



ULAB
UNIVERSITY OF LIBERAL ARTS
BANGLADESH

CSE 499

Development Project Report on

**Mobile Based Handwritten Text Recognition and Digitisation
System**

Submitted By

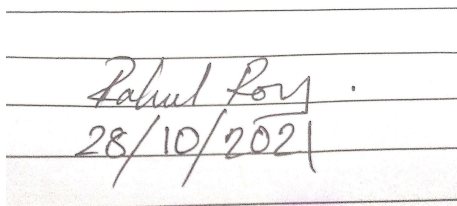
Rahul Roy (172014022)

October 28, 2021

DECLARATION

I do hereby declare that this project report entitled “Mobile Based Handwritten Text Recognition and Digitisation System ” has been prepared by me under the guidance of Dr. Muhammad Abul Hasan for the partial fulfillment of CSE program from the Department of Computer Science & Engineering (CSE), University of Liberal Arts Bangladesh (ULAB). I also assure that this report is original in nature and has not been submitted elsewhere for any other purpose.

Signature



Rahul Roy .
28/10/2021

Rahul Roy

ID: 172014022

Department of Computer Science & Engineering

Date: 28th October, 2021

ACKNOWLEDGEMENT

I would like to express my gratitude and appreciation to my supervisor Dr. Muhammad Abul Hasan Sir for all his diligence, support and encouragement during the development of the whole project. I am indebted to his cooperation, guidance, and advice- without which it would be considerably difficult for me to complete this One semester long project.

In addition, I am also grateful to my family members and friends for their cooperation, support and encouragement during this project work.

APPROVAL

The project entitled “Mobile Based Handwritten Text Recognition and Digitisation System” is submitted to the Department of Computer Science and Engineering at University of Liberal Arts Bangladesh (ULAB) in partial fulfillment of the requirements for the degree of Bachelor of Science.

Dated: 27th October, 2021

Muhammad Abul Hasan, PhD
Assistant Professor
Computer Science and Engineering
University of Liberal Arts Bangladesh

Signature & Date

Syed Akhter Hossain, PhD
Professor and Head of Department
Computer Science and Engineering
University of Liberal Arts Bangladesh

Signature & Date

Table of Contents

Chapter 1: Introduction	7
1.1 Introduction	7
1.2 Motivation	8
1.3 Objectives	9
1.4 Expected Outcomes	9
1.5 Project Management	9
1.6 Report Layout	11
Chapter 2: Background	12
2.1 Preliminaries/Terminologies	13
2.2 Scope of the Problem	16
2.3 Challenges	16
Chapter 3: Requirement Specification	17
3.1 Application Requirement Collection and Analysis	17
3.2 Dataset Information	18
Chapter 4: Design Specification	21

4.1 Application Design and Implementation	21
4.2 Machine Learning Design	26
4.3 Implementation Requirements	27
Chapter 5: Implementation and Testing	28
5.1 Implementation of Machine Learning Model	28
5.2 Deployment of Machine Learning Model	29
5.3 Testing Implementation	30
5.4 Test Results and Evaluation	33
Chapter 6: Impact on Society, Environment and Sustainability	36
6.1 Impact on Society	36
6.2 Impact on Environment	37
6.3 Ethical Aspects	37
Chapter 7: Conclusion and Future Scope	38
7.1 Discussion and Conclusion	38
7.2 Scope for Further Developments	39
References	40

Chapter 1: Introduction

1.1 Introduction

Scanned handwritten documents have been a major part of online education during the COVID-19 pandemic. Many online based applications like CamScanner have been widely used during this pandemic for scanning handwritten homeworks and assignments. However, constant internet activity is not prevalent in all geographical areas- and it is this inconsistency that may cause disruptions in the functioning of these online based applications. Therefore, the purpose of this project is to develop a prototype of a Flutter based offline application that is able to scan, extract and digitise handwritten text. For this purpose, a custom machine learning architecture will be developed, trained, tested and finally deployed in order to achieve the desired goal.

1.2 Motivation

Scanned handwritten documents have proved to be a major part of online education during the COVID-19 pandemic all over the globe- not only in Bangladesh. The inability to attend classes in the university premises. The motivation of this project is to create a lightweight machine learning model that can be deployed offline to scan, extract and digitise handwritten documents via a mobile application. This will reduce the problem encountered as a result of illegible handwriting and thereby reduce the mistakes made due to poor handwriting understanding.

