

**Text Recognition (OCR) for Patient
Records Digitization using CNN**

BY

ONG ZI LEONG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

For the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2022

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Text Recognition (OCR) for Patient Records Digitization
using CNN

Academic Session: JAN 2022

I ONG ZI LEONG
(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,



(Author's signature)



(Supervisor's signature)

Address:

2A, Lorong Impian Indah 2,
Taman Impian Indah,
14000 BM, Pulau Pinang

Dr. Aun Yichiet

Supervisor's name

Date: 21 April 2022

Date: 21 April 2022

Universiti Tunku Abdul Rahman			
Form Title: Submission Sheet for FYP			
Form Number: FM-IAD-004	Rev No.: 0	Effective Date: 21 JUNE 2011	Page No.: 1 of 1

**FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TUNKU ABDUL RAHMAN**

Date: 21 April 2022

SUBMISSION OF FINAL YEAR PROJECT

It is hereby certified that Ong Zi Leong (ID No: 18ACB02522) has completed this final year project entitled “Text Recognition (OCR) for Patient Records Digitization using CNN” under the supervision of Dr. Aun YiChiet (Supervisor) from the Department of Computer and Communication Technology, Faculty of Information and Communication Technology.

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.


Yours truly,



ONG ZI LEONG

DECLARATION OF ORIGINALITY

I declare that this report entitled “**Text Recognition (OCR) for Patient Records Digitization using CNN**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature :  _____

Name : Ong Zi Leong

Date : 21 April 2022

ACKNOWLEDGEMENTS

I would like to thank my project supervisor, Dr. Aun YiChiet, for providing me with an opportunity to showcase my skills on this project. Dr. Aun was always there to help and support me whenever I encountered difficulties.

I would also like to thank my family and friends who have often shown me support and kept me motivated to complete this project.

ABSTRACT

Optical character recognition (OCR) is widely used to transcribe texts from images in computer vision. Although current OCR methods can accurately transcribe printed text (structured), they often fall short on unstructured or handwritten text recognition. This project proposed a text recognition method to recognize handwritten text on patients' clinical data using a convolutional neural network (CNN). We compiled custom handwriting datasets from *MNIST 0-9* and *Kaggle A-Z datasets* to add more handwriting diversity in training a more robust OCR model. The CNN has 3-convolutional layers to learn high-level features and a dropout layer to prevent overfitting. The preliminary results showed that the proposed model achieved 93.75% classification accuracy while Tesseract (the state-of-the-art OCR) scored 69.79%. The data will be transformed from handwritten text to computer-readable text and then stored in files in xml form for further development.

TABLE OF CONTENTS

TITLE PAGE	I
DECLARATION OF ORIGINALITY	II
ACKNOWLEDGEMENTS	III
ABSTRACT	IV
TABLE OF CONTENTS.....	V
LIST OF TABLES	VII
LIST OF FIGURES	VIII
LIST OF ABBREVIATIONS	IX
Chapter 1 Introduction	1
1.1 Problem Statement and Motivation.....	1
1.2 Project Scope.....	2
1.3 Project Objectives	2
1.4 Impact, significance and contribution	3
1.5 Background information	3
1.5.1 Patient Management System.....	3
1.5.2 Convolutional Neural Network (CNN).....	4
1.5.3 Optical Character Recognition (OCR).....	5
1.6 Report Organization	6
Chapter 2 Literature Review	7
2.1 Overview of existing OCR engines.....	7
2.1.1 Tesseract	7
2.1.2 Google Cloud Vision	8
2.2 Comparison between existing OCR engines.....	9
2.3 Review on related work.....	10
Chapter 3 Proposed Approach	14
3.1 Datasets	14
3.2 Methodology	15
3.3 Model Overview.....	16

3.4	Matrix Evaluation.....	20
3.5	Tools and Technologies implementation	21
3.6	Implementation Issues and Challenges	21
3.7	Timeline	22
Chapter 4 Experimental Setup and Result		23
4.1	Overview	23
4.2	Performance evaluation.....	23
4.3	Text Predictions on Plain Text.....	26
4.3.1	Comparison between the proposed model and Tesseract engine.....	26
4.4	Text Predictions on Patient Medical Record Form	27
4.4.1	Extracting Region of Interest	27
4.4.2	Evaluation of Model Prediction for Patient Medical Records	30
4.5	Time consumed on manual transcribe and AI transcribe.....	31
4.6	Error rate between manual transcribe and AI transcribe.....	34
4.7	Future Remarks	35
Chapter 5 Conclusion.....		36
5.1	Project Review	36
5.2	Future Work	36
Bibliography		37
Appendices.....		A-1

LIST OF TABLES

Table Number	Title	Page
Table 2-1	Features Comparison of Existing OCR engine	9
Table 2-2	Summaries of reviewed papers	13
Table 3-1	Library used for the proposed model	21
Table 4-1	Classification report for the proposed model	25
Table 4-2	Summary of the prediction's accuracy by a text	26
Table 4-3	Summary of the prediction's accuracy by a single letter	26
Table 4-4	Few examples outputs from the predictions	27
Table 4-5	Predictions result on patient medical form	30
Table 4-6	Predictions result on patient medical records by fields	30
Table 4-7	Comparison between manual transcribe time and AI transcribe time	34
Table 4-8	Error rate between manual transcribe and AI transcribe	34

LIST OF FIGURES

Figure Number	Title	Page
Figure 1-1	Architecture of CNN.	4
Figure 1-2	Input and Filter in Convolutional Layer.	5
Figure 2-1	OCR's Process Flow	7
Figure 2-2	Text Detection Result on Tesseract and Google Cloud Vision	9
Figure 3-1	MNIST 0-9 and Kaggle A-Z datasets	14
Figure 3-2	Proposed Methodology Process	15
Figure 3-3	Architecture of CNN	16
Figure 3-4	Input, Filter, and Feature Map	17
Figure 3-5	Model Summary	26
Figure 3-6	Amount of the datasets for each class	21
Figure 3-7	Timeline	22
Figure 4-1	Training and Validation loss	23
Figure 4-2	Training and Validation Accuracy	23
Figure 4-3	Confusion Matrix	24
Figure 4-4	Original Form of Patient Medical Record	28
Figure 4-5	Mask created over Region of Interest	29
Figure 4-6	Mask over interested region of patient medical form	29
Figure 4-7	Screenshot of website inputted data	31
Figure 4-8	XML file generated by website	32
Figure 4-9	XML file generated using AI transcribe	32
Figure 4-10	Process time taken for Manual Transcribe	33
Figure 4-11	Process time taken for AI Transcribe	33

LIST OF ABBREVIATIONS

<i>CNN</i>	Convolutional Neural Network
<i>ECOC</i>	Error Correcting Output Code
<i>FCS</i>	Fuzzy Character Search
<i>LSTM</i>	Long-Short-Term Memory
<i>MSER</i>	Maximally Stable External Regions
<i>OCR</i>	Optical Character Recognition
<i>ReLU</i>	Rectified Linear Unit
<i>ROI</i>	Region of Interest
<i>RPN</i>	Region Proposal Network
<i>SVM</i>	Support Vector Machine

Chapter 1 Introduction

1.1 Problem Statement and Motivation

Even in the 21st century, most clinics or hospitals still rely on traditional paperwork to access the medical record or patients' information for their daily operations. A traditional paper-based record system involving recording the patients' personal information, and medical records into paper, disk, or films, and the file will be stored in a physical storage facility. For the paper-based record, people need to go through a time-consuming process to retrieve desired data from the paper [1]. In addition, paper-based records are not scalable and cannot be easily replicated for backup purposes due to the sheer volume of records. Therefore, electronic health records (EHR) were introduced to facilitate data sharing between organizations and also lubricate the daily operations of hospitals and clinics. However, moving from a traditional paper-based database system to an electronic-based database system is a daunting task. This process is both time-consuming and costly, as data such as patient records and medical records need to be manually entered into the computer to continue the digitization process.

Thanks to the development of advance's technologies, today technologies has the ability on detecting and extracting text from documents or table form and export it to the computer. Many solutions have been proposed such as Optical Character Recognition (OCR) to automate the data extraction from printed or written text from scanned documents or image files and then converting it into the machine-encoded file for further editing or processing. Traditional OCR uses patterns and correlations to distinguish text from other elements. However, this technique does not yield high accuracy in some complex text or blurred images [2].

Today, deep learning is applied to OCR systems that provide highly accurate text recognition and detection. Datasets of handwritten numbers and letters are used to train a model to predict the text. These new OCR systems will gain knowledge and learn how to recognize an unlimited number of characters, rather than being limited to a predetermined number of character sets [3]. The Convolutional neural network (CNN) is a popular neural

network architecture for image recognition and processing. It is also a state-of-the-art model for handwritten character recognition. Back in 1998, researchers used a seven-layer convolutional neural network excluding the input layer for MNIST datasets which are the handwritten text datasets, and came up with an error rate of 0.9% [4]. As a result, there is a focus on deep learning-based OCR because it can be more accurate than traditional OCR.

1.2 Project Scope

The project aims to use computer vision to digitize patient's data from paper records to develop a patient management system to manage general tasks in the clinic and. A deep learning model will be trained to extract text from the patient's physical records and convert it into a computer editable format referring to the existing OCR system. The datasets used in this project are the standard MNIST 0-9 dataset and Sachin Patel's Kaggle A-Z dataset, which is based on a special NIST database 19. These datasets were combined into a unified character dataset for training using the CNN model to predict digits and alphabets. OpenCV was also used to help in the process of training the model. It is an open-source library for image processing, computer vision, and machine learning that can process images or videos to recognize human handwriting. Finally, the project will be deployed on docker to provide better scalability for future needs.

1.3 Project Objectives

The objectives of this project are as follows.

1. To build a custom hand-writing dataset that includes a more diversified types of handwriting sample.
2. To train a robust OCR model for hand-written text recognition using a CNN.
3. To transcribe text from patients' report cards using the proposed OCR using region-based text recognition.

1.4 Impact, significance and contribution

Medical documentation is an essential part of a clinical process. Storing data by using a paper-based record is not an ideal way in terms of time, cost, and even security. For people who need to search the information need to go through the time-consuming process and have a hard time on sharing with other organization for analyzing. However, data digitization for a patient management system is not an easy task. To transform paper-based records into the computer-editable format, a CNN-based OCR system can be implemented. OCR is a solution of translating the scanned/captured documents into machine-encoded text type by using a webcam or camera. Patients' information in the paper-based record can be scanned and converted into the machine-encoded file. With this technology, the time required for data transformation can be reduced at the same time decrease the human error in the data processing.

CNN is a deep artificial neural network architecture for image recognition. CNN can learn relevant features from an image at different levels similar to the human brain. It allows you to extract useful image features and is powerful for image classification and recognition because of its high accuracy.

