applied sciences

MDPI

Article

A Pipeline Approach to Context-Aware Handwritten Text Recognition

Yee Fan Tan [1], Tee Connie [1,*], Michael Kah Ong Goh [1] and Andrew Beng Jin Teoh [2,*]

[1] Faculty of Information Science and Technology, Multimedia University, Melaka 75450, Malaysia; 1171202077@student.mmu.edu.my (Y.F.T.); michael.goh@mmu.edu.my (M.K.O.G.)
[2] School of Electrical and Electronic Engineering, College of Engineering, Yonsei University, Seoul 03722, Korea
* Correspondence: tee.connie@mmu.edu.my (T.C.); bjteoh@yonsei.ac.kr (A.B.J.T.); Tel.: +60-62523592 (T.C.)

Featured Application: The proposed handwritten text recognition pipeline can be used for practical documents transcription and context recognition.

Abstract: Despite concerted efforts towards handwritten text recognition, the automatic location and transcription of handwritten text remain a challenging task. Text detection and segmentation methods are often prone to errors, affecting the accuracy of the subsequent recognition procedure. In this paper, a pipeline that locates texts on a page and recognizes the text types, as well as the context of the texts within the detected region, is proposed. Clinical receipts are used as the subject of study. The proposed model is comprised of an object detection neural network that extracts text sequences present on the page regardless of size, orientation, and type (handwritten text, printed text, or non-text). After that, the text sequences are fed to a Residual Network with a Transformer (ResNet-101T) model to perform transcription. Next, the transcribed text sequences are analyzed using a Named Entity Recognition (NER) model to classify the text sequences into their corresponding contexts (e.g., name, address, prescription, and bill amount). In the proposed pipeline, all the processes are implicitly learned from data. Experiments performed on 500 self-collected clinical receipts containing 15,297 text segments reported a character error rate (CER) and word error rate (WER) of 7.77% and 10.77%, respectively.

Keywords: handwritten text recognition; Residual Network; Transformer model; object detection; named entity recognition

Citation: Tan, Y.F.; Connie, T.; Goh, M.K.O.; Teoh, A.B.J. A Pipeline Approach to Context-Aware Handwritten Text Recognition. *Appl. Sci.* **2022**, *12*, 1870. https://doi.org/10.3390/app12041870

Academic Editors: Cheonshik Kim and Byung-Gyu Kim

Received: 17 December 2021
Accepted: 10 February 2022
Published: 11 February 2022

1. Introduction

Handwritten text recognition (HTR) has gained enormous research interest due to the potential benefits that can be derived from accurate text transcription that eases attempts to digitize handwritten content [1,2]. An HTR system is applicable to a myriad of scenarios, ranging from reading bank cheque amounts to transcribing medical records and notes [3]. Although highly desirable in practical applications, HTR is faced with a number of challenges.

The current HTR systems generally apply a Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) model for text transcription [3]. However, a variety of text styles, such as printed texts, handwritten texts, scribble, and images, exists in real-life documents. Therefore, using a single text recognition model is insufficient for the text transcription task. Ingle et al. proposed a text style classification approach using an LSTM-based and fully feed-forward network model for line-level segmentation. An optimal model was determined when the calculated probability of a particular class was greater than a predefined threshold [3]. In addition, Singh and Karayev presented a study on HTR by decomposing an image into one or more regions as texts, mathematical equations, tables, and scratched texts using a Residual Network (ResNet) [4]. These studies

have demonstrated a way for precise text transcription by ignoring unknown or unreadable regions, and the proposed approaches are applicable in a wide range of applications, covering simple to complex scenarios. Nevertheless, the line-based segmentation methods [3] sometimes fail to recognize the texts correctly due to difficulty in segmenting the image into lines accurately. Full page-based models that only transcribe a particular region of texts in the page while skipping others [4], on the other hand, can only work well on a balanced dataset, such as when the layout of the page is the same.

In the literature, the combination of CNN with Recurrent Neural Network (RNN) [5,6] and LSTM [7] are widely applicable for sequence modeling in HTR. However, the RNN variations face vanishing and exploding gradient problems, where the models fail to learn the long sequence information [8]. Recently, the Transformer model with an attention mechanism has been introduced, and it has demonstrated superior performance over the conventional RNN and LSTM models for long sequence information processing [4,9]. The Transformer model yields outstanding performance on public benchmark datasets, especially on the IAM dataset [10]. For example, a CNN and LSTM architecture integrated with a Transformer model was applied on the IAM dataset and achieved a character error rate (CER) of 8.50% [2]. Another study that applied the Transformer model to handwriting document recognition reported a CER of 6.30% on the IAM dataset at paragraph-level [4].

In general, an HTR system development pipeline comprises two stages: (1) text localization and (2) text recognition [11]. Real-life documents generally contain a combination of text types such as printed texts, handwritten texts, signatures, and others. How to correctly localize and recognize these text types has become pivotal to avoid bias and ensure the right data sample distribution. Many solutions have been proposed to better perform these tasks [3,4,9], but there is still room for improvement. Apart from accurately recognizing the handwritten texts, how to associate the meaning or context of the recognized texts is also crucial to enable the automatic documentation of the transcribed texts. Being able to meet the computational requirement of a real-life HTR system is also of paramount importance.

In this paper, a context-aware HTR pipeline is proposed to overcome the limitations of the existing HTR systems by considering the accuracy and efficiency of text type classification and localization, text recognition, and text context recognition on real-life documents as a whole. Clinical receipts are used as the subject of study as they contain a combination of printed items on the receipts, handwritten texts of the clinicians, as well as non-text elements, such as the logo of the clinics. Towards this end, a dataset containing 500 samples of clinical receipts has been collected. The documents are further segmented based on regions such as patient names, address, prescription, and bill amount, yielding a total of 15,297 text segments.

The proposed HTR pipeline consists of a You Only Look Once v5 (YOLOv5) model for text localization and type classification, followed by a Residual Network with Transformer (RESNET-101T) for text recognition, and a Named Entity Recognition (NER) model for text context recognition. An integrated model comprising ResNet-101 and Transformer, which is coined as ResNet-101T, is introduced in this paper. ResNet-101 acts as a feature extractor, whose output is fed into the Transformer model to perform text recognition. ResNet-101 is well-known for its ability to alleviate the effect of vanishing gradient and avoid performance degradation when the network's depth is increased. The Transformer model is selected due to its outstanding performance in handling sequential data, and it has a low inductive bias compared to conventional RNN architectures. Nevertheless, the Transformer model requires a huge dataset for training. Therefore, data augmentation is performed to ensure the model is properly trained with a sufficient amount of data. The proposed pipeline aims to study the applicability of data-driven DL models on a close-to-real-life dataset, where it contains much noisier and challenging data compared to the existing benchmark datasets. The contributions of this paper are highlighted as follows:

- A context-aware HTR pipeline made up of a series of carefully chosen pre-processing, text recognition, and context interpretation funnels is presented to deal with a close-to-real-life handwritten text dataset. The pipeline is designed to locate texts on a page

and is able to recognize the text types such as handwritten text, printed text, non-text, as well as the text context.
- A ResNet-101T model that has a better ability in handling sequence data compared to RNN variations is proposed for text recognition. The proposed model is compared with the state-of-the-art HTR methods, including CNN-LSTM and Vision Transformer.
- A NER model is proposed to complement the pipeline to recognize the context of the transcribed texts. Transcription of the document can be performed in a fully automatic way.

The remaining paper is organized as follows. Section 2 presents the literature review. Section 3 describes the data and theoretical backgrounds of the methods applied for the proposed pipeline. Then, Section 4 presents the experimental result, and Section 5 discusses the insights into the proposed pipeline. Finally, the conclusion and future works are drawn in Section 6.

