[1]st Proceeding of International Conference on Science and Technology UISU (ICST)
"Contribution of Engineering Disciplines to Achieving Sustainable Development Goals Research Realizing the Vision of Golden Indonesia 2045"
ISSN: 3063-7929 (Online)
DOI: https://doi.org/10.30743/icst

Document Image Analysis for Deep Learning-Based Text Recognition

Firman Styono[1*], Bob Subhan Riza[2], Mhd Furqan[3]
[1]Master of Computer Science Program, Faculty of Engineering and Computer Science, Universitas Potensi Utama, Medan. Indonesia
[2]Department of Computer Science, Faculty of Information Systems, Universitas Potensi Utama, Medan. Indonesia
[3]Department of Computer Science, Faculty of Science and Technology, UIN Sumatera Utara, Medan. Indonesia
[*]Corresponding author: fstyono43@gmail.com

Abstract -This study evaluates the performance of two models, OCR + CNN and Hybrid CNN-RNN, using the MNIST dataset, which consists of 60,000 training samples and 10,000 test samples, each sized 28 x 28 pixels. The results show that the Hybrid CNN-RNN model significantly outperforms the OCR + CNN model in terms of Overall Accuracy, F1-Score, Character Error Rate (CER), and Word Error Rate (WER). The Hybrid CNN-RNN achieved an overall accuracy of 99.18%, compared to 93.14% for OCR + CNN, and demonstrated much lower error rates (CER and WER of 0.82%) compared to OCR + CNN (CER and WER of 6.86%). In terms of training and validation accuracy, the Hybrid CNN-RNN also performed better, reaching 99.53% training accuracy and 99.18% validation accuracy at Epoch 5, while OCR + CNN achieved 95.89% and 95.97%, respectively. Despite the superior accuracy, the Hybrid CNN-RNN model required more inference time, taking 7.46 seconds for 10,000 samples, as opposed to 5.22 seconds for the OCR + CNN model. In conclusion, while the Hybrid CNN-RNN model offers better accuracy and stability, the OCR + CNN model is more efficient in terms of inference time, and the choice of model depends on whether higher accuracy or faster inference is prioritized.

Keywords: ***Deep Learning; CNN; CNN-RNN; OCR; Text Recognition.***

1. INTRODUCTION

Text recognition in document images is one of the major challenges in the field of computer vision and text processing. It plays an important role in various applications, such as document digitisation, automated form processing, and image-based data analysis [1]. With the increasing need for accurate and efficient text recognition systems, various deep learning-based approaches have been developed. One of the key technologies widely used in text recognition is Optical Character Recognition (OCR). OCR enables text extraction from document images to be converted into digital text that can be further processed [2]. With the development of deep learning, Convolutional Neural Network (CNN) and CNN-RNN (Recurrent Neural Network) based approaches have significantly improved the accuracy and robustness of OCR systems over conventional methods based on traditional algorithms [6].

Convolutional Neural Networks (CNN) have become the dominant method for feature extraction in images due to their ability to capture spatial patterns. However, CNN models tend not to be effective enough in capturing sequential information, which is important in the context of text recognition. To overcome this limitation, a hybrid architecture that combines CNN with Recurrent Neural Networks (RNN) has been introduced. This combination utilises the advantages of CNN in feature extraction and the ability of RNN in capturing temporal dependencies, resulting in better performance in document image-based text recognition [9], [10].

Thus, this research not only contributes to the development of deep learning technology but also has a direct impact on operational effectiveness in the industrialised world. The analysis is conducted using metrics such as accuracy, inference time, and robustness to noise, which is expected to provide clear guidance in choosing the best approach for text recognition applications in the industrial sector.

Previous research has discussed various approaches to improve OCR performance but still has limitations on documents with font variations or text structure. Cheng et al. (2020) evaluated the effectiveness of CNN-RNN in handling noise in distorted documents and found that this approach was able to improve accuracy by up to 15% compared to CNN alone [2]. Research by Li et al. (2021) also compared deep learning approaches for OCR and found that CNN-RNN excelled in text analysis of documents that require context, such as tables and forms [9].

This research aims to compare the performance of CNN and CNN-RNN in the task of text recognition on document images. In addition to exploring the strengths and weaknesses of both approaches, this research is designed to provide practical solutions for the industrialised world, particularly companies that operate with high volumes of digital documents. In the business world, many important decisions have to be made based on the information contained in documents such as financial statements, contracts, and invoices that are often stored in PDF format. An efficient text recognition system can help companies to extract information automatically, thus speeding up the decision-making process and increasing productivity [5], [6].