1.5 Background information

1.5.1 Patient Management System

A patient management system is a computing system that controls all of the data related to health care providers, helping them to perform their jobs more efficiently. Various departments can collect, store, process, extract and transmit all kinds of data to generate different information to increase the hospital's daily operational performance and provide automated management for its overall operation. Before a patient management system has been created, patients' information will be recorded on paper and stored in the physical storage facilities. People need to search the data from one file to another file causes data searching to become a time-consuming process. In the late 1960s, when the computer system was getting attention, the industry sees the development of electronic medical records to record patients' information. This allows the physicians to improve patient care

by using the computer. Thanks to the advancement of technology, developers today can develop a management system involving electronic medical records for the hospital. It can help manage all the tasks in the hospital and real-time decisions in only one place. Healthcare management even begins from the patients' hands through their cell phones and easily fulfills their needs in today's world. This system can significantly help the whole hospital function paperless and integrate all the information regarding patients, staff, doctors, finances, etc. An appropriate patient management system can help reduce a ton of work and increase overall performance for the health care organization. Although many organizations start to digitize their workflow, still many out there rely on the paper-based record. The transformation from paper-based to computer-based need a lot of time and effort but thanks to technology today, digitalization can be simplified.

1.5.2 Convolutional Neural Network (CNN)

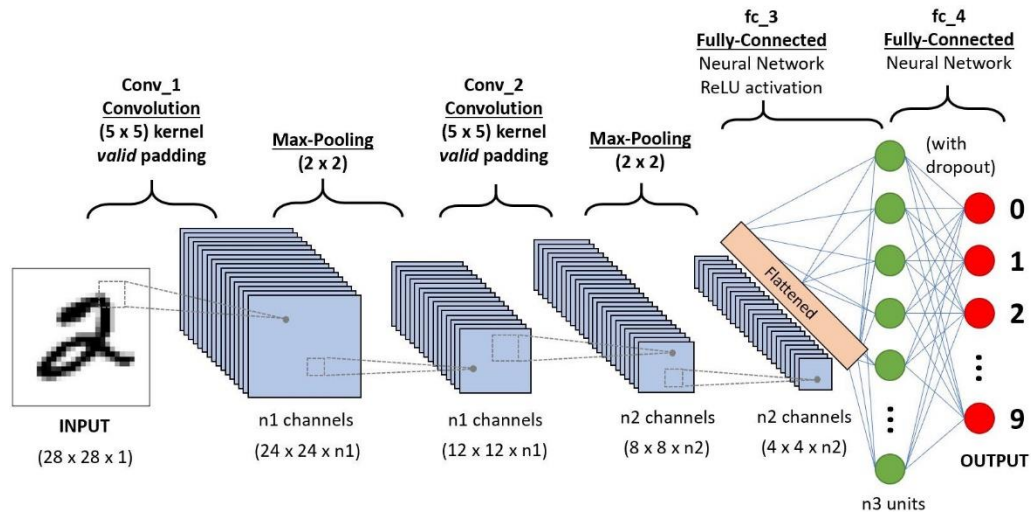


Figure 1-1 Architecture of CNN

Convolutional Neural Network (CNN) is a well-known deep learning algorithm proposed by Yann LeCun in the late 90s. CNN is used for image classification and recognition due to its high accuracy. It can detect the important features of the image without any human supervision. CNN is so powerful because of its non-linearity. The convolution operation will pass through the activation function to learning complex patterns in the data. There

are three basic components to define basic CNN, the convolutional layer, the pooling layer, and the fully connected layer.

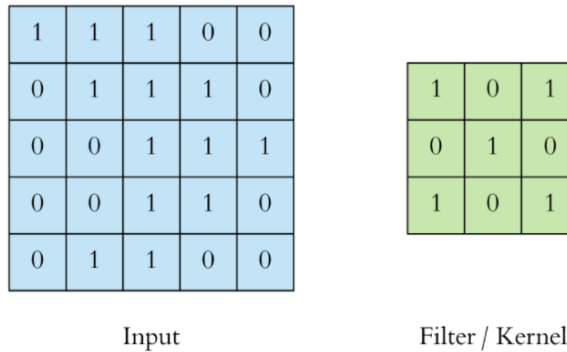


Figure 1-2 Input and Filter in Convolutional Layer

The convolutional layer is the main building block for the CNN model. The filter or weight matrix will slide over the input image to do element-wise matrix multiplication, sum the result, and adding the result into the feature map. The filter will extract different features on the image such as color, edge, shape, and stack together after performing padding to produce the final output of the convolution layer. The weight will be changed throughout the learning process to minimize the loss function which helps the network in correct prediction.

Finally, the output from the final pooling and convolution layer will be flattened into 1D vector numbers and the node from each layer will be connected to produce a fully connected layer. In this layer, the data is classifier into various classes by using the fully connected layer.

1.5.3 Optical Character Recognition (OCR)

Optical Character Recognition (OCR) was invented in the late 1920s. OCR is the process of converting text images into the machine-encoded text to digitize the data for further editing, presentation, or searching. OCR became popular in the early 1990s intending to digitize newspapers. By using OCR, text can be extracted in the form of image files or PDF files. OCR can be divided into two steps, pre-processing and character recognition. For pre-processing, irrelevant information in the image such as distortions are suppressed and

important image features are enhanced to improve the accuracy of recognition. For character recognition, two other methods can be emphasized, namely pattern recognition and feature detection. OCR recognizes the stored version of the character and compares it with the scanned image to find the correct character. This can have many limitations because not everyone has the same handwriting, and even on computers, there are different fonts. Feature detection is a more advanced method of character recognition. The features of each character are detected to find the correct letter. Today, most modern OCR programs use neural networks to automatically extract features in a brain-like manner. In this rapidly evolving world, OCR has been used in many sectors to improve their business performance due to increased digitization. For example, OCR can be used to digitize medical records. By replacing paper records with electronic records, OCR can greatly benefit the healthcare industry. Introduce electronic records can benefit the healthcare industry by improving day-to-day operational performance. Patient information can also be stored more securely and prevent any record be destroyed.

1.6 Report Organization

The report organization is as follows: Literature Review in chapter 2; system design in chapter 3; Proposed Approach in chapter 3; Experiment Setup and Result in chapter 4; Conclusion in chapter 5.

Chapter 2 Literature Review

2.1 Overview of existing OCR engines

This section discusses the OCR engines available on the web and compares the gaps in functionality based on the features offered.

2.1.1 Tesseract

Tesseract is a powerful open-source OCR engine that has gained high popularity among many OCR developers as it is considered to be one of the top three OCR engines in the world. It can only be accessed by programmers via the command line as it does not support a graphical user interface (GUI). Tesseract can support several output formats such as txt, pdf, hocr, and tsv. However, Tesseract is only trainable in version 4.0 by using Long Short-Term Memory (LSTM). LSTM neural networks are implemented on the Tesseract engine in version 4.0. LSTM is a recurrent neural network (RNN) that focuses on order-dependent prediction problems which can remember previous useful information for the current input process [5]. This new implementation helps improve the accuracy of the Tesseract OCR engine, and programmers can now train a new model from scratch or fine-tune an existing model for better system performance. It now can be trained to recognize up to more than 100 languages.

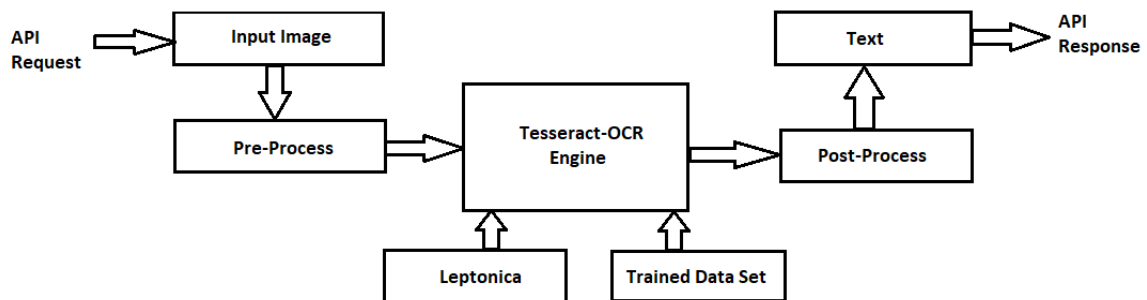


Figure 2-1 OCR's Process Flow

Figure 2-1 shows the flow of the OCR system. Each input image needs to go through a pre-processing process to improve the quality of the image. Techniques like rescaling, binarization, noise removal and rotation of the image help to modify the image quality for better further processing. Removing irrelevant information from the image, such as distortion, helps to get better OCR output.

Although Tesseract is a powerful engine for converting text to machine-encoded formats, it does not have the same accuracy in recognizing handwritten text. Compared to printed text, handwritten text is recognized with low accuracy. Such problems have led to some challenges for programmers implementing Tesseract in projects of recognition of handwritten text.

2.1.2 Google Cloud Vision

The Google Cloud Vision API enables developers to integrate visual detection capabilities into their applications, not only for OCR but also for image tagging, face and landmark detection. A Google Compute Engine Account needs to be created before using these functions. Except for text detection, it also supports language identification and up to more than 200 languages. There are 2 main annotations to help with text recognition which are `TEXT_ANNOTATION` and `DOCUMENT_TEXT_DETECTION` [6]. Text annotation is intended to be used in a variety of lighting conditions, and it can read the text in a variety of styles, but at a sparser level. Besides, the JSON file will return entire strings, as well as individual words and their corresponding bounding boxes. For `DOCUMENT_TEXT_DETECTION`, it is the same as `TEXT_ANNOTATION`, except that it is built specifically for densely displayed text files such as scanned books and the output JSON file will contain the information in the paragraphs, breaks, and blocks. Similar to the Tesseract engine, `DOCUMENT_TEXT_DETECTION` in Google Cloud Vision can recognize handwritten text, but the output is not expected to be very high and accurate.

2.2 Comparison between existing OCR engines

OCR \ Features	Tesseract	Google Cloud Vision
Online/Offline	offline	online
Handwritten detection	Yes, but not accurate	Yes, but not accurate
Trainable	Yes	No
Free Tier	Open Source	Paid Services
API Integration	Available	Available

Table 2-1 Features Comparison of Existing OCR engines

Based on Table 2-1, both of the OCR engines can recognize handwritten text from the image but the accuracy is not as high as expected. However, compare to Tesseract, Google Cloud Vision has better performance on recognizing handwritten text.

