUBLBK	87,5%	REPUBLBK	87.5%
NUSWANTORO	NUSWANTORO	100%	NUSWANTORO	100%
UJIAN	UJIB	60%	UJIB	60%
Average		82,2%	Average	92,1%

Based on Table 1, the testing results of the test show an average accuracy of 82.2, with the highest accuracy at 100% and the lowest at 0%. This is because the image is not given a Gaussian filter, resulting in noise being created on the background when binarized, which reduces the accuracy of text recognition. Without using a Gaussian filter, noise cannot be removed, and the quality of the photos is reduced, which also reduces the accuracy of text recognition and can cause errors in text recognition. Gaussian filters are commonly used in image processing to smooth and blur photos, and they can be used to reduce noise and improve the quality of photos. By applying a Gaussian filter to the input image before processing it with the OCR system, the quality of the image can be improved, which can help improve the accuracy of the OCR system. The testing using Gaussian filter show an average accuracy of 92.1%, with the highest accuracy at 100% and the lowest at 60%. This is because the objects being predicted are too close together, so when segmenting the letters, the letters that are touching are only read as one. Images will be difficult to read if the background of the image is not clean, dark, and uneven. Therefore, applying a Gaussian filter in the preprocessing stage is important because it removes noise from the image, which improves the accuracy of text recognition and the quality of the image. By improving the quality of the input image, the OCR system is more likely to accurately recognize the text in the image.

CONSLUSION

This research was conducted on text recognition using the neural network method with and without adding a Gaussian filter. From the results of this research and testing, it can be concluded that the system using a Gaussian filter has a success rate of text recognition with an accuracy of 92.1%, while the system without using a Gaussian filter has a success rate of text recognition with an accuracy of 82.2%. It can therefore be concluded that text recognition using the neural network method assisted by a Gaussian filter will produce a higher accuracy than text recognition without using a Gaussian filter. Gaussian filters can improve the quality of the input image by removing noise and smoothing the image, which can help improve the accuracy of the OCR system. It is important to consider the use of preprocessing techniques such as Gaussian filtering to improve the accuracy of OCR systems. For the future research, Neural Network may be can implemented using Wiener filter or Adaptive Threshold to increase best pixel enhancement before weighting and is expected to produce higher accuracy.

REFERENCES

[1] V. V Mainkar, J. A. Katkar, A. B. Upade, and P. R. Pednekar, "Handwritten Character Recognition to Obtain Editable Text," 2020 Int. Conf. Electron. Sustain. Commun. Syst., pp. 599–602, 2020.

[2] Y. Wang, W. Xiao, and S. Li, "Offline Handwritten Text Recognition Using Deep Learning: A Review," J. Phys. Conf. Ser., vol. 1848, no. 1, p. 012015, Apr. 2021, doi: 10.1088/1742-6596/1848/1/012015.

[3] M. Wang, S. Niu, and Z. Gao, "A Novel Scene Text Recognition Method Based on Deep Learning," Comput. Mater. Contin., vol. 60, no. 2, pp. 781–794, 2019, doi: 10.32604/cmc.2019.05595.

[4] R. Parthiban, R. Ezhilarasi, and D. Saravanan, "Optical Character Recognition for English Handwritten Text Using Recurrent Neural Network," in 2020 International Conference on System, Computation, Automation and Networking (ICSCAN), Jul. 2020, pp. 1–5. doi: 10.1109/ICSCAN49426.2020.9262379.

[5] M. B. Bora, D. Daimary, K. Amitab, and D. Kandar, "Handwritten Character Recognition from Images using CNN-ECOC," Procedia Comput. Sci., vol. 167, pp. 2403–2409, 2020, doi: 10.1016/j.procs.2020.03.293.

[6] Y. Wang, L. Dang, and J. Ren, "Forest fire image recognition based on convolutional neural network," J. Algorithm. Comput. Technol., vol. 13, p. 174830261988768, Jan. 2019, doi: 10.1177/1748302619887689.

[7] S. Aqab and M. Usman, "Handwriting Recognition using Artificial Intelligence Neural Network and Image Processing," Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 7, 2020, doi: 10.14569/IJACSA.2020.0110719.

[8] W. Mo, X. Luo, Y. Zhong, and W. Jiang, "Image recognition using convolutional neural network combined with ensemble learning algorithm," J. Phys. Conf. Ser., vol. 1237, no. 2, p. 022026, Jun. 2019, doi: 10.1088/1742-6596/1237/2/022026.

[9] Y. B. Hamdan and A. Sathesh, "Deep Learning based Handwriting Recognition with Adversarial Feature Deformation and Regularization," J. Innov. Image Process., vol. 3, no. 4, pp. 367–376, Dec. 2021, doi: 10.36548/jiip.2021.4.008.

