# Enhancing Accuracy in Optical Character Recognition of Sensor Readings: A Comparative Study of Tesseract and CRNN Models with Emphasis on Image Preprocessing

by

Aidan Dennehy [R00145278]

For the module DATA9003 - Research Project as part of the
Master of Science in Data Science and Analytics, Department of Matematics

Supervisor: Dr Alex Vakaloudis

July 2023

# Declaration of Authorship

I, Aidan Dennehy , declare that this thesis titled, ”Enhancing Accuracy in Optical Character Recognition of Sensor Readings: A Comparative Study of Tesseract and CRNN Models with Emphasis on Image Preprocessing” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for an undergraduate degree at Munster Technological University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at Munster Technological University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this project report is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ____________________________________

Date: ____________________________________

Munster Technological University

# *Abstract*

Department of Matematics

Master of Science in Data Science and Analytics

by Aidan Dennehy [R00145278]

This research is primarily dedicated to the formulation of an innovative method for accurately interpreting sensor data obtained from digitized images. Confronting inherent challenges such as diminished contrast and subpar image quality, often associated with sensor readings, the study exploits Optical Character Recognition (OCR). This is accomplished employing two distinct techniques: Tesseract and Convolutional Recurrent Neural Network (CRNN) models.

An unique feature of the research lies in its novel image preprocessing steps, specifically the masking of red and green colors prior to conversion to grayscale. This process considerably augments the efficacy of OCR. Additionally, the study underlines the critical importance of correct font selection for each sensor to enhance reading accuracy.

The findings highlight the essential role of image quality and contrast in OCR, while presenting an innovative approach to image preprocessing for improved results. The potential implications of this research are extensive and could shape future undertakings in the fields of OCR and sensor digitization. The research underscores the vital aspects of image preprocessing and reveals how precise interventions can markedly improve sensor data interpretation from digitized images.

# *Acknowledgements*

Acknowledgements here . . .

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| **LAH** | **L**ist **A**bbreviations **H**ere |

*For/Dedicated to/To my…*

# Chapter 1

# Introduction

## 1.1 Area of Interest

The area of interest for this literature review is the intersection of computer vision, optical character recognition (OCR), and deep learning, with particular emphasis on the Tesseract OCR engine and Convolutional Recurrent Neural Networks (CRNNs). These technological advancements have revolutionized the way machines recognize and understand visual information, especially digits. Given their diverse and significant applications, ranging from digitizing written documents to aiding autonomous vehicle navigation, they hold vast potential for transforming many sectors. This research focuses on exploring the principles that underlie these tools, their performance in real-world applications, and the possibilities they offer for future development. This involves assessing the strengths of these systems, identifying their limitations, and suggesting potential areas of improvement. Moreover, it considers how these technologies are pushing the boundaries of OCR, paving the way for more sophisticated and versatile tools that can better navigate the complexities and variations in text size, font, and orientation often encountered in different visual scenes.

Optical Character Recognition (OCR) technology has seen substantial advancements in recent years, transforming the process of data extraction from visual mediums to digital formats. This technology, crucial in numerous fields ranging from document

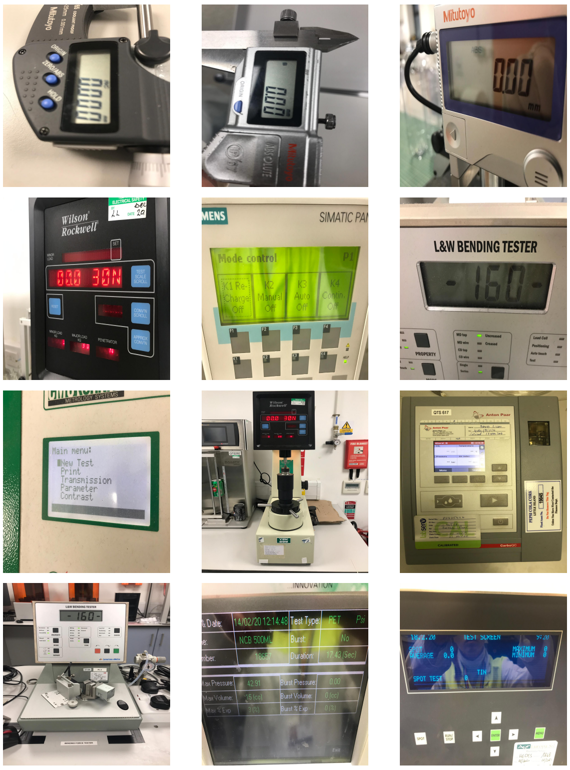


Table 1.1: Nimbus Sensor Images

digitization to automated data entry systems. OCR holds specific importance when it comes to interpreting sensor readings, a key aspect of data-driven industries. The necessity for accurate, efficient, and automated reading of sensor-generated data has led to the investigation of various techniques and models within the OCR domain.

Two models which feature prominently emerged as potential solutions, namely Tesseract, an open-source OCR engine sponsored by Google, and Convolutional Recurrent Neural Network (CRNN), a combination of CNN, RNN, and Connectionist Temporal Classification that offers promising results in scene text recognition tasks.

In OCR applications, image preprocessing has a pivotal role. It prepares an image for further processing by reducing noise and unnecessary details and enhancing features that are important for later stages, thereby directly influencing the accuracy of the final output. Among various preprocessing techniques, the novel approach of red and green color masking, followed by conversion to grayscale, has shown to significantly improve the accuracy of digit recognition.

## 1.2 Motivation

The motivation behind this research stems from the challenges encountered in the manual and infrequent readings of environmental sensors in various operational settings such as factories. These sensors, while accurate and essential, lack a means for continuous data capture. Typically, an individual manually reads the sensor outputs at fixed intervals, which could range from hourly to daily. This method, while necessary, is prone to human error, potentially leading to inaccuracies in the recorded data and subsequent analysis reports. Furthermore, the infrequency of readings may result in delays in responding to critical sensor data, which could precipitate further issues. These complications could be mitigated with the implementation of Optical Character Recognition (OCR) technology. By enabling continuous, automated readings of these sensors, OCR has the potential to not only reduce errors but also ensure timely reaction to important sensor changes, optimizing the overall operation and efficiency of the systems.