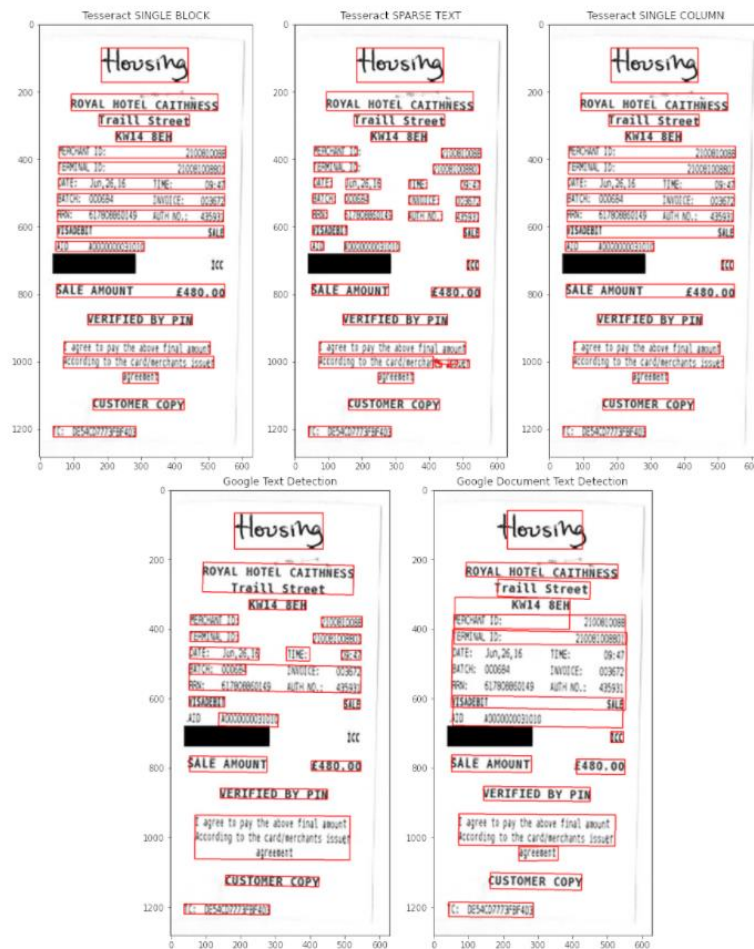


Figure 2-2 Text Detection Result on Tesseract and Google Cloud Vision

According to Figure 2-2, although the Google text detection model did not detect some lines of text, the segmentation of text detection was better compared to document text detection. Besides, Tesseract successfully detected both printed and handwritten text in all three models. However, Google Cloud Vision recognized the correct word "Housing", while Tesseract recognizes it as "stflrg" [7]. Therefore, in terms of accuracy, Google Cloud Vision has better results than Tesseract.

In summary, both OCR algorithms are excellent at recognizing and detecting written text in images. However, both engines must be further enhanced to obtain better accuracy in handwritten text recognition.

2.3 Review on related work

Many OCR engine was developed a long time ago even before the boom of deep learning in 2012 [8]. OCR has an excellent result in detecting and recognizing the structure or printed text but it has a hard time dealing with the unstructured text. Therefore, deep learning-based OCR has more attention in recent years. In this section, some deep learning-based OCRs-related work and their limitations will be discussed.

Timmaraju and Khanna *et al* (2015) have proposed a method for detecting and recognizing the text in natural images using two CNNs, one for detection and another for recognition. They use a sliding window technique to find characters at the center of the image patch by a CNN detection engine. The output on the detection engine passes into recognition CNN to determine which character was presented. They augmented their training data with negative examples to prevent the CNN from starting detection in a window including transitions between two adjacent characters in a word, resulting in high scores of false predictions. In addition, after assigning the bounding boxes of the words, Fuzzy Character Search (FCS) was utilized. The detection CNN examined the peaks of each word that corresponded to the three patches with the maximum confidence level. To avoid incorrect predictions, erroneous peaks containing non-central characters were marked as uncertain peaks. Finally, they achieved 98.53% detection accuracy and 86.53% recognition accuracy using these methods.

Yang *et al.* (2019) have proposed a Faster R-CNN-based method for handwritten text recognition. The Faster R-CNN performs well in object detection and can be considered as a hybrid of Region Proposal Networks (RPNs) and Fast R-CNN [11]. They pre-processed handwritten text with VGG-19 as a pre-training model for Faster R-CNN and utilize CNN for character recognition. The text is segmented into word regions and background regions using Faster R-CNN, and the word regions are further split into character regions and background regions. After the feature is extracted from the input images, RPNs are used to generate potential candidate frames for the targets and insert the results into the Region of Interest (ROI) pooling layer to compute the feature maps. As the consequence of this, they achieved high accuracy in character segmentation by converting character segmentation to object detection using Faster R-CNN. They achieve 99% for character segmentation rate of the word, 95% for character segmentation rate of letter, and average 97% in character recognition. In conclusion, they divided the problems into several sub-problems, and the results proved to be effective for complex handwritten text recognition work.

Bora *et al.* (2020) have modified the traditional CNN model by using the Error Correcting Output Code (ECOC) classifier in CNN-ECOC to replace the softmax layer in the CNN model. This approach aids in the conversion of multi-class classification problems into binary classification problems to obtain high prediction accuracy. The ECOC classifier is trained with the features extracted from the input image and feeds them to all binary learners trained with a linear Support Vector Machine (SVM), where one category is positive and the other categories are negative. SVM is a machine learning method that examines data for classification and regression analysis and divides the data into two categories. Then, the results in the ECOC classifier are translated to codewords and compared to the generated coding matrix table. Besides, they investigated several pre-trained models to determine the best model to combine with ECOC to generate the highest accuracy. Among all the implemented models, they identified AlexNet as the most suitable CNN for combining with ECOC. They noticed a 0.6% improvement in testing accuracy and achieved 97.71% accuracy in both training and testing after the features of the trained AlexNet were fed into the ECOC classifier.

Choudhary *et al.* (2018) have proposed an approach using Maximally Stable External Regions (MSER) for text region detection and CNN for recognizing the characters. MSER is a feature detector that looks for features surrounded by a consistent contrasting background. Because MSER is susceptible to image blurring, Canny edges are used to improve image quality. Detected text is extracted from the image and any non-text regions are removed. Each discovered region character is isolated as a single binary image for recognition by the CNN model. Due to the use of MSER, images with uneven contrast and color variations do not affect the recognition of text regions. The proposed technique achieves a recognition rate of 85-90% for individual characters and 70-75% for the overall text recognition rate of a single image.

Hassan *et al.* (2019) have utilized the bi-directional Long-Short-Term Memory (LSTM) as classification and CNNs for character segmentation. For handwritten text, raw pixels values from a column of text line images are fed into 1D-LSTM for learning and classification. While for handwritten text, the sliding window technique is used to traverse over the text line for character segmentation and the values are fed into RNN for learning the shape of the character [14]. Golovko *et al.* (2019) have suggested using Faster R-CNN to detect the block in the documents. To overcome the difficulty of recognizing objects in a single network, single-shot detectors or YOLO are used. These methods make it easier to locate individual blocks in a document and feed the detected region into the CNN for recognition. In summary, classification and localization using these algorithms can produce high accuracy.

Study	Techniques	Text Recognition Accuracy
Timmaraju <i>et al</i>	CNN	86.53%
Yang <i>et al</i>	Faster R-CNN	97%
Bora <i>et al</i>	CNN-ECOC	96.71%
Choudhary <i>et al</i>	CNN with MSER	85-90% for a single character. 70-75% for a text
Hassan <i>et al</i>	CNN with Bi-directional LSTM	83.69%

Table 2-2 Summarize of reviewed papers

Based on Table 2-2, Faster R-CNN proposed by Yang *et al* has achieved the highest accuracy in text recognition among the reviewed paper. They extract the characters from the background and use RPNs to find out the potential area for further processing. RPNs are more efficient than sliding windows compared to the method described by Timmaraju and Khanna, which requires a lot of computational power but is less accurate. Their proposed method also has limitations on recognizing characters “0”, “O”, “1”, and “I”. Choudhary *et al.* used CNN and MSER for text recognition. the main drawback of MSER is that it is susceptible to the quality of the input image. Blurred images can affect their performance, so the images need to be pre-processed before using this technique.

Chapter 3 Proposed Approach

3.1 Datasets

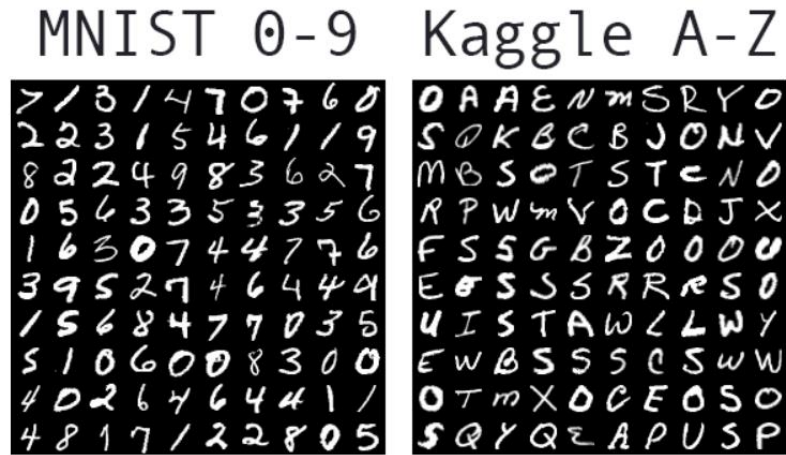


Figure 3-1 MNIST 0-9 and Kaggle A-Z datasets

This project utilized the MNIST 0-9 dataset and Sachin Patel's Kaggle A-Z dataset which is based on a special NIST-based database.¹⁹ These datasets were used to train a CNN model to predict handwritten which contains 26 characters and digits from 0 to 9. There is a total of 372 451 data in A-Z datasets and 70 000 data in 0-9 MNIST datasets. The A-Z dataset was being stored in a CSV file in pixel values form. These data were combined into one unified dataset which contains a total of 442 451 data. Moreover, the data were then split into a training set and validation set which contain 353 960 data and 88 491 data respectively. The original image shape of the A-Z dataset was converted from 441x658 to 28x28 to standardize with the 0-9 datasets for model training.

