International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V73I6P123

Volume 73 Issue 6, 281-291, June 2025
© 2025 Seventh Sense Research Group®

Original Article

Enhancing Credit Card Registration Form Processing with Fine-Tuned Transformer-Based OCR Models

Rafi Surya[1], Amalia Zahra[2]

[1,2]*Computer Science Department, BINUS Graduate Program, Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia.*

[1]*Corresponding Author : rafi.surya@binus.ac.id*

Received: 11 October 2024 Revised: 15 May 2025 Accepted: 07 June 2025 Published: 28 June 2025

Abstract - *The widespread adoption of credit cards has significantly advanced universal payment methods. As the popularity of credit cards continues to grow, so does the number of applications that need to be processed. Conventional application processes rely on manual data entry from physical forms, resulting in significant inefficiencies. Therefore, an optimized registration scheme is needed to solve this problem. Optical Character Recognition (OCR) is a potential method to solve this problem. Automating data entry with OCR makes the registration process faster and easier since the user input is minimal. This paper reports experiments on fine-tuning a transformer-based OCR model, TrOCR, for credit card application forms. The enhanced model is tested on the IAM dataset and later on real forms, i.e., credit card application forms from Bank XYZ. The results from the experiments on 50 credit card registration forms are gathered in Table. The model that achieved the lowest CER of 0.3620 was TrOCR_Model_4, which was trained with data pre-processing and tuning beam search parameters. The results indicate that handwritten text is correctly recognized by this modified TrOCR model, with relatively low CER rates, which allows for a more efficient credit card application process.*

Keywords - *Optical Character Recognition, Transformer-based OCR, Credit card registration forms, Handwritten text recognition, Character Error Rate.*

1. Introduction

In the current digital era, credit cards are one of the most frequently used payment methods. Based on data from Bank Indonesia (BI), the number of credit cards distributed in Indonesia reached 17.19 million units in 2022, showing an increase compared to 16.51 million cards in 2021. In 2023, the number of credit card users continued to rise to 17.82 million cards [1]. According to information from Bank Indonesia (BI), credit card transactions faced an increase, amounting to Rp 323.602 million in 2022 compared to Rp 244.516 million in 2021. Additionally, the volume of credit card transactions in Indonesia reached 342.7 million times in 2022, reflecting an increase compared to 281.90 million times in 2021.

From January to August 2023, credit card transactions amounted to Rp 265.77 million, with a recorded transaction volume of 256.5 million times, which is expected to increase by the end of 2023 [2]. The credit card registration process is a significant aspect of obtaining a credit card, where prospective cardholders must fill out registration forms containing various important data such as full name, address, phone number, and other financial details. Nevertheless, it is often time consuming and subject to human error to process the credit card sign-up forms from bank employees or financial institutions on a manual basis. Optical Character Recognition (OCR) has been playing a major role in automating registration form processing in this scenario. One of the most important procedures in functioning a credit card is the registration process, which requires individuals to provide their personal and financial information via usual forms. Usually, through a manual entry process, where these data are handled by bank staff, these steps are time-consuming and may also be one source of error. A remedy to these problems is the introduction of Optical Character Recognition (OCR), a system for form processing, which, therefore, the article considers a valuable innovation. This study employs the TrOCR model, which stands for Transformer-based Optical Character Recognition with Pre-trained Models. One of the pioneering papers that proposed and popularized the use of the transformer model is the well-known "'Attention Is All You Need." This model relies entirely on self-attention, avoiding recurrent networks, GRUs, or LSTMs. The transformer utilizes an Encoder-Decoder structure [3]. The TrOCR model is a text recognition system that uses the Transformer architecture and also takes advantage of pre-trained models in the field of both computer vision and natural language processing. It is an easy-to-understand and successful method that does not use a CNN backbone [4].

