MINI PROJECT REPORT

OCR APP ON iOS USING CoreML

by

NIKHIL NAYYER (17BCS035)

VIBHANSH GUPTA (17BCS059)

Under the guidance of Prof. Sanjay Sharma

Submitted in partial fulfilment of the requirements for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

SHRI MATA VAISHNO DEVI UNIVERSITY, KATRA

(School of Computer Science & Engineering)

JAMMU & KASHMIR – 182 320

Session 2020-21

CERTIFICATE

SHRI MATA VAISHNO DEVI UNIVERSITY

It is hereby certified with confirmation that this project report on 'OCR App on iOS using CoreML' is the work of Nikhil Nayyer and Vibhansh Gupta who completed the work under my mentorship.

Sanjay Kumar Sharma
Assistant Professor
Department of CSE

Submitted for Viva on : 18th December 2020

Examiner : _____

ACKNOWLEDGEMENT

It is our pleasure to acknowledge a deep sense of gratitude to Prof. Sanjay Sharma, our project guide, whose guidance, encouragement and insightful conversation have helped us to finish our project.
In order to solve all the difficulties in the execution of this project, his advice proved to be the most helpful.

We are grateful to Prof. Anuj Mahajan for his direct and indirect assistance in completing this project.

Finally, without making our sincere appreciation to all those who have supported us in the completion of this mission, this acknowledgment would be incomplete.

ABSTRACT

Optical Character Recognition, usually referred to as OCR, is the way to convert the image created by scanning a text or a document into a machine-editable format. A computer system that is fitted with such an OCR system would improve the speed of input operation and reduce any possible human errors.

As the modification of the same character occurs due to the change of fonts or the introduction of different types of noise, the recognition of written characters is itself a hard issue.

The key concept is to make use of common libraries to construct an ML model that can be used in the mobile application to deal with the differential problems in font and sizes, because if pre-processing, feature extraction and recognition are not robust, it makes the recognition task difficult.

Once the model is ready for deployment, integrate it inside the application container with the help of popular libraries, high level APIs and frameworks.

Various Natural Language Processing (NLP) libraries and Machine Learning APIs have been used researched with service requests from TensorFlow and implementation of other such Deep Learning solutions.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

OCR: Optical Character Recognition

ANN: Artificial Neural Networks

R&D: RESEARCH AND DEVELOPMENT

ML: Machine Learning

DFD: Data Flow Diagram

CHAPTER 1: INTRODUCTION

In the running world, as information is scanned through paper records, there is an increasing demand for software systems to identify characters in the computer system as we understand that we have a range of newspapers and books related to various subjects in printed format. There is a massive demand these days to "store the information accessible in these paper documents on a computer storage disc and then reuse this information by searching process later." One convenient way to store information in these systems from these paper documents is to scan the documents first and then store them as images.

However, the contents of these documents are very difficult to read and to scan line-by-line and word-by-word for reuse. This is because the characteristics of real-world documents differ from the character characteristics of the computer documents.

There are different techniques in the world for this method. We have selected Optical Character Recognition as the key fundamental technique for identifying characters from all these techniques. In many organisations, especially in the field of research and development (R&D), in large business enterprises, in government agencies etc. the conversion of paper documents into electronic formats is an ongoing activity.

A simplified solution to capture notes via optical character recognition using artificial neural networks has been identified. Artificial neural networks have some benefits in the back-propagation network and classification, considering the computational difficulty involved, in the sense of emulating adaptive human intelligence to a small degree.

1.1 PURPOSE

The main purpose of the grid-based OCR system is to perform document image analyses and to process electronically transformed document formats from paper formats more easily and efficiently.

This improves the accuracy of character detection during document processing in contrast with various existing character recognition systems. The OCR approach derives from its bit-mapped images the context and font properties of the characters.

The key objective is to speed up the process of character recognition in the document's development. As a result, a large number of documents can be handled in less time and thereby saves time in the sorting process.

1.2 PROBLEM STATEMENT

There is increasing demand in the world of running that users convert the printed documents to electronic papers to safeguard their data. Thus, the fundamental OCR method was developed to convert the data on papers to computer-compatible documents, allowing editing and re-usability of documents. The previous OCR framework on a grid infrastructure is just OCR without grid features.

This was tried out using manual programming techniques but could not deliver the expected results either due to a high computational requirement or due to the complexity of the requirements.

1.3 GOAL AND VISION

1. An Artificial Neural Networks (ANNs) based Optical Character recognition system. Using the Back Propagation algorithm, the ANN will be trained. The binary numbers used as an input for a simple feature extracting method, the output of which is supplied to an ANN in addition to the input, are provided from each typed English letter of the system. The Feed Forward algorithm then offers insight into the entry of a neural network, supplemented by the Back Propagation algorithm that includes planning, error evaluation and mass adjustment.

2. Then, we wanted to build a mobile application that could carry the task of capturing an image and storing it to be passed on to the ML model for OCR analysis. We wanted to have a minimalistic app that would be very user-friendly but still could have all the functionalities we needed it to perform.

3. Using the trained Neural Network inside a mobile application. While various number of methods have been made available to deploy local copies of the models of ANNs with saved weights. But, since we wanted to deploy a solution to Apple's, iOS mobile operating system, we wanted to use their VisionKit Library and convert our ML model into the format suitable for deployment. This is where CoreML comes in. It will help us integrate our ML model in the Xcode iOS app we've prepared. After that, it all comes down to improving the User Experience.