2. Literature Review

The emergence of deep learning (DL) has brought significant advances to HTR. The DL models have progressively improved the performance of HTR transcription over the years. This section discusses the different ideas proposed to solve HTR and the performance achieved.

Bluche presented an approach for joint line segmentation for HTR transcription [11]. The dataset was taken from paragraph-level images from the RIMES and IAM datasets. The author proposed the integration of the attention mechanism with the Multi-dimensional Long Short-Term Memory Recurrent Neural Network (MDLSTM-RNN) for implicit text segmentation and transcription. The collapse layer of the model was modified with an attention mechanism to provide the weights in identifying the input positions on a paragraph image iteratively to enable a free segmentation method for text transcription. As a result, the model achieved a CER of 4.9 and 2.5 on the IAM and RIMES datasets, respectively, containing images with 300 dpi resolution.

Wigington et al. proposed a model of Start, Follow, and Read (SFR) for historical HTR [12]. The SFR model could identify the text position by using a Region Proposal Network and a CNN-LSTM model for text transcription. The proposed SFR model was composed of a Start-of-Line (SOL) finder, which identified the text line of a given image, a Line Follower (LF), which segmented the position identified by the SOL finder iteratively and, lastly, the HTR model. The 2017 ICDAR full-page competition dataset was applied in the study of German handwriting. The proposed model achieved outstanding performance, with a BLEU score of 72.3. The model was also evaluated on the RIMES and IAM datasets, achieving a CER and WER of 2.1 and 9.3, as well as 6.4 and 23.2, respectively.

There was another study that applied CNN to a Kannada handwritten document [13]. In the paper, the author proposed CNN for training the data. In the experiment setup, the Chars74K dataset containing over 657 classes was applied. Each class has 25 handwritten characters. Data augmentation techniques, including denoising, contrast normalization, segmentation, gray-scale conversion, and binarization, were performed to expand the dataset size. After a hundred epochs of training, the model achieved 99% accuracy on the Chars74k dataset and 96% on a self-collected handwritten document.

Ingle et al. conducted a study for a scalable HTR system [3]. The authors integrated the proposed HTR system into a larger-scale OCR system. In the study, the authors applied LSTM-based models and gated recurrent convolutional layers (GRCL) as a fully feed-forward network model for line-level text recognition and classification. Online handwritten data from the IAM offline and online databases and a self-collected online handwritten sample were used. The authors trained separate models for both printed and handwritten words. The dataset used for both printed and handwritten text consisted of 508 and 433 images. After hyperparameter tuning, the proposed GRCL achieved a character error rate (CER) and word error rate (WER) of 4.0 and 10.8, respectively.

Wu et al. presented a method to detect and recognize handwritten text and text-line [14]. The authors presented a method named Multi-Level Convolutions Convolutional and Recurrent Network (MLC-CRNN), which combined different deep learning techniques, including CNN, RNN, and Connectionist Temporal Classification (CTC) loss function. In the paper, the Connectionist Text Proposal Network (CTPN) was used in training a new model for a handwriting text-line detector. Following handwriting text recognition, the team applied a refined CRNN model. To make it a multi-layer convolution (MLC), two more branches were added linearly on the original convolutional layers. The datasets used for training were obtained from three hundred students. The participants were asked to write on a standard answer sheet. The training set contained 3883 images, and the testing set contained 297 images. The proposed MLC-CRNN model, when integrated with two MLC modules, achieved the best performance by obtaining an accuracy of 91.4%.

Singh and Karayev presented a study that applied full-page handwriting document recognition using the Transformer model [4]. The authors aimed to recognize handwritten texts in a full-page manner. The model consists of a CNN network to extract the features from the document, followed by Transformer as an image-to-sequence model, which learns to map an image to a sequence. The model was trained on various datasets, including IAM, WIKITEXT, FREE FORM ANSWER, ANSWERS2. The model was then evaluated on the FREE FORM ANSWERS dataset and obtained a CER of 7.6%. Despite the promising performance, the method suffers from a biased multi-task problem. For example, if the model is trained using datasets that only contain one transcribed text region per sample (like the Free Form dataset), the model will have a tendency to transcribe only one main text region while skipping the others due to its full-page recognition nature. This is a challenge for text recognition. It is important to have a robust text recognition system that can deal with different text types, including printed texts, handwritten texts, and non-texts. Table 1 summarizes the different papers discussed in this section.

Table 1. Summary of included studies.

Author	Subject of Study	Proposed Solution	Dataset	Experimental Results
Bluche (2016) [11]	Joint line segmentation and transcription	MDLSTM-RNN	RIMES and IAM database	CER of 4.9 and 2.5 on IAM and RIMES data, respectively
Wigington et al. (2018) [12]	Historical document processing	SFT	ICDAR 2017 competition dataset, IAM, RIMES	BLEU score of 72.3 on ICDAR dataset. CER and WER of 2.1 and 9.3, 6.4 and 23.2 on both IAM and RIMES datasets, respectively
Asha and Krishnappa (2018) [13]	Kannada Handwritten Document Recognition	CNN	Chars74K dataset	99% of accuracy
Ingle et al. (2019) [3]	Line-level text style classification and recognition	GRCL	IAM online and offline database, self-collected handwritten online samples	CER and WER of 4.0 and 10.8
Wu et al. (2020) [14]	Recognition of handwritten text and text-line	CTPN to detect text lines, MLC-CRNN for text recognition	3883 training images and 297 testing images	Accuracy of 91.4%
Sign and Karayev (2021) [4]	Full-page handwritten document recognition	Transformer	IAM, WIKITEXT, FREE FORM ANSWERS, ANSWERS2	CER of 7.6%

3. Proposed Method

3.1. Data Collection

In this study, a dataset consisting of 500 clinical receipts is collected. The dataset is composed of 10 variants of medical receipts with 50 samples each. The empty receipt templates are obtained from online resources, with a resolution of 300 dpi and above. During data collection, empty medical receipts were distributed to the participants to fill with their own handwriting. No restriction was imposed on how the participants should write on the receipts. The participants come from various backgrounds and professions, were aged between 12 and 50 at the time of the study, and are from Malaysia. Every participant is literate and can write independently, with no known disability. A blank printed copy of the receipt template was given to them to fill. After the form was filled, the filled form was collected and scanned. Figure 1 shows some samples of the collected handwritten receipts.

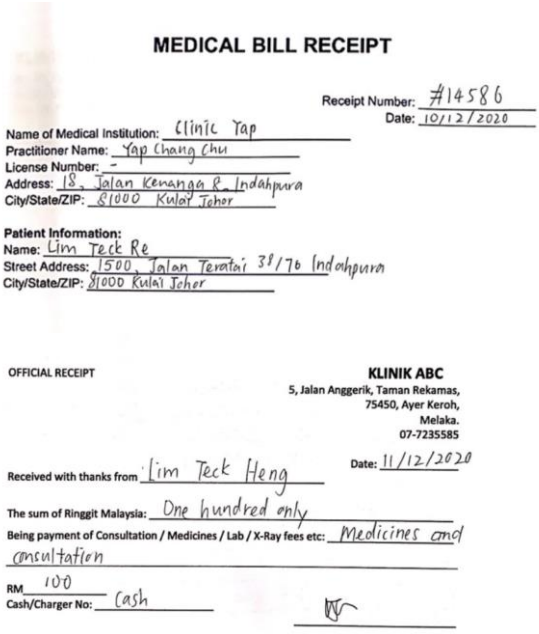


Figure 1. Samples of collected handwritten clinical receipts.