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

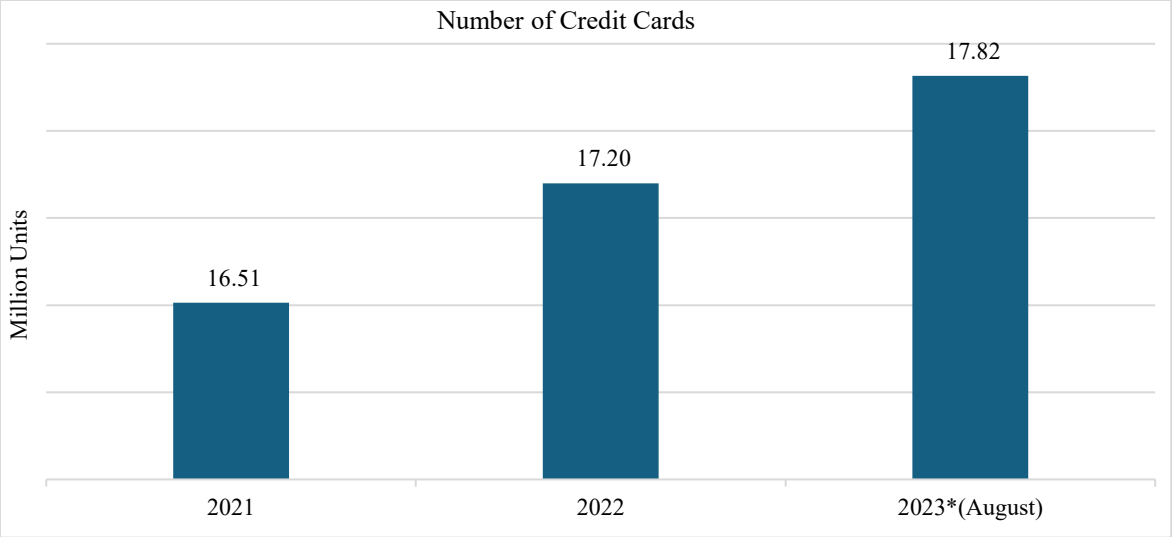

Fig. 1 Number of credit card users in indonesia

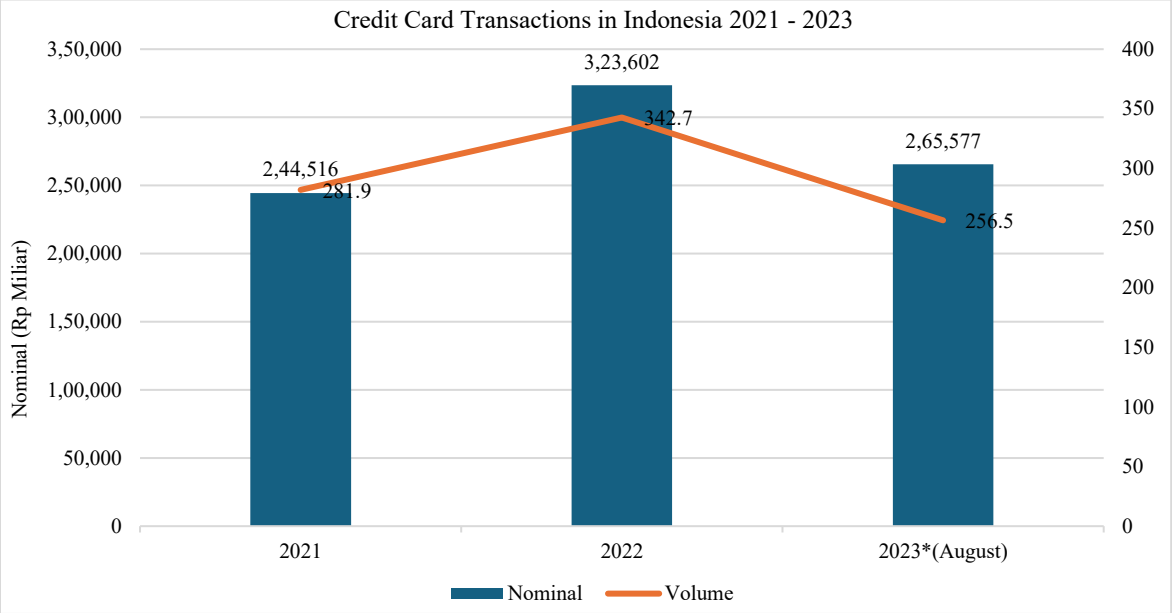

Fig. 2 Credit card user transactions in indonesia

The TrOCR model was chosen since it performed the best in 2022 on the OCR tasks involving printed or handwritten texts. Furthermore, another benefit of not using CNN as a backbone for TrOCR is its lack of dependence on convolutive and burdensome pre-processing or post-processing logistics. In the results of the experiment, it can be inferred that a model named TrOCR has managed to beat all the agents who were training their models with a data-bronze-collective schema built either from synthetic or IAM data and not used for training the T-models by means of the CER as the evaluation metric. The best result (2.89 CER) was achieved with a Transformer model by TrOCR. Moreover, the members, without any assistance from humans, reached the quality of performance of the results obtained by Diaz. [5] study that used additional human-labeled internal datasets. The TrOCR model incorporates an encoder and decoder, which is designed as follows: the Encoder utilizes a pre-trained ViT-style [6] model, and the decoder uses a pre-trained BERT-style [7] model. This trained version of the model will be evaluated on a sample registration form collection, which contains different samples of handwriting for testing purposes. The outcome of this study is expected to enhance the quality of efficiency and accuracy achieved in processing credit card registration forms. By using the developed OCR system, banks and financial institutions can automate the form processing process, reduce the time and costs involved, and minimize potential human errors. Therefore, this study aims to refine the TrOCR model for implementation into a system capable of efficiently processing Bank XYZ's credit card registration forms and achieving accurate OCR results, particularly for the credit card

registration process. By adjusting preprocessing stages, such as data normalization and noise removal, and fine-tuning the TrOCR model to recognize the specific format of the bank's forms, this system is expected to overcome challenges in recognizing complex characters and expedite the overall form processing. Moreover, it seeks to improve upon prior techniques regarding the accuracy and efficiency of the results.

2. Related Works

The research described here includes a study on banking cases showing a systematic pattern of goals and processes resembling the objectives of this study. The research presents a study on banking cases exhibiting patterns of objectives and steps akin to this research. This research is about the OCR process on bank cards, utilizing the VGG16 model from this research to get a model accuracy of 86% [8]. In a similar research in 2019, OCR processes on bank cards were undertaken utilizing OpenCV for identification, yielding subpar accuracy at 13.4%. Nevertheless, using Faster Regional-based Convolutional Neural Network (RCNN) + LeNet-5, the loss achieved during training was 0.09 and the accuracy was 90.5%. When we implemented the DCNN + CRNN experiments, the loss diminished to 0.11 with an accuracy of 89.9%. Likewise, CTPN+CRNN experiments resulted in a loss decrease to 0.05 and an accuracy of 95.5% [9]. In the study presented by Sun and You (2020), OCR was carried out on bank card numbers by an artificial neural network model called CRNN-BLSTM model after training the model through dataset augmentation. The result was the bank card numbers that we could derive. As demonstrated in experiments, the model could rapidly pinpoint higher numbers more accurately [10].

Another study in 2019 devised a system for automating the extraction of information from bank checks, including payee name, payment amount, date, and bank name, among others. In this paper, we use deep learning architecture such as CNN and Recurrent Neural Networks (RNN). The IAM dataset is exploited in this setting to learn a handwriting recognition model using a neural network. OpenCV takes the handwritten text from images, and the sentences are split into words, which are then used as input to the model. This model contains 5 layers of CNN before 3 layers of RNN. The CNN uses ReLU with several types of kernel sizes , and its output is subsequently entered into an RNN (LSTM). The predicted text is decoded using Connectionist Temporal Classification (CTC), and noisy reading is completed with the Adam optimizer. If the performance is not up to the mark, further layers of augmented data can be used in training phases [11].

Another research in 2020 determined a machine learning model for SNR based on serial number recognition on banknotes. It started with extracting the whole serial number's region from the image and applying per-character recognition over it once it was segmented. The workflow of this research is a kind of serial number recognition network based on deep learning, which can be end-to-end trained, eliminating initial character segmentation by three steps as follows. First, a deep convolutional neural network is employed to sequentially extract image features from the input image. Second, this feature sequence is fed into Bidirectional Recurrent Neural Networks (BRNNs), where character segmentation is not required. Finally, label recognition is performed using Connectionist Temporal Classification to decode the BRNNs' output. In terms of accuracy and efficiency, the model achieves character and renminbi (RMB) serial number recognition with accuracies of 99.96% and 99.56%, respectively [12].

In 2024, Watson W. and colleagues researched table extraction processes in the financial sector. Financial documents often lack consistent formatting and frequently exhibit high variances in layout across pages and files. This paper focuses on the image domain, where the only machine-readable data is pixelated. The dataset was sampled from 10,000 pages drawn from 9,985 financial documents, including annual reports, prospectuses, and shareholder meeting minutes published by international companies across various sectors, such as commodities, banking, and technology. The model employed in this study was based on the U-Net architecture with DenseNet-169. In this classification model, recall was prioritized over precision, as false positives could be corrected by the subsequent U-Net model.