3.2 Methodology

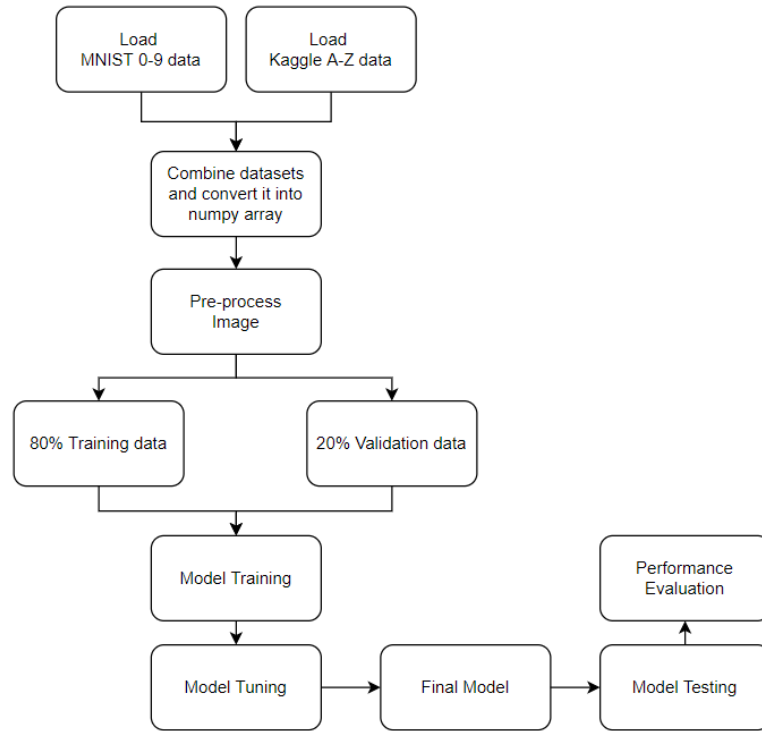


Figure 3-2 Proposed Methodology Process

Based on Figure 3-2, the first step of the proposed method is data collection. MNIST 0-9 data is load from the Jupyter notebook directly while the Kaggle A-Z is downloaded online and load from the local storage. To simplify the loading process, the character data were stored in CSV files, separated by commas by the authors. Both datasets were loaded and stacked together as one unified dataset. After loading the data, it was split into label columns and image columns. Then, using `train_test_split` in the sklearn model, the data is split into 80% test data and 20% training data. Finally, the data were scaled to 1 by dividing by 255 on the dataset to normalize the data.

Pre-processing image. Before training the model, all the images are pre-processed to remove some irrelevant information in the images and prepare for model input. Then, the datasets are being reshaped into 28x28 to standardize all the images for training. Images are also being transformed into a grayscale image with 1-dimension to allow the characters be more pop up from the background and remove the noise from the images that might

influence the model training. The accuracy of model training can be enhanced by pre-processing the images, and the model can extract features from the images more efficiently.

After pre-processing the images, the CNN model is built by assigning all the training parameters. The parameters in the model are important, and single values in them can affect the overall results. The weights or biases are then changed throughout the learning process and backpropagation is performed to reduce the losses by using a loss function. After the model is trained, it is tested using a pre-prepared test dataset. If the output is unsatisfactory, certain modifications, such as hyperparameter tuning and data augmentation, are made to improve the accuracy of the model. Finally, performance evaluation is performed to check the results of the model using several methods such as confusion matrix, F1 score, accuracy and recall.

3.3 Model Overview

In this project, the CNN model was used for training. The CNN model contains numerous critical layers that influence the training process and the final model output. The convolutional layer, pooling layer, activation function, and fully connected layer are among these layers. In this part, we will go through the model's layers and the functions that are employed.

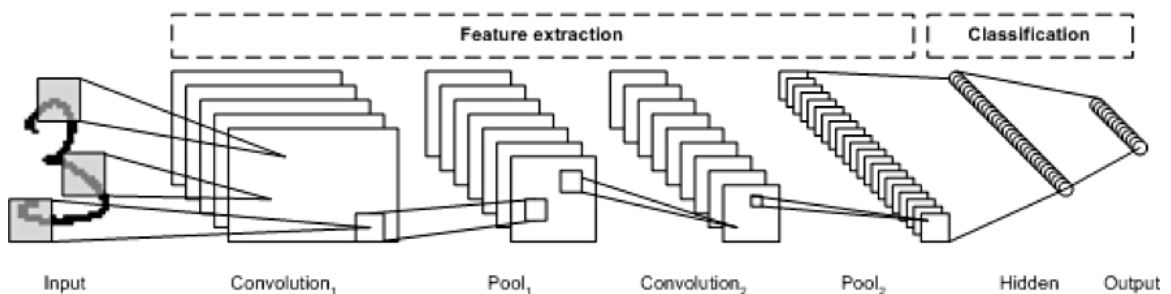


Figure 3-3 Architecture of CNN

Refer back to the architecture of CNN. The input image or test data is the initial element of the model. In our approach, the input image is resized to 28x28x1 beforehand, which means that the height and width are 28 and 1 dimensional, respectively. The resolution of 28x28x3 is generally RGB, while the 1-dimensional image is grayscale. The input images

are then processed through a series of convolutional layers with filters/kernels, pooling layers, fully connected layers, and a Softmax function that classifies the images with probability values from 0 to 1.

The convolutional layer is the first layer in CNN which is used to extract the features from the input image for the learning process. The filter slides over the input image and performs a dot product to generate a feature map containing all image features such as edges, colors, curves, etc.

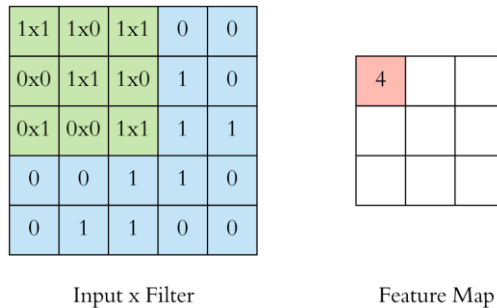


Figure 3-4 Input, Filter, and Feature Map

32 (3x3) filter was assigned to slide across the input image to extract the feature of the image and generate the feature map. The values in the filter were not defined, but it was taught to produce various values for extracting more specific characteristics in the picture throughout the learning process. The depth of the feature map is determined by the number of filters set for our layer. Thus, as a result of the convolutional layers, a total of 32 layers of feature maps are produced for our case.

To ensure the non-linearity, the Rectified Linear Unit (ReLU) was used as the activation function to let the convolution network learn non-negative linear values. The ReLU function is shown as below:

$$f(x) = \max(0, x)$$

The results of the convolution layers are provided to the ReLU function to ensure that the final feature map result is a ReLU function applied to them and not a sum value. There will be no negative results because the output will always find the maximum value between the

input value and zero. When using the ReLU function, neurons are triggered if the value exceeds 1. Because not all neurons are involved, it is more computationally efficient than other functions such as the sigmoid and tanh functions.

After that, the data is downsampled using the maximum pooling method. The max-pooling reduces the spatial size of the image and the parameters needed to be trained, hence control the training process to prevent overfitting. It selects the maximum value, or the most important feature, in the feature map region covered by the filter. The pooling size is set to (2x2) with a stride of 2. Thus, after the second hidden layer, the output size of the pooled convolutional layer changes from (24x24x26) to (12x12x64) for our case. The difference between the filter we mentioned before and max-pooling is that max-pooling only downsamples the data, while the filter gets different learning weights during the learning process, extracts the feature in the image, and downsamples the data if no padding is used.

Finally, the Softmax function is used as the activation function for the output layer of our model. Usually, the Softmax function is used to solve multi-class classification problems, and it is most suitable for our case since we have 26 characters and 10 digits. The output layer scores of the model are then fed into this function to calculate the classification probabilities for each class. The Softmax function is shown as below:

$$f(z_i) = \frac{e^{z_i}}{\sum_k e^{z_k}}$$

The score for each class from the output layer is passed into a function that calculates its exponential value and divides it by the sum of the exponential values of all classes to obtain the final probability. The z_i represents the score we want to calculate, and the denominator is the sum of the exponential of all z_k values to obtain the final result. The result of the Softmax function is changed by the scores of the other classes in the output layer. Therefore, if the scores in other classes are changed, the results will change as well. It is worth noting that the Softmax function calculates the sum of probabilities as 1. A total of 36 possibilities was generated for all 36 classes.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 26, 26, 32)	320
conv2d_1 (Conv2D)	(None, 24, 24, 64)	18496
max_pooling2d (MaxPooling2D)	(None, 12, 12, 64)	0
conv2d_2 (Conv2D)	(None, 10, 10, 64)	36928
flatten (Flatten)	(None, 6400)	0
dense (Dense)	(None, 128)	819328
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 36)	4644
Total params: 879,716		
Trainable params: 879,716		
Non-trainable params: 0		

Figure 3-5 Model Summary

Figure 3-3 shows that the output shape was 26x26 with a depth of 32 after the first convolutional layer. Because a 3x3 filter was used in the first layer to glide over the input image to extract features, the shapes were decreased from 26x26 to 24x24. In this research, we employed three convolutional layers for training our model. Following the completion of all convolution layers, a 10x10x64 output shape was generated. Before proceeding to the completely linked layer, the image must be flattened by multiplying all of the values (10x10x64) to form a 6400 one-dimensional array. Finally, the fully connected layer links all preceding layers in order to give learnt features derived from the combination of all characteristics. We have defined 30 epochs numbers and the batch size in 32 for training our data.

3.4 Matrix Evaluation

Evaluation measures are used to determine the performance of the model based on our training. First, the confusion matrix was used to express quantitatively the accuracy of the classification. The confusion matrix has two dimensions and contains information about the actual and predicted class obtained from the classification. Each row represents an actual class, while each column represents the predicted class. For the confusion matrix, TP represents true positives, FP represents false positives, TN represents true negatives, and FN represents false negatives. Each of these components can be used to calculate the accuracy, precision, recall, and F1 score of the classifier.

$$\textit{Accuracy} = \frac{\textit{TP} + \textit{TN}}{\textit{TP} + \textit{TN} + \textit{FP} + \textit{FN}}$$

For calculating the accuracy of our classifier, the total true predicted number is divided by the total number of the data.

$$\textit{Precision} = \frac{\textit{TP}}{\textit{TP} + \textit{FP}}$$

The precision then indicates how many of the positive predictions in the class are correct.

$$\textit{Recall} = \frac{\textit{TP}}{\textit{TP} + \textit{FN}}$$

Then, recall is defined as the proportion of all relevant results that are correctly classified by our model. The higher the recall rate, the lower the false negatives and vice versa.

$$\textit{F1-Score} = 2 \times \frac{\textit{Recall} \times \textit{Precision}}{\textit{Recall} + \textit{Precision}}$$

The F1 score is defined as the weighted average of accuracy and recall. The numerator is the recall multiplied by the accuracy and the denominator is the recall plus the accuracy.

3.5 Tools and Technologies implementation

Python Library	Version
Keras	2.5.0
Tensorflow	2.5.0
Sklearn	0.24.1
Cv2	4.5.3
Matplotlib	3.3.4

Table 3-1 Library used for the proposed model

The OpenCV library and matplotlib library are used to visualize our data and for pre-processing the images while the sklearn library help in splitting the data and also print the classification report.