3.2. The Proposed Context-Aware HTR Pipeline

The proposed HTR pipeline, from pre-processed input to context recognition, is illustrated in Figure 2. We would like to have a model that is aware of the receipt's content for better recognition accuracy in the pipeline. Towards this end, the printed texts and handwritten texts are separated into two different processing funnels. The You Only Look Once v5 (YOLOv5) [15] model is applied to distinguish the handwritten texts from printed texts and non-text elements for more precise region of interest (ROI) localization. After that, an Optical Character Recognition (OCR) model is used to identify the printed texts, while the handwritten texts are recognized using a Residual Network (ResNet-101) and Transformer architecture (ResNet-101T). In this paper, Tesseract [16], a matured open-source OCR model, is applied for printed text recognition, and thus, no further evaluation is made for the model. Subsequently, both outputs from the text recognition models are processed by a Named Entity Recognition (NER) model to identify the context of the texts.

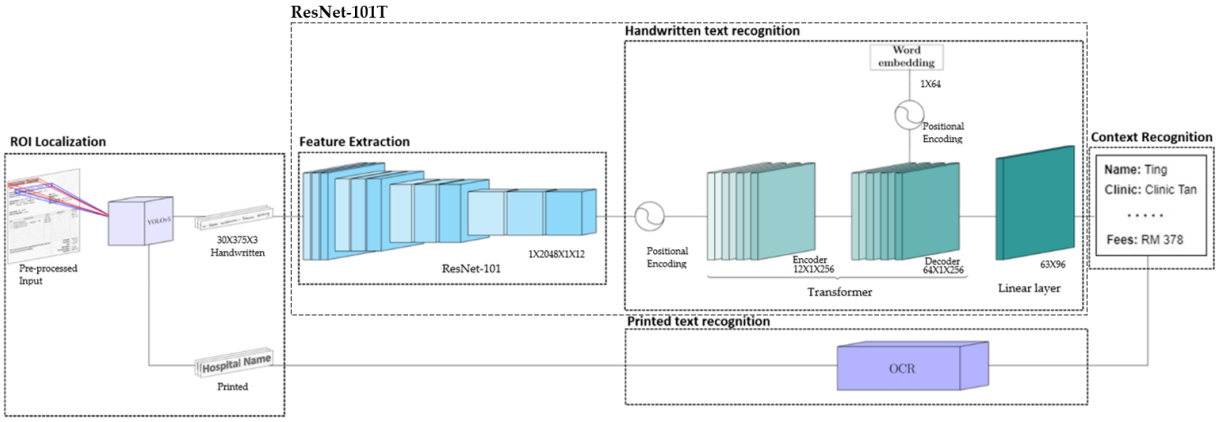


Figure 2. HTR pipeline process and architecture.

3.2.1. Pre-Processing

During the data pre-processing stage, various procedures were applied to ensure that the data were in an appropriate form for further model training. Considering different noises that might occur in a real-life document image, we took several aspects into consideration. We identified two problems: (a) the image is taken at a slanted or skewed angle; (b) there are lines in the image that could affect classification performance. Progressive Probabilistic Hough Line Transform (PPHLT) [17,18] was applied to rotate the image to the correct angle. The method first detects the image's edges by applying canny edge detection. The image is then rotated accordingly based on the computed angle from the PPHLT algorithm. Next, morphological operations and image inpainting [19] are applied to remove the lines from the image. After that, trimming is used to remove excessive white pixels, such as in the border regions in the image.

A YOLOv5 model has been trained to identify the text region of interest (ROI). A text ROI is categorized into three types: printed texts, handwritten texts, and non-text. More information about ROI localization and categorization is given in the next section. The segmented regions are then padded into the same size. For model training, the text ROIs are labeled manually as printed, handwritten, and others. There are a total of 5099 handwritten text segments, 7445 printed text segments, and 261 non-text segments in this study. Figure 3 shows a sample of the labeled image.

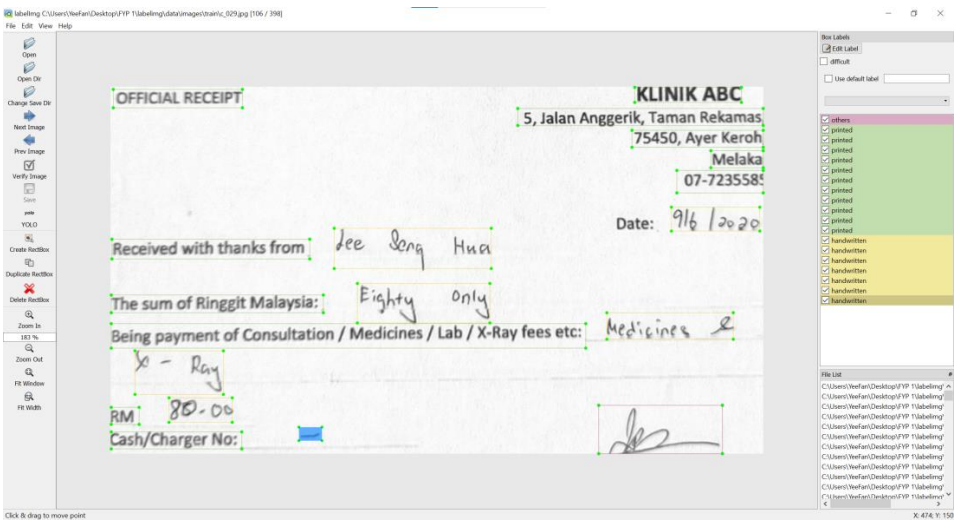


Figure 3. Sample labeled image.

The text segment is augmented randomly by reducing or adding line width, adding Gaussian noise, and blurring the images. The final dataset is composed of 15,297 segments of handwritten text images. Figure 4 presents the flow of the proposed pre-processing approach. Some samples of the text segment and the corresponding augmented images are shown in Figure 5.

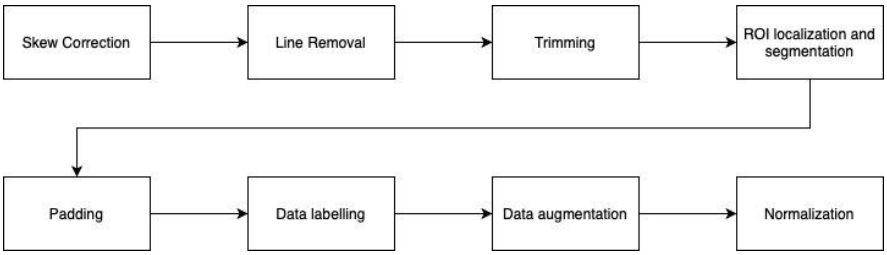


Figure 4. The flow of pre-processing approach.

Figure 5. Samples of segmented and augmented images.

3.2.2. ROI Localization and Categorization

Text ROI location and categorization are crucial in making sure that the input fed to the HTR model is of good quality. Towards this end, YOLOv5, which is a successful object classification and detection method, is deployed. YOLOv5 works by dividing an image into a grid system, and the object will be detected within the cell of the grid. YOLOv5 is established and refined based on the YOLOv3 method presented by Joseph and Ali [20]. No academic publication for YOLOv5 is available; hence, the theoretical background of YOLOv3 is provided.