## 1.3 Aims and Objectives

The primary aim of this research is to improve the efficiency and accuracy of Optical Character Recognition (OCR) on images of sensor readings by applying novel preprocessing steps and optimizing image capture settings. This project focuses on two OCR methods: Tesseract OCR and Convolutional Recurrent Neural Network (CRNN) models, both widely used for text recognition tasks.

1. **Objective 1: Assess the Performance of Tesseract OCR and CRNN Models**

   The initial phase of the project involves a baseline assessment of the existing OCR systems. The Tesseract OCR and CRNN models will be employed on multiple sets of image files, with each set undergoing a 'global run'. This step aims to establish a baseline for the performance of these systems without any preprocessing measures.

2. **Objective 2: Design and Implement Image Preprocessing Techniques**

   In an attempt to enhance the quality of the images and subsequently improve the

OCR results, various image preprocessing methods will be introduced. A primary focus will be the implementation of color masking (specifically for green and red) prior to the conversion to grayscale. This approach aims to make the images clearer and more conducive to OCR.

3. **Objective 3: Identify Optimal Image Capture Settings**

   In parallel with image preprocessing, the research will aim to identify the optimal parameters for image capture to further enhance OCR performance. The specific parameters of focus will include camera contrast, distance, and lighting.

4. **Objective 4: Compare and Evaluate the Effects of Preprocessing and Optimized Capture Settings on OCR Results**

   Once preprocessing measures and optimized image capture settings have been implemented, the images will undergo OCR using both Tesseract and CRNN models. This step aims to ascertain the joint impact of preprocessing and optimal capture parameters on the performance of OCR systems.

5. **Objective 5: Analyze and Report Findings**

   The final objective of the research is to analyze the findings and draw conclusions on the effectiveness of the proposed preprocessing techniques and optimal capture parameters. This analysis aims to fill a gap in the literature, which currently lacks comprehensive studies on the potential benefits of image preprocessing and capture settings optimization for OCR of sensor readings.

In conclusion, this research seeks not only to enhance our understanding of how image preprocessing and capture optimization can improve OCR outcomes, but also to provide practical insights that could inform the future development of OCR systems.

## 1.4 Structure of the Thesis

This thesis is organized into five main chapters, each covering a specific aspect of the study:

1. **Chapter 1: Introduction**

   This chapter provides an overview of the research, outlining the area of interest and motivation behind the study. It also presents the aims and objectives that guide the research.

2. **Chapter 2: Literature Review**

   This chapter reviews previous research relevant to this study. It begins with a general introduction to the field, followed by specific sections on Tesseract OCR, CRNN OCR, and other OCR systems, examining their strengths, weaknesses, and applications.

3. **Chapter 3: Methodology**

   This chapter presents the research methodology, including the design and implementation of image preprocessing techniques and the methods used to identify optimal image capture settings. It also details how the Tesseract and CRNN OCR systems are applied in this research.

4. **Chapter 4: Results**

   This chapter presents the findings of the study. It includes an analysis of the OCR performance before and after the application of the preprocessing methods and optimized image capture settings.

5. **Chapter 5: Discussion and Conclusion**

   This final chapter discusses the implications of the research findings, drawing conclusions about the effectiveness of the proposed techniques for improving OCR performance. It also highlights potential areas for future research.

# Chapter 2

# Literature Review

## 2.1 Introduction

As we stand on the precipice of a future molded by artificial intelligence and machine learning, one domain that is experiencing considerable progress is Optical Character Recognition (OCR). In this dynamic and continuously evolving field, there are many techniques which have emerged among the significant game-changers, two of these are the Tesseract OCR engine and Convolutional Recurrent Neural Networks (CRNNs). Tesseract, initially developed by Hewlett-Packard and later adopted by Google, is a pioneering engine that converts images of text into machine-encoded text, offering utilities across numerous applications. On the other hand, CRNNs, a deep learning-based approach, combine the spatial feature extraction capabilities of Convolutional Neural Networks (CNNs) with the sequential data processing capacity of Recurrent Neural Networks (RNNs). These networks have set new benchmarks in the realm of scene text recognition, overcoming the challenges posed by variations in text sizes, fonts, and orientations. This literature review delves into the intricacies of these advanced tools, shedding light on their principles, applications, strengths, and potential areas for improvement, thereby enriching our understanding of current trends in OCR technology and pointing to the future possibilities.

This literature review explores the current state of OCR technologies, with a particular focus on Tesseract and CRNN models. It delves into various image preprocessing techniques, emphasizing the unique method of red and green color masking before conversion to grayscale. Lastly, it investigates the role of font selection in enhancing OCR accuracy, thereby setting the context for the subsequent research.

While this review focuses on the capabilities of Tesseract OCR and Convolutional Recurrent Neural Networks (CRNNs) in the OCR domain, it's important to acknowledge that the OCR landscape is not limited to these technologies. Many other methods play equally significant roles in expanding the OCR frontiers and opening up new avenues for research and application. Long Short-Term Memory Networks (LSTMs), Transformers, attention-based OCR models, rule-based systems, Support Vector Machines (SVMs), Hidden Markov Models (HMMs), K-Nearest Neighbors (KNN), and template matching are some of these diverse methodologies that provide unique perspectives and solutions in the OCR realm. Each of these methods has its distinctive advantages, making them optimal for certain types of tasks, as well as its limitations, requiring continuous research and development for enhancement. However, the scope of this review will mainly revolve around Tesseract and CRNNs, while the mentioned methods provide an essential context for understanding the broader OCR ecosystem.