[10] Tapotosh Ghosh, M.-H.-Z. Abedin, H. Al Banna, N. Mumenin, and M. Abu Yousuf, "Performance Analysis of State of the Art Convolutional Neural Network Architectures in Bangla Handwritten Character Recognition," Pattern Recognit. Image Anal., vol. 31, no. 1, pp. 60–71, Jan. 2021, doi: 10.1134/S1054661821010089.

[11] G. A. Robby, A. Tandra, I. Susanto, J. Harefa, and A. Chowanda, "Implementation of Optical Character Recognition using Tesseract with the Javanese Script Target in Android Application," Procedia Comput. Sci., vol. 157, pp. 499–505, 2019, doi: 10.1016/j.procs.2019.09.006.

[12] E. A. Al-Zubaidi, M. M. Mijwil, and A. S. Alsaadi, "Two-Dimensional Optical Character Recognition of Mouse Drawn in Turkish Capital Letters Using Multi-Layer Perceptron Classification," J. Southwest Jiaotong Univ., vol. 54, no. 4, 2019, doi: 10.35741/issn.0258-2724.54.4.4.

[13] H. M. Najadat, A. A. Alshboul, and A. F. Alabed, "Arabic Handwritten Characters Recognition using Convolutional Neural Network," in 2019 10th International Conference on Information and Communication Systems (ICICS), Jun. 2019, pp. 147–151. doi: 10.1109/IACS.2019.8809122.

[14] A. Rao, A. Arpitha, C. Nayak, S. Meghana, S. Nayak, and S. S., "Exploring Deep Learning Techniques for Kannada Handwritten Character Recognition: A Boon for Digitization," vol. 29, pp. 11078–11093, Jul. 2020.

[15] I. Khandokar, M. Hasan, F. Ernawan, S. Islam, and M. N. Kabir, "Handwritten character recognition using convolutional neural network," J. Phys. Conf. Ser., vol. 1918, no. 4, p. 042152, Jun. 2021, doi: 10.1088/1742-6596/1918/4/042152.

[16] I. Uddin et al., "Benchmark Pashto Handwritten Character Dataset and Pashto Object Character Recognition (OCR) Using Deep Neural Network with Rule Activation Function," Complexity, vol. 2021, pp. 1–16, Mar. 2021, doi: 10.1155/2021/6669672.

[17] Y. B. Hamdan and Sathish, "Construction of Statistical SVM based Recognition Model for Handwritten Character Recognition," J. Inf. Technol. Digit. World, vol. 3, no. 2, pp. 92–107, Jun.