YOLOv3 predicts the coordinates of a bounding box, t_x, t_y, t_w, t_h, where x, y, w, h represent the x and y coordinates, width, and height. Sum squared error is used for training the model. Additionally, the model uses logistic regression to measure the objectness score, also known as the probability of being classified into a particular object. The feature extractor used consists of 53 layers (DarkNet-53) and is more efficient in utilizing the GPU for faster evaluation. Generally, YOLOv3 predicts the bounding boxes at three different scales, and features are extracted from those scales. The outcome is a 3d-tensor of the bounding box, objectness score, and classes prediction. The bounding boxes b_x, b_y, b_w, b_h are defined in Equation (1).

$$\begin{aligned} b_x &= \sigma(t_x) + c_x \\ b_y &= \sigma(t_y) + c_y \\ b_w &= p_w e^{t_w} \\ b_h &= p_h e^{t_h} \end{aligned} \tag{1}$$

where c_x, c_y are the offset from the top left corner of the image, and p_w, p_h are width and height of the bounding box prior. σ stands for the sigmoid function. Built upon the fundamentals of YOLOv3, different variants for YOLOv5 have been introduced [15]. Some examples include YOLO-v5n with the least parameters of 1.9 million, YOLO-v5s with 7.2 million parameters, YOLO-v5m with 21.2 million parameters, YOLO-v5l with 46.5 parameters, and YOLO-v5x with the most parameters of 86.7 million.

3.2.3. Residual Network with Transformer (ResNet-101T)

Resnet is proposed by He et al. [21] as a residual learning framework for better neural network training with a deeper architecture. The Residual network has shortcut connections inserted to the network to make a counterpart of the Residual version. There are ResNets with different lengths of layers, which can be implied from their names; for example, ResNet-101 stands for a ResNet architecture containing 101 layers.

In residual learning, the stacked layers are the building block for feature mapping. A building block can be defined as:

$$y = F(x, \{W_i\} + x) \tag{2}$$

where x is the input vector, y is the output vector, and the function, F stands for residual mapping. The operation for F can be performed by applying a shortcut connection or element-wise addition.

On the other hand, Transformer is a neural network proposed by Vaswani et al. [9], that applies an attention mechanism to connect the encoder and decoder. The encoder maps the input to a sequence of continuous representation. Given the mapped sequence, the decoder generates the output of one element at a time, where the model is autoregressive at each step as it uses the generated output at the previous step as additional input for output generation. Transformer with attention mechanism is introduced to tackle the problem of memory constraints in RNN variations by modeling the dependencies without considering the input and output sequence distances. Transformer has an advantage in that it has a relatively low inductive bias compared to RNN.

The encoder stack contains a multi-head attention mechanism, which is followed by a feed-forward neural network layer, and each layer is followed by a normalization layer. As opposed to the encoder, the decoder contains more layers. The output of the encoder is inserted into the third layer of the decoder. Additionally, a linear layer is appended to the end of the decoder for output prediction. Figure 6 shows the decoder–encoder structure of the Transformer model.

The attention in the Transformer model is known as scaled-dot attention. The attention function is defined as:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

where Q is a matrix of a set of queries of the attention function, and K and V stand for keys and values. Dot-product attention is used as it is faster and has better space efficiency compared to additive attention. However, the attention is implemented as a multi-head mechanism. Thus, the outputs of the attention are concatenated as:

$$Multiheaad\ attention(Q, K, V) = Concatenate(head_1, \ldots, head_i)W^O \tag{4}$$

where $head_i$ stands for $Attention\left(QW_i^Q, KW_i^K, VW_i^V\right)$. The multi-head attention is applied in the encoder–decoder of the Transformer model to allow the decoder to look through every position of the input sequences, coming from the queries of previous layers and the memory keys and values. Additionally, the self-attention layers in the encoder and decoder allow the output of previous layers to be attended for its position.

The Transformer model uses positional encoding, PE, to make use of the sequence order by providing information of the token of the sequences,

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{\mathrm{d}_{\mathrm{model}}}}}\right),\ PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{\mathrm{d}_{\mathrm{model}}}}}\right) \tag{5}$$

where pos and i are the position and dimension of the input, and $\mathrm{d}_{\mathrm{model}}$ is the hidden size of the Transformer model.

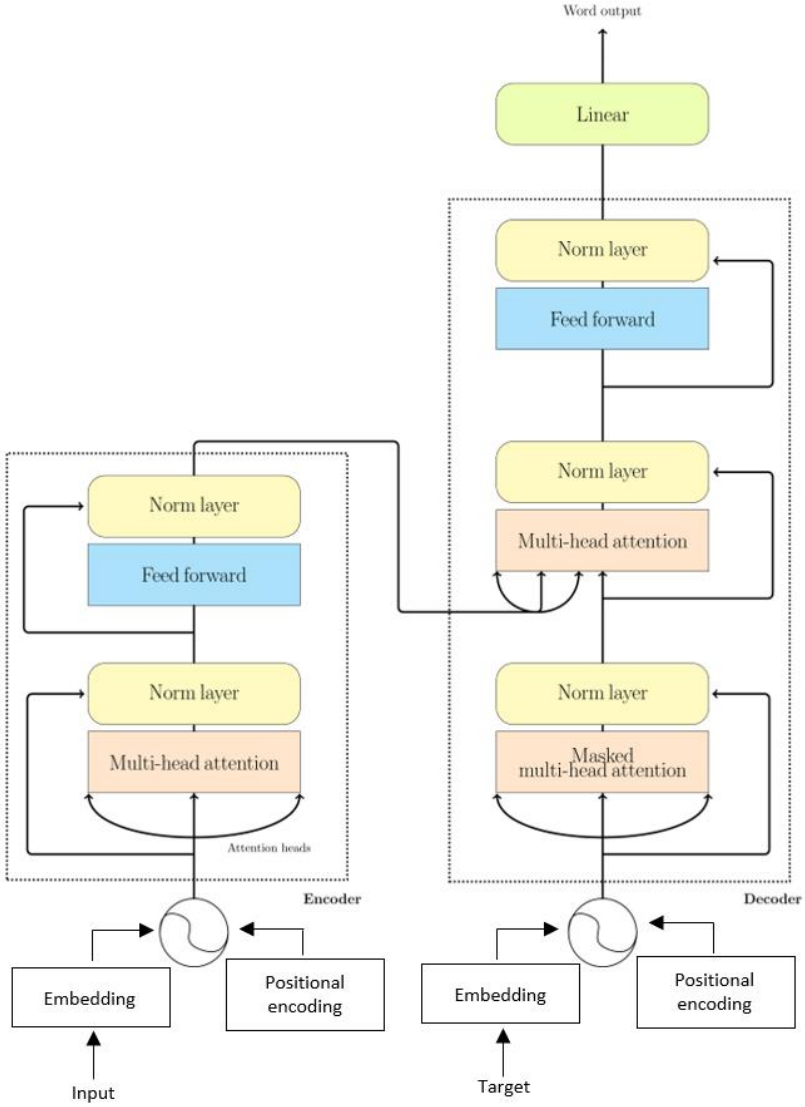


Figure 6. Structure of the Transformer model.

3.2.4. Named Entity Recognition (NER)

The NER model is used to identify the context of the recognized texts from the transcribed receipt. In this paper, a Natural Language Processing (NLP) model called spaCy [22] is applied. spaCy allows the model to recognize a wide range of words or entities. The NER model is composed of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) models [23], and it is processed based on a transition-based model described in the paper by Lample et al. [24].

A transition-based model directly identifies the suitable representations of a multi-token. The model is built as a stacked data structure to obtain the input's chunks in predicting the following actions. Lample et al. used a stacked LSTM model, enabling the stacked object embedding through the push and pop operations [24]. Stack LSTM is used to compute the dimensional embedding of the stack content, buffer, output, and actions taken at each time step, which represents the distribution of the possible action at each time step. Thus, the model aims to maximize the conditional probability of action sequences based on the given sentence input. The maximum probability of the action is chosen until the chunking algorithm meets the termination condition.

3.3. Implementation Details

This section describes the training procedure of the proposed pipeline. Firstly, the YOLO-v5 model is used for multi-text type classification. In this study, the YOLO-v5x model was chosen as it has shown a very promising result from the established results. An annotation software called LabelImg [25] was used to annotate the collected dataset to train the YOLOv5 model. The image is segmented into three classes (printed, handwritten, and non-text), as shown in Figure 7. The non-text category contains unreadable texts such as logos, signatures, and others. Three hundred receipt images were used, with a train and validation split of 5:1. After that, the model was trained for 100 epochs, where the best model with the highest precision score was saved. The total number of characters included in the experiment was 96, with a maximum length of 64.