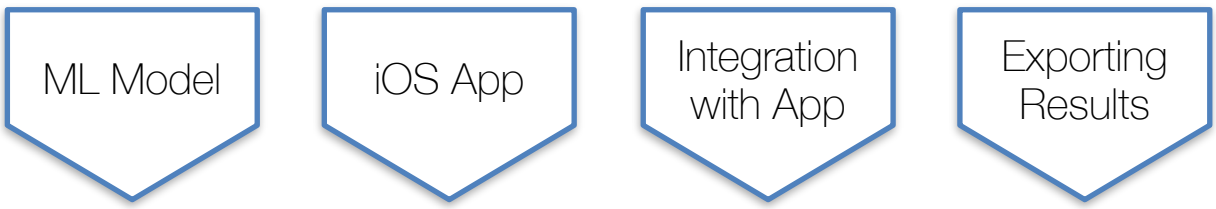

Figure 1.1: Task Division and step by step completion

CHAPTER 2: REQUIREMENTS SPECIFICATION

There is a massive demand these days to store the information available in physical data around us on a computer so that it can be edited or that this information can later be reused by looking for its data entry.

DATA FLOW DIAGRAM

A data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system.

For the visualisation of data analysis, DFDs may also be used. The chart for our system in the form of a data flow diagram is represented here :

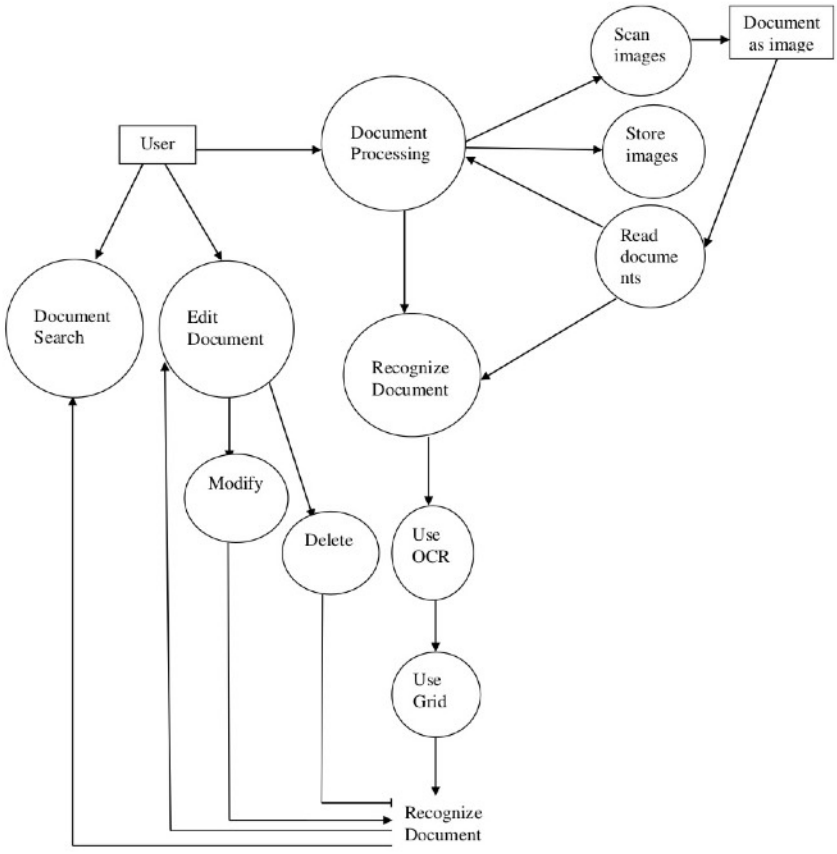

Figure 2.1: DFD model

The problem is that when information is scanned into digital copies, the software systems will identify characters in the computer system, since we understand that we have a variety of journals and books in written format, which are connected to various subjects. The documents are stored in the computer system as images such as jpeg, gif, etc., whenever we scan the documents through the scanner. The user can not read or edit these images. The individual contents of these documents are difficult to read and search for reuse.

In order to understand unstructured linguistic data, NLP uses algorithms to define and interpret natural language laws. To make sense of the data, computers use computer programmes such as Java and C++. Human beings speak English, Spanish, mandarin, of course, and many other natural human languages.
This computer-many speech distance is being bridged by NLP. Regrettably, language cannot be carefully taught for computers in Excel tablets, so NLP relies on algorithms to improve comprehension.

People have tried to solve the OCR problem with many traditional computer vision techniques such as image filters, contour detections and imaging classifications that have worked well on small template-based datasets that don't differ greatly in their direction, image quality, etc.

Deep learning methods have improved in recent years to reawaken interest in the OCR problem which allows neuronal networks to combine text-finding tasks within an image with text-understanding. The use of deep-rooted neural networks and attention mechanisms and repetitive networks has gone a long way in that regard.

Once the image is given to the computer, the computer will use the algorithms to extract text associated with each word. Then it collects data from them. Using this data, the computer will determine an accurate response - or at least, hazard a guess as to what the complete word would be.

2.1 USE CASES

What we have to do is use a Machine Learning Model to create predictions about what text is available in the document when there is one. It uses deep learning and attention mechanism methods.

A few of the planned use cases include :

1. "Transcribe Text from Images"
   Identify and extract text within an image.
2. "Transcribe Invoices"
   Save time and money by storing important information from invoices much more efficiently and accurately.
3. "Transcribe Receipts"
   Extract key data from within your travel receipts and watch them automatically organise into sharable information snippets.