## 2.2 Tesseract OCR

Optical character recognition (OCR) is the process of converting images of text into machine-readable text. Tesseract is an open-source Optical Character Recognition (OCR) engine that is widely used for a variety of tasks, including document digitization, machine translation, and data entry. Tesseract was developed at HP between 1984 and 1994. Initially conceived as a PhD research project to improve OCR performance for HP's scanners, it outperformed contemporary commercial OCR engines but never became an HP product. Its development was mainly focused on improving rejection efficiency rather than base-level accuracy. Despite being shelved in 1994, Tesseract proved its prowess in the 1995 UNLV Annual Test of OCR Accuracy. HP made Tesseract open-source in 2005, hosted at Google Code. Tesseract's architecture assumes binary images as input and uses a step-by-step pipeline for processing, including a unique connected component analysis for detecting inverse text. Its recognition process is two-pass: initial recognition of words feeds an adaptive classifier which subsequently improves text recognition further down the page. A final phase resolves fuzzy spaces and checks alternative hypotheses for the x-height to locate smallcap text. [1]

Figure 2.1: Tesseract OCR
[2]

In this literature review, we will discuss five research papers that have been published on Tesseract OCR. These papers cover a wide range of topics, including the accuracy of Tesseract, the performance of Tesseract on different types of documents, and the use of Tesseract for specific applications. Each study presents unique findings, and together they create a comprehensive overview of the current state and potential trajectory of Tesseract OCR. Through a detailed exploration of these works, we aim not only to comprehend the nuances of Tesseract OCR but also to contribute to the burgeoning discourse around its implications and opportunities in our increasingly digitized world

In the paper "Benchmarking Object Detection Algorithms for Optical Character Recognition of Odometer Mileage" Andersson et al. compare two state-of-the-art object detection models, Faster R-CNN and YOLO, with an open-source OCR model, Tesseract, for reading the odometer mileage from car dashboard images. They use a dataset of 2,389 images of car odometers and evaluate the models based on mean average precision, prediction accuracy, and Levenshtein distance.

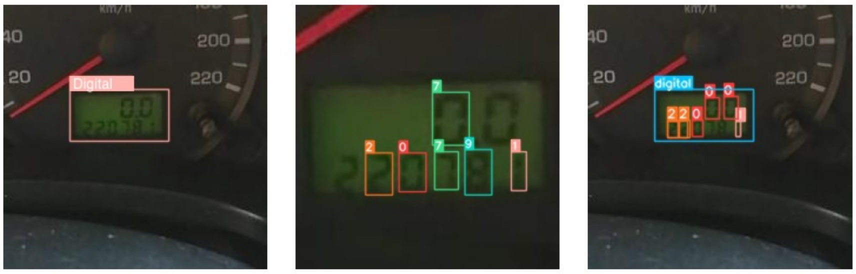


Figure 2.2: Visualization of Boundary box predictions, and digit classification from YOLO-models. left - YOLO Singlebox-model. middle - YOLO Digitbox-model. right - YOLO Multibox-model.
[3]

They find that the object detection models outperform Tesseract on all metrics, and that Faster R-CNN has the highest mAP and accuracy, while YOLO has the lowest Levenshtein distance. [3]

Ahuja et al.'s paper "Detecting Vehicle Type and License Plate Number of different Vehicles on Images" discusses the development of a model that can locate a particular vehicle that the user is looking for depending on two factors 1. the Type of vehicle and the 2. License plate number of the car. The proposed system uses a unique mixture consisting of Mask R-CNN model for vehicle type detection, WpodNet and Tesseract for License Plate detection and Prediction of letters in it. [4]

The first stage of Ahuja et al.'s project involved annotating 2,650 images with a custom dataset using the open-source tool VGG Annotator. The images were annotated with

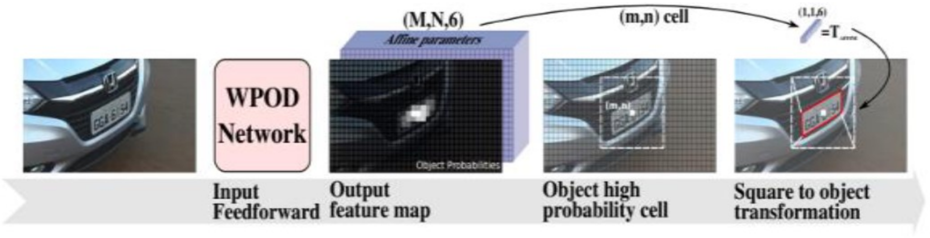


FIGURE 2.3: WpodNet Detection Method [4]

rectangular shapes instead of polygons, and the categories were 2-Wheelers, 3-Wheelers, 4-Wheelers, and vehicles with more than four wheels. The trained Mask R-CNN model achieved an F1 score of 0.72399 on the training data and 0.68 on the test data.

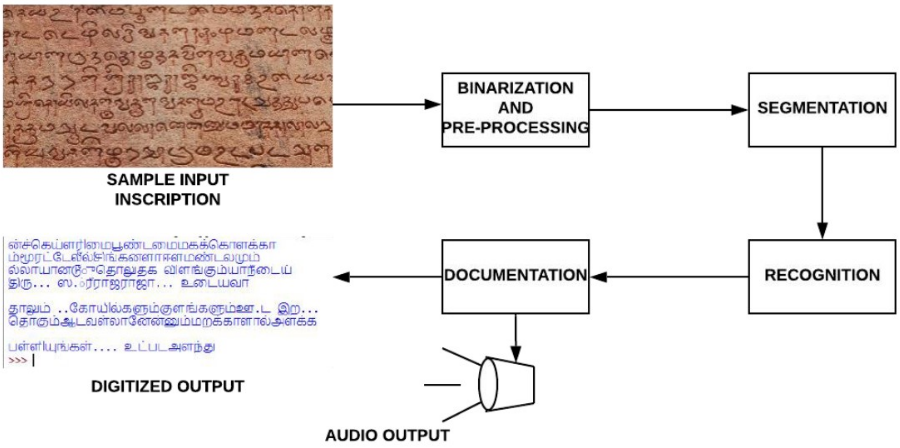
The second stage was split into two parts: license plate detection and letter classification. The WpodNet model was used for license plate detection, achieving an accuracy rate of over 90%. The Tesseract model was then used to predict the letters on the license plate, enhancing speed and accuracy.