In contrast, false negatives would result in pages containing tables being discarded too early in the pipeline. The model was fine-tuned to achieve a recall of 0.995 and a precision of 0.871 on the validation set. On the test set, consisting of 1,000 pages, the recall was recorded at 0.993 and precision at 0.910. This method was evaluated at the table level and the page level; 83% of the pages were entirely correct, 9.5% included partial tables, 5.9% contained surrounding text, and 1.6% represented other errors, although these cases were still considered partially correct [13].

Another study used as a literature reference is the research conducted by Rusli and colleagues in 2020. This study developed an OCR model using CNN with the addition of NLP for post-processing. The data used for testing consisted of 50 ID card photos, divided into two types: 25 photos taken using a camera and 25 taken using a scanner. The researchers conducted three experiments with different models. The first model combined OCR with NLP for post-processing, achieving an F-score of 0.78. The second model, with the help of SVM, had an F1-score of 0.63, and the third model, by using CNN, reached an F1-score of 0.84. From the results, it is seen that the CNN model was the best performer [14]. A paper by Zhang et al. in 2022 reveals that the research used Tesseract OCR version 5.0.0 alpha. A set of 100 ID card photos was the data that was tested. It turned out that the

photos were mainly taken with mobile phone cameras, which impacted not only the image quality but also the accuracy of the identification (casual conversation). A Generative Adversarial Network (GAN) was used in the first processing stage to remove the noise from the images, thus enhancing the performance of Tesseract OCR. The authors also utilized shadow removal and image binarization in the preprocessing phase. They saw that the combination of Tesseract OCR with the Threshold TRUNC method for binarization performed very well. After applying DeblurGAN, shadow removal effectively reduced the Character Error Rate (CER). Combining the three pre-processing approaches, the scholars succeeded in the lowest CER, 18.82% [15].

The researchers Ma and Yan performed serial number recognition on banknotes using deep learning in 2021. The sample consisted of digital images that had branding and data augmentation performed on them. The study described a deep learning-based algorithm that ensured the stability of the technology to recognize the serial numbers of the banknotes even in the presence of complex backgrounds. Also, a deep neural network framework was introduced that could be employed for serial number recognition on banknotes. "In financial applications, it is always better to be safe than to be accurate" Hence, with the DenseNet as the main classifier, the scaling transformation SegLink for character detection, their usage for character detection, the result was in agreement, and a detection rate up to 95.80% were reached. The study also proposed using a convolutional neural network with an attention residual model for serial recognition number recognition, reaching a precision rate of 97.09% [16]. The study by Srivastava et al. in 2019 was about developing a model using machine learning to solve a problem in bank check processing by automating the manual character recognition process. The adoption of neural networks is a vital part of achieving the accurate and efficient recognition of handwriting. The images are broken down into fixed-sized pixels, and these segmented characters are then given to the neural network, which gets the characters from the checks and writes them in text format. Verifying the amount in words can be cross-checked with the amount in numbers, ensuring that the two fields match and the check is valid. An error message is displayed if any field is empty, informing the user of the missing field. OCR accuracy may depend on external factors such as damage (which may affect the written part) to the check's surface. Nevertheless, the model has been created in a manner that does not let factors such as lighting or orientation disrupt OCR's accuracy. The model's test accuracy was around 95.71% when conducted on bank checks, whereas the model's validation accuracy during the training was 98% for digits and 97% for the alphabet [17].

In a research study by Parthiban et al. 2020, a Recurrent Neural Network (RNN) model was used for Optical Character Recognition (OCR) on handwritten English text. The database was made up of the writers' transcriptions. The experiment displayed the RNN model as a good learner who could understand handwriting and provide an accuracy of 90% when converted to printed text. The individuality of the handwriting style of each person was taken into account from the start of the study [18]. Recently, in 2021, Narayan and Muthalagu introduced a paper where they developed a Convolutional Neural Network (CNN) model to recognize English handwritten characters from the images of the text. The model had shown that it could be capable of real-time character recognition, which widens its usefulness. They identified preprocessing as the most crucial factor in maintaining the model's high performance. The image preprocessing approach improved the image features and increased recognition accuracy. As a result of the model's implementation, the identification accuracy achieved was at a remarkably high level of 97.59%, with a loss of 6.6% [19].

Gao et al. 2019 verified an Optical Character Recognition (OCR) method for bank cards. The method was based on YOLO v3 as an object detector. The research suggested that it is possible to realize OCR with a minimum requirement of the number of layers in the implementation of the fully connected layers in the last stages of the model. The credit card number came from the bounding boxes, and the one with the highest probability was used. Using a Convolutional Neural Network (CNN), the detected card area with the card number was processed to get feature sequences. A Long Short-Term Memory (LSTM)-based neural network was further used to encode features extracted from the CNN into a probability matrix most likely to result from a combination of the card number. By means of Connectionist Temporal Classification (CTC), the process of finding the observed sequence of the card number and outputting it was accomplished; that is, the alignment and the transformation of the complex character sequence to the correct order and form of characters was utilized so that a relevant and accurate outcome could be achieved. The core of our task is YOLO for object detection and then CNN+LSTM+CTC for the recognition. This combines optimizing and improving them and excels in training and recognition speed. The evaluation metrics achieved were a Recall of 75.20% and a Precision of 86.32% [20].

Referring to the previous research, those studies still employed CNN as the backbone. Transformer-based models effectively understand long-range relationships between characters or words in a text image. This is crucial when recognizing long text or text with a complex structure, such as complete sentences or paragraphs. CNNs are less effective than Transformers in handling the context or relationships between distant elements in an image. In CNNs, the convolution process is done locally, which can slow down processing if the text is scattered or there are long dependencies between characters in the image. Transformers are more flexible in adapting to various text types in vertical, horizontal, or even curved formats. This is because

Transformer models are not dependent on local spatial patterns like CNNs but rather on the overall relationships between characters throughout the image. OCR using Transformers is better suited for handling long or continuous text, as each input segment is processed simultaneously with attention to its global context. Therefore, this study will focus on building an OCR model that targets Bank XYZ's credit card registration forms. TrOCR will be employed as the method for text extraction from the registration forms, as this method is relatively new and represents the state-of-the-art in OCR models today [4]. A key advantage of TrOCR is that it does not use CNN as its backbone, freeing the model from convolution processes and reducing the need for complex pre-processing or post-processing steps. Hence, this study aims to fine-tune the TrOCR model for implementation in a system that can improve processing time and achieve accurate OCR results, particularly for processing Bank XYZ's credit card registration forms. The proposed model will address challenges in recognizing complex characters and accelerate the overall form processing.