USE CASE DIAGRAMS

In each of the below-explained use cases, the functionality is described regarding the following points :

- Use Case Name
- Details about the use case
- Actors using this use case
- The flow of events that occur in this use case
- The conditions that occur in this use case

There are various use case diagrams that altogether finish the complete picture :

1. **Training the Machine Learning Model**

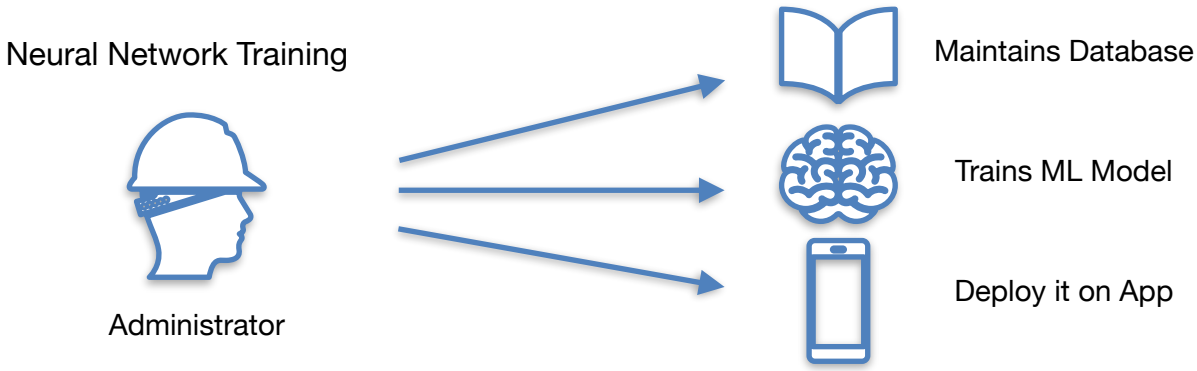

Figure 2.2 : Use-Case Diagram for Neural Network Training

The administrator or developer maintains the database, completes the pre-processing and trains it for further deployment.

2. **Document Processing**

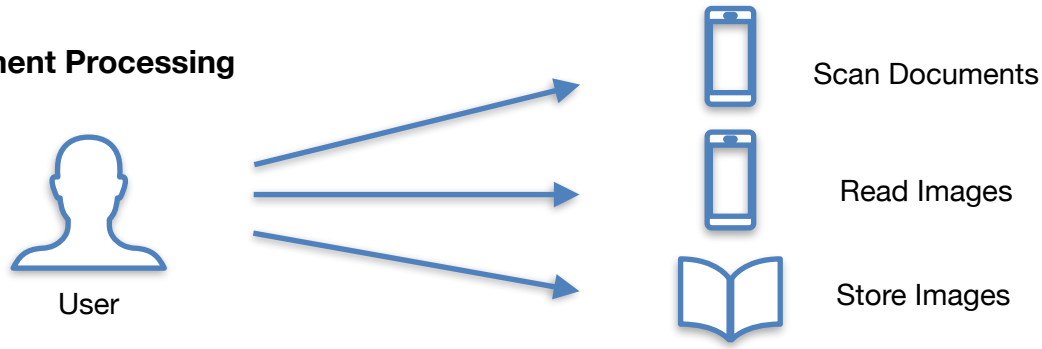

Figure 2.3 : Use-Case Diagram for Document Processing

The documents are read as photos when the user scans the documents. In the device memory, the read images are then saved and transferred to the model to generate the results.

3. Document Recognition

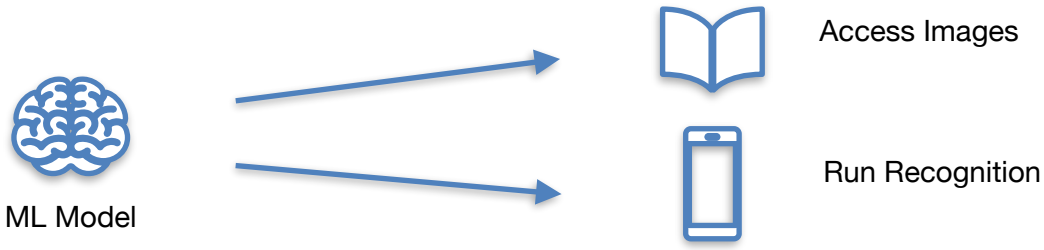

Figure 2.4: Use-Case Diagram for Optical Character Recognition

The ML Model is the only entity who participates in the document recognition. The images stored by the user are accessed and are run through the model's training to output the results of the OCR.

4. Document Sharing

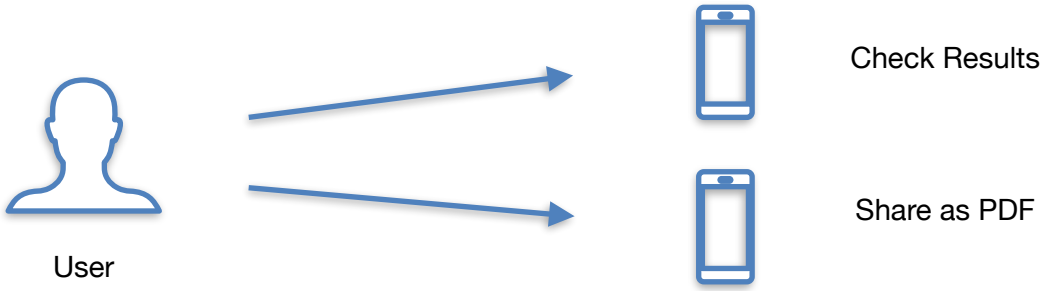

Figure 2.5: Use-Case Diagram for Document Exporting

User has the option to see the select text that is recognised from the image and perform operations on it like copy, search, call etc. The user can also share the result as a pdf and export it to any app he/she wishes.

2.2 DEPENDENCIES

Dependency of a project is a rational, constraint-based or preferential relationship between two activities or projects, so that the completion or initiation of one depends on the completion or initiation of the other.

The relationship of cause and effect between dependencies and constraints. In its simplest form, a limitation is a constraint that must be done or performed within its limits. Lack of resources such as money and human capital, lack of time and even a lack of experience can lead to restrictions.

Below are some of the software dependencies explained in detail regarding what they do and why are they absolutely essential in the scope of this project.

2.2.1 Xcode

Xcode is Apple's integrated development environment to Build, Test, and Submit applications for Apple's Operating Systems.