2. METHODS

This research method follows a systematic approach to extract valuable, relevant, and understandable knowledge from data. The process involves several interconnected stages, as illustrated in Figure 1.

[1]st Proceeding of International Conference on Science and Technology UISU (ICST)
"Contribution of Engineering Disciplines to Achieving Sustainable Development Goals Research Realizing the Vision of Golden Indonesia 2045"
ISSN: 3063-7929 (Online)
DOI: https://doi.org/10.30743/icst

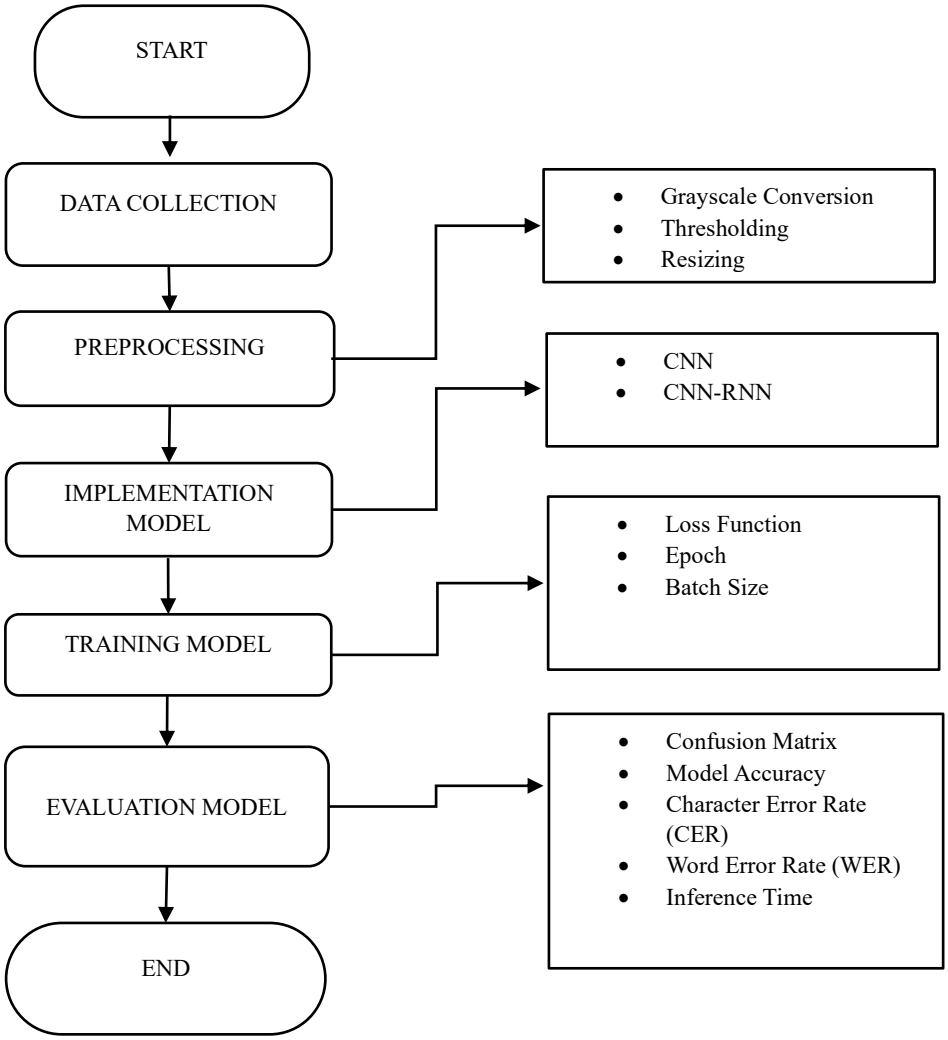

Figure 1. Research Stages

2.1. Data Collection

The dataset used in this research is the MNIST (Modified National Institute of Standards and Technology) dataset, which is a widely recognized benchmark for evaluating image recognition models. It consists of a large collection of handwritten digits (0-9), each image being 28x28 pixels in grayscale.

2.2. Preprocessing

The preprocessing stage is carried out to ensure consistent quality of input data. Here is an explanation of each of these steps:

a. Conversion to grayscale
To reduce the dimensionality of the data without losing important information and sets the colormap to 'gray', ensuring the images are displayed in grayscale.