1.3 Objectives

The objective of the project is to develop and deploy an application that can scan, extract and digitise handwritten text documents which would be characteristic of handwritten documents for homework and exam assignments encountered during the online classes conducted during the COVID-19 pandemic. The result of this project is to enable ordinary users - particularly students- to use an offline mobile application to scan, extract and digitise handwritten documents in the context of a low resource technological availability.

1.4 Expected Outcomes

The expected outcome of this project is to develop a Flutter based offline mobile application that can be used to scan, extract and digitise handwritten documents- leading to better storage, editability, accessibility and searchability of these original handwritten documents. The application will use Intelligent Word Recognition (IWR)- a neural network based approach to Optical Character Recognition (OCR).

1.5 Project Management

The project is expected to be finished within 8 weeks, with the following tasks :

- i. Literature Review
- ii. Requirement Engineering
- iii. Mobile Application Design
- iv. Front End Development
- v. BackEnd Development
- vi. Training of Machine Learning Model
- vii. Deploy Machine Learning Model
- viii. Application Testing

ix. Report Writing

The Gantt Chart for the planned activities - and their expected duration- is given below:

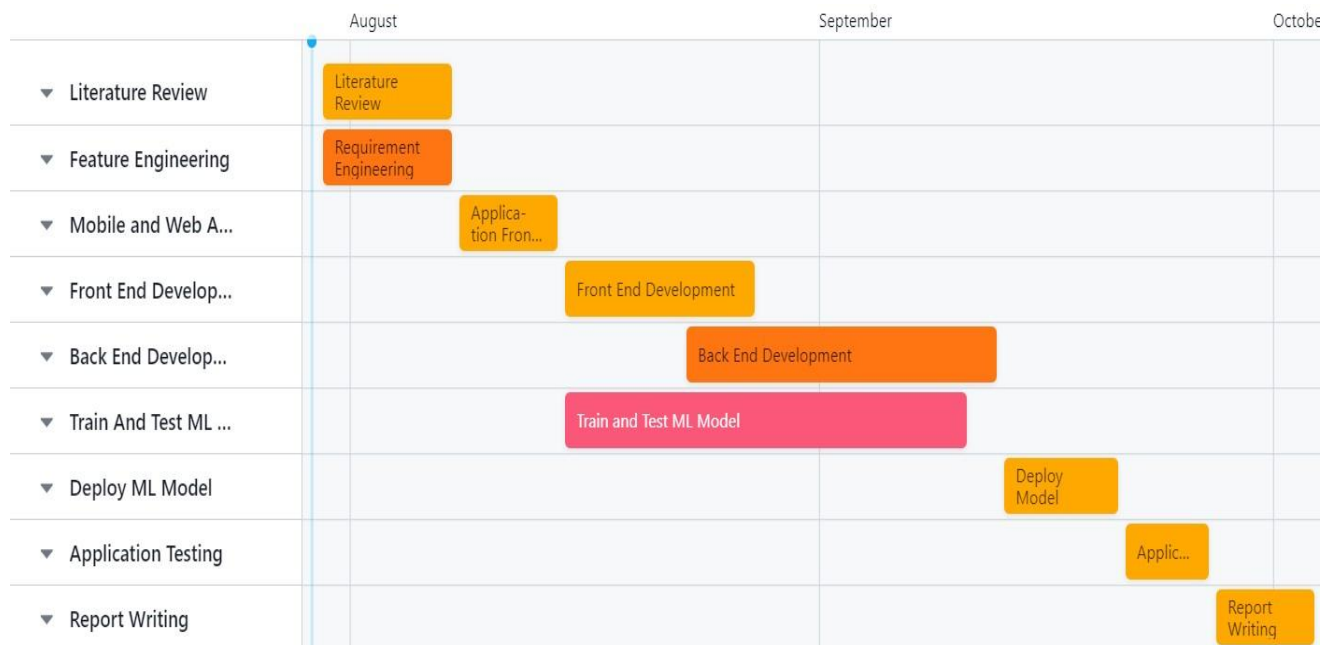


Figure 1: Gantt Chart

1.6 Report Layout

In **Chapter 2**, the technology stack- that will be used to carry out the project to completion- will be laid out. The names of basic terminologies that will be used throughout the report- and their respective descriptions- will also be written. The scope of the problem statement of this project will also be talked about - along with the challenges that are to be expected in the undertaking of this project.