Xcode consists of a suite of software used by developers to construct applications for Apple platforms. From designing your software to checking, optimising, and uploading it to the App Store, use Xcode to handle your entire development workflow.

The main highlighted features of Xcode are :

- If a real device is not available, use the Simulator to easily prototype and test the app in a simulated environment. The Simulator offers various configurations, folders, and versions of the operating system for the iPhone, iPad, Apple Watch, and Apple TV users.
- To profile and evaluate your app, boost performance, and find memory issues, use instruments. Instruments gather information and present the findings using various methods called instruments.
- To build and train customised machine learning models for your app, make use of Create ML and all of it’s powerful Machine Learning tools.
- Use the Reality Composer to create 3D compositions and experiences with augmented reality (AR).

2.2.2 SwiftUI

SwiftUI is a creative, incredibly easy way of designing user interfaces with the power of Swift across all Apple platforms. Create user interfaces using only one set of software and APIs for any Apple computer.
SwiftUI fits well with modern Xcode design tools, featuring an easy to read and natural to write declarative Swift syntax that retains code and the project draft perfectly synchronised.

Anything you edit is absolutely in line with the code in the adjacent editor while you work on the design canvas. When you type, the code is instantly available as a sample, and any changes you make to that preview appear immediately in your code. Xcode immediately recompiles your modifications and inserts them, accessible and editable at all times, into a running version of your software.

2.2.3 Core ML

The Core ML mobile machine learning system of Apple is the user-friendly face of one of the latest industries in recent years to draw tech headlines: machine learning on smartphones and mobile devices with a limited form factor.

Figure 2.6: Core ML Functionality

At present, it is a field torn between many business and logistical imperatives and constraints, but promising to turn AI development from an abstract and simplified user interface into a profoundly personal and local' one without the cloud-based data collection's adverse privacy implications.

2.2.4 MODEL BUILDING VIA TENSORFLOW

TensorFlow is an open-source machine learning framework. It has a robust, scalable ecosystem of tools, libraries and community resources that enables researchers to quickly create and deploy ML driven applications to push the state-of-the-art in ML and developers using TensorFlow.

- Easy model building

  Build and train ML models easily using intuitive APIs like Keras that allow immediate model iteration and easy debugging.

- Robust ML production anywhere

  Easily train and deploy cloud, on-prem, browser, or on-device templates regardless of the language you are using.

- Powerful experimentation for research

  To take new concepts from abstract to code, to modern models, all with a simple and scalable architecture.

Figure 2.7: The TensorFlow Logo

CHAPTER 3: DESIGN

This chapter will deal with the processes involved behind the making of the application. The whole system was used to run OCR on captured images by running them through the Core ML model, which can be fetched by a simple API call.
The sections will be divided according to the tasks they pertain to and in the order they've been shown in the figures in Chapter 2.
Tasks may be related to one or more other sections and that has been specified everywhere applicable.

3.1 FUNCTION ORIENTED DESIGN

3.1.1 Procedural Approach Modelling

Declarative models characterise inter-entity relationships. They are very common, but can lead to reasoning algorithms that are complex and inefficient. When information includes a strong element of directionality, procedural models are more suitable. They tend to be highly trained in a particular task or situation and can generate outputs for inputs effectively.

During design, the purpose is to prepare for the system a gross, conceptual blueprint or schematic. Therefore one job is to modularise the system and demarcate the major components of the processing and data repository. Defining the interfaces between the components is another job. Describing the relationships of the components is the final mission.
Therefore during design, the goal is to turn the overall requirements of the system into precise, detailed requirements for the components of the system.