3. Dataset

This study used two different datasets for model development and testing. The first dataset is the IAM Dataset, which is widely used for handwriting recognition tasks. This dataset contains various labeled handwriting samples and is used in training, testing, and validation. The IAM Dataset provides a variety of handwriting styles that help the model better recognize patterns. The second dataset is the Bank XYZ Forms, a collection of handwritten form data from the XYZ banking institution. This dataset is used exclusively for the final testing of the model, allowing for the evaluation of the model's generalization ability on data different from the training dataset, thus providing insight into the model's performance in real-world scenarios.

3.1. Training Dataset

This study uses the IAM dataset as their data source, a set of English handwritten text that is not primary but based on previous research. The IAM dataset is frequently employed for the training, validating, and testing of handwriting recognition systems and writer identification and verification research. The handwriting samples have been given by 657 persons who were behind the making of 13,353 text line images. These transcriptions result from the Lancaster-Oslo/Bergen Corpus of British English. Specifically, the dataset consists of 1,539 pages of handwritten text with 115,320 words in total. These samples are from a recent collection of English and are part of annotations at the sentence, line, and word levels [21].

3.2. Testing Dataset

The data utilized in the testing phase consists of authentic information in the form of registration forms for XYZ bank credit cards. The data to be employed exclusively encompasses customer account information and personal data. Further elaboration on the specifics of the data employed in these sections will be provided in Figure 3 and Figure 4.

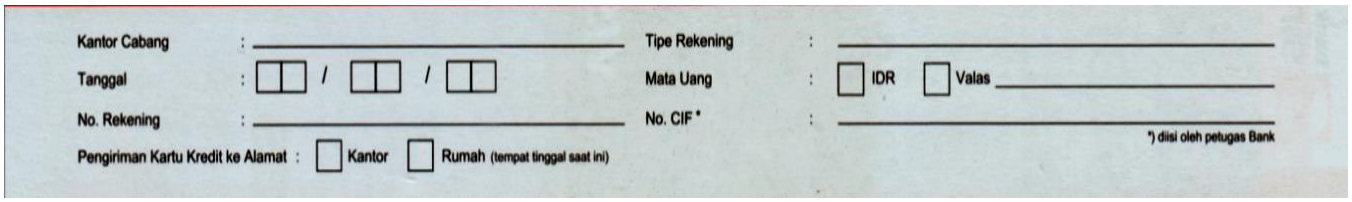

Fig. 3 Image form of account information section

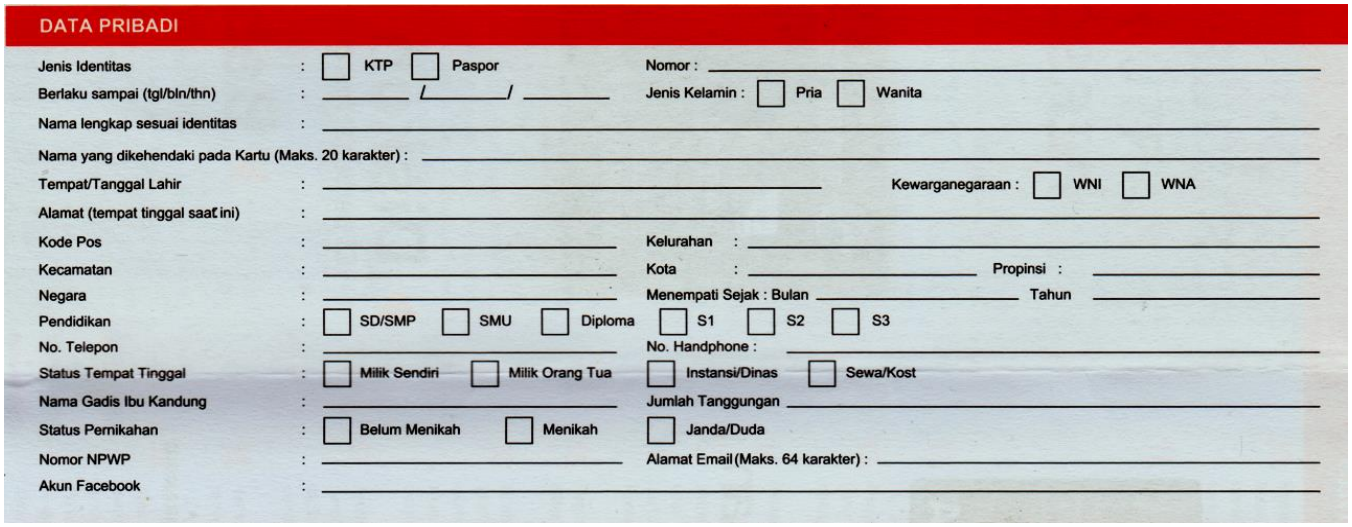

Fig. 4 Image form of personal data section

4. Method Proposed

This study delineates two main stages: (1) Model Training can be illustrated using Figure 5, and (2) Model Testing. The first stage will elucidate the fine-tuning process of the TrOCR model using the IAM dataset.

In contrast, the second stage will involve testing the constructed model using forms from XYZ bank credit cards, which can be illustrated in Figure 6.

4.1. TrOCR Acrhitecture

The TrOCR model utilized in this study is a transformer-based OCR model designed for image text extraction. In this research, the TrOCR model was first introduced by Li et al. [4] in 2022.

The TrOCR model leverages ViT [6] and DeiT [22] models as encoders, with RoBERTa [23] utilized as the decoder. Figure 5. explains the TrOCR architecture