In **Chapter 3**, the requirement specification of the application will be talked about, and the design requirements of the mobile application.

In **Chapter 4**, the planned front end design and the back end design of the application will be laid out- along with the implementation requirements.

In **Chapter 5**, the implementation of the front end and the backend of the application are given, along with the results acquired from the application and testing.

In **Chapter 6**, the impact sustainable aspects of this project- and the underlying technology- including the effects and impact on the society, environment and the world.

In **Chapter 7**, the conclusions are discussed. Finally, the scope for further developments of this project is discussed.

Chapter 2: Background

2.1 Preliminaries/Terminologies

Convolutional Neural Network (CNN) is a type of artificial neural network architecture widely used in both descriptive and generative works related to image processing and recognition.

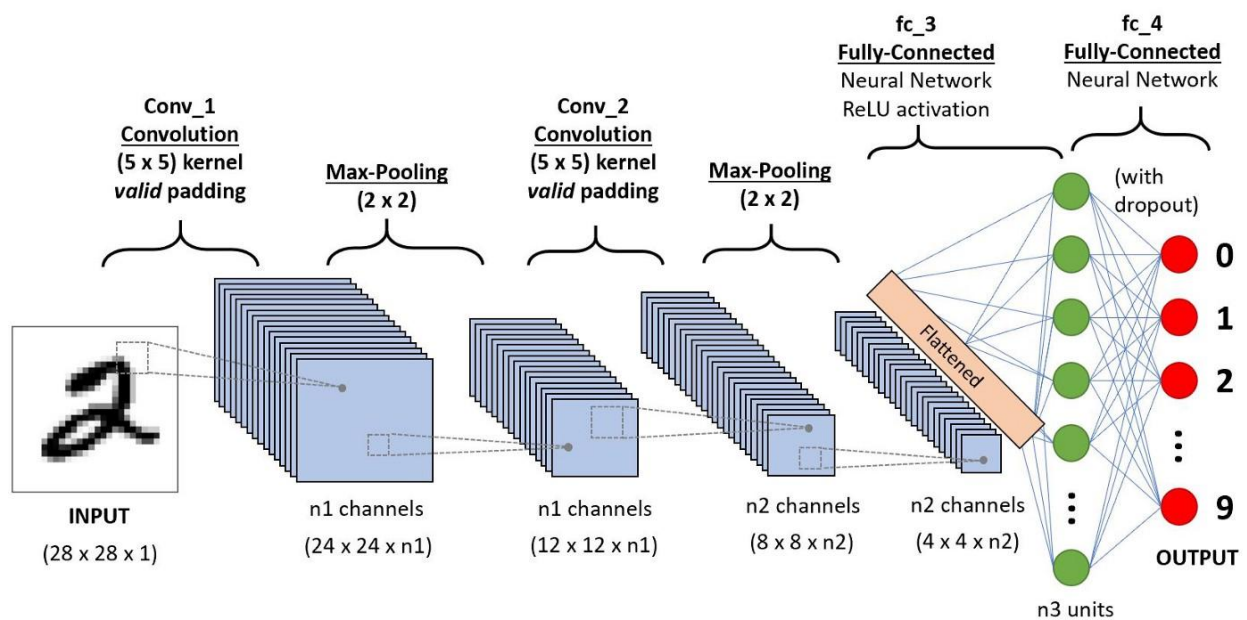


Figure 2: An example of convolutional neural network used to identify handwritten digits

Recurrent Neural Network (RNN) is a particular type of artificial neural network architecture which is used for time series data, involving sequences. RNNs- by having the concept of memory- can store the states/information of prior inputs- of varying lengths- to generate the next output of the sequence.