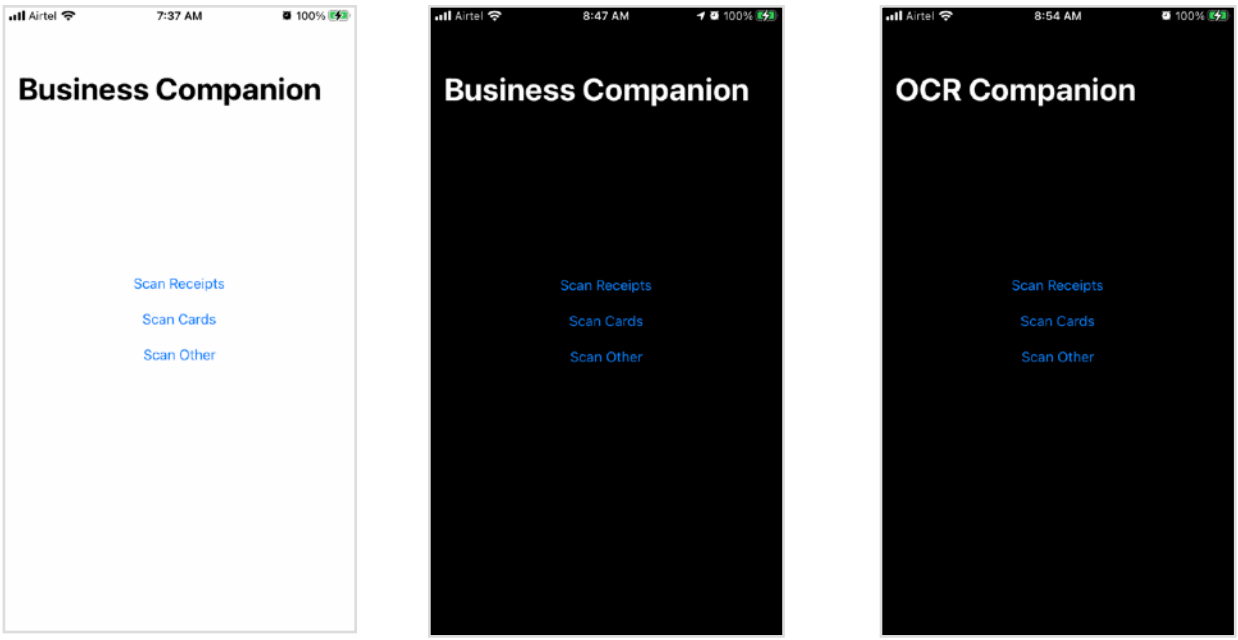

Figure 3.1: Initial Draft around Mid Semester Evaluation

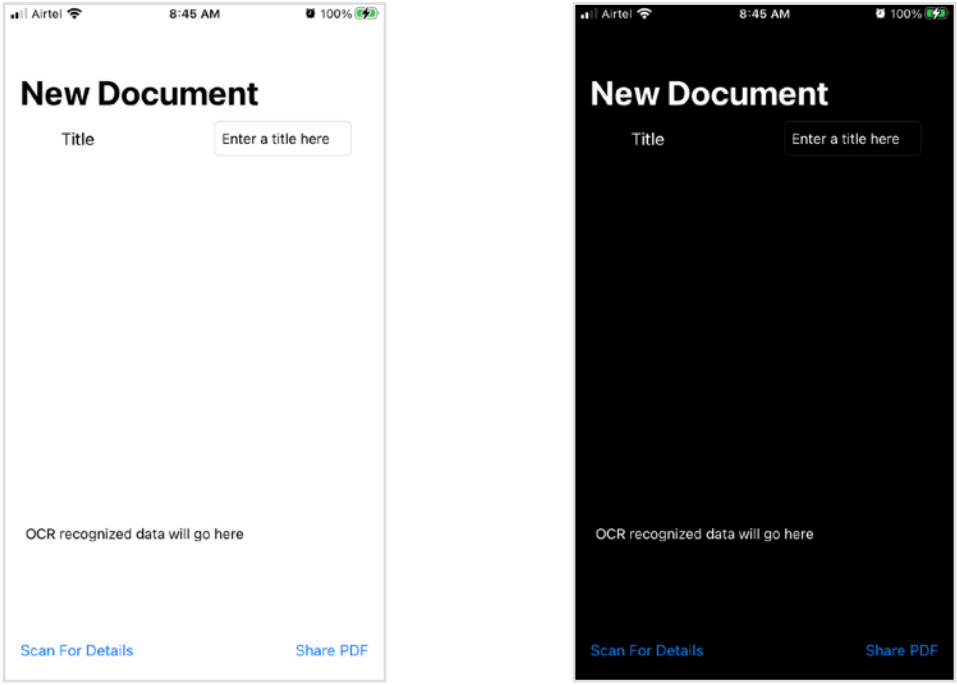

Figure 3.2: Final Draft during submission

3.1.2 Structure Charts

In software engineering, a Structure Chart is a chart that displays a system's breakdown to its lowest manageable parts. In order to group programme modules into a structure.

Each module is represented by a box containing the name of that module. The layout of the structure visualises the relationships between modules, showing data transfer with arrows between modules.

The structure displays the relationship between modules, illustrating the transfer of data between models.

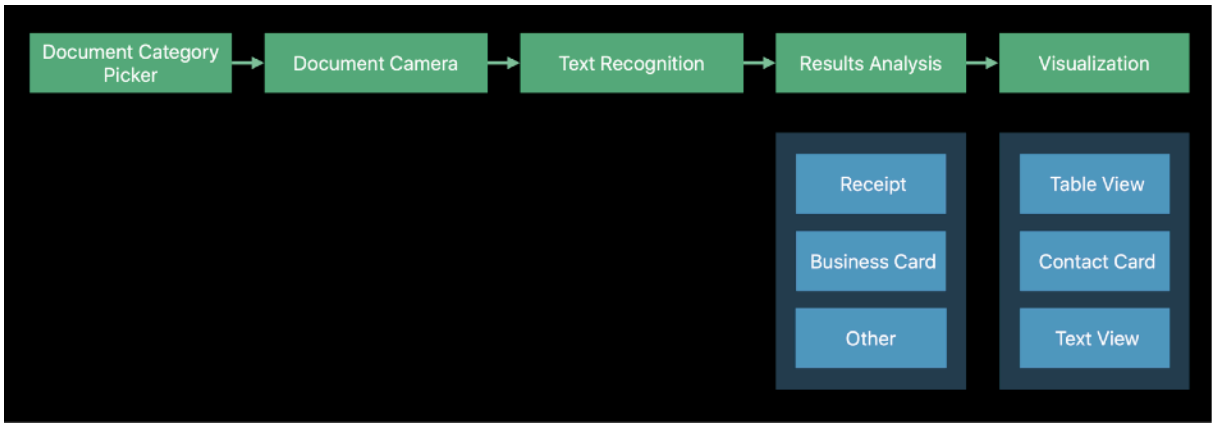

Figure 3.3: Structure Chart for Initial Draft

3.2 GUI DESIGN

3.2.1 Interface Design

The design principles of the User Interface (UI) are the methods/process for designing the software application's front end view of which the client or user can interact/use the software application easily without any hazards. The software application would be most helpful if the UI view of the given software application is appealing, easy to use, quick, understandable, sensitive.