ISSN: 3063-7929 (Online)
DOI: https://doi.org/10.30743/icst

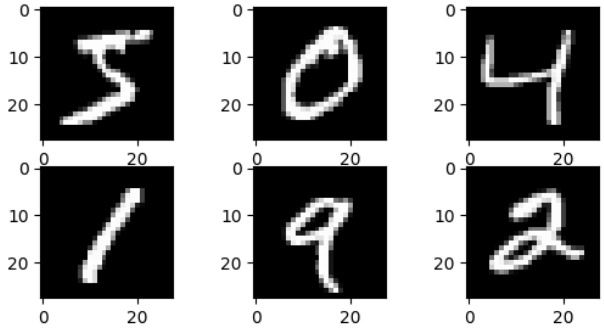

Figure 2. Results of Grayscale Dataset

b. Reshape

Reshape changes the structure of the data to have 4 dimensions. The first dimension (X_train.shape[0]) represents the number of images (samples). It remains unchanged. The next two dimensions (28, 28) specify the height and width of each image. The last dimension (1) indicates that the images are grayscale (single color channel). This is required by many deep learning models.

Figure 3. Reshape Process Results

c. Normalize

Pixel values in the original MNIST dataset range from 0 to 255. Dividing by 255.0 scales the values to a range between 0.0 and 1.0. Normalization helps improve the performance and stability of the model training process.

Figure 4. Results of Normalize Process

2.3. Model Implementation

a. ***OCR + Convolutional Neural Network* (CNN)**

CNN model is designed for spatial feature extraction from document images. This CNN consists of multiple convolution and pooling layers with ReLU activation function. By adding sequential creates a linear stack of layers. Conv2D is a convolutional layer that extracts features from the images. MaxPool2D reduces the spatial dimensions of the feature maps. Flatten converts the multi-dimensional feature maps into a single vector. Dense layers are fully connected layers that perform computations. The activation functions 'relu' and 'softmax' are used to specify the activation functions for these layers.

b. *OCR + Convolutional Recurrent Neural Network* (CRNN)

The CRNN model combines CNN for feature extraction and RNN, specifically *Long Short-Term Memory* (LSTM), to capture sequential information from text features. This architecture uses a convolution layer to extract features, which are then passed to the LSTM for temporal processing.

2.4. Training Model

The model was trained using the preprocessed dataset. The training parameters include:

a. Loss Function

Connectionist Temporal Classification (CTC) loss to accommodate varying text length.

b. Epochs

5 epochs with early stopping based on accuracy validation. This indicates the number of times the entire training dataset is used for training.

c. Batch_size

It set to 128 specifies the number of samples processed before updating the model's weights.

The framework used is TensorFlow and Keras and Sklearn to accelerate model training.

[1]st Proceeding of International Conference on Science and Technology UISU (ICST)
"Contribution of Engineering Disciplines to Achieving Sustainable Development Goals Research Realizing the Vision of Golden Indonesia 2045"
ISSN: 3063-7929 (Online)
DOI: https://doi.org/10.30743/icst

2.5. Evaluation

The model was evaluated based on Confusion Matrix that helps analyze the model's prediction performance in detail. Loss measures how far the model's predictions are from the actual values, guiding model improvement. Accuracy gives the overall percentage of correct predictions but may not fully reflect performance in certain cases, especially with imbalanced data. Character Error Rate (CER): Measures the error rate of recognised characters, Word Error Rate (WER): Measures the error rate of recognised words, Inference Time: The time taken to process one document.

3. RESULTS AND DISCUSSION

3.1. Data description

The total data used is 60000 consistent with a size of 28 x 28 pixels for train data and 10000 test data derived from the MNIST dataset.