3.6 Implementation Issues and Challenges

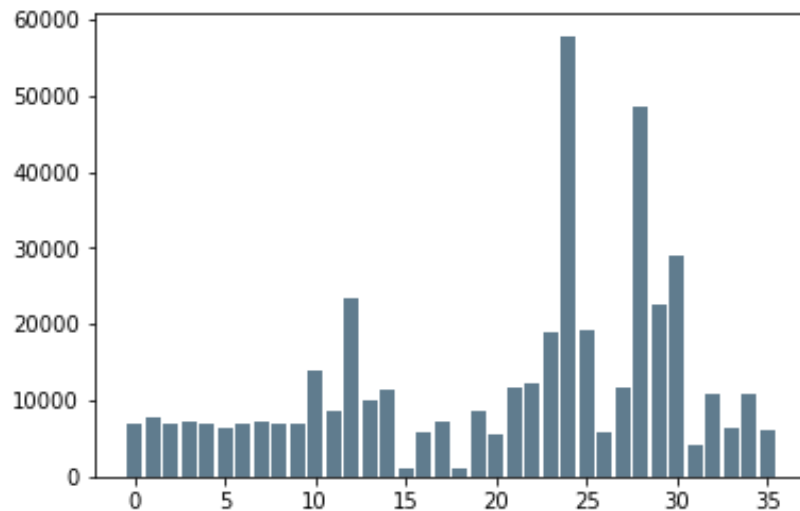


Figure 3-6 Amount of the datasets for each class

The first implementation issue is the unbalanced nature of the data set. According to the histogram in Figure 3-4, the numbers 0-9 have balanced datasets with the same amount of data overall. However, we can notice that the data set for A-Z is unbalanced, with the character "0" having 57,825 data, the character "S" having 48,491 data, and the characters

"I" and "I" having only about 1,000 data. This situation leads to the fact that the model will focus on the data set with more data. In addition, this can lead to poor prediction results and reduce the accuracy of the prediction.

In addition, the process of training CNN models with low-specification devices is time-consuming and computationally intensive, reducing the efficiency of the model training process. The model needs to be tuned to get better results, which is a daunting process. In most cases, the GPU in the device is the best case for training the model because it can speed up the training process. To solve this problem, we can use Google Colab, which provides free online GPU to run their models. However, Google Labs has limited access to the free GPU.

3.7 Timeline

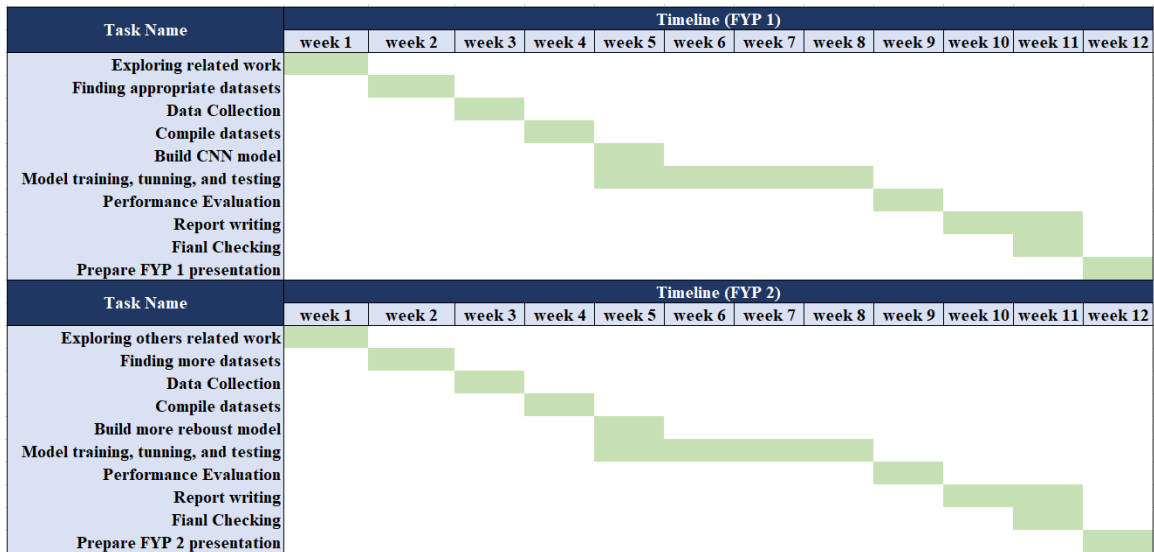


Figure 3-7 Timeline

Chapter 4 Experimental Setup and Result

4.1 Overview

This chapter will cover the analysis of our model and the performance of the proposed model and the existing Tesseract OCR engine in recognizing handwritten text. Various evaluation methods will be used and graphs will be shown to help visualize the results. In addition, some of the shortcomings of our proposed model will be highlighted and outline possible future work to enhance our model to make it a more robust handwriting recognition system.

4.2 Performance evaluation

Figures 4-1 and 4-2 describe the training and testing losses over the epochs, as well as the training and testing accuracies over epochs. During the training process, we achieved a validation accuracy of 0.9866 and also a validation loss of 0.0571.

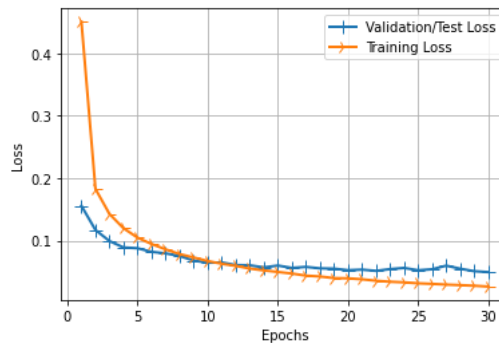


Figure 4-1 Training and Validation loss

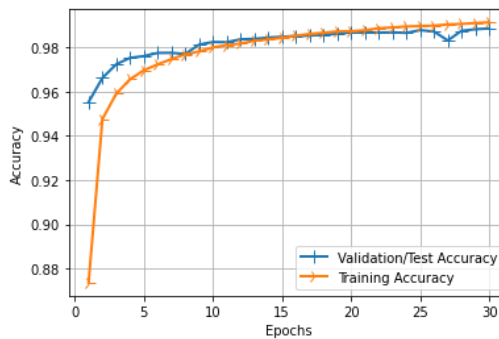


Figure 4-2 Training and Validation Accuracy

A rise in losses in the validation data indicates slight overfitting of the model. Measures such as reducing the model capacity or regularizing the result in the model can be taken to avoid overfitting the model. In our case, only one dropout layer was applied in the model with a value of 0.25. To avoid overfitting, we can either add another dropout after the convolutional layer to reduce the capacity or slim down the network during training.

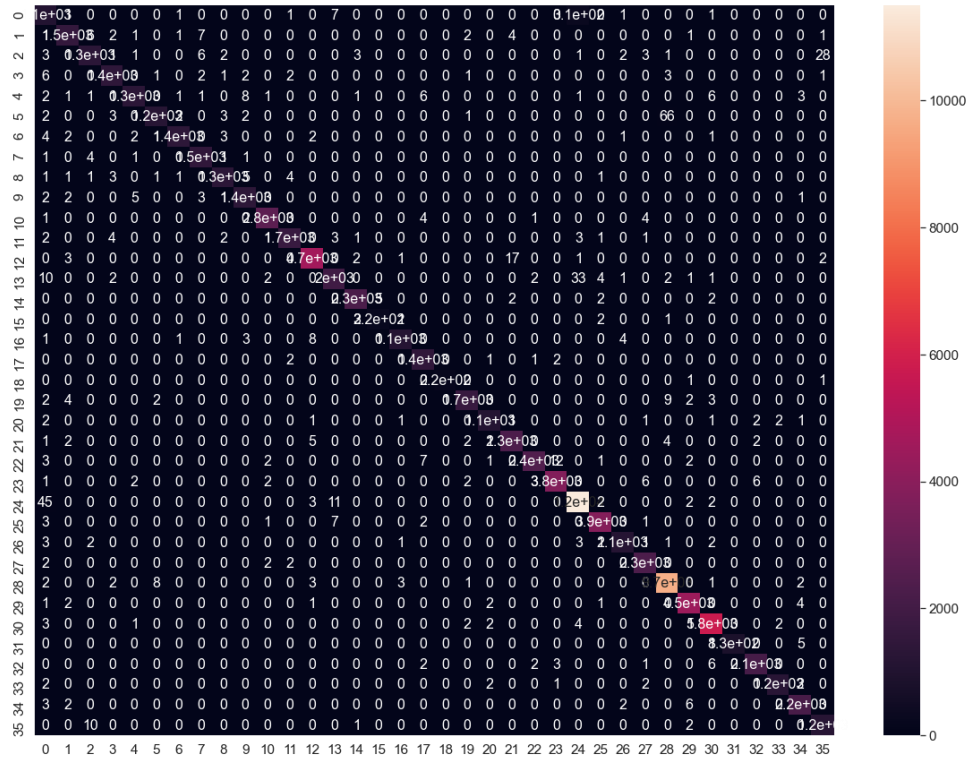


Figure 4-3 Confusion matrix

Figures 4-3 show the confusion matrix of the test set for our model. The diagonal of the confusion matrix shows the number of accurate classifications while others show the inaccurate classification. The model predicts well in all the characters and achieved an average of 99% of accuracy. However, the model is confusing on the word "0" with index 0 and the word "O" with index 26. Since the patterns and features of "O" and "0" are identical, the model does not predict well for these two words.

	precision	recall	f1-score	support
0	0.95	0.74	0.83	1381
1	0.99	0.99	0.99	1575

2	0.98	0.98	0.98	1398
3	0.99	0.99	0.99	1428
4	0.98	0.98	0.98	1365
5	0.99	0.90	0.94	1263
6	0.99	0.99	0.99	1375
7	0.99	0.99	0.99	1459
8	0.98	0.98	0.98	1365
9	0.98	0.98	0.98	1392
a	0.99	1.00	1.00	2774
b	0.99	0.99	0.99	1734
c	1.00	0.99	1.00	4682
d	0.98	0.97	0.98	2027
e	1.00	1.00	1.00	2288
f	0.97	0.97	0.97	232
g	1.00	0.98	0.99	1152
h	0.97	0.99	0.98	1444
i	1.00	0.99	0.99	224
j	0.99	0.99	0.99	1699
k	0.99	0.99	0.99	1121
l	0.99	1.00	0.99	2317
m	0.99	0.99	0.99	2467
n	0.99	0.99	0.99	3802
o	0.96	0.99	0.98	11565
p	0.99	0.99	0.99	3868
q	0.99	0.98	0.98	1162
r	1.00	0.99	0.99	2313
s	0.99	1.00	0.99	9684
t	0.99	1.00	1.00	4499
u	0.99	1.00	0.99	5802
v	1.00	0.99	0.99	836
w	0.99	0.99	0.99	2157
x	1.00	0.99	0.99	1254
y	0.99	1.00	0.99	2172
z	0.98	0.98	0.98	1215
accuracy			0.99	88491
macro avg	0.99	0.98	0.98	88491
weighted avg	0.99	0.99	0.99	88491

Table 4-1 Classification report for the proposed model

The above table shows the classification report of the proposed models, including accuracy, F1 score, precision and recall for each class. Note that all models have fairly good food results except for digit 0.