The third and final stage of the project involved the model accepting user input for vehicle type, license plate, and image. The model then compared this input with its output to determine if the searched vehicle had been identified.

The final hybrid model uses Mask R-CNN (F1 score: 0.72399 training, 0.68 testing), WpodNet, and pytesseract for car and license plate identification. This can aid in tracking stolen vehicles and assessing parking lot availability.

In an interesting paper, Giridhar et al.'s ”A Novel Approach to OCR using Image Recognition based Classification for Ancient Tamil Inscriptions in Temples” describes a novel approach to OCR using image recognition based classification for ancient Tamil inscriptions in temples. The proposed work focuses on improving optical character recognition techniques for ancient Tamil script which was in use between the 7th and 12th centuries. A data set has been curated using cropped images of characters found on certain temple inscriptions, specific to this time period as a case study.

After using Otsu thresholding method for binarization of the image, a two-dimensional convolution neural network is defined and used to train, classify, and recognize the



```
ன்ச்கெய்ளர்ளைமபூண்டளைம்மகக்கொளக்கா
ம்மூரட்டேளீல்சிங்கனளாஈளமண்டலமும்
ல்லாயானடு◌ாதொலுதக விளங்கும்யாந்டைய்
திரு... ஸ.◌்ர்ராஜராஜா... உடையவா

தாலும் ..கோயில்களும்குளங்களும்ஊ.ட இற...
தொகும்ஆடவள்லானேன்னும்மறக்காளால்அளக்க

பள்ளியுங்கள்.... உட்படஅளந்து
>>> |
```

FIGURE 2.4: Giridhar's proposed architecture [5]

ancient Tamil characters. To implement the optical character recognition techniques, the neural network is linked to the Tesseract using the Pytesseract library in Python. As an added feature, this work also incorporates Google's text to speech voice engine to produce an audio output of the digitized text. Various samples for both modern and ancient Tamil were collected and passed through the system.

The author used Otsu Thresholding Method for binarization of the image and a Two-dimensional Convolution Neural Network to train, classify, and recognize the ancient Tamil characters. To implement the optical character recognition techniques, the neural network is linked to the Tesseract using the Pytesseract library in Python. As an added feature, this work also incorporates Google's text to speech voice engine to produce an audio output of the digitized text.

On average, the combined efficiency of the system for ancient Tamil script was 77.7%, although it varied based on the specific challenges encountered with different types of text and inscriptions. The study acknowledged that further work is needed to overcome these issues and increase the system's accuracy and efficiency.[5]

In Cakic et al's paper "The Use of Tesseract OCR Number Recognition for Food Tracking and Tracing", they discuss the use of computer vision and optical character recognition (OCR) on mobile devices to read serial numbers from wine labels in order to enable applications based on tracking and tracing of each individual wine bottle. The research focuses on the implementation and image processing that improved detection accuracy.

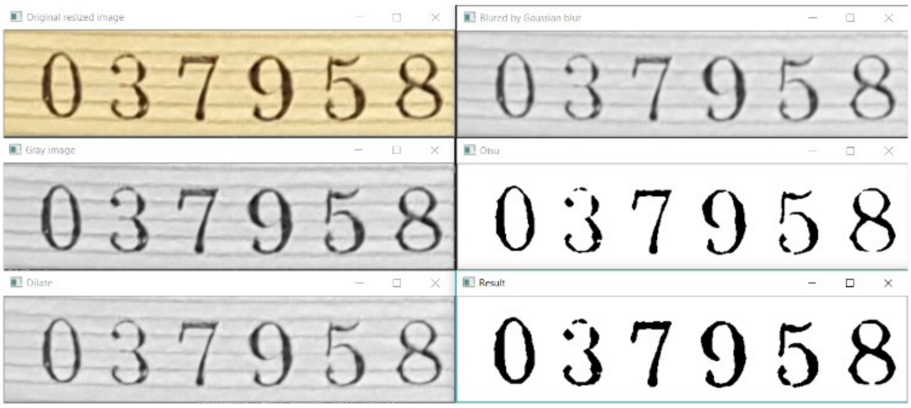


FIGURE 2.5: Numbers change through the preprocessing stage [6]

Cakic et al.'s research involved testing a script to read serial numbers from about 150 wine label images, some of which were of poor quality. The initial attempt, which didn't involve any image preprocessing, managed to correctly read the full serial numbers on around 62% of the images. After preprocessing the images, the success rate increased significantly to 87.5%. This suggests that the quality of the preprocessing and the images used in the test dataset significantly affects the accuracy of the results.

In 2019, Robby et al.'s paper "Implementation of Optical Character Recognition using Tesseract with the Javanese Script Target in Android Application" discusses the challenges and methods of recognising non-Latin scripts, especially Javanese, which have complex shapes and structures. They present their dataset collection, training, and testing process using Tesseract OCR tools. They report their results and analysis, showing that their model achieved the highest accuracy by combining single boundary box and separate boundary boxes for different parts of the characters.

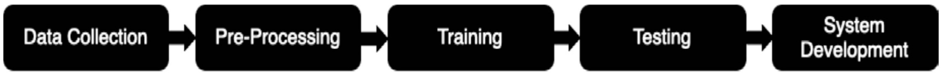


FIGURE 2.6: Robby et al.'s Proposed Method [7]

The authors of this paper used a variety of methods to develop a Javanese character recognition system. First, they collected 5880 Javanese characters from various sources and cropped and resized them to 32x32 pixels. Then, they applied several preprocessing steps to the images, such as binarization, noise removal, contrast enhancement, and skew correction. Next, they augmented the data by applying random rotations, translations, scaling, and shearing to the images. Finally, they used Tesseract, an open source OCR tool, to train different models with different boundary box methods.