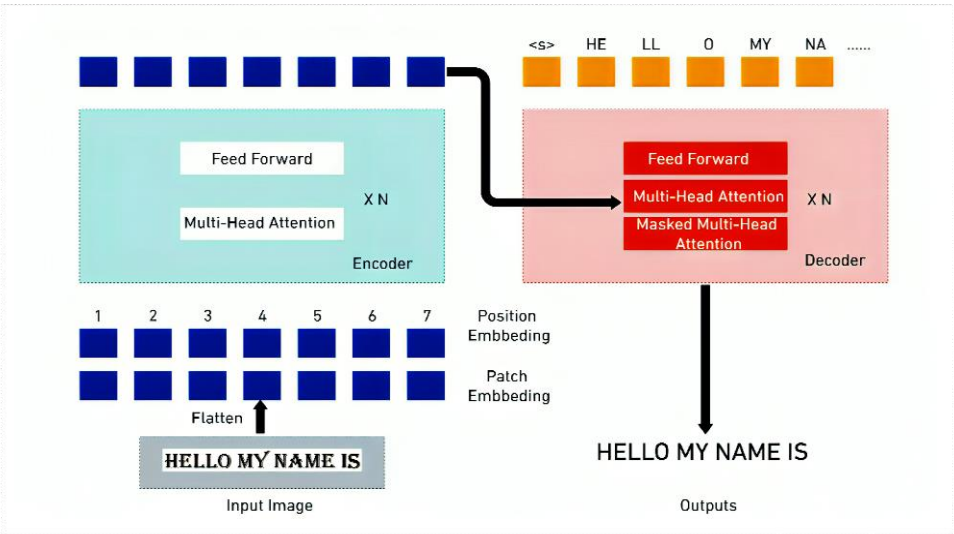

Fig. 5 Tranformer-based architecture

4.1.1. Image Encoder

In the TrOCR architecture, the Encoder receives input images ∈ R3×H0×W0 with dimensions of 3×H0×W0, which are then resized to a fixed size (H, W). Since the Transformer encoder cannot process raw images unless they are in the form of input token sequences, the Encoder divides the input images into a set of small boxes of size N = HW/P^2 with a fixed size (P, P), where the width W and height H of the resized images can be evenly divided by the box size P. Subsequently, these small boxes are converted into vectors and linearly projected into a D-dimensional vector, referred to as Patch Embedding. D represents the hidden size used by the transformer in all its layers.

$$Self - Att(Q,K,V) = V\ softmax\left(\frac{K^T Q}{\sqrt{D_k}}\right) \quad (1)$$

$$MultiHead(Q,K,V) = W_o[head_1; \dots; head_n] \quad (2)$$

$$head_i = Self - Att(Q_i, K_i, V_i) \quad (3)$$

$$\forall i \in \{1, \dots, h\}, Q_i = W_q^i Q, K_i = W_k^i K, V_i = W_v^i V \quad (4)$$

Equation 1 describes the self-attention mechanism in transformers. It works by first calculating the dot product between the query (Q) and key (K) matrices to measure the similarity between them. This result is then scaled by the square root of the key dimension (D_k) to avoid large values that could slow down learning. The computed scores are run through a softmax function to get an attention distribution that will help the model decide which parts of the input are more important and, in the end, use those to calculate the attention-weighted sum of the value (V) matrix. The explicit, refined computational mechanism allows the model to filter out irrelevant concepts while producing output [24].

Equation 2 signifies that the multiple outputs of attention heads (each for different input aspects) are first concatenated. Then, a linear transformation is performed to aggregate the information and create the output. This method boosts the model's capability to grasp a greater variety of relationships among the input data.

Equation 3, "$head_i$ = Self-Att(Q_i, K_i, V_i)", first tell us that for every attention head, it is necessary to do a separate self-attention computation using its own derived queries, keys, and values. This way, each head can gather diverse data relationships, which, in turn, are summed up in the multi-head attention mechanism, and it grants the model a deeper and more comprehensive reading of the input data.

Formula 4 outlines that each attention head i will take the input matrices Q, K, and V and the specific weight matrices W_(q^l)^i W_(k^l)^i and W_(v^.)^i are used to change these matrices linearly. This method enables each head to select some parts of the input information related to clever groupings, and thereby, the model gains better performance to represent all available relations module, enabling the decoder to focus on different aspects of the encoder output. Similar to the architectures of ViT and DeiT, this model preserves a special token "[CLS]" typically used for image classification tasks. The "[CLS]" token aggregates all information from the entire Patch Embedding and represents the overall image.

Additionally, the model retains distillation tokens in the input sequence when utilizing the pre-trained DeiT model to initialize the Encoder, allowing the model to learn from the teacher model. Patch Embedding and these two special tokens are equipped with 1D positional embeddings that can be learned based on their absolute positions.

In contrast to features extracted by CNN-like networks, the Transformer model lacks specific inductive biases for images and processes images as sequences of small boxes. This enables the model to easily allocate different attention either to the entire image or to individual small boxes independently.

4.1.2. Text Encoder

The decoder consists of identical layers, similar to those in the Encoder, but with the addition of an "encoder-decoder attention" fbeammodule inserted between the multi-head self-attention and the feed-forward network. This technology directs the decoder's focus to several parts of the encoder output. In order to stem the appearance of overfitting during training, the decoder applies attention blackout in the self-attention mechanism. This attention limitation guarantees that each position in the decoder output can only access the previous output positions; the method prevents future inputs from being seen during training. The hidden states of the decoder are linearly transformed into the vocabulary size dimension, and the softmax function is used to calculate the vocabulary to get the probability over the words. By then, a more reliable and accurate generation process is possible due to applying the beam search algorithm, which leads to the final output.

4.1.3. Beam Search Algorithm

Beam search is a search algorithm that creates a number of hypotheses in a left-to-right manner. It does so by creating and saving a set of active hypotheses to be used for the final result by comparing their best values. Each iteration takes the ending of the End-of-Sentence (EOS) symbol not as a sign that the hypothesis is made, and this lets the best *n* things be picked each time for the new active set of hypotheses, with *n* being the number of hypotheses in the beam.

The beam search algorithm stops when the hypothesis with the best value in the set ends with the EOS symbol[25]. Beam search has become the standard approach during the decoding phase of neural translation and other text generation tasks to deal with the multiple potential predictions of a word at a given time. With beam search, the model does not select only the word with the highest prediction score but also considers several possible predictions. This allows the model to explore more potential outcomes, thereby improving the quality of the final result [26].

4.2. Proposed Framework

The fine-tuning and training process was conducted using the pre-trained trocr-base-handwritten model. To achieve optimal results, specific training parameter settings were required during the training process. Figure 6 illustrates the fine-tuning process of the model using two scenarios: fine-tuning the model with pre-processing and fine-tuning the model without pre-processing. Table 1 details the parameters used for pre-processing, while Table 2 presents the hyperparameters employed by the researchers for training the TrOCR model. In this initial stage, training, testing, and validation data are separated, and the dataset is divided into three main subsets. One subset is employed for training the model (training data), another subset is reserved for evaluating the performance of the trained model (testing data), and the remaining subset is utilized for validating the trained model (validation data).