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
Shape of the dataset.
Train: X=(60000, 28, 28), y=(60000,)
Test: X=(10000, 28, 28), y=(10000,)
```

Figure 8. Total training and testing data

3.2. Discussion

This report evaluates the performance of OCR + CNN and OCR + CRNN systems using key metrics such as accuracy, F1-score, Character Error Rate (CER), and Word Error Rate (WER). The results also highlight the system's training progress, validation performance, and inference efficiency. Below are the detailed findings and analysis.

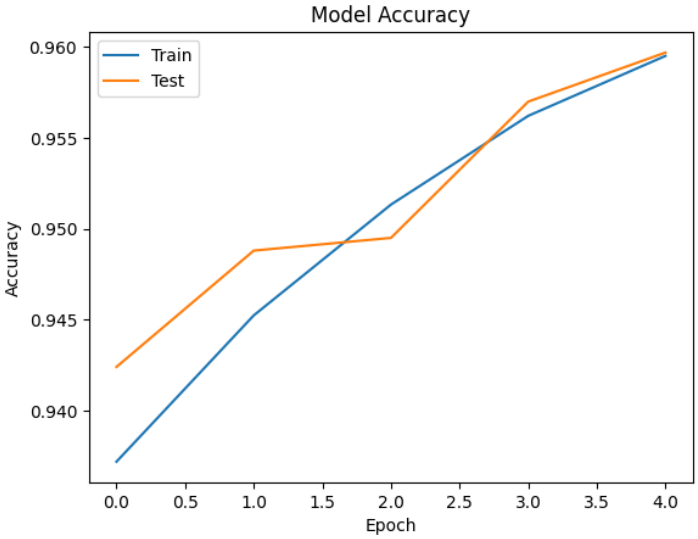

Figure 9. Model Accuracy of OCR + CNN

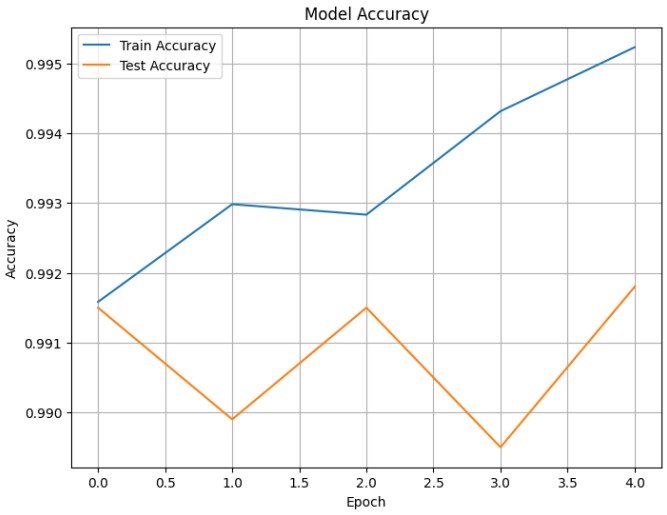

Figure 10. Model Accuracy of OCR + CRNN

[1]st Proceeding of International Conference on Science and Technology UISU (ICST)
"Contribution of Engineering Disciplines to Achieving Sustainable Development Goals Research Realizing the Vision of Golden Indonesia 2045"
ISSN: 3063-7929 (Online)
DOI: https://doi.org/10.30743/icst

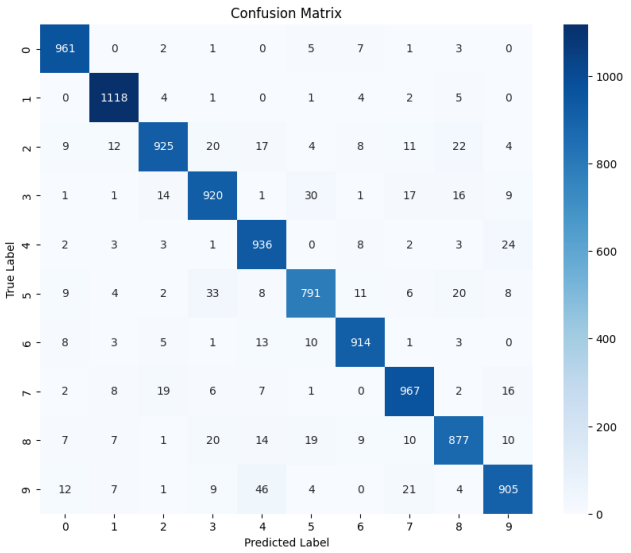

Figure 11. Confusion Matrix of OCR + CNN

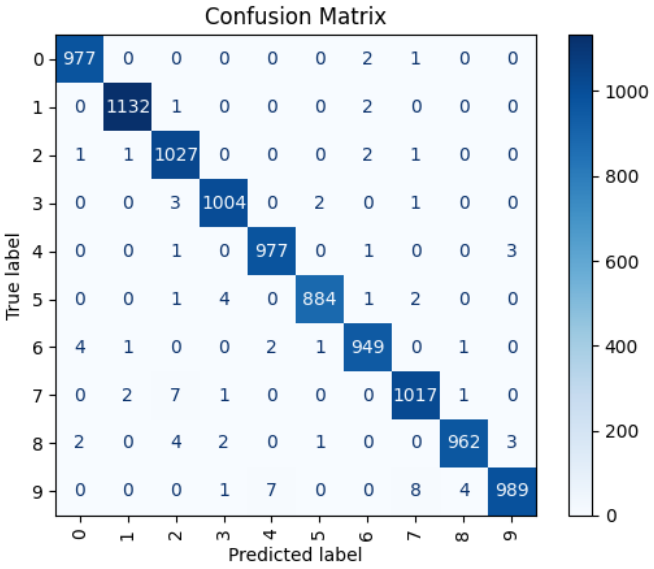

Figure 11. Confusion Matrix of OCR + CRNN

Table 1. Evaluation model of OCR + CNN and OCR + CRNN

Metode Evaluasi	OCR + CNN	OCR + CRNN
Overall Accuracy	93.14%	99.18%
F1-Score	0.931	0.9918
Character Error Rate (CER)	6.86%	0.82%
Word Error Rate (WER)	6.86%	0.82%
Training Accuracy (Epoch 5)	95.89%	99.53%
Validation Accuracy (Epoch 5)	95.97%	99.18%
Total Inference Time (10,000 samples)	5.22s	7.46s
Average Inference Time per Sample	0.000522s	0.000746s

4. CONCLUSION

This study has yielded the result that the OCR + CRNN model significantly outperforms the OCR + CNN model in terms of Overall Accuracy, F1-Score, Character Error Rate (CER), and Word Error Rate (WER). The overall accuracy of Hybrid CNN-RNN is 99.18%, compared to 93.14% for OCR + CNN, indicating that the Hybrid model is more efficient in recognizing and processing data. Furthermore, Hybrid CNN-RNN shows a much lower error rate, with both CER and WER at 0.82%, while OCR + CNN has a CER and WER of 6.86%. This demonstrates that the Hybrid model makes fewer mistakes in character and word recognition. In terms of training and validation accuracy, Hybrid CNN-RNN also outperforms OCR + CNN with training accuracy at 99.53% and validation accuracy at 99.18% in Epoch 5, compared to 95.89% and 95.97%, respectively, for OCR + CNN. This suggests that Hybrid CNN-RNN is more stable and faster in the training process. However, despite its superior accuracy and error rate performance, Hybrid CNN-RNN requires more inference time, both in total (7.46 seconds for 10,000 samples) and per sample (0.000746 seconds), compared to OCR + CNN, which takes 5.22 seconds for 10,000 samples and 0.000522 seconds per sample. In conclusion, while Hybrid CNN-RNN offers better accuracy, lower error rates, and more stable training performance, OCR + CNN is more efficient in terms of inference time. The choice of model ultimately depends on whether the priority is higher accuracy or faster inference.