The author's found that the best model was the one that combined single boundary box and separate boundary boxes for different parts of the characters, which achieved an accuracy of 97.50% and an error rate of 2.50% on the test set.

## 2.3 Convolutional Recurrent Neural Networks (CRNNs)

CRNN, an abbreviation for Convolutional Recurrent Neural Network, is a unique fusion of the advantages of convolutional neural networks (CNN) and recurrent neural networks (RNN), which are different kinds of neural network architectures.

CRNNs are usually applied in the classification and analysis of sequential data like text, speech, and images. Due to their ability to handle variable-length sequential data and recognize long-term dependencies, they are extremely useful for tasks that need to comprehend contextual and temporal information. They have displayed excellence in modeling and processing sequential data across diverse tasks, marking them as an efficient instrument in this domain.

### 2.3.1 Working of CRNNs

The functionality of CRNNs is outlined as follows:

1. **Input:** The primary input to a CRNN is a sequence of data, which could be images or audio samples.

2. **Convolutional Layers:** The incoming sequence is channeled through convolutional layers, akin to those in CNNs. These layers extract features from the input and are particularly efficient for image-based inputs.

3. **Recurrent Layers:** The output from a convolutional layer is then sent through one or more recurrent layers. These layers preserve a hidden state that memorizes information from previous entries in the sequence, making them ideal for sequential data processing.

4. **Bridge between Convolutional and Recurrent Layers:** Usually, the output from a convolutional layer is sampled before it is introduced to a recurrent layer. This strategy helps to minimize the network's computational complexity while maintaining the core characteristics of the input.

5. **Output:** Finally, the output from the last recurrent layer is processed through a fully connected layer. This final layer produces a prediction for the input sequence, which could be a string of characters, words, or any other output related to the task.

## 2.4 Other OCR Methods

### 2.4.1 Long Short-Term Memory Networks (LSTMs)

Long Short-Term Memory Networks (LSTMs) are a special kind of recurrent neural network capable of learning long-term dependencies, which makes them highly suitable for OCR tasks. They've been used successfully to decode sequences of characters from images.[8]

Breuel et al. in the paper "High-Performance OCR for Printed English and Fraktur using LSTM Networks" write about a novel application of bidirectional Long Short-Term Memory (LSTM) networks to the problem of machine-printed Latin and Fraktur recognition, without segmentation, language modelling or post-processing.

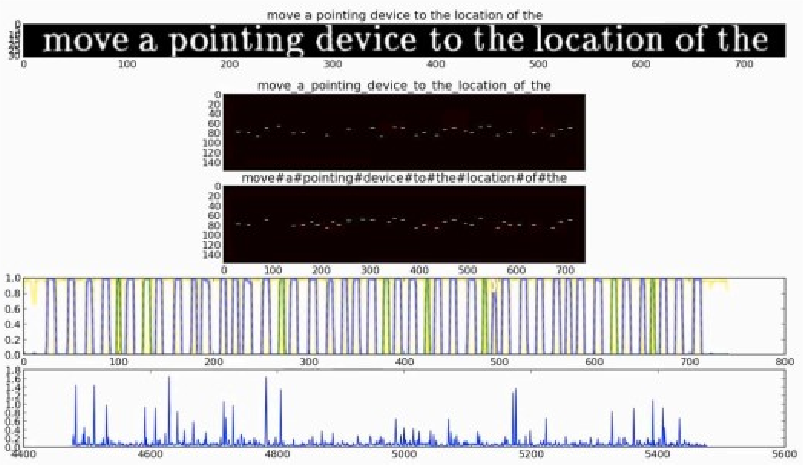


FIGURE 2.7: Greyscale image with background cleaning
[8]

A preprocessing step for text-line normalisation that uses a dictionary of connected component shapes and associated baseline and x-height information to map the input text lines to a fixed size output image.

A comparison of the LSTM-based system with other OCR systems on printed English and Fraktur texts, showing that LSTM achieves very low error rates and generalizes well to unseen data.

### 2.4.2 Transformers

Originally developed for natural language processing tasks, Transformer models have been adapted for OCR. They treat the OCR problem as a sequence-to-sequence translation task, translating the input image into a sequence of characters.

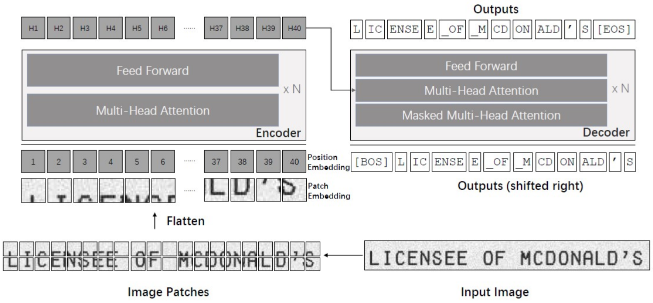


FIGURE 2.8: Li's Architecture of TrOCR
[2]

M.Li et al.'s "TrOCR: Transformer-Based Optical Character Recognition with Pre-trained Models" paper proposes an end-to-end text recognition approach with pre-trained image Transformer and text Transformer models, which leverages the Transformer architecture for both image understanding and wordpiece-level text generation. [2]

Transformer based OCR models have the advantage of being able to handle long sequences of text, which is useful for OCR tasks. However, they are computationally expensive and require large amounts of training data.

CRNNs are more suitable for this project because they are faster and require less training data and are better at handling spatial information

### 2.4.3 Attention-based OCR models

Attention mechanisms allow models to focus on different parts of the input image while predicting each character in the output sequence, similar to how humans read. This can improve accuracy, especially on more complex images.

Li et al.'s "Attention Based RNN Model for Document Image Quality Assessment" paper proposes a novel method for document image quality assessment (DIQA). The method integrates convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to capture spatial features and attention mechanisms. It also uses reinforcement learning to train a locator network that selects the optimal regions for quality evaluation.