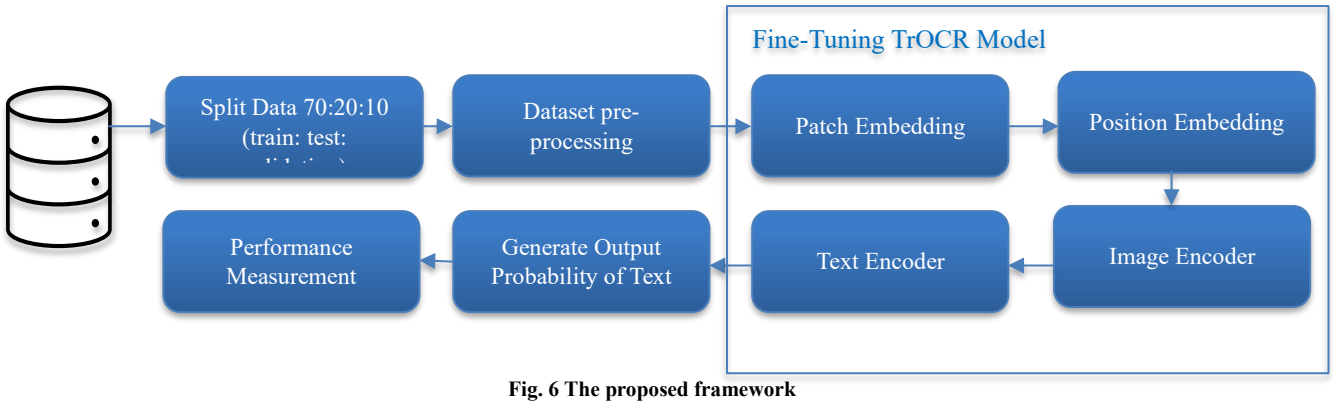

Fig. 6 The proposed framework

Table 1. Preprocessing parameters

Parameter	Value
Image Format	RGB
Brightness	0.5
hue	0.3
GaussianBlur	kernel_size=(5, 9), sigma=(0.1, 5)

The training data to be used for training must undergo a pre-processing stage first. The images will be converted to the RGB format. Then, the brightness parameter will be adjusted to 0.5, meaning the brightness of the image will randomly vary between 0.5 and 1.5 times its original brightness. The hue parameter will be set to 0.3, indicating that the hue of the image will randomly vary between 0.3 and 0.7 times its original hue. The kernel_size parameter will be set to (5, 9), indicating that the Gaussian filter will have a size of either 5x5 or 9x9. The sigma parameter will be set to (0.1, 5), meaning the standard deviation of the Gaussian will vary randomly between 0.1 and 5. Images generated by these transformations may exhibit varying levels of blurriness.

Table 2. Hyperparameter tuning

Parameter	Value	
	Scheme 1	Scheme 2
Input Size	(3, 384, 384)	(3, 384, 384)
max_length	5	10
no_repeat_ngram_size	3	3
length_penalty	2.0	2.0
num_beams	2	4
evaluation_strategy	epoch	epoch
epoch	20	20
batch_size	24	32
learning_rate	0.00005	0.001
weight_decay	adam	adam
Optimizer	0.0005	0.0005
early stopping	True	True

The hyperparameter tuning process will use a combination of the hyperparameters listed in Table 2. During this process, fine-tuning of the beam search hyperparameters will be performed, where adjustments to the beam search hyperparameters are made to enhance the model's performance in generating text from images. The fine-tuned model will be evaluated using the IAM test dataset. The evaluation process of the fine-tuned model using CER begins by comparing the text generated by the model with the actual reference text. In the case of character recognition, this may involve comparing the characters recognized by the model with the characters that should exist in the text. Subsequently, character errors are calculated by comparing incorrectly generated characters by the model with the total number of characters in the reference text.

$$CER = \frac{S + D + I}{N} \quad (5)$$

S is the number of substitutions, I is the number of insertions, D is the number of deletions, and N is the total number of characters in the real text. The lower the CER, the better the model performance, which means the model can recognize characters more accurately. The use of CER provides several advantages. First, it provides a metric that is more sensitive to character errors frequently occurring in OCR or other character recognition tasks. Second, CER allows researchers and developers to understand common errors, such as insertions, deletions, or character substitutions. The model with good performance in the test data evaluation of IAM will be utilized for the identification process in the XYZ bank form. Figure 7 will elucidate the stages of the evaluation process in the XYZ bank Form. XYZ bank's credit card registration form intended for final testing must undergo preprocessing stages first. In the initial phase, form images must undergo an image alignment process to address misaligned or skewed registration form images. Image alignment involves adjusting the position and orientation of the images.

Figure 8 shows the image alignment process. Through this step, misalignments that may be caused by various factors, such as shooting angle or errors in data acquisition, can be corrected.

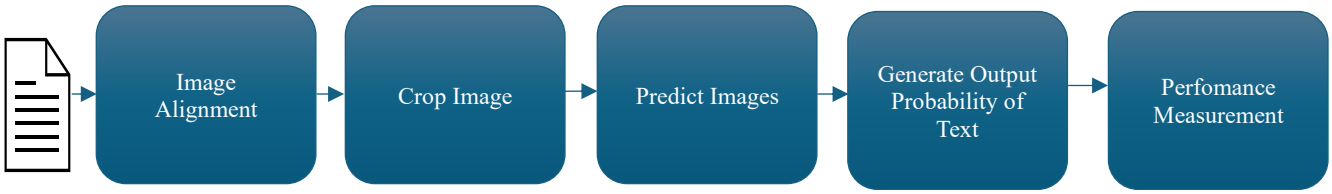

Fig. 7 Evaluation on bank XYZ Form

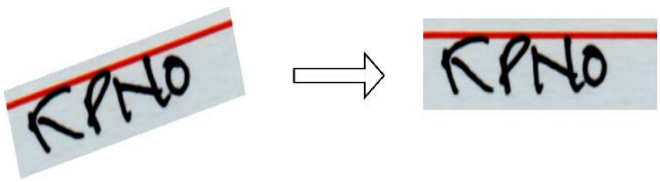

Fig. 8 Image alignment process

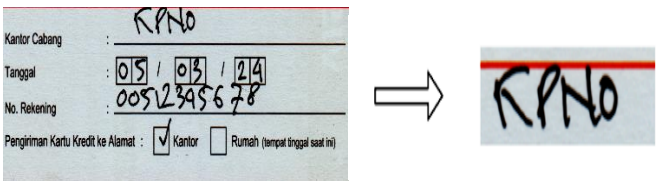

Fig. 9 Image cropping process

To obtain accurate results for predicting Columns containing information and image cropping processes must be performed to obtain prediction outcomes focused on specific image parts. The image cropping process uses coordinates X, Y, W, and H, where these four values define the bounding box on the image to be cropped-Figure 9 shows the process. Here, X denotes the horizontal position from the left edge of the bounding box, Y denotes the vertical position from the top edge of the bounding box, W (width) represents the horizontal length of the bounding box, indicating the extent of an object along the X axis, and H (height) represents the vertical length of the bounding box, indicating the extent of an object along the Y axis. The prediction results generated by the model will be stored as a dataframe for comparison with the actual values present in the form and subsequent computation of the CER evaluation matrix.