The final draft for the UI Design was :

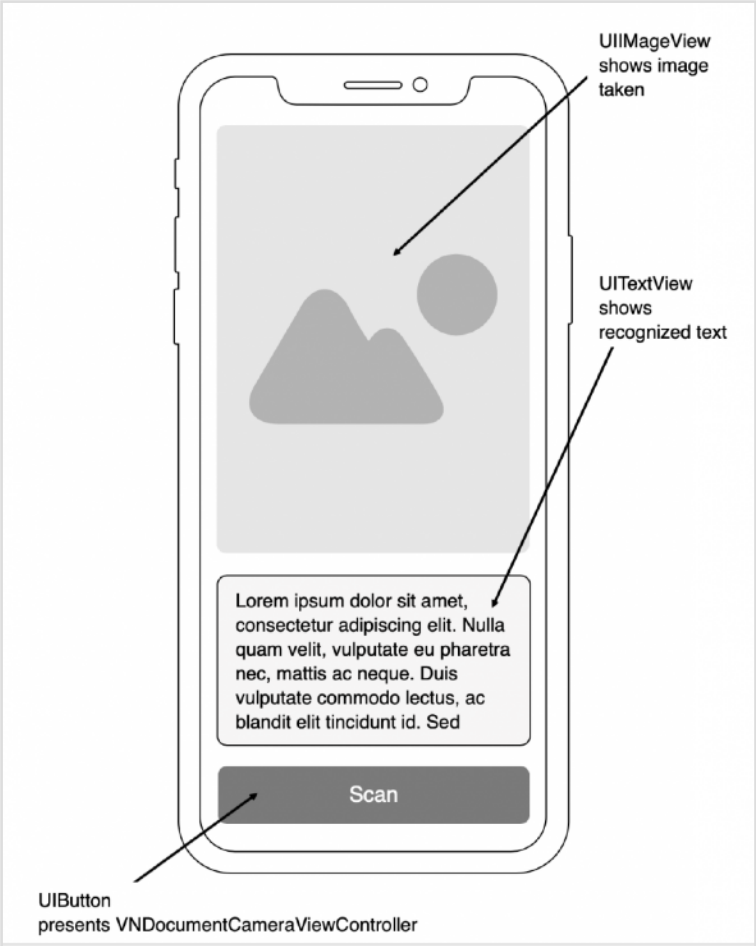

Figure 3.4: UI Design Idea

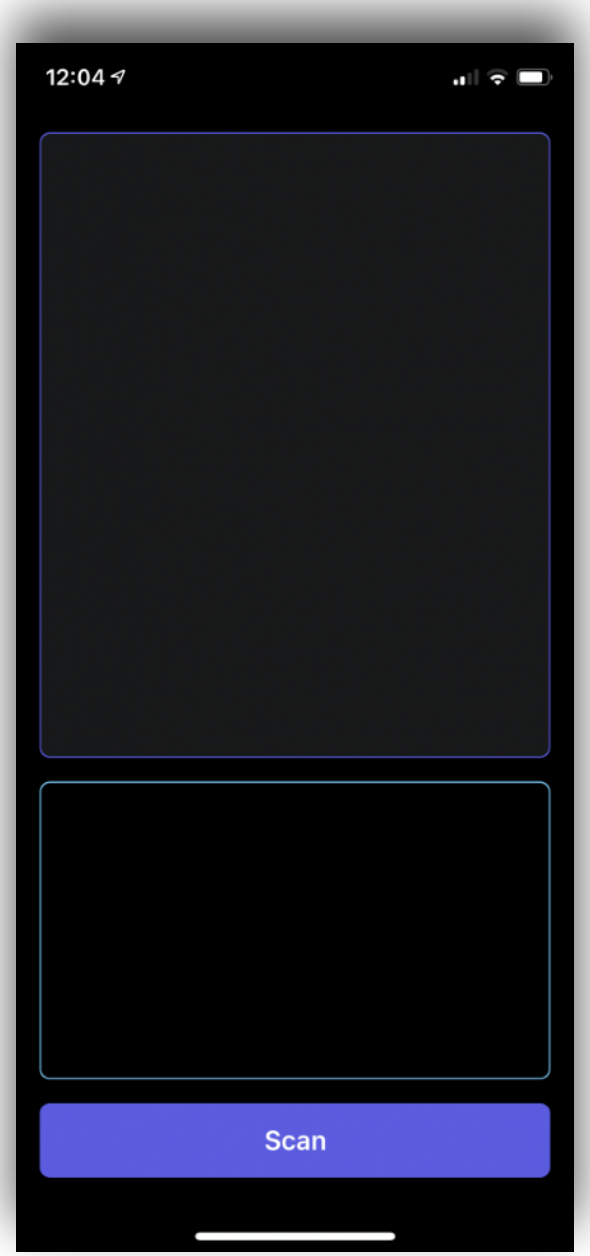

Figure 3.5: App Interface for Final Draft submission