The CNNs are used to extract spatial features from the document images. The RNNs are used to capture the temporal dependencies between the features. The locator network is used to select the optimal regions for quality evaluation. The regions are selected based on the attention mechanism, which identifies the most important regions in the document images. [9]

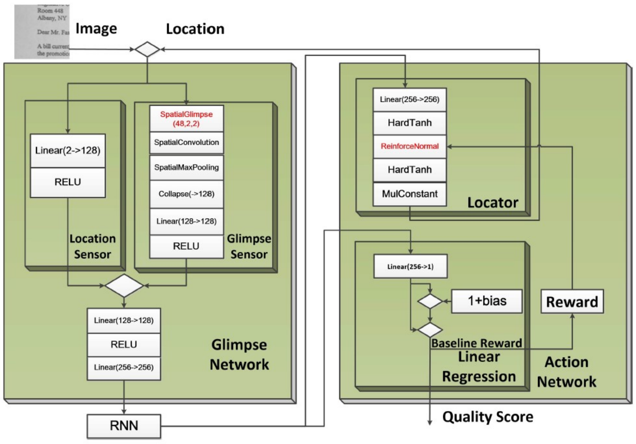


Figure 2.9: Architecture of Li's proposed network [9]

RNNs are good at handling sequential information but are poor at handling spatial information. CRNN's are more complex and combine the strengths of CNNs and RNNs which is more suitable for the this paper's OCR task.

### 2.4.4 Rule-based systems

These were some of the earliest methods for OCR and use specific rules for identifying characters based on their shape, size, and relative position. They are now less commonly used due to their limitations with complex and diverse inputs.

Doush et al.'s paper "A novel Arabic OCR post-processing using rule-based and word context techniques" developed a rule-based OCR system for Arabic text that uses a combination of horizontal and vertical projections to segment characters and then classifies them based on their shape and relative position. [10]

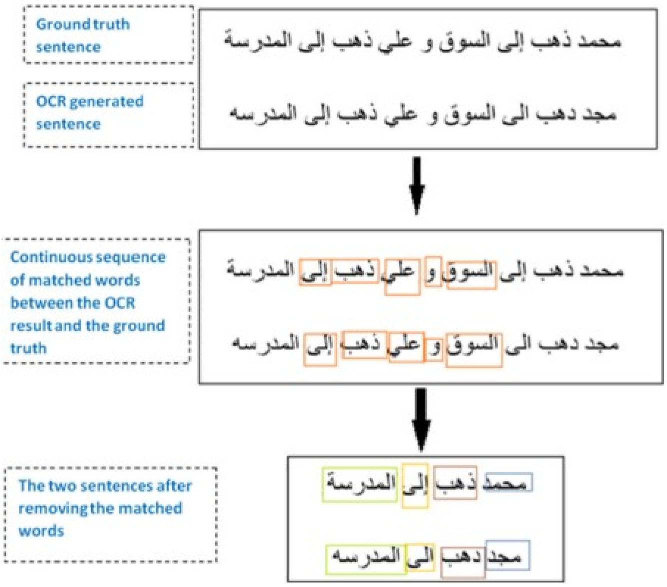

Ground truth sentence
محمد ذهب إلى السوق و علي ذهب إلى المدرسة
OCR generated sentence
مجد دهب الى السوق و علي ذهب إلى المدرسه
Continuous sequence of matched words between the OCR result and the ground truth
محمد ذهب إلى السوق و علي ذهب إلى المدرسة
مجد دهب الى السوق و علي ذهب إلى المدرسه
The two sentences after removing the matched words
محمد ذهب إلى المدرسة
مجد دهب الى المدرسه

Figure 2.10: Example of applying the Rule Based FAHTA Algorithm [10]

The FAHTA algorithm is a novel alignment technique that is used to match the ground truth text with the OCR misrecognized text. The paper also says that the FAHTA algorithm is fast, accurate, and can handle different types of OCR errors, such as over-segmentation, under-segmentation, and merging words. The paper claims that the FAHTA algorithm can be used for other languages as well.

For the purposes of this project, the rule-based system is not suitable because it requires a large number of rules to be defined for each character, which is not feasible for the large range iof digit fonts.

### 2.4.5 Support Vector Machines (SVMs)

Support Vector Machines (SVMs) are used for character recognition in OCR due to their effective high-dimensional mapping and classification abilities. They work best when text is clearly segmented. In the their paper ”Development of an Image Processing Techniques for Vehicle Classification Using OCR and SVM”, Joshua et al. used SVMs to classify characters in a license plate image and achieved an accuracy of 98.3% using a local dataset of 10,000 images.[11]

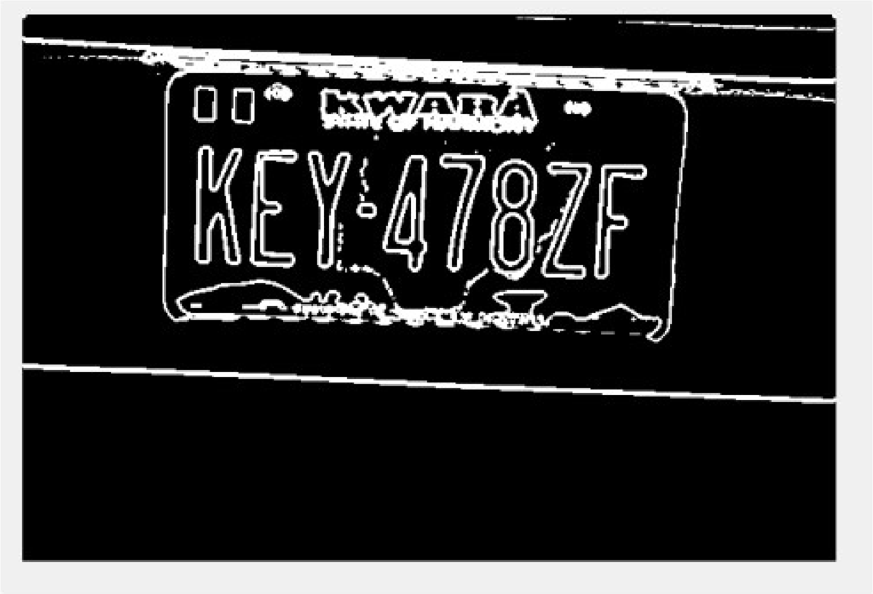


FIGURE 2.11: Greyscale image with background cleaning
[11]