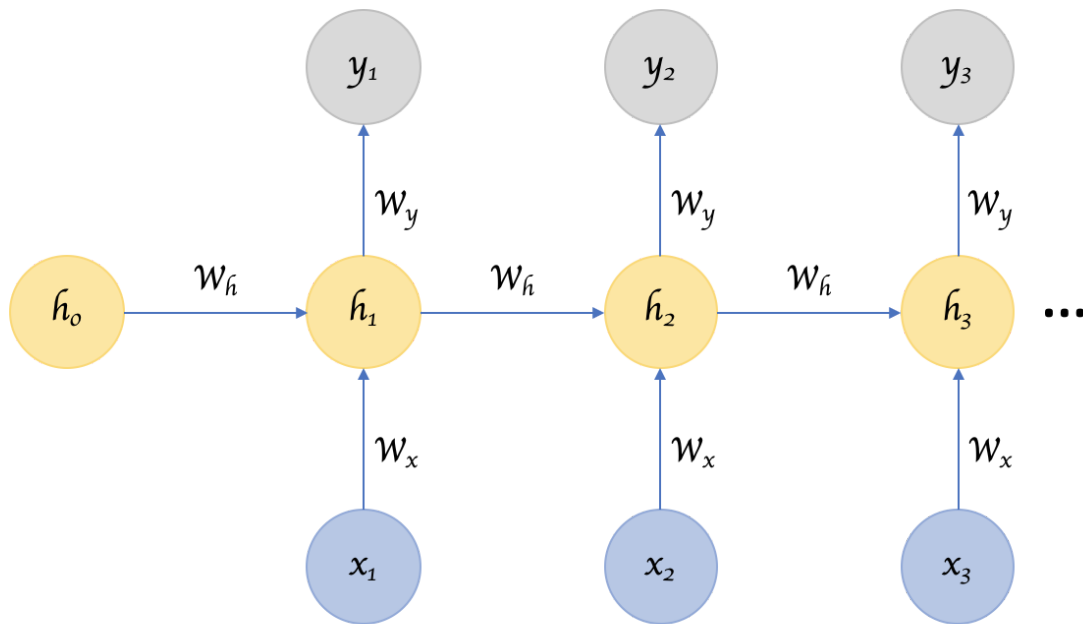


Figure 3: A recurrent neural network with a hidden state.

Convolutional Recurrent Neural Network is a deep learning architecture that involves a combination of CNN and RNN. The architecture involves a 2D CNN, batch normalization, RELU activation, max-pooling and dropout. Three to 5 CNN

layers are sequentially placed followed by a reshape layer, which is then followed by RNN layers (bidirectional GRU).

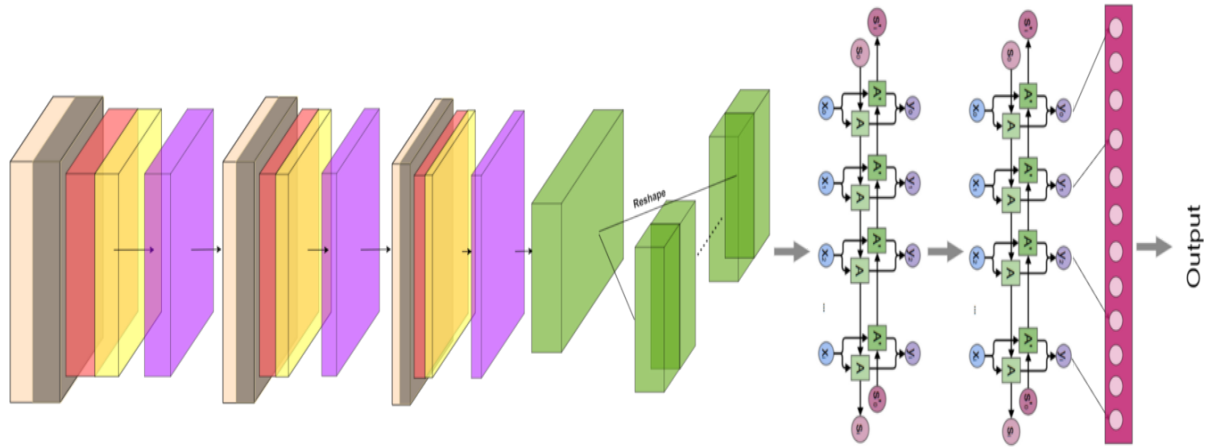


Figure 4: A sample CRNN architecture^[3]

Connectionist Temporal Classification(CTC) is an algorithm which is used to train artificial deep neural networks for handwriting recognition, speech recognition and other sequence related problems. The algorithm is suitable for its use as a loss function- and also for the purpose of making inferences- is because it does not require the input and output sequence to be aligned.