4.3 Text Predictions on Plain Text

4.3.1 Comparison between the proposed model and Tesseract engine

To compare with the suggested handwritten text recognition model, the Tesseract engine is employed. The comparison will test the accuracy of recognizing the word and also the accuracy of recognizing the single letter. There is a total of 20 testing data are being tested by both programs and the following shows the summary of the result.

System	Total Predictions	Correct Prediction	Accuracy
Proposed Model	20	16	80%
Tesseract engine	20	10	50%

Table 4-2 Summary of the prediction's accuracy by a text

Although the Tesseract engine has good image-to-text capabilities, it still does not perform well in recognizing handwritten text. Compared to the proposed model, we achieve an accuracy of 80%, while the Tesseract engine achieves only 50%. This result shows that the proposed method has higher accuracy in recognizing handwritten words compared to the existing Tesseract engine.

System	Total Predictions	Correct Prediction	Accuracy
Proposed Model	96	90	93.75%
Tesseract engine	96	67	69.79%

Figure 4-3 Summary of the prediction's accuracy by a single letter

There are 96 characters in all images and the proposed model successfully recognized 90 out of 96 characters, achieving an accuracy of 93.75% in the comparison. Compared to the tesseract engine, it has an accuracy of 69.79% and only correctly recognizing 67 out of 96 characters. The main problem of the Tesseract engine is that it cannot detect if the letters are not on a line. For the proposed model, it has great difficulty in distinguishing the characters "O" and "0" and "Z" and "2".

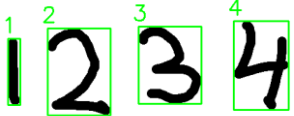

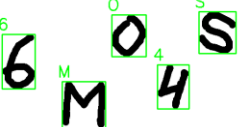
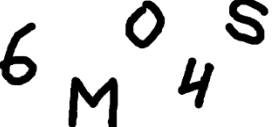




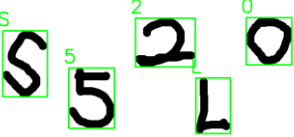
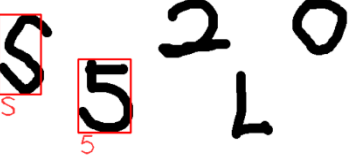
Proposed method	Tesseract engine
	
	
	
	
	

Table 4-4 Predictions on plain text

The table above shows a few examples of results from the testing between the proposed method and the Tesseract engine.

4.4 Text Predictions on Patient Medical Record Form

4.4.1 Extracting Region of Interest

A region of interest (ROI) represents a specific area where you tend to do something. To avoid extracting duplicate or unneeded data, the system must know the exact placement of the detected text. In our situation, because the patient's medical form contains a variety of information, such as name, age, gender, diagnosis, and so on. As a result, we must crop the area of interest and feed it into the model for prediction.

PATIENT MEDICAL RECORD
KLINIK FYP
CONFIDENTIAL

Name:	Gender:	IC:	DOB:
Sex:	Address:		
Final Diagnosis:			
Summary:			
Name of MO:	Signature	Date:	

Figure 4-4 Original Form of Patient Medical Record

Figure 4-4 shows the original empty form of the patient's medical record. Note that there are different headings above the region that do not need to be extracted. The headings of this information should be skipped and only the information in the area where people write is needed for text recognition. In order to crop the region of interest, a ROI point needs to be specified beforehand, and after looping through the ROI, a mask needs to be placed over the region of interest and cropped it. The cropped image can now be sent to the model for prediction.

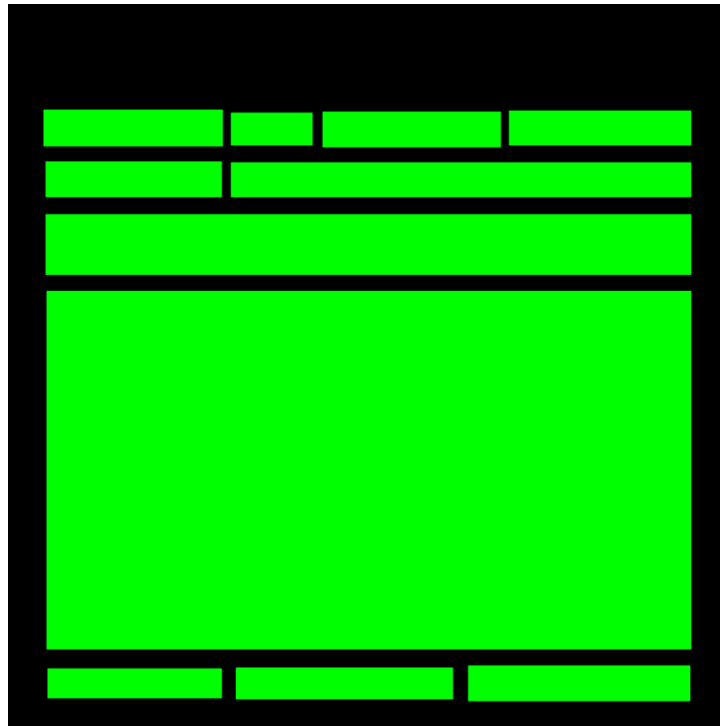


Figure 4-5 Mask created over Region of Interest

PATIENT MEDICAL RECORD
KLINIK FYP
CONFIDENTIAL

Name:	Gender:	IC:	DOB:
SAM	MALE	0001237833	11 JAN 2022
Sex:	Address:		
MAN	KAMPAR 123		
Final Diagnosis:			
CANCER			
Summary:			
Name of MO:	Signature	Date:	
ADAM		12 FEB 2022	

Figure 4-6 Mask over interested region of patient medical record

According to Figure 4-5, the region of interest is masked by a rectangle that indicates the location where text detection and recognition is needed. This is the mask that needs to be hovered over the patient medical form to crop out the region so that only useful data is ready for prediction to prevent any unrelated data has been detected.

4.4.2 Evaluation of Model Prediction for Patient Medical Records

A total of 50 patient medical forms with all the different data are sent to the model for prediction. The cropped images will be sent to the model and then after some pre-processing process, the accuracy of text recognition will be tested. The following are the results of the predictions.

Total Predictions	Correct Prediction	Accuracy
50	44	88%

Table 4-5 Predictions result on patient medical records by form

According to Tables 4-5, the proposed model achieves an accuracy of 80% in detecting correct patient medical data. Some incorrect results exist due to the confusion of the system for characters o or 0, z or 2. In addition to this, characters that are close to each other are difficult to be detected by the system. This resulted in the system not being able to fully detect the correct text from the form.

Total Predictions	Correct Prediction	Accuracy
450	415	92.2%

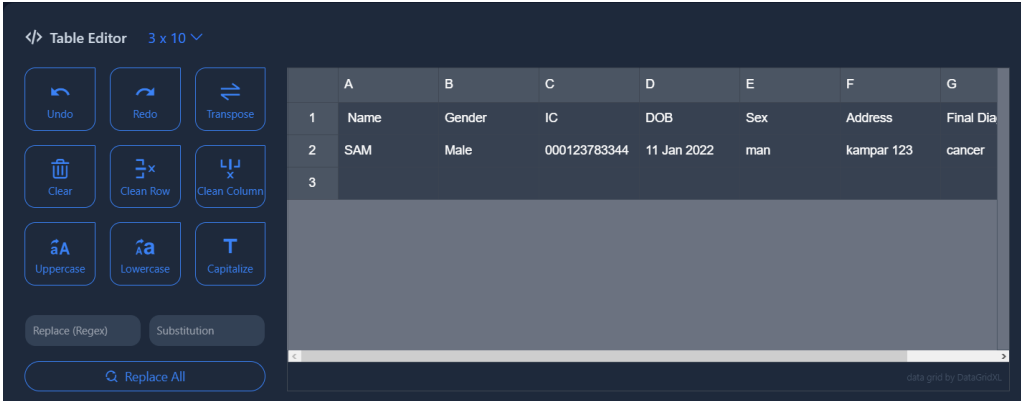
Table 4-6 Predictions result on patient medical records by fields

In total, there are 450 fields to be detected and identified. The proposed system achieved 85% accuracy in predicting each field in all patients' medical records. Out of 450 fields, 360 fields were successfully predicted without any errors. The characters in the remaining fields are either not recognized or are incorrectly predicted by the system due to confusion of characters.

4.5 Time consumed on manual transcribe and AI transcribe

The model was evaluated by comparing the process time of manual transcription and artificial intelligence transcription. To evaluate the efficiency of the model built on transcribing the patient's medical records into xml files, 50 examples of medical reports were provided as input to the model. The trained model will perform word recognition and then write the recognition results to the xml file. The entire process time of each example is recorded.

To obtain process time using the manual transcription method, 50 participants over the age of 18 were randomly selected to participate in the manual transcription process. They were each given an example of a patient record that was also randomly selected from the set used to obtain AI Transcribe process time. They were then asked to manually translate the patient records into xml files using an online website at <https://tableconvert.com/xml-generator>. This website provided a simple interface so that users could form an xml file with the expected value without regard to syntax or formatting. In other words, the user simply reads the values from the patient record example and enters them into the provided interface. The time spent by the participant in this action will be collected as the process time using the manual transcription method.



The screenshot shows a web interface titled "Table Editor" with a "3 x 10" dropdown. On the left, there are several utility buttons: Undo, Redo, Transpose, Clear, Clean Row, Clean Column, Uppercase, Lowercase, Capitalize, Replace (Regex), Substitution, and a "Replace All" button. The main area displays a table with the following data:

	A	B	C	D	E	F	G
1	Name	Gender	IC	DOB	Sex	Address	Final Dia
2	SAM	Male	000123783344	11 Jan 2022	man	kampar 123	cancer
3							

Figure 4-7 Screenshot of website inputted data

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <root>
3 <row>
4 <Name>SAM</Name>
5 <Gender>Male</Gender>
6 <IC>000123783344</IC>
7 <DOB>11 Jan 2022</DOB>
8 <Sex>man</Sex>
9 <Address>kampar 123</Address>
10 <Final Diagnosis>cancer</Final Diagnosis>
11 <Summary>aaa</Summary>
12 <MO>adam</MO>
13 <Date>12 Feb 2022</Date>
14 </row>
15 </root>
16
```

Figure 4-8 XML file generated by website