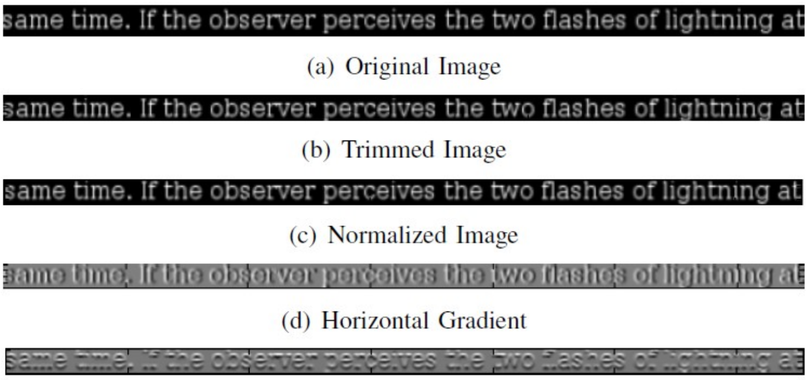
Joshua et al. describe the steps of their proposed system, which include image preprocessing, feature extraction, OCR, and SVM classification. They also explain how they collected and labeled their dataset of Nigerian vehicle plate numbers.

### 2.4.6 Hidden Markov Models (HMMs)

HMMs have been used in OCR for recognizing sequential data. HMMs are statistical models that assume an underlying process to be a Markov process with hidden states.

In Rashid et al.'s "An evaluation of HMM-based Techniques for the Recognition of Screen Rendered Text" paper, they evaluate Hidden Markov Model (HMM) techniques for optical character recognition (OCR) of low resolution text from screen images and compares them with other OCR systems.

The paper uses two data sets of screen rendered characters and text-lines, and extracts two types of features from them: gray scale raw pixel features and gradient based gray level intensity features.


same time. If the observer perceives the two flashes of lightning at
same time. If the observer perceives the two flashes of lightning at
same time. If the observer perceives the two flashes of lightning at

(a) Original Image

(b) Trimmed Image

(c) Normalized Image

(d) Horizontal Gradient

(e) Vertical Gradient

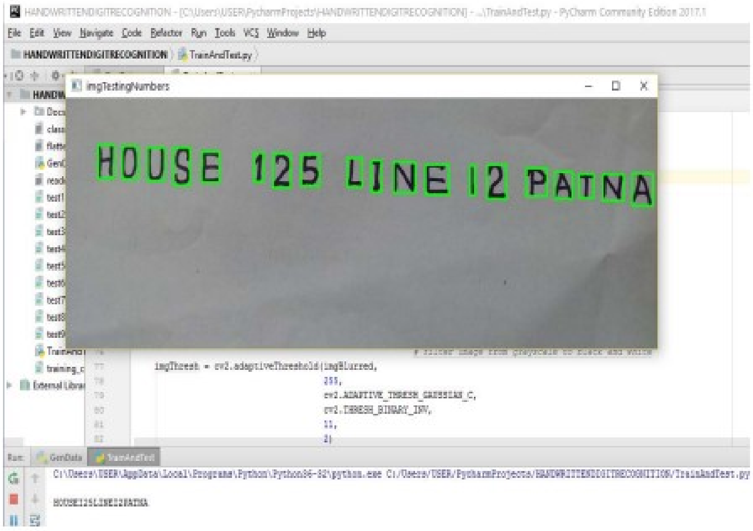
FIGURE 2.12: Rashid's Extraction steps from screen rendered text-lines [12]

The paper reports the character recognition accuracy of the HMM-based methods and other OCR engines on the two data sets. It shows that the HMM-based methods reach the performance of other methods on screen rendered text and achieve above 98% accuracy.[12]

HMMs are a good choice for tasks where simplicity and interpretability are important. CRNNs are a good choice for tasks where accuracy is more important, and where the sequences are long or complex.

### 2.4.7 K-Nearest Neighbors (KNN)

KNN is a simple, instance-based learning algorithm used for OCR, particularly for isolated character recognition. Hazra et al. develop an optical character recognition (OCR) system that uses a custom image to train a k-nearest neighbor (KNN) classifier. They claim that their system can recognize handwritten or printed text in any language by changing the training image and labels. [13]


HANDWRITTENDIGITRECOGNITION - [C:\Users\USER\PycharmProjects\HANDWRITTENDIGITRECOGNITION] - ...\TrainAndTest.py - PyCharm Community Edition 2017.1
File Edit View Navigate Code Refactor Run Tools VCS Window Help
HANDWRITTENDIGITRECOGNITION
TrainAndTest.py
imgTestingNumbers
HOUSE 125 LINE 12 PATNA
Run:
GenData
TrainAndTest

```
                                                # filter image from grayscale to black and white
imgThresh = cv2.adaptiveThreshold(imgBlurred,
                                  255,
                                  cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                                  cv2.THRESH_BINARY_INV,
                                  11,
                                  2)
```

```
C:\Users\USER\AppData\Local\Programs\Python\Python36-32\python.exe C:/Users/USER/PycharmProjects/HANDWRITTENDIGITRECOGNITION/TrainAndTest.py

HOUSE125LINE12PATNA
```

Figure 2.13: Characters and Digits regocnised [11]

Hazra et al. explain the steps of their algorithm, which include image processing, feature extraction, and KNN classification. They also discuss the advantages of KNN over other classification methods, such as ease of interpretation, low computation time, and high predictive power. In this paper the authors started with clear images of known fonts, which is not the case in this project.