3.2.2 App Screenshots

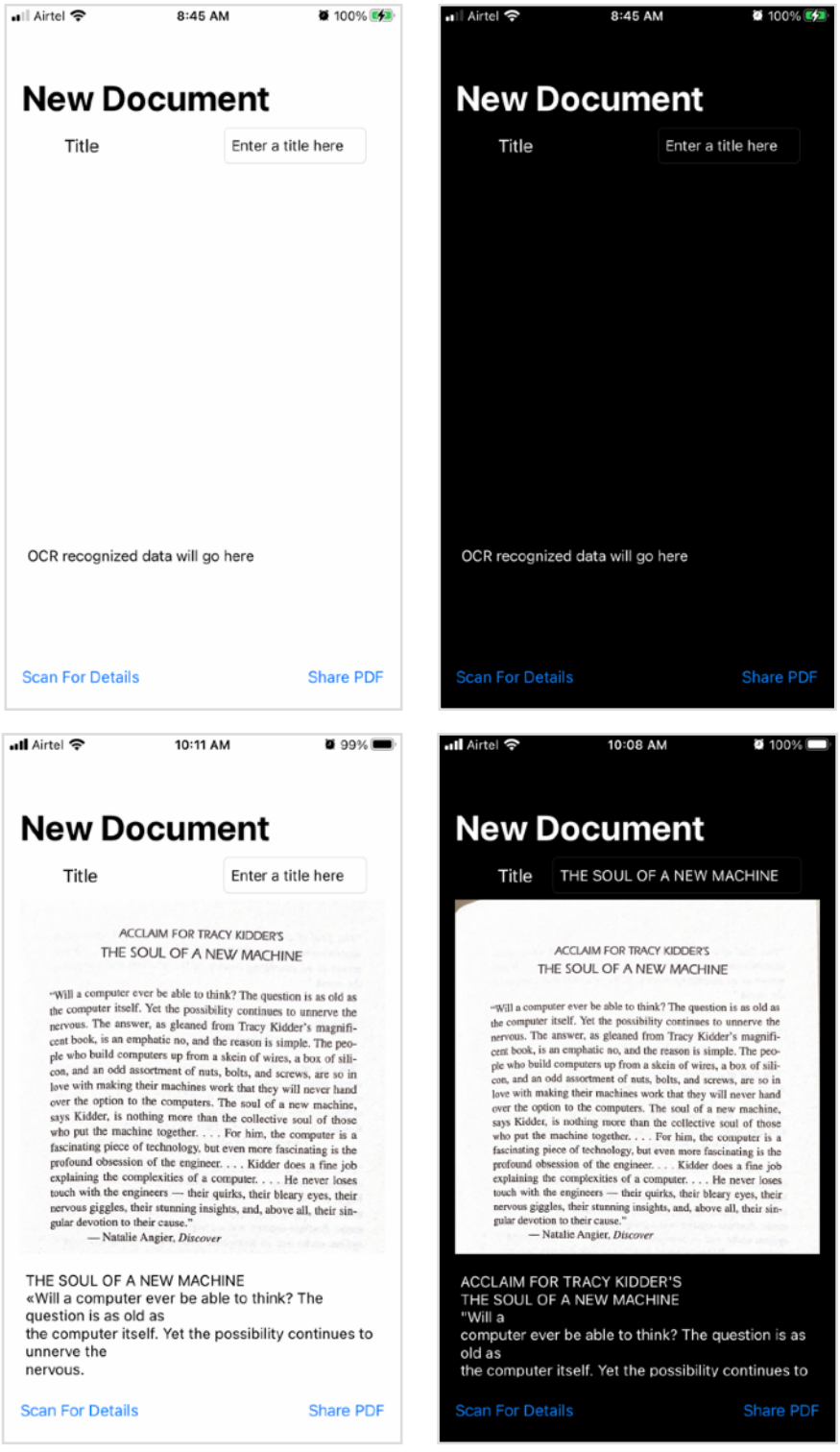

Figure 3.6: Main Screen (before scan vs after scan)

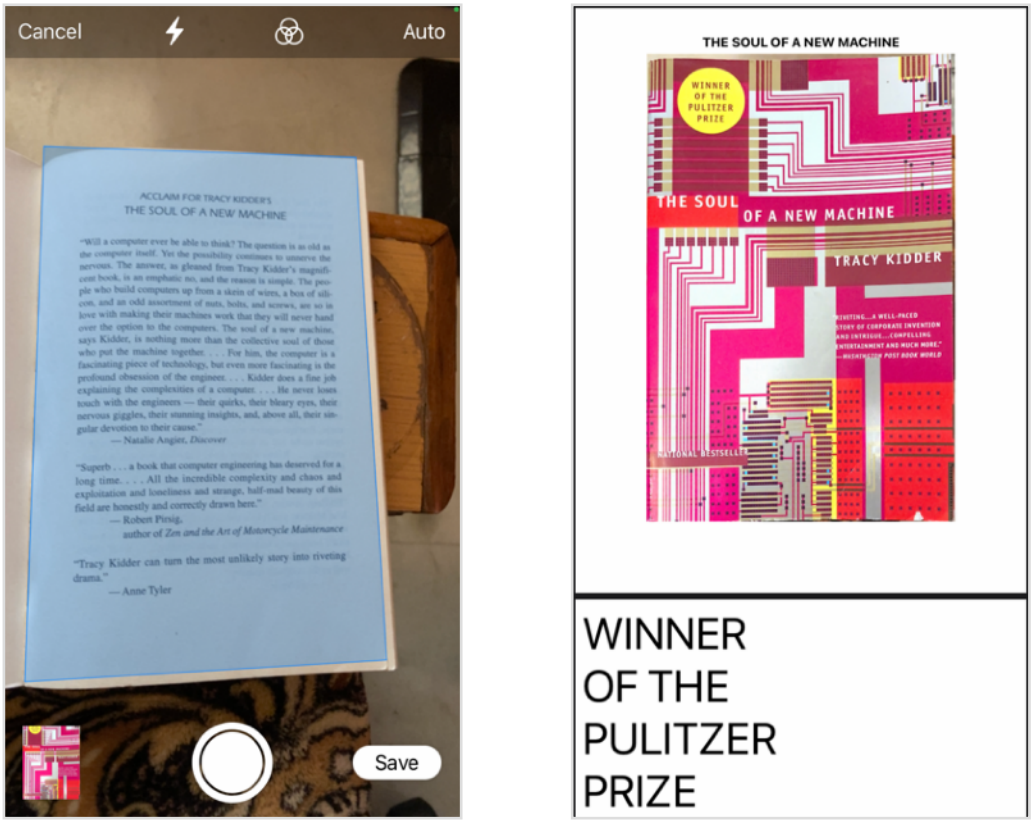

Figure 3.7: During Scan vs Exported Scan (PDF)