```
<?xml version="1.0" ?>
<data>
  <medical_records>
    <name>Sam</name>
    <gender>male</gender>
    <ic>333-11-2222</ic>
    <dob>11 Jan 2000</dob>
    <sex>m</sex>
    <address>12 Kampar</address>
    <diagnosis>diabetes</diagnosis>
    <summary>abc</summary>
    <MO>Adam</MO>
    <date>12 Jun 2022</date>
  </medical_records>
</data>
```

Figure 4-9 XML file generated using AI transcribe

After obtaining required data from both transcribe methods, the process time taken has been illustrated into bar chart so that any significant difference can be observed more easily.

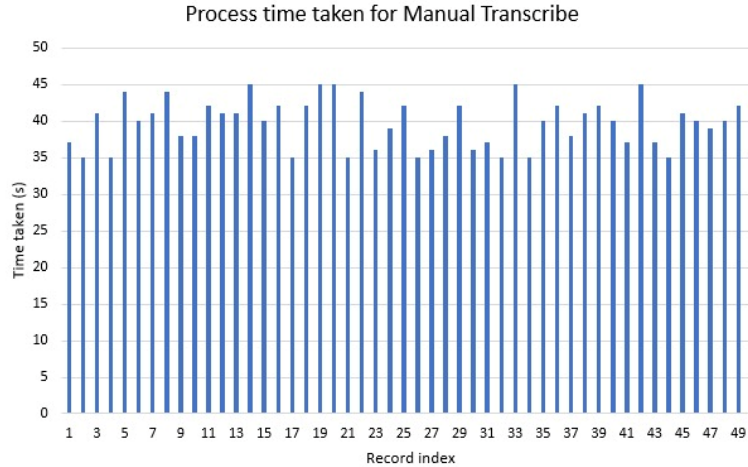


Figure 4-10 Process time taken for Manual Transcribe

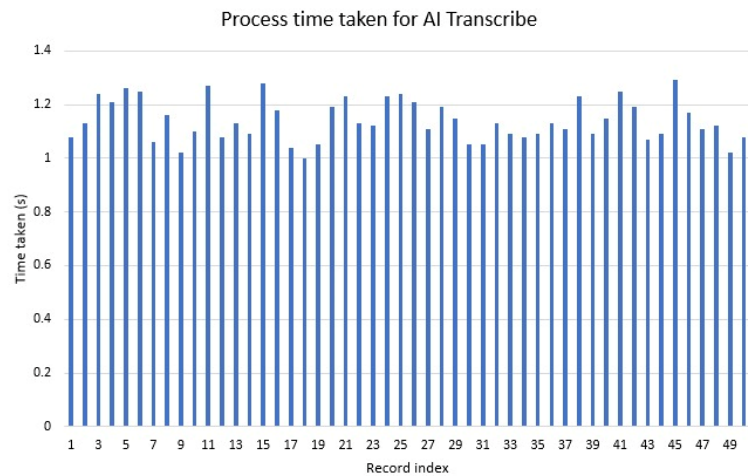


Figure 4-11 Process time taken for AI Transcribe

From Figure 4-10, it can be observed that the translation of patient medical record into xml file using Manual Transcribe method required between 35 to 45 seconds, and took an average of 39.66 seconds. On the other hand, the AI Transcribe methods has taken around 1 to 1.3 second, with an average processing time of 1.14 second, having a significant difference with the process time of Manual Transcribe.

Methods	Manual Transcribe	AI Transcribe
Minimum time taken (s)	35	1
Maximum time taken (s)	45	1.3
Total time taken (s)	1983	57
Average time taken (s)	39.66	1.14

Table 4-7 Comparison between manual transcribe time and AI transcribe time

The above table shows the maximum and minimum times for transcription using manual transcribe and AI transcribe. It also shows that the average time for AI transcription is much faster than manual transcribe, which suggests that the proposed model is a better choice compared to manual transcription.

4.6 Error rate between manual transcribe and AI transcribe

Methods	Total Transcription	Total Correct Transcription	Error	Accuracy
Manual Transcribe	50	39	22%	78%
AI Transcribe	50	44	12%	88%

Table 4-8 Error rate between manual transcribe and AI transcribe

According to Tables 4-8, among the 50 transcriptions of patient records, the error rate for manual transcription was 22%, while the error rate for manual transcription was 12%. The higher error rate of manual transcription was due to human errors that occurred during the transcription process. Throughout the comparison, AI transcriptions can effectively reduce the error rate caused by human errors, presenting a more reliable nature because AI models have great potential for accuracy if the model is trained and refined with a sufficient amount of data, but human errors are inconsistent and unavoidable.

4.7 Future Remarks

Overall, the proposed model is still not a very robust classification model to detect the word that connect to each other. In the future, various techniques can still be applied to enhance the system. There are several reasons that may lead to the inaccuracy of the predictions.

- There may be unbalanced model training datasets.
- The data in the alphabet dataset may have confused words.
- There may be some bad hyperparameter settings to train the model.

Chapter 5 Conclusion

5.1 Project Review

We present a CNN-based OCR engine for extracting text from pictures and converting it to machine-readable text. Many online OCR systems are incapable of detecting and extracting handwritten text from the images. Therefore, the goal of this project is to create a deep learning-based OCR model that can identify handwritten text in images and convert it to machine-encoded format. The MNIST 0-9 dataset and the Kaggle A-Z dataset were used to train our model, and the CNN model was chosen as our based model for this project. Furthermore, the ReLU and Softmax functions were utilized as activation functions to direct our model's learning of complicated patterns in the input pictures. Our model has a validation accuracy of 0.9866 and a validation loss of 0.0571 throughout 30 epochs in 32 batch sizes. Furthermore, in 20 test sets, our proposed model beats the existing Tesseract OCR engine, attaining 80 percent accuracy in identifying a word and 93.75 percent accuracy in predicting a single letter. In addition, medical forms for 50 patients have been tested and 80% of accuracy has been achieved.

In conclusion, the project's goal is to create an OCR system that can extract handwritten text from images. The project was able to achieve excellent training accuracy while also outperforming the previous OCR engine at detecting handwritten text.

5.2 Future Work

Some restrictions might be improved further to increase the usefulness and accuracy of our proposed product. For starters, the model predicts characters "O" and "0," as well as "Z" and "2." To address this issue, a model that predicts words with letters or numbers can be employed to avoid the recognition model from being confused between characters. Furthermore, the dataset only contains capital characters 0-9 and A-Z, with no lowercase letters a-z. The a-z dataset may be gathered and trained to improve the model's ability to predict additional characters. In addition, symbolic predictions can be added to enhance the model, which will increase the usefulness of the model for predicting variety data.

Bibliography

- [1] N. Liu *et al.*, “A New Data Visualization and Digitization Method for Building Electronic Health Record,” *Proc. - 2020 IEEE Int. Conf. Bioinforma. Biomed. BIBM 2020*, 2020, doi: 10.1109/BIBM49941.2020.9313116.
- [2] Gilad David Maayan, “Optical Character Recognition using Deep Neural Network,” *International Journal of Computer Applications*, Apr. 2020.
- [3] Abhinav Somani, “The Future Of OCR Is Deep Learning,” *Forbes*, Sep. 2019.
- [4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, 1998, doi: 10.1109/5.726791.
- [5] Shipra Saxena, “Introduction to Long Short Term Memory (LSTM),” *Analytics Vidhya*, 2021. <https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory- lstm/> (accessed Aug. 14, 2021).
- [6] T. Cheng, “Introduction to Google Cloud Vision,” *Nanonets*, 2021. <https://nanonets.com/blog/google-cloud-vision/>.
- [7] Misha Iakovlev, “The Battle of the OCR Engines,” *FUZZY LABS*, 2020. <https://fuzzylabs.ai/blog/the-battle-of-the-ocr-engines/> (accessed Aug. 15, 2021).
- [8] Rahul Agarwal, “Deep Learning Based OCR for Text in the Wild,” *Nanonets*, 2021. <https://nanonets.com/blog/deep-learning-ocr/#but-why-really> (accessed Aug. 14, 2021).
- [9] A. S. Timmaraju and V. Khanna, “Detecting and Recognizing Text in Natural Images using Convolutional Networks,” 2015.
- [10] J. Yang, P. Ren, and X. Kong, “Handwriting Text Recognition Based on Faster R-CNN,” *Proc. - 2019 Chinese Autom. Congr. CAC 2019*, 2019, doi:

10.1109/CAC48633.2019.8997382.

- [11] R. Girshick, “Fast R-CNN,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, 2015, doi: 10.1109/ICCV.2015.169.
- [12] M. B. Bora, D. Daimary, K. Amitab, and D. Kandar, “Handwritten Character Recognition from Images using CNN-ECOC,” *Procedia Comput. Sci.*, vol. 167, no. 2019, 2020, doi: 10.1016/j.procs.2020.03.293.
- [13] S. Choudhary, N. K. Singh, and S. Chichadwani, “Text Detection and Recognition from Scene Images using MSER and CNN,” *Proc. 2018 2nd Int. Conf. Adv. Electron. Comput. Commun. ICAECC 2018*, 2018, doi: 10.1109/ICAIECC.2018.8479419.
- [14] S. Hassan, A. Irfan, A. Mirza, and I. Siddiqi, “Cursive Handwritten Text Recognition using Bi-Directional LSTMs: A Case Study on Urdu Handwriting,” *Proc. - 2019 Int. Conf. Deep Learn. Mach. Learn. Emerg. Appl. Deep. 2019*, 2019, doi: 10.1109/Deep-ML.2019.00021.
- [15] V. Golovko *et al.*, “Deep convolutional neural network for recognizing the images of text documents,” *CEUR Workshop Proc.*, vol. 2386, pp. 297–306, 2019.

Text Recognition (OCR) for Patient Records Digitization using CNN

Introduction

- OCR is a system that translate the text in the image into machine-encoded format for further processing.

Problem Statement

- Moving from a traditional paper-based database system to an electronic-based database system is a daunting task.
- Existing OCR engine not so accurate in recognizing handwritten text.

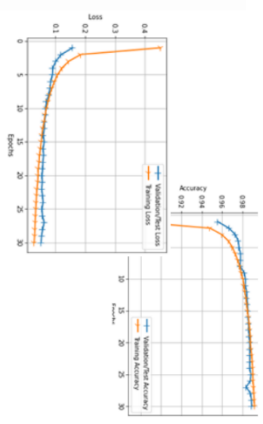
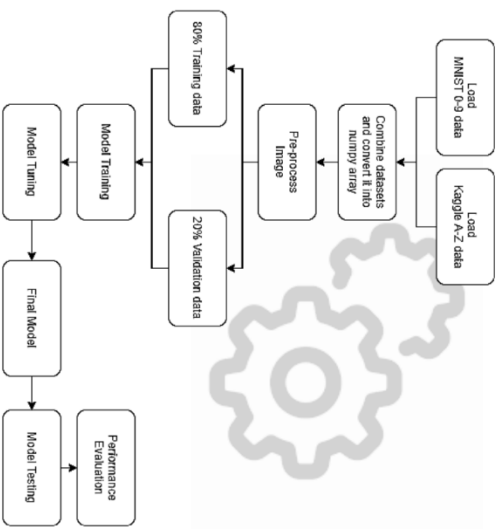
Result

- we achieved a validation accuracy of 0.9866 and also a validation loss of 0.0571
- 93.75 percent accuracy in predicting a single letter.
- 80 percent accuracy in identifying a word

Objectives

1. To build a custom hand-writing dataset that includes a more diversified types of handwriting sample.
2. To train a robust OCR model for hand-written text recognition using a CNN.
3. To transcribe text from patients' report cards using the proposed OCR using region-based text recognition.

Methodology



Conclusion

- Manage to build a CNN model to extract text from an image to machine-encoded format.
- Have high accuracy result in recognizing handwritten text

Appendices

Poster

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	Study week no.: 2
Student Name & ID: Ong Zi Leong 18ACB02522	
Supervisor: Dr. Aun Yichiet	
Project Title: Text Recognition (OCR) for Patient Records Digitization using CNN	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Revise report to caught up current project status.
- Discussion with supervisor on current project status

2. WORK TO BE DONE

- Data collecting

3. PROBLEMS ENCOUNTERED

- No problem encountered so far

4. SELF EVALUATION OF THE PROGRESS

Self-assigned tasks are completed within the time.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	Study week no.: 3
Student Name & ID: Ong Zi Leong 18ACB02522	
Supervisor: Dr. Aun Yichiet	
Project Title: Text Recognition (OCR) for Patient Records Digitization using CNN	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Data collecting
- Preprocessing data

2. WORK TO BE DONE

- Start model training and model testing

3. PROBLEMS ENCOUNTERED

- No problem encountered so far

4. SELF EVALUATION OF THE PROGRESS

Self-assigned tasks are completed within the time.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	Study week no.: 4