ACKNOWLEDGMENT

The authors would like to thank the Master of Computer Science Program for its support and sponsorship. And we also thank the research collaboration of Universitas Potensi Utama and UIN Sumatera Utara for supporting this research.

ISSN: 3063-7929 (Online)
DOI: https://doi.org/10.30743/icst

REFERENCES

[1] Dessy Tri, N. (2022). *"Recent Developments in Recurrent Neural Networks for Sequence Modeling."* Journal of Computational Intelligence and Applications.

[2] G.R. Hemanth, M., & Rao, P. (2023). *"Hybrid Approaches for Handwritten Text Recognition: Recent Trends and Future Directions."* IEEE Transactions on Neural Networks and Learning Systems.

[3] Favour Olaoye, O., & Adepoju, A. (2024). "Recent Advances in Handwritten Text Recognition with Deep Learning." *International Journal of Artificial Intelligence Research.*

[4] Kasyfi Ivanedra, B., & Metty Mustikasari, S. (2019). "Improving Handwritten Text Recognition Using CNN and Data Augmentation." *Journal of Data Science and Analytics.*

[5] Mayur Bhargab Bora, R., & Ghosh, A. (2019). "Deep Learning Approaches for Handwritten Character Recognition." *Proceedings of the International Conference on Machine Learning and Data Engineering (ICMLDE).*

[6] Hanan, M., & Kumar, V. (2021). "Handwritten Text Recognition Using CNN and Transfer Learning Techniques." *Journal of Machine Learning Research.*

[7] Zhang, J., & Wallace, B. C. (2019). "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification." *Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP)*.

[8] Ali Firdaus, M., Riawan, B., & Hadi, S. (2021). "Deep Learning for Handwritten Text Recognition with CNN and Attention Mechanisms." *International Journal of Computer Vision*.

[9] Suryo Hartanto, A., & Yuliana, L. (2020). "RNN and Attention Mechanisms for Handwritten Text Recognition." *Journal of Computer Engineering and Applications*.

[10] Gabrani, M., & Joshi, R. (2021). "Hybrid CNN and Transformer Models for Handwritten Text Recognition." *Journal of Artificial Intelligence Research*.

[11] Hassan, M., & Li, X. (2020). "Recent Advances in Recurrent Neural Networks for Time Series Prediction." *International Journal of Data Science and Analytics*.

[12] I Wayan Suartika E. P., et al (2016). "Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101." *JURNAL TEKNIK ITS Vol. 5, No. 1*.

[13] Soheila Gheisari, et al (2021). "A combined convolutional and recurrent neural network for enhanced glaucoma detection." *Scientific Reports*(11-1945). DOI: 10.1038/s41598-021-81554-4.