Table 3. Performance model on IAM data test

Experiment Model	Preprocessing	Hyperparameter Tuning	CER	Accuracy	Precision	Recall	F1
TrOCR Model 1		✓	0.1150	0.8850	0.875	0.885	0.855
TrOCR Model 2		✓	0.0851	0.9149	0.915	0.908	0.909
TrOCR Model 3	✓	✓	0.0795	0.9205	0.921	0.920	0.922
TrOCR Model 4	✓	✓	0.0607	0.9393	0.949	0.929	0.959

5. Result and Discussion

The previous section explained the architecture and process of fine-tuning the TrOCR model in research. To assess the effectiveness of the model, two types of testing are needed to evaluate the performance of the detection model and recognition approach.

5.1. Evaluation of IAM Data Test

The dataset used to fine-tune the TrOCR base handwritten model is the IAM Handwriting Database, which is of a size of 10,373 and consists of forms of handwritten English text. Training, validation, and testing are the three stages the dataset is passed through, and their fraction from 10,373 is 70:10:20 accordingly. The TrOCR model was trained using Google Collaboratory with Nvidia A100 GPU RAM GPU 15 GB. The trained TrOCR model will be loaded for evaluation on the IAM test dataset. The model prediction results on IAM data will be compared to the original labels and evaluated with the Character Error Rate (CER) evaluation matrix. The results of this comparison are summarized in Table 3. Based on the experimental results of the four models presented in Table 3, the model configuration with dataset preprocessing and hyperparameter tuning outperformed in terms of Character Error Rate (CER). The lowest CER was 0.0851 for the model with hyperparameter tuning without dataset preprocessing.

Additionally, the model Configuration with both dataset preprocessing and hyperparameter tuning achieved a CER of 0.0607.

5.2. Evaluation of form XYZ Bank

In the second part of the evaluation, the experimental model will be evaluated with real data in the form of an XYZ bank credit card registration form. Are 50 forms used to evaluate the model, and each form will go through an image alignment process. Each handwriting on the form will be cropped using coordinates and evaluated using CER. Figure 10 will illustrate the evaluation process in this section.

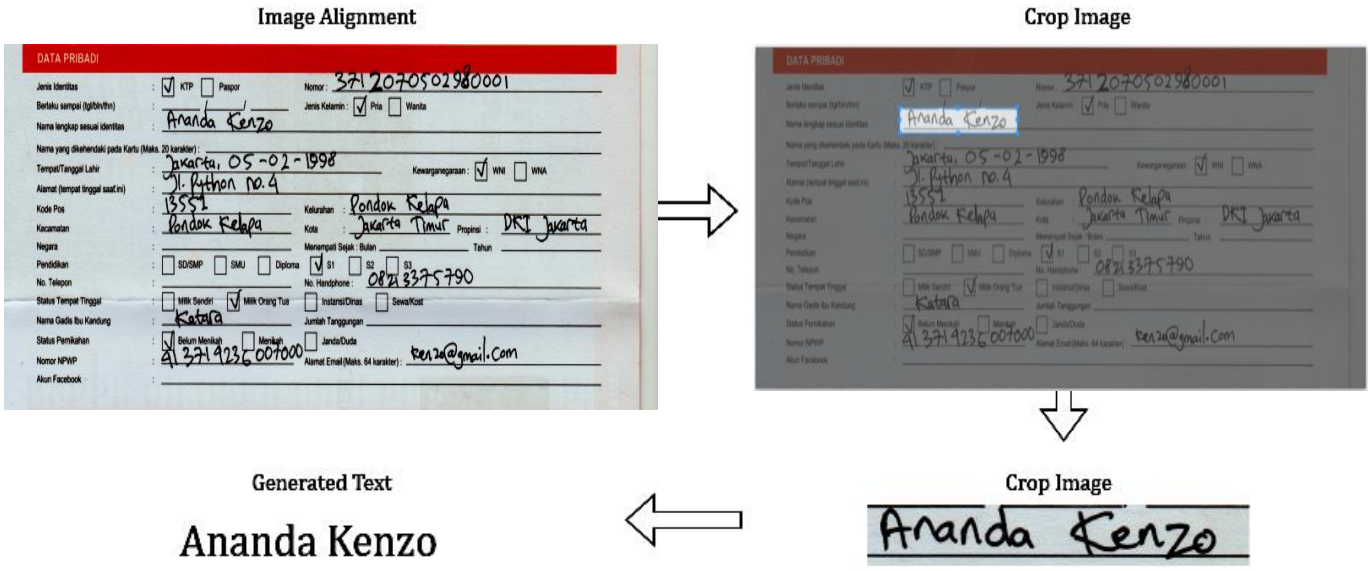

Fig. 10 Model evaluation process on bank XYZ Form

Table 4. Performance model on bank XYZ Form

Experiment Model	CER	Accuracy	Precision	Recall	F1
TrOCR_Model_1	0.5213	0.4707	0.4767	0.4587	0.4685
TrOCR_Model_2	0.4985	0.5015	0.5275	0.5345	0.5500
TrOCR_Model_3	0.4816	0.5264	0.5174	0.5424	0.5184
TrOCR_Model_4	0.3620	0.6880	0.6380	0.6743	0.6301

The text successfully predicted by the model in each form will be saved in dataframe form and compared with the original label. Table 4 shows model evaluation results on 50 XYZ bank forms using CER. Based on Table 4, the model with preprocessing and hyperparameter tuning shows a very low Character Error Rate (CER) of 0.3620. In contrast, the model without preprocessing and hyperparameter tuning has a significantly higher CER of 0.5213.

This indicates that preprocessing and hyperparameter tuning effectively reduce character recognition errors and improve the accuracy of the predictive model. Adjusting brightness helps address images with low brightness levels, while the hue parameter enhances image quality by increasing contrast. Additionally, kernel and sigma parameters help reduce noise in the images and assist in separating objects from the background. Beam search parameter tuning allows the model to identify the most likely sequence of words from the character recognition output. The beam search parameters are set to achieve high accuracy while limiting word searches that could slow down the recognition process.