Student Name & ID: Ong Zi Leong 18ACB02522	
Supervisor: Dr. Aun Yichiet	
Project Title: Text Recognition (OCR) for Patient Records Digitization using CNN	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Model training
- Model testing

3. WORK TO BE DONE

- Solving data imbalanced issues
- Discussion with supervisor about the current status

3. PROBLEMS ENCOUNTERED

- Data imbalanced

4. SELF EVALUATION OF THE PROGRESS

Self-assigned tasks are completed within the time.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	Study week no.: 5
Student Name & ID: Ong Zi Leong 18ACB02522	
Supervisor: Dr. Aun Yichiet	
Project Title: Text Recognition (OCR) for Patient Records Digitization using CNN	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Discussion with supervisor
- Solved data imbalanced issues
- Model training and model testing

2. WORK TO BE DONE

- Fixing overfitting issues

3. PROBLEMS ENCOUNTERED

- Model overfitting

4. SELF EVALUATION OF THE PROGRESS

Self-assigned tasks are completed within the time.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	Study week no.: 6
Student Name & ID: Ong Zi Leong 18ACB02522	
Supervisor: Dr. Aun Yichiet	
Project Title: Text Recognition (OCR) for Patient Records Digitization using CNN	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Managed to solve the model overfitting issues.

2. WORK TO BE DONE

- Model tuning to increase accuracy
- Model testing

3. PROBLEMS ENCOUNTERED

- No problem encountered so far

4. SELF EVALUATION OF THE PROGRESS

Self-assigned tasks are completed within the time.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	Study week no.: 7
Student Name & ID: Ong Zi Leong 18ACB02522	
Supervisor: Dr. Aun Yichiet	
Project Title: Text Recognition (OCR) for Patient Records Digitization using CNN	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Model testing

2. WORK TO BE DONE

- Restructure the code
- Finalize the model

3. PROBLEMS ENCOUNTERED

- Currently no problem

4. SELF EVALUATION OF THE PROGRESS

Self-assigned tasks are completed within the time.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	Study week no.: 8
Student Name & ID: Ong Zi Leong 18ACB02522	
Supervisor: Dr. Aun Yichiet	
Project Title: Text Recognition (OCR) for Patient Records Digitization using CNN	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Preparing the patient medical form for testing

2. WORK TO BE DONE

- Extract ROI of the form
- Developed a ROI generator system

3. PROBLEMS ENCOUNTERED

- No problem encountered so far

4. SELF EVALUATION OF THE PROGRESS

Self-assigned tasks are completed within the time.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	Study week no.: 9
Student Name & ID: Ong Zi Leong 18ACB02522	
Supervisor: Dr. Aun Yichiet	
Project Title: Text Recognition (OCR) for Patient Records Digitization using CNN	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Extracted the data form ROI using self-developed system
- Writing code to translate extracted data to xml file

2. WORK TO BE DONE

- Transform data to xml form

4. PROBLEMS ENCOUNTERED

- No problem encountered so far

4. SELF EVALUATION OF THE PROGRESS

Self-assigned tasks are completed within the time.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	Study week no.: 10
Student Name & ID: Ong Zi Leong 18ACB02522	
Supervisor: Dr. Aun Yichiet	
Project Title: Text Recognition (OCR) for Patient Records Digitization using CNN	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Completed all source code problem

2. WORK TO BE DONE

- Writing report.

3. PROBLEMS ENCOUNTERED

- No problem encountered so far

4. SELF EVALUATION OF THE PROGRESS

Self-assigned tasks are completed within the time.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	Study week no.: 11
Student Name & ID: Ong Zi Leong 18ACB02522	
Supervisor: Dr. Aun Yichiet	
Project Title: Text Recognition (OCR) for Patient Records Digitization using CNN	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Writing a part of the report.

2. WORK TO BE DONE

- Complete the report writing
- Attach requirement documents for submission

3. PROBLEMS ENCOUNTERED

- No problem encountered so far

4. SELF EVALUATION OF THE PROGRESS

Self-assigned tasks are completed within the time.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	Study week no.: 12
Student Name & ID: Ong Zi Leong 18ACB02522	
Supervisor: Dr. Aun Yichiet	
Project Title: Text Recognition (OCR) for Patient Records Digitization using CNN	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Report completed

2. WORK TO BE DONE

- Prepare for presentation

3. PROBLEMS ENCOUNTERED

- No problem encountered so far

4. SELF EVALUATION OF THE PROGRESS

Self-assigned tasks are completed within the time.



Supervisor's signature




Student's signature

Turnitin report

Optical character recognition (OCR) is widely used to transcribe texts from images in computer vision. Although current OCR methods can accurately transcribe printed text (structured), they often fall short on unstructured or handwritten text recognition. This project proposed a text recognition method to recognize handwritten text on patients' clinical data using a convolutional neural network (CNN). We compiled custom handwriting datasets from *MNIST 0-9* and *Kaggle A-Z datasets* to add more handwriting diversity in training a more robust OCR model. The CNN has 3-convolutional layers to learn high-level features and a dropout layer to prevent overfitting. The preliminary results showed that the proposed model achieved 93.75% classification accuracy while Tesseract (the state-of-the-art OCR) scored 69.79%. The data will be transformed from handwritten text to computer-readable text and then stored in files in xml form for further development.

Even in the 21st century, most clinics or hospitals still rely on traditional paperwork to access the medical record or patients' information for their daily operations. A traditional paper-based record system involving recording the patients' personal information, and



The screenshot shows a Turnitin Match Overview window. At the top, it displays a similarity score of 6%. Below this, there is a list of 9 matches, each with a rank, a source name, and a similarity percentage. The matches are as follows:

Rank	Source	Similarity
1	www.mdpi.com (Internet Source)	1%
2	dokumen.pub (Internet Source)	<1%
3	www.commonlounge.c... (Internet Source)	<1%
4	www.nanonets.com (Internet Source)	<1%
5	sucra.repo.nii.ac.jp (Internet Source)	<1%
6	"Computer Networks a... (Publication)	<1%
7	Shahbaz Hassan, Ayes... (Publication)	<1%
8	www.researchgate.net (Internet Source)	<1%
9	"Artificial Neural Netwo... (Publication)	<1%

At the bottom of the screenshot, there are controls for 'Text-Only Report' and 'High Resolution' (set to 'On').

Text Recognition (OCR) for Patient Records Digitization using CNN

ORIGINALITY REPORT

6%	4%	4%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.mdpi.com Internet Source	1%
2	dokumen.pub Internet Source	<1%
3	www.commonlounge.com Internet Source	<1%
4	www.nanonets.com Internet Source	<1%
5	sucra.repo.nii.ac.jp Internet Source	<1%
6	"Computer Networks and Inventive Communication Technologies", Springer Science and Business Media LLC, 2022 Publication	<1%
7	Shahbaz Hassan, Ayesha Irfan, Ali Mirza, Imran Siddiqi. "Cursive Handwritten Text Recognition using Bi-Directional LSTMs: A Case Study on Urdu Handwriting", 2019 International Conference on Deep Learning	<1%

and Machine Learning in Emerging Applications (Deep-ML), 2019

Publication

8	www.researchgate.net Internet Source	<1 %
9	"Artificial Neural Networks and Machine Learning – ICANN 2017", Springer Science and Business Media LLC, 2017 Publication	<1 %
10	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1 %
11	www.diva-portal.org Internet Source	<1 %
12	Submitted to University of Western Sydney Student Paper	<1 %
13	"Intelligent Computing Theories and Application", Springer Science and Business Media LLC, 2017 Publication	<1 %
14	"Natural Language Processing and Chinese Computing", Springer Science and Business Media LLC, 2021 Publication	<1 %
15	Maryam Kouzehgar, Yokhesh Krishnasamy Tamilselvam, Manuel Vega Heredia, Mohan Rajesh Elara. "Self-reconfigurable façade-	<1 %

cleaning robot equipped with deep-learning-based crack detection based on convolutional neural networks", Automation in Construction, 2019

Publication

16	digitalintellectuals.hypotheses.org Internet Source	<1 %
17	www.ijrst.com Internet Source	<1 %
18	heartbeat.fritz.ai Internet Source	<1 %
19	link.springer.com Internet Source	<1 %
20	mdpi-res.com Internet Source	<1 %
21	tech-related.com Internet Source	<1 %
22	"Artificial Neural Networks in Pattern Recognition", Springer Science and Business Media LLC, 2018 Publication	<1 %
23	ebin.pub Internet Source	<1 %

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date:	Page No.: 1of 1



FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Ong Zi Leong
ID Number(s)	18ACB02522
Programme / Course	Bachelor of Computer Science
Title of Final Year Project	Text Recognition (OCR) for Patient Records Digitization using CNN

Similarity	Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)
Overall similarity index: <u> 6 </u> % Similarity by source Internet Sources: <u> 4 </u> % Publications: <u> 4 </u> % Student Papers: <u> 1 </u> %	
Number of individual sources listed of more than 3% similarity: <u> 0 </u>	
Parameters of originality required and limits approved by UTAR are as Follows: (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8</i>	

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor
 Name: _____
 Date: 21 April 2022

Signature of Co-Supervisor
 Name: _____
 Date: _____



UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	18ACB02522
Student Name	Ong Zi Leong
Supervisor Name	Dr Aun Yi Chiet

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Plastic Cover (for hardcopy)
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

(Signature of Student)

Date: 21 April 2022