### 2.4.8 Template Matching

Template Matching is a technique used to locate small-parts of the bigger image which match a template image. This can be useful in OCR when the set of possible characters is known and limited. In Hossain et al.'s "Optical Character Recognition based on Template Matching" paper, they use template matching to recognize characters in a license plate image.

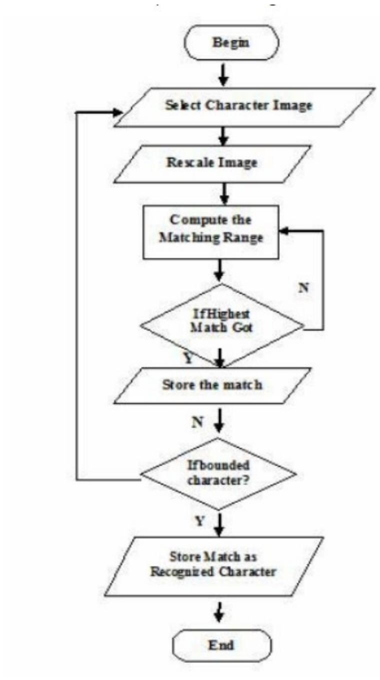


Figure 2.14: Flowchart of Hossain's TM OCR [14]

Their system prototype was tested on different text images with different fonts and sizes. The accuracy was calculated based on the character recognition accuracy. Their results show that Calibri and Verdana fonts had the highest accuracy, while Cambria and Times New Roman fonts had the lowest accuracy. The accuracy can be improved by training the system with more fonts and features. [14]

## 2.5 Conclusion

Where has tesseract and CRNN been found to be suitable for OCR tasks?

# Chapter 3

# Methodology

## 3.1

# Chapter 4

# Results

## 4.1 Introduction

# Chapter 5

# Discussion and Conclusion

## 5.1 Discussion

# Bibliography

[1] R. Smith, "An Overview of the Tesseract OCR Engine," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2*. IEEE, pp. 629–633. [Online]. Available: http://ieeexplore.ieee.org/document/4376991/

[2] M. Li, T. Lv, J. Chen, L. Cui, Y. Lu, D. Florencio, C. Zhang, Z. Li, and F. Wei, "TrOCR: Transformer-Based Optical Character Recognition with Pre-trained Models," vol. 37, no. 11, pp. 13 094–13 102. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/26538

[3] E. Andersson, "Benchmarking Object Detection Algorithms for Optical Character Recognition of Odometer Mileage." [Online]. Available: https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1688125&dswid=462

[4] A. Ahuja and A. Chaudhuri, "Detecting Vehicle Type and License Plate Number of different Vehicles on Images."

[5] L. Giridhar, A. Dharani, and V. Guruviah, "A Novel Approach to OCR using Image Recognition based Classification for Ancient Tamil Inscriptions in Temples," p. 8. [Online]. Available: https://arxiv.org/abs/1907.04917

[6] S. Cakic, T. Popovic, S. Sandi, S. Krco, and A. Gazivoda, "The Use of Tesseract OCR Number Recognition for Food Tracking and Tracing," in *2020 24th International Conference on Information Technology (IT)*. IEEE, pp. 1–4. [Online]. Available: https://ieeexplore.ieee.org/document/9070558/

[7] G. A. Robby, A. Tandra, I. Susanto, J. Harefa, and A. Chowanda, "Implementation of Optical Character Recognition using Tesseract with the Javanese Script Target in Android Application," vol. 157, pp. 499–505. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877050919311640

[8] T. M. Breuel, A. Ul-Hasan, M. A. Al-Azawi, and F. Shafait, "High-Performance OCR for Printed English and Fraktur Using LSTM Networks," in *2013 12th International Conference on Document Analysis and Recognition*. IEEE, pp. 683–687. [Online]. Available: http://ieeexplore.ieee.org/document/6628705/

[9] P. Li, L. Peng, J. Cai, X. Ding, and S. Ge, "Attention Based RNN Model for Document Image Quality Assessment," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, pp. 819–825. [Online]. Available: http://ieeexplore.ieee.org/document/8270070/

[10] I. A. Doush, F. Alkhateeb, and A. H. Gharaibeh, "A novel Arabic OCR post-processing using rule-based and word context techniques," vol. 21, no. 1-2, pp. 77–89. [Online]. Available: http://link.springer.com/10.1007/s10032-018-0297-y

[11] I. O. Joshua, M. O. Arowolo, M. O. Adebiyi, O. R. Oluwaseun, and K. A. Gbolagade, "Development of an Image Processing Techniques for Vehicle Classification Using OCR and SVM," in *2023 International Conference on Science, Engineering and Business for Sustainable Development Goals (SEB-SDG)*. IEEE, pp. 1–9. [Online]. Available: https://ieeexplore.ieee.org/document/10124622/

[12] S. F. Rashid, F. Shafait, and T. M. Breuel, "An Evaluation of HMM-Based Techniques for the Recognition of Screen Rendered Text," in *2011 International Conference on Document Analysis and Recognition.* IEEE, pp. 1260–1264. [Online]. Available: http://ieeexplore.ieee.org/document/6065512/

[13] T. K. Hazra, D. P. Singh, and N. Daga, "Optical character recognition using KNN on custom image dataset," in *2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON).* IEEE, pp. 110–114. [Online]. Available: http://ieeexplore.ieee.org/document/8079572/

[14] M. A. Hossain and S. Afrin, "Optical Character Recognition based on Template Matching," pp. 31–35. [Online]. Available: https://globaljournals.org/GJCST_Volume19/4-Optical-Character-Recognition.pdf

# Appendix A

# Code Snippets

Put appendix material in this section e.g. code snippets

USE THE APPENDICES