6. Conclusion

Based on Table 4, the model that underwent preprocessing and fine-tuning with beam search parameters exhibited a relatively small Character Error Rate (CER) of 0.3620 compared to the model without preprocessing and fine-tuning, which had a considerably higher CER of 0.5213. This indicates that preprocessing and fine-tuning with beam search parameters can effectively reduce errors in character identification and enhance the accuracy of the model predictions. This paper proposes fine-tuning TrOCR based on the handwritten text for handwriting recognition in the registration forms of XYZ bank credit cards, intended for use in system automation for data input. The research combines preprocessing stages on the data and fine-tuning of beam search parameters. The experimental results demonstrate that the TrOCR model, when fine-tuned and subjected to preprocessing stages, achieved a significantly low CER on the IAM dataset test data, with a CER of 0.0607 based on Table 3 and 0.3620 on 50 XYZ bank forms based on Table 4. This research contributes to developing models for automated data input systems for credit card forms. There are various potential areas for future development, such as: 1) the TrOCR base handwritten model utilized has a substantial size of 4.96 GB, where Parameter-Efficient Fine-Tuning (PEFT) techniques could be applied to reduce the number of trainable parameters used, and 2) preprocessing stages during form prediction could be enhanced, with image resolution and brightness levels potentially increased to improve model accuracy. Beyond the credit card registration domain, this study's approach and findings have broader applications in various document automation and digitization tasks. The fine-tuned TrOCR model, optimized for handwritten text recognition, can be adapted for processing a wide range of handwritten forms, such as government applications, hospital intake forms, academic records, and legal documents. These fields often face challenges with manual data entry, similar to banking, and could benefit from automated OCR systems that improve efficiency, accuracy, and scalability. Furthermore, the techniques employed here - particularly preprocessing strategies and beam search tuning - provide a replicable blueprint for improving OCR performance in diverse document-processing environments.

Acknowledgements

This research is supported by funding from the Directorate of Research, Technology, and Community Service of the Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia, under the scheme of Hibah DRTPM 2024 with the main contract number 105/E5/PG.02.00.PL/2024; 784/LL3/AL.04/2024; 092/VRRTT/VI/2024 on 30th May 2024.

References

[1] Bank Indonesia, Jumlah Pengguna Kartu Kredit di Indonesia, 2024. [Online]. Available: https://www.bi.go.id/id/statistik/ekonomi-keuangan/spip/Pages/SPIPSeptember-2023.aspx

[2] Bank Indonesia, Regional Credit Card/KK, Indonesia, 2024. [Online]. Available: https://www.bi.go.id/id/statistik/ekonomi-keuangan/spip/Documents/TABEL_5d.pdf#search=kartu%20kredit

[3] Ashish Vaswani et al., "Attention is All You Need," *arXiv Preprint*, pp. 1-15, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[4] Minghao Li et al., "TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models," *arXiv Preprint*, pp. 1-10, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[5] Daniel Hernandez Diaz et al., "Rethinking Text Line Recognition Models," *arXiv Preprint*, pp. 1-11, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[6] Bhavesh Kumar, "ViT Cane: Visual Assistant for the Visually Impaired," *arXiv Preprint*, pp. 1-4, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[7] Jacob Devlin et al., "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding," *arXiv Preprint*, pp. 1-16, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[8] Cheng Shuiqi, Guo Yanliang, and Chen Yue, "A Bank Card Number Recognition Method Based on Convolutional Neural Network," *International Conference on Communications, Information System and Computer Engineering*, Beijing, China, pp. 826-830, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[9] Yuanxue Xin, Pengfei Shi, and Song Han, "An Automatic Location and Recognition Method for Bank Card Number," *Proceedings of the 2019 International Conference on Robotics, Intelligent Control and Artificial Intelligence*, Shanghai China, pp. 728-732, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[10] Gege Sun, and Fucheng You, "Bank Card Number Recognition System based on Deep Learning," *Proceedings of the 2020 4th International Conference on Electronic Information Technology and Computer Engineering*, Xiamen China, pp. 745-749, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[11] Mukesh Jha et al., "Automation of Cheque Transaction using Deep Learning and Optical Character Recognition," *International Conference on Smart Systems and Inventive Technology*, Tirunelveli, India, pp. 309-312, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[12] Zhijie Lin et al., "SNRNet: A Deep Learning-Based Network for Banknote Serial Number Recognition," *Neural Process Letters*, vol. 52, no. 2, pp. 1415-1426, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[13] William Watson, and Bo Liu, "Financial Table Extraction in Image Documents," *Proceedings of the First ACM International Conference on AI in Finance*, New York, pp. 1-8, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] Firhan Maulana Rusli, Kevin Akbar Adhiguna, and Hendy Irawan, "Indonesian ID Card Extractor Using Optical Character Recognition and Natural Language Post-Processing," *arXiv Preprint*, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[15] Emilda Zhang, Vincent Ardyan Putra, and Gede Putra Kusuma, "Improving Optical Character Recognition Accuracy for Indonesia Identification Card Using Generative Adversarial Network," *Journal of Theoretical and Applied Information Technology*, vol. 100, no. 8, pp. 2424-2437, 2022. [Google Scholar] [Publisher Link]

[16] Xin Ma, and Wei Qi Yan, "Banknote Serial Number Recognition Using Deep Learning," *Multimedia Tools and Applications*, vol. 80, no. 12, pp. 18445-18459, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[17] Shriansh Srivastava et al., "Optical Character Recognition on Bank Cheques Using 2D Convolution Neural Network," *Applications of Artificial Intelligence Techniques in Engineering*, vol. 697, pp. 589-596, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[18] R. Parthiban, R. Ezhilarasi, and D. Saravanan, "Optical Character Recognition for English Handwritten Text Using Recurrent Neural Network," *International Conference on System, Computation, Automation and Networking*, Pondicherry, India, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[19] Adith Narayan, and Raja Muthalagu, "Image Character Recognition using Convolutional Neural Networks," *Seventh International conference on Bio Signals, Images, and Instrumentation*, Chennai, India, pp. 1-5, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[20] Yefan Gao et al., "Bank Card Number Recognition System based on Deep Learning," *Proceedings of the 2019 International Conference on Artificial Intelligence and Advanced Manufacturing*, Dublin, Ireland, pp. 1-6, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[21] U.-V. Marti, and H. Bunke, "The IAM-Database: An English Sentence Database for Offline Handwriting Recognition," *International Journal on Document Analysis and Recognition*, vol. 5, no. 1, pp. 39-46, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[22] Hugo Touvron et al., "Training Data-Efficient Image Transformers & Distillation through Attention," *arXiv Preprint*, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[23] Yinhan Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *arXiv Preprint*, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[24] Da Chang, and Yu Li, "Mixed Text Recognition with Efficient Parameter Fine-Tuning and Transformer," *arXiv Preprint*, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[25] Felix Stahlberg, and Bill Byrne, "On NMT Search Errors and Model Errors: Cat Got Your Tongue?," *arXiv Preprint*, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[26] Philipp Koehn, and Rebecca Knowles, "Six Challenges for Neural Machine Translation," *arXiv Preprint*, 2017. [CrossRef] [Google Scholar] [Publisher Link]