(**a**) (**b**) (**c**)

Figure 7. Sample segmented images: (**a**) printed (**b**) handwritten (**c**) non-text.

After the handwritten segments were obtained from YOLOv5, we combined the ResNet-101 and Transformer models, named ResNet-101T, for HTR. ResNet-101 is used as the feature extractor, the backbone of the proposed architecture where the linear projection layer is excluded, and Transformer is used to analyze the extracted features, as inspired by [26]. In [26], CNN is used together with Transformer to achieve the object detection goal. In this paper, ResNet-101 is responsible for learning 2D representations that encompass shape/outline and positional information of the texts/words in the image. The last feature map by ResNet-101 serves as the input to Transformer, which is then flattened with 2D-positional encoding and passed to the encoder. The ResNet-101 feature map also serves as the input for the decoder, which is the target in this case. The output embeddings of the decoder are then passed to the final linear layer for predictions.

Both the ResNet-101 and Transformer (ResNet-101T) are jointly trained. Two inputs are fed into the model: handwritten segments and the label. The image is fed into ResNet-101T with the shape $30 \times 375 \times 3$. The label length is 64 characters. The input shape and the corresponding number of parameters for each layer of ResNet-101T are given in Table 2. The total number of text segments used was 15,297, and the train, test, validation split ratio was 8:1:1. The model was trained for 100 epochs, and the hyperparameters were tuned to find the optimal result in terms of character error rate (CER) and word error rate (WER). The hyperparameters that are tuned in the experiments include cell units of the ResNet-101 and the number of decoders, encoders, attention heads of the Transformer model.

Table 2. Output shape and number of parameters of each layer in the proposed ResNet-101T model.

Layer (Type)	Input Shape	Parameter
ResNet	[1, 3, 30, 375]	9408
Embedding layer	[1, 63]	25,344
Positional Encoding	[1, 63, 256]	0
Transformer Encoder	[12, 1, 256]	5,260,800
Transformer Decoder	[12, 1, 256], [63, 1, 256]	6,315,520
Linear	[1, 63, 256]	25,443
Total parameter: 12,151,651		
Trainable parameter: 12,151,651		
Non-trainable parameters: 0		

In training the NER model, a tag editor is used to annotate the text data, where the text data is obtained from the annotated text labels. Unlike the labeled data for the HTR

model recognition, the text labels are annotated for both printed and handwritten texts. The model is trained for 100 epochs, and the best model with the highest accuracy is saved. The dataset contains a total of 54,522 tokens, with 954 sentences. Tokens refer to chunks within a sentence, or the string between spaces and punctuation symbols, while sentences represent the sequence of tokens. Eleven attributes, such as the clinic's name, address, and contact details, were identified in the study.

4. Experimental Results

This section presents the experimental results of the proposed HTR pipeline. The experimental setup and performance of each model are provided in the following sections.

4.1. Experimental Setup

The proposed HTR pipeline consists of three funnels: ROI localization, handwritten word recognition, and context recognition. Each of these funnels is evaluated separately. The experiments are conducted using a The experiments are conducted using a laptop with a processor of Intel(R) Core (TM) i7-10875H CPU @2.30 GHZ and an GeForce RTX 3060 GPU manufactured by NVIDIA corporation, purchased from the supplier Illegear, Johor, Malaysia.

4.2. You Only Look Once v5 (YOLOv5)

The YOLOv5 model has demonstrated promising results in accurately identifying the ROIs of the printed text, handwritten text, and non-text. Figure 8 shows a sample output of the detected regions from a receipt, along with the confidence score. The model has a very high mean average precision (mAP) of 0.5 confidence, at 91.78%, where the mAP of confidence score of 0.5 and above show a lower expectancy, at 52.75%. Moreover, the model can achieve a precision of 91.61% and a recall of 86.24%. Throughout the training, the loss graphs of the model in detecting the bounding boxes, the error between the detected objects, and the classification loss showed a steady decreasing trend. The learning rate also decreased steadily after 20 iterations. Figure 9 illustrates the metric changes throughout the training period.

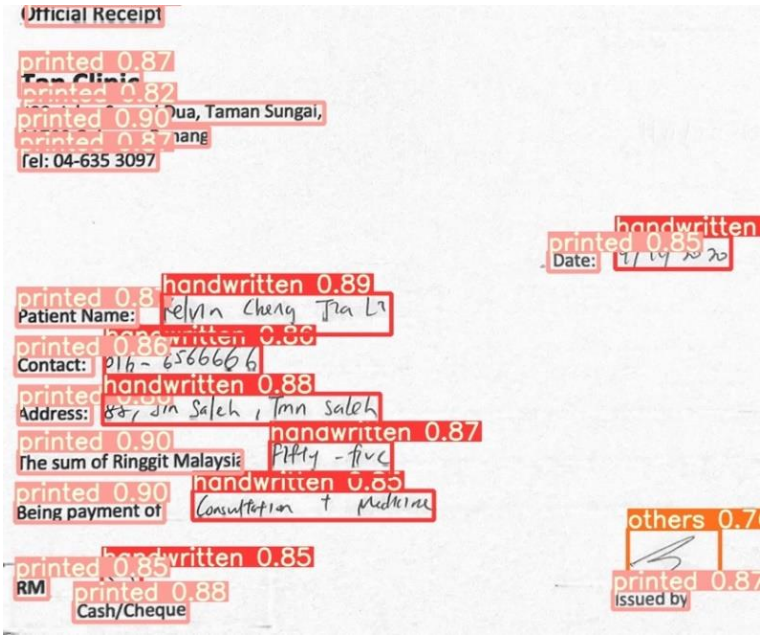


Figure 8. Sample detected output of the YOLOv5 model.