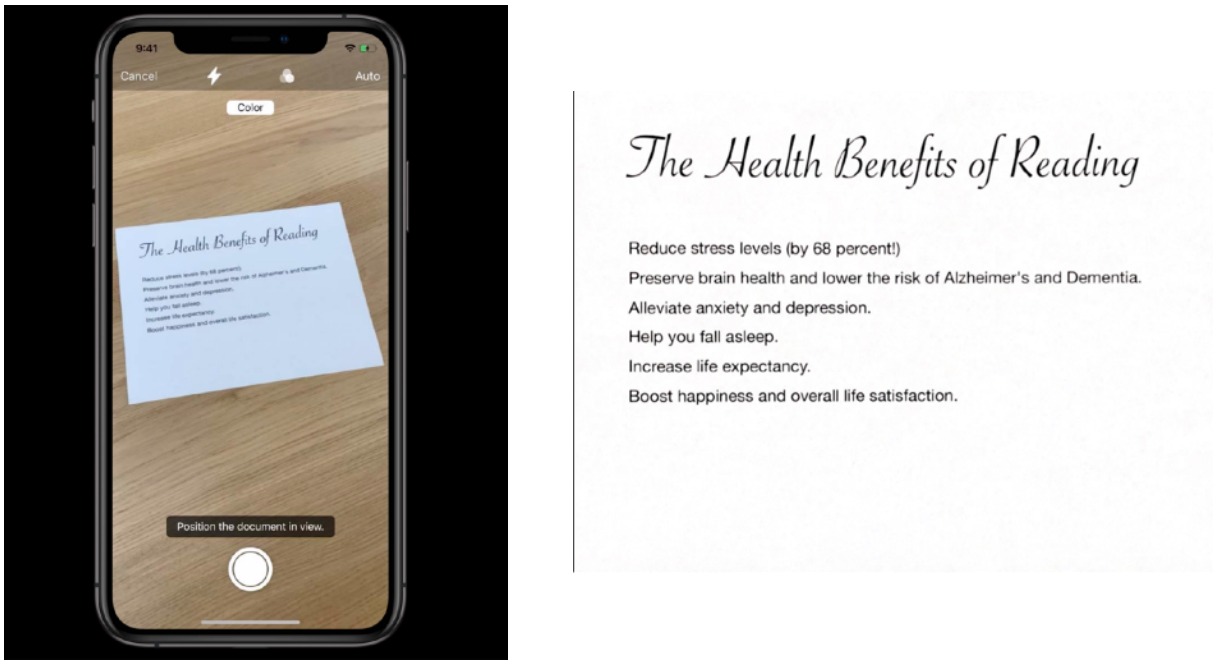

Figure 3.8: Demo Test of the app's OCR

3.3 MODEL PERFORMANCE

The aim of testing research is to discover mistakes. Testing is the mechanism through which any possible defect or flaw in a functioning product is found. It offers a way for parts, sub-assemblies, assemblies and/or a finished product to check their functionality. It is the software exercise phase in order to ensure that the software system meets and does not fail in an inappropriate way to fulfil the specifications and user expectations.

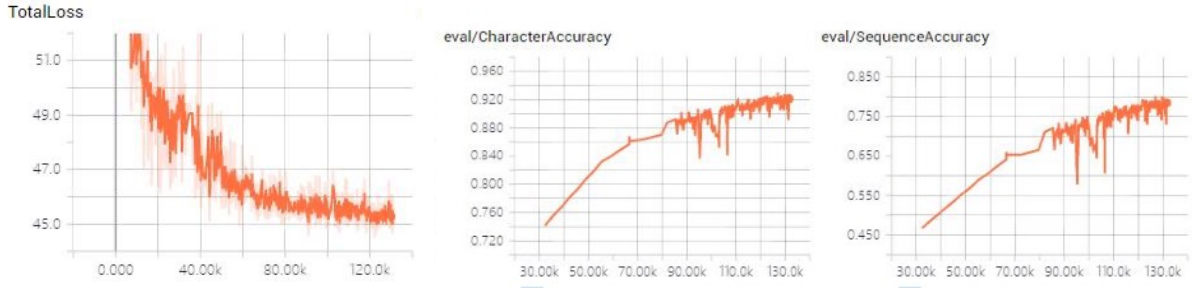

Figure 3.9: Measuring the (i) Total Loss and (ii) Accuracy of the model

Figure 3.10: Testing the model on a sample image

CHAPTER 4: CONCLUSION AND FUTURE WORK

This chapter summaries the entire work and presents possibilities for future work, ideas to be implemented in the field as well as thoughts on how the work could be improved going forward.

4.1 CONCLUSION

In the context of this report, a structure was given for a Deep Learning model to be created as a solution to tackle the traditional approach of making an Optical Character Recognition (OCR) app. The design of the applications was revamped numerous times as feedback came through, while the backend working of the model was researched thoroughly and reached a conclusive decision.

A Deep Learning model was trained and subsequently, converted in an app-deployable format, to detect text in any given scanned image.

The ability to share the scanned image as well as the result of the OCR as a PDF file was implemented in the app.

4.2 FUTURE WORK

Multiple OCR Scans as a Spreadsheet

The team plans to use CSV spreadsheets to store data automatically in case of a known data entry type. For example, receipts and business card scans should be detected and be automatically parsed into a spreadsheet for the user's ease of use.

In order to determine what the outcomes will be under various conditions, numbers can be changed, which is a function widely used by businesses in Spreadsheets to forecast future developments. It can also be used for evaluating a hypothesis with other data types, including empirical and statistical data.

REFERENCES

1. (2012) “Optical Character Recognition Using Artificial Neural Network” by Sameeksha Barve
   Available: ISSN: 2278 – 1323
2. (2019) Advances in Natural Language Framework [Online].
   Available: WorldWide Developers Conference
3. VisionKit - Official Framework Documentation [Online].
   Available: Apple Developer
4. Core ML - Official Framework Documentation [Online].
   Available: Apple Developer
5. (2012) “A Review on the Various Techniques used for Optical Character Recognition” by
   Available: K L University
6. Keras - High Level Library for Deep Learning [Online].
   Available: Module: tf.keras

—— END OF REPORT ——