2021, doi: 10.36548/jitdw.2021.2.003.
[18] M. H. KESİKOĞLU, S. Y. ÇİÇEKLİ, and T. KAYNAK, “THE IDENTIFICATION OF SEASONAL COASTLINE CHANGES FROM LANDSAT 8 SATELLITE DATA USING ARTIFICIAL NEURAL NETWORKS AND K-NEAREST NEIGHBOR,” Turkish J. Eng., vol. 4, no. 1, pp. 47–56, Jan. 2020, doi: 10.31127/tuje.599359.
[19] S. Ahlawat, A. Choudhary, A. Nayyar, S. Singh, and B. Yoon, “Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN),” Sensors, vol. 20, no. 12, p. 3344, Jun. 2020, doi: 10.3390/s20123344.
[20] A. K. Nugroho, I. Permadi, and M. Faturrahim, “Improvement Of Image Quality Using Convolutional Neural Networks Method,” Sci. J. Informatics, vol. 9, no. 1, pp. 95–103, May 2022, doi: 10.15294/sji.v9i1.30892.
[21] T. Q. Vinh, L. H. Duy, and N. T. Nhan, “Vietnamese handwritten character recognition using convolutional neural network,” IAES Int. J. Artif. Intell., vol. 9, no. 2, p. 276, Jun. 2020, doi: 10.11591/ijai.v9.i2.pp276-281.
[22] N. B. Muppalaneni, “Handwritten Telugu Compound Character Prediction using Convolutional Neural Network,” in 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), 2020, pp. 1–4. doi: 10.1109/ic-ETITE47903.2020.349.
[23] J. Liu and J. Hong, “Design of Handwritten Numeral Recognition System Based on BP Neural Network,” J. Phys. Conf. Ser., vol. 2025, no. 1, p. 012016, Sep. 2021, doi: 10.1088/1742-6596/2025/1/012016.
[24] A. Mohsin and M. Sadoon, “Developing an Arabic Handwritten Recognition System by Means of Artificial Neural Network,” J. Eng. Appl. Sci., vol. 15, no. 1, pp. 1–3, Oct. 2019, doi: 10.36478/jeasci.2020.1.3.
[25] Y. Zhuang et al., “A Handwritten Chinese Character Recognition based on Convolutional Neural Network and Median Filtering,” J. Phys. Conf. Ser., vol. 1820, no. 1, p. 012162, Mar. 2021, doi: 10.1088/1742-6596/1820/1/012162.
[26] R. B. Lincy and R. Gayathri, “Optimally configured convolutional neural network for Tamil Handwritten Character Recognition by improved lion optimization model,” Multimed. Tools Appl., vol. 80, no. 4, pp. 5917–5943, Feb. 2021, doi: 10.1007/s11042-020-09771-z.
[27] N. P. Sutramiani, N. Suciati, and D. Siahaan, “MAT-AGCA: Multi Augmentation Technique on small dataset for Balinese character recognition using Convolutional Neural Network,” ICT Express, vol. 7, no. 4, pp. 521–529, Dec. 2021, doi: 10.1016/j.icte.2021.04.005.
[28] A. Susanto, C. Atika Sari, I. U. W. Mulyono, and M. Doheir, “Histogram of Gradient in K-Nearest Neighbor for Javanese Alphabet Classification,” Sci. J. Informatics, vol. 8, no. 2, pp. 289–296, Nov. 2021, doi: 10.15294/sji.v8i2.30788.
[29] D. Beohar and A. Rasool, “Handwritten Digit Recognition of MNIST dataset using Deep Learning state-of-the-art Artificial Neural Network (ANN) and Convolutional Neural Network (CNN),” in 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), Mar. 2021, pp. 542–548. doi: 10.1109/ESCI50559.2021.9396870.
[30] I. Prihandi, I. Ranggadara, S. Dwiasnati, Y. S. Sari, and Suhendra, “Implementation of Backpropagation Method for Identified Javanese Scripts,” J. Phys. Conf. Ser., vol. 1477, no. 3, p. 032020, Mar. 2020, doi: 10.1088/1742-6596/1477/3/032020.
[31] A. Hazra, P. Choudhary, S. Inunganbi, and M. Adhikari, “Bangla-Meitei Mayek scripts handwritten character recognition using Convolutional Neural Network,” Appl. Intell., vol. 51, no. 4, pp. 2291–2311, Apr. 2021, doi: 10.1007/s10489-020-01901-2.
[32] R. KARAKAYA and S. KAZAN, “Handwritten Digit Recognition Using Machine Learning,” Sak. Üniversitesi Fen Bilim. Enstitüsü Derg., vol. 25, no. 1, pp. 65–71, Feb. 2021, doi: 10.16984/saufenbilder.801684.
[33] M. Jain, G. Kaur, M. P. Quamar, and H. Gupta, “Handwritten Digit Recognition Using CNN,” in 2021 International Conference on Innovative Practices in Technology and Management (ICIPTM), Feb. 2021, pp. 211–215. doi: 10.1109/ICIPTM52218.2021.9388351.
[34] S. Ahlawat and A. Choudhary, “Hybrid CNN-SVM Classifier for Handwritten Digit Recognition,” Procedia Comput. Sci., vol. 167, pp. 2554–2560, 2020, doi: 10.1016/j.procs.2020.03.309.
[35] S. P. Deore and A. Pravin, “Devanagari Handwritten Character Recognition using fine-tuned Deep Convolutional Neural Network on trivial dataset,” Sādhanā, vol. 45, no. 1, p. 243, Dec. 2020, doi: 10.1007/s12046-020-01484-1.
[36] P. (Dr. . R. S. Anuj Bhardwaj, “HANDWRITTEN DEVANAGARI CHARACTER RECOGNITION USING DEEP LEARNING - CONVOLUTIONAL NEURAL NETWORK

(CNN) MODEL," PalArch's J. Archaeol. Egypt / Egyptol., vol. 17, no. 6 SE-, pp. 7965–7984, Dec. 2020, [Online]. Available: https://archives.palarch.nl/index.php/jae/article/view/2203

[37] M. S. Alam, K.-C. Kwon, M. A. Alam, M. Y. Abbass, S. M. Imtiaz, and N. Kim, "Trajectory-Based Air-Writing Recognition Using Deep Neural Network and Depth Sensor," Sensors, vol. 20, no. 2, p. 376, Jan. 2020, doi: 10.3390/s20020376.

[38] R. Dixit, R. Kushwah, and S. Pashine, "Handwritten Digit Recognition using Machine and Deep Learning Algorithms," Int. J. Comput. Appl., vol. 176, no. 42, pp. 27–33, Jul. 2020, doi: 10.5120/ijca2020920550.

[39] A. Anton, N. F. Nissa, A. Janiati, N. Cahya, and P. Astuti, "Application of Deep Learning Using Convolutional Neural Network (CNN) Method For Women's Skin Classification," Sci. J. Informatics, vol. 8, no. 1, pp. 144–153, May 2021, doi: 10.15294/sji.v8i1.26888.

[40] I. F. Katili, F. D. Esabella, and A. Luthfiarta, "Pattern Recognition Of Javanese Letter Using Template Matching Correlation Method," J. Appl. Intell. Syst., vol. 3, no. 2, pp. 49–56, Dec. 2018, doi: 10.33633/jais.v3i2.1954.