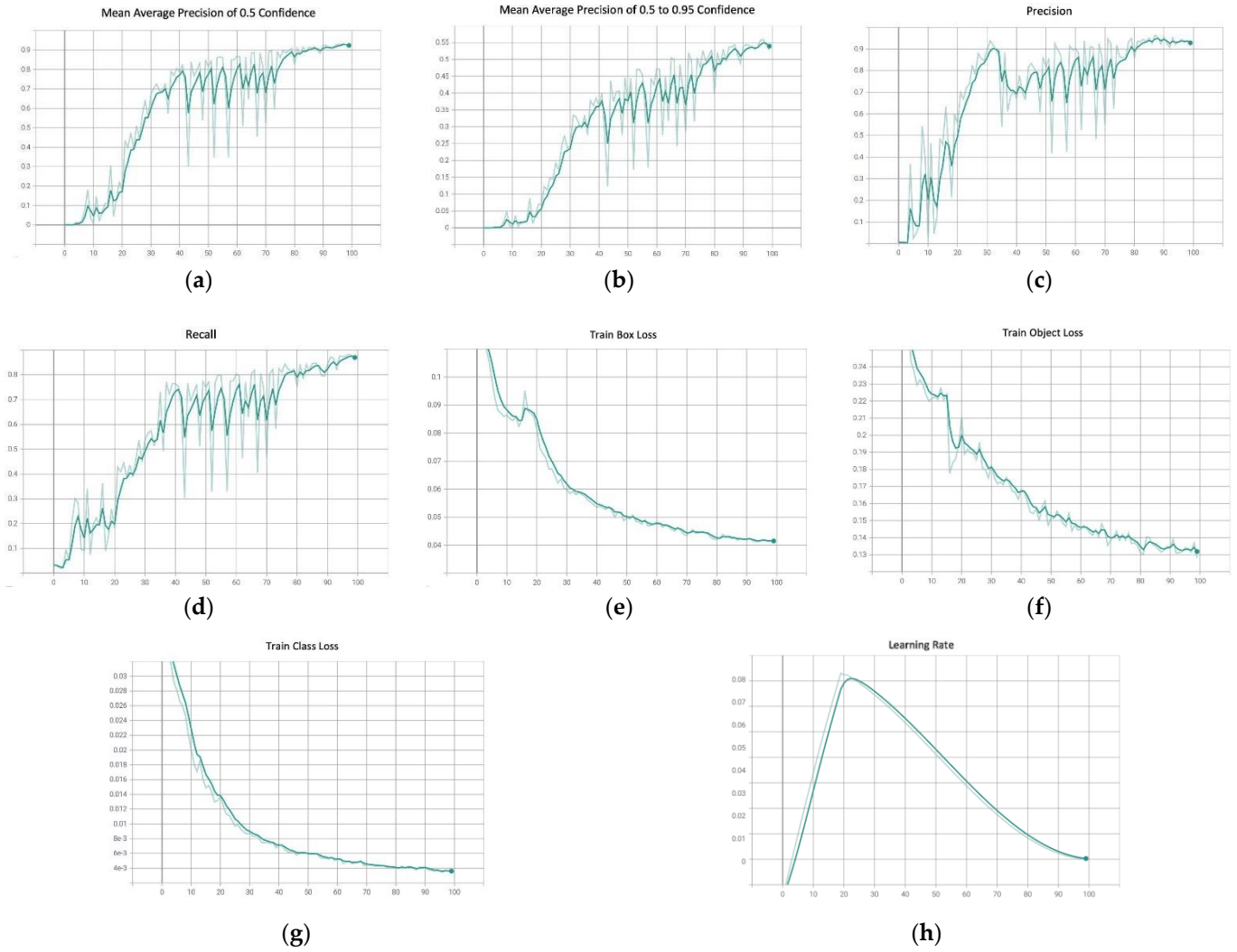


Figure 9. Experimental results of the Yolov5 model: (**a**) mean average precision of 0.5 confidence; (**b**) mean average precision of 0.5 to 0.95 confidence; (**c**) precision; (**d**) recall; (**e**) train box loss; (**f**) train object loss; (**g**) train class loss; (**h**) the learning rate.

4.3. ResNet-101 with Transformer (ResNet-101T)

The proposed ResNet-101T model is trained for 100 epochs. ResNet-101 is used for feature extraction, where the extracted features are fed as input to the Transformer to identify the underlying information based on the image pixels. The ResNet output is in a 2D format as the last two layers are dropped. Figure 10 shows some examples of the extracted feature maps from the first layer of ResNet. The topology of the words is still visible in the ResNet output, which encodes positional information, i.e., word sequence. After that, a 2D positional encoding is used to flatten the 2D representation into a 1D sequence. We believe the feature vector possesses some sequential information. This is where the Transformer model plays a role in processing the sequential data.

Figure 10. Example of extracted features of the first layer of ResNet-101.

The model achieved a character error rate (CER) and word error rate (WER) of 7.77% and 10.77% on the testing data. In addition, the model demonstrated a stable decrease in both the training and validation losses. To improve the performance of the proposed model,

hyperparameter tuning was carried out. The hyperparameters being tuned included the number of heads, encoders, decoders, and the hidden dimension size of the ResNet-101T. The hyperparameter tuning process took a very long time, as a more complex structure is more computationally intensive. Due to training time constraints, the training epoch was fixed at 100 for each setting. Table 3 shows the experimental result of hyperparameter tuning in terms of CER, WER, and time in seconds. The initial model, with the setting of 4 encoders, decoders, and attention heads, yields the best performance among the competing models. This is possible as models with a more complex structure would require a longer convergence time. Additionally, a too-complex structure might lead to overfitting. Therefore, the two models with the settings of the number of encoders, decoders, attention heads of (6, 6, 4) and (8, 8, 8), and cell units of 1024 were terminated earlier due to the extremely long training time, and no significant loss reduction was noticed after ten epochs.

Table 3. Hyperparameter tuning results of the proposed ResNet-101T model.

Number of Encoders, Decoders, Attention Heads	Unit Dimension		
	256	**512**	**1024**
4, 4, 4			
CER	7.77	8.66	20.93
WER	10.77	11.81	29.02
Times (Second)	350,864	392,254	592,497
6, 6, 4			
CER	9.15	11.87	86.76
WER	13.08	16.93	87.39
Times (Second)	388,399	487,445	73,817 (Stopped at 11)
8, 8, 8			
CER	12.96	13.35	1.0
WER	17.15	19.25	1.0
Times (Second)	429,401	539,650	87,148 (Stopped at 11)

4.4. Named Entity Recognition (NER)

There are a total of 11 attributes contained in the NER model, including medicine, payment, clinic name, address, contact, website, receipt number, name, email, and service. The model's performance is measured in terms of accuracy, entities precision, recall, and f1-score. Table 4 summarizes the model performance of each entity. Generally, the trained NER model achieved a promising result, with a full score for accuracy, precision, recall, and f1-score for all the attributes except payment. The results demonstrate that the NER model can perform well on the context recognition task.

Table 4. NER model performance on entities recognition.

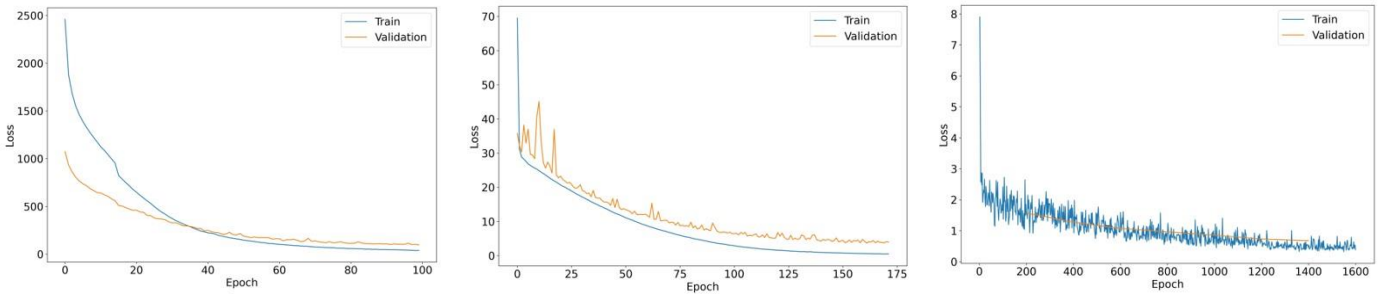
Entities	Precision	Recall	F1-Score
Medicine	1	1	1
Payment	0.9967	1	0.9983
Clinic Name	1	1	1
Address	1	1	1
Contact	1	1	1
Website	1	1	1
Receipt Number	1	1	1
Name	1	1	1
Email	1	1	1
Service	1	1	1

4.5. Comparisons with State-of-the-Art Methods

To validate the effectiveness of the proposed model, a comparison is made with LSTM [27] and Visual Transformer (ViT) [28] models. LSTM was selected for benchmark comparison as it is the state-of-the-art sequential processing method in HTR. Moreover, ViT was investigated to demonstrate the superiority of the proposed ResNet-101T method over the sole Transformer model. For a fair comparison, the same setting is imposed in all the experiments, such as the dataset split ratio. ViT was trained with a total of 1600 steps and evaluated at every 200 steps. The LSTM model achieved a CER and WER of 11.55 and 26.64, and ViT had a CER and WER of 10.60 and 18.41, where the performance of both models was inferior to ResNet-101T, as presented in Table 5. In terms of speed comparison, the LSTM model used the least computational time for training while ViT took a much longer time to train. Note that although LSTM has a faster training speed due to its simpler architecture, its accuracy is much lower than the proposed ResNet-101T model. Figure 11 shows the training and validation loss of the models throughout training.