Therefore, given an input series X and an output series Y , CTC gives an output distribution over all possible Y for a particular input of X . The CTC objective for a single (X,Y) pair is:

$$p(Y | X) = \sum_{A \in \mathcal{A}_{X,Y}} \prod_{t=1}^T p_t(a_t | X)$$

The CTC conditional probability	marginalizes over the set of valid alignments	computing the probability for a single alignment step-by-step.
---	--	--

2.2 Scope of the Problem

The scope of this project is to develop a prototype of a mobile application that can access and use the machine learning model to scan, extract and digitise handwritten documents. The model will be deployed and used offline- as opposed to many other existing applications that require online connectivity for their use.

2.3 Challenges

The primary challenge of this project is to be able to develop and deploy a lightweight machine learning model that can be deployed to the Flutter based application effectively and efficiently.

Chapter 3: Requirement Specification

3.1 Application Requirement Collection and Analysis

The application should be able to perform the following operations:

- i. Select a picture [of handwritten documents] from the mobile storage
- ii. Display the image on the mobile screen.
- iii. Crop the image to adjust for the display of handwritten texts
- iv. Display the handwritten words and the respective digitised texts.
- v. Edit the displayed digitised text - to allow for corrections, in light of incorrect recognition.

3.2 Dataset Information

The IAM dataset has been used for the purpose of training, testing and deploying the custom machine learning model. The dataset has been divided into a train-validation-test set according to the following ratio:

Train Dataset → 75

Validation Dataset → 15

Test Dataset → 15

The original dataset has been preprocessed and cleaned before dividing the dataset into the above three sets. Firstly, a word list was created from the dataset that contains only the correct results of the image segmentation (labelled **ok** in the dataset). Secondly, all the images were converted into rectangular shapes for better OCR effectiveness. The total number of words contained in the cleaned dataset is 96, 456 words.

As a result the total number of words contained in each of the split dataset is as follows:

- i. Train Dataset : 72, 342 words
- ii. Validation Dataset: 12, 057 words
- iii. Test Dataset: 12, 057 words.

An illustration of the train set images- along with the labels- is given below:

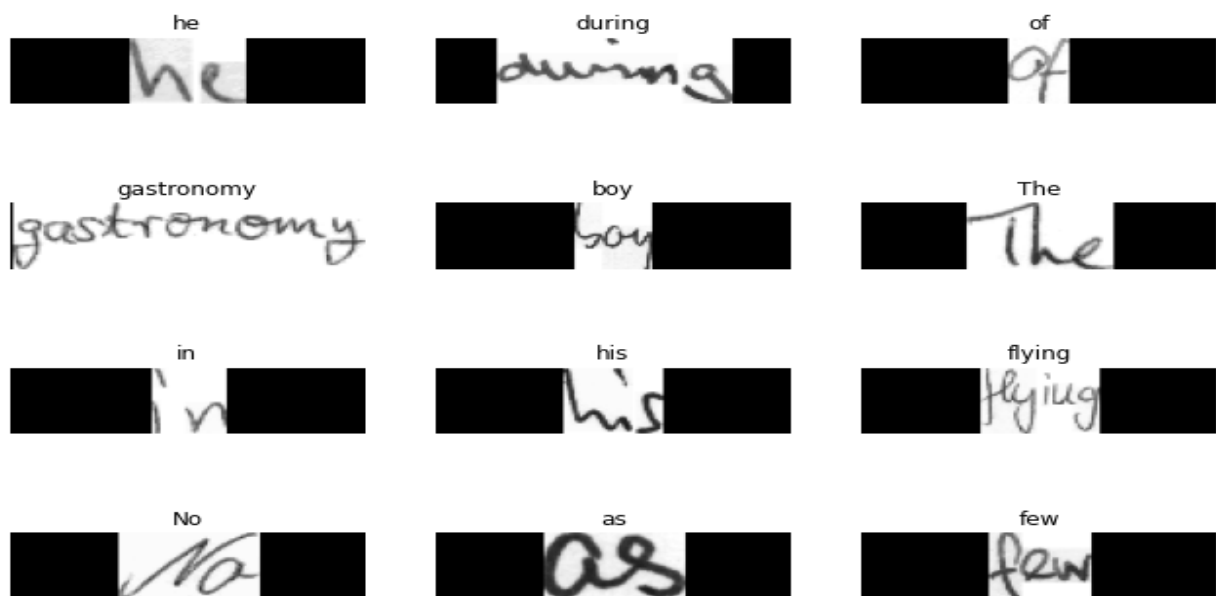