Table 5. Result comparison of Transformer and LSTM.

Model	CER	WER	Computational Time for Model Training (s)
Proposed ResNet-101T	7.77	10.77	350,864
LSTM [28]	11.55	26.64	87,148
ViT [29]	12.47	20.18	571,428

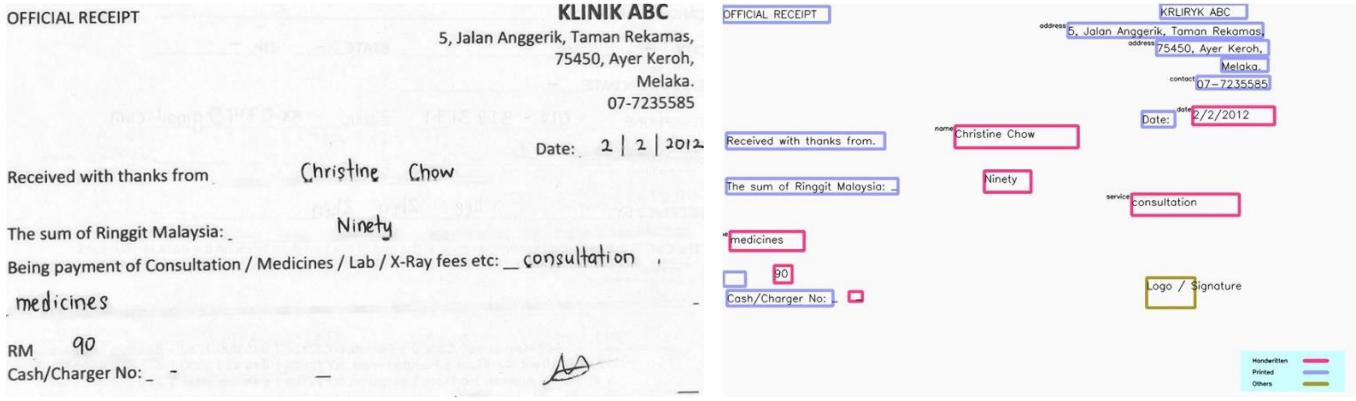


(**a**) (**b**) (**c**)

Figure 11. Experimental results of the ResNet-101T: (**a**) training and validation loss of the Transformer model; (**b**) training and validation loss of the LSTM model; (**c**) training and validation loss of ViT.

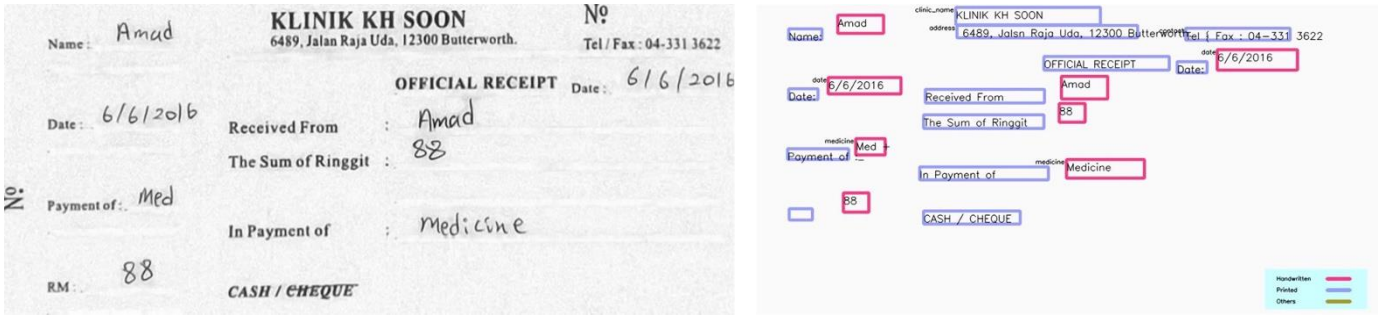
4.6. Demonstration

This section demonstrates some of the sample input and output of the proposed method. Figures 12 and 13 show the transcription results of the proposed HTR pipeline for different receipt templates. The printed text region is highlighted in blue, while the handwritten text region is bounded with a red square. A label is displayed at the top left corner of the square if the NER model identifies any underlying information from the text regions. We observe that the proposed model can correctly recognize the context of the transcribed texts. However, the application of OCR to printed texts sometimes failed to recognize the texts appropriately. This is considered a future work to enhance the model's performance.



(**a**) (**b**)

Figure 12. Sample input and output I: (**a**) input; (**b**) output.



(**a**) (**b**)

Figure 13. Sample input and output II: (**a**) input; (**b**) output.

5. Discussion

Some interesting findings have been discovered in this study. Real-life documents contain a substantial amount of noise (the noise can occur before and after the digitization process). Thus, the document layout should be properly analyzed, and data pre-processing plays an important role in treating different types of documents. In contrast to the line segmentation technique [3] and the full-page document recognition method [4], we applied an object detection approach for implicit ROI localization. The proposed approach was able to perform text type classification at the same time. There generally exist different text types in handwritten receipts, such as printed, handwritten, and non-text. Different text types should be treated individually to ensure optimal performance, rather than considering all text types in one go, which might result in inferior performance. Thus, explicitly segmenting different ROIs according to the text types would help to increase recognition accuracy and is more suitable for real-life applications.

Along this line, we find that a single HTR model cannot cope with the different text types. For example, a model that is good at recognizing printed texts does not necessarily work well with handwritten texts. Moreover, a separate model needs to be developed to recognize non-text attributes such as signature, which is considered vital for documentation. Therefore, a pipeline approach was proposed by training different models to deal with different text types. The transcribed texts are not directly useful for practical applications. Hence, a NER module was introduced to assign the transcribed texts into their corresponding entities/groups (e.g., name, date, address, phone number). The pipeline approach ensures a fully automated workflow with better efficiency, recognition accuracy, and data management ability in a practical application.

We also wish to highlight that the ResNet-101T model is proposed due to the following reasons. First, ResNet-101 has the advantage of being able to minimize negative effects

when the depth of the network is increased. Second, the Transformer model can better model the words input with the attention mechanism compared to the other RNN variations. Moreover, it also has a relatively low inductive bias compared to its RNN counterpart. In addition, the experimental result suggests the application of ResNet can effectively extract the 2D representative features, such as the shape/outline and positional information of the words for training the Transformer model. The result of feeding ResNet output to the Transformer model is better than using raw text input for the Transformer model. Empirical results show that the proposed model has clearly outperformed LSTM and ViT in terms of CER and WER. Although LSTM takes less time for training due to its relatively simpler architecture, its accuracy is far inferior to the proposed ResNet-101T method.

6. Conclusions and Future Works

A system that can recognize human handwritten text is significantly essential in automatic information storage and management. This paper presents a pipeline approach towards HTR. The proposed approach is composed of ROI localization, text type classification, text recognition, and context recognition funnels. A ResNet-101T model is introduced to recognize handwritten texts. The proposed model, trained using a self-collected clinical receipts dataset containing 15,297 text segments, achieves a CER and WER of 7.77% and 10.77%, respectively. In addition, more experimental studies can be carried out to investigate the use of ViT for HTR, such as fine-tuning and cross-validation.

For future endeavors, more training data will be collected to enhance the system's efficiency and accuracy. More receipt samples will be distributed to collect different handwritten styles, such as the clinicians and medical staff in the hospital, who might have messier handwritten styles. In this way, the proposed approach can be fine-tuned and applied in real-life scenarios. The HTR pipeline can be further improved and extended to different domains, such as clinical reports and receipts updates, insurance records, industrial documents, and others. In addition, Explainable AI techniques [29,30] are considered for future works to enhance the model's explainability and to learn meaningful representations to improve the model's performance.

Author Contributions: Conceptualization, Y.F.T. and T.C.; Methodology, Y.F.T., T.C. and M.K.O.G.; Software, M.K.O.G.; Validation, A.B.J.T.; Writing—original draft, Y.F.T. and T.C.; Writing—review and editing, A.B.J.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NO. NRF-2019R1A2C1003306).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are available at https://github.com/yeefantan/ResNet-101T-for-HCR/blob/main/README.md (accessed on 22 January 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chowdhury, A.; Vig, L. An Efficient End-to-End Neural Model for Handwritten Text Recognition. *arXiv* **2018**, arXiv:1807.07965. Available online: http://arxiv.org/abs/1807.07965 (accessed on 22 October 2021).
2. Chung, J.; Delteil, T. A Computationally Efficient Pipeline Approach to Full Page Offline Handwritten Text Recognition. *arXiv* **2020**, arXiv:1910.00663. Available online: http://arxiv.org/abs/1910.00663 (accessed on 22 October 2021).
3. Ingle, R.R.; Fujii, Y.; Deselaers, T.; Baccash, J.; Popat, A.C. A Scalable Handwritten Text Recognition System. In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, Australia, 20–25 September 2019; IEEE: New York, NY, 2019; pp. 17–24. [CrossRef]
4. Singh, S.S.; Karayev, S. Full Page Handwriting Recognition via Image to Sequence Extraction. *arXiv* **2021**, arXiv:2103.06450. Available online: https://arxiv.org/abs/2103.06450 (accessed on 22 October 2021).
5. Zhang, X.; Yan, K. An Algorithm of Bidirectional RNN for Offline Handwritten Chinese Text Recognition. In *Intelligent Computing Methodologies*; Huang, D.-S., Huang, Z.-K., Hussain, A., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 423–431.

6. Hassan, S.; Irfan, A.; Mirza, A.; Siddiqi, I. Cursive Handwritten Text Recognition using Bi-Directional LSTMs: A Case Study on Urdu Handwriting. In Proceedings of the 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML), Istanbul, Turkey, 26–28 August 2019; IEEE: New York, NY, USA, 2019; pp. 67–72. [CrossRef]
7. Nogra, J.A.; Romana, C.L.S.; Maravillas, E. LSTM Neural Networks for Baybáyin Handwriting Recognition. In Proceedings of the 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), Singapore, 23–25 February 2019; IEEE: Singapore, 2019; pp. 62–66. [CrossRef]
8. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [CrossRef] [PubMed]
9. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762. Available online: http://arxiv.org/abs/1706.03762 (accessed on 22 October 2021).
10. Marti, U.-V.; Bunke, H. The IAM-database: An English sentence database for offline handwriting recognition. *Int. J. Doc. Anal. Recognit. IJDAR* **2002**, *5*, 39–46. [CrossRef]
11. Bluche, T. Joint Line Segmentation and Transcription for End-to-End Handwritten Paragraph Recognition. *arXiv* **2016**, arXiv:1604.08352.
12. Wigington, C.; Tensmeyer, C.; Davis, B.; Barrett, W.; Price, B.; Cohen, S. Start, Follow, Read: End-to-End Full-Page Handwriting Recognition. In *Computer Vision—ECCV 20180*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 372–388. [CrossRef]
13. Asha, K.; Krishnappa, H. Kannada Handwritten Document Recognition using Convolutional Neural Network. In Proceedings of the 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS), Bengaluru, India, 20–22 December 2018; IEEE: New York, NY, USA, 2018; pp. 299–301. [CrossRef]
14. Wu, K.; Fu, H.; Li, W. Handwriting Text-line Detection and Recognition in Answer Sheet Composition with Few Labeled Data. In Proceedings of the 2020 IEEE 11th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 16–18 October 2020; IEEE: New York, NY, USA, 2020; pp. 129–132. [CrossRef]
15. Jocher, G.; Stoken, A.; Chaurasia, A.; Borovec, J.; Xie, T.; Kwon, Y.; Michael, K.; Changyu, L.; Fang, J.; Abhiram, V.; et al. Ultralytics/Yolov5: v6.0—YOLOv5n "Nano" Models, Roboflow Integration, TensorFlow Export, OpenCV DNN Support, Zenodo. 2021. Available online: https://zenodo.org/record/5563715#.YgYOB-pByUk (accessed on 22 October 2021).
16. Thakare, S.; Kamble, A.; Thengne, V.; Kamble, U. Document Segmentation and Language Translation Using Tesseract-OCR. In Proceedings of the 2018 IEEE 13th International Conference on Industrial and Information Systems (ICIIS), Rupangar, India, 1–2 December 2018; IEEE: New York, NY, USA, 2018; pp. 148–151. [CrossRef]
17. Cantoni, V.; Mattia, E. Hough Transform. In *Encyclopedia of Systems Biology*; Dubitzky, W., Wolkenhauer, O., Cho, K.-H., Yokota, H., Eds.; Springer: New York, NY, USA, 2013; pp. 917–918. [CrossRef]
18. Galamhos, C.; Matas, J.; Kittler, J. Progressive probabilistic Hough transform for line detection. In Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149), Fort Collins, CO, USA, 23–25 June 1999; Volume 1, pp. 554–560. [CrossRef]
19. Bradski, G. The openCV library. *Dr. Dobb's J. Softw. Tools Prof. Program.* **2000**, *25*, 120–123.
20. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767. Available online: http://arxiv.org/abs/1804.02767 (accessed on 9 November 2021).
21. Tzutalin, LabelImg. Available online: https://github.com/tzutalin/labelImg (accessed on 22 October 2021).
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; IEEE: New York, NY, USA; pp. 770–778. [CrossRef]
23. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. *arXiv* **2020**, arXiv:2005.12872. Available online: http://arxiv.org/abs/2005.12872 (accessed on 1 December 2021).
24. Honnibal, M.; Montani, I.; Van Landeghem, S.; Boyd, A. spaCy: Industrial-Strength Natural Language Processing in Python, (2020). Available online: https://zenodo.org/record/5764736#.YgYOaupByUk (accessed on 1 December 2021).
25. Tarcar, A.K.; Tiwari, A.; Dhaimodker, V.N.; Rebelo, P.; Desai, R.; Rao, D. NER Models Using Pre-Training and Transfer Learning for Healthcare. *arXiv* **2019**, arXiv:1910.11241. Available online: http://arxiv.org/abs/1910.11241 (accessed on 22 October 2021).
26. Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; Dyer, C. Neural Architectures for Named Entity Recognition. *arXiv* **2016**, arXiv:1603.01360. Available online: http://arxiv.org/abs/1603.01360 (accessed on 23 October 2021).
27. Wigington, C.; Stewart, S.; Davis, B.; Barrett, B.; Price, B.; Cohen, S. Data Augmentation for Recognition of Handwritten Words and Lines Using a CNN-LSTM Network. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; pp. 639–645. [CrossRef]
28. Li, M.; Lv, T.; Cui, L.; Lu, Y.; Florencio, D.; Zhang, C.; Li, Z.; Wei, F. TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models. *arXiv* **2021**, arXiv:2109.10282. Available online: http://arxiv.org/abs/2109.10282 (accessed on 6 December 2021).
29. Zhou, X.; Jin, K.; Shang, Y.; Guo, G. Visually Interpretable Representation Learning for Depression Recognition from Facial Images. *IEEE Trans. Affect. Comput.* **2018**, *11*, 542–552. [CrossRef]
30. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning deep features for discriminative localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2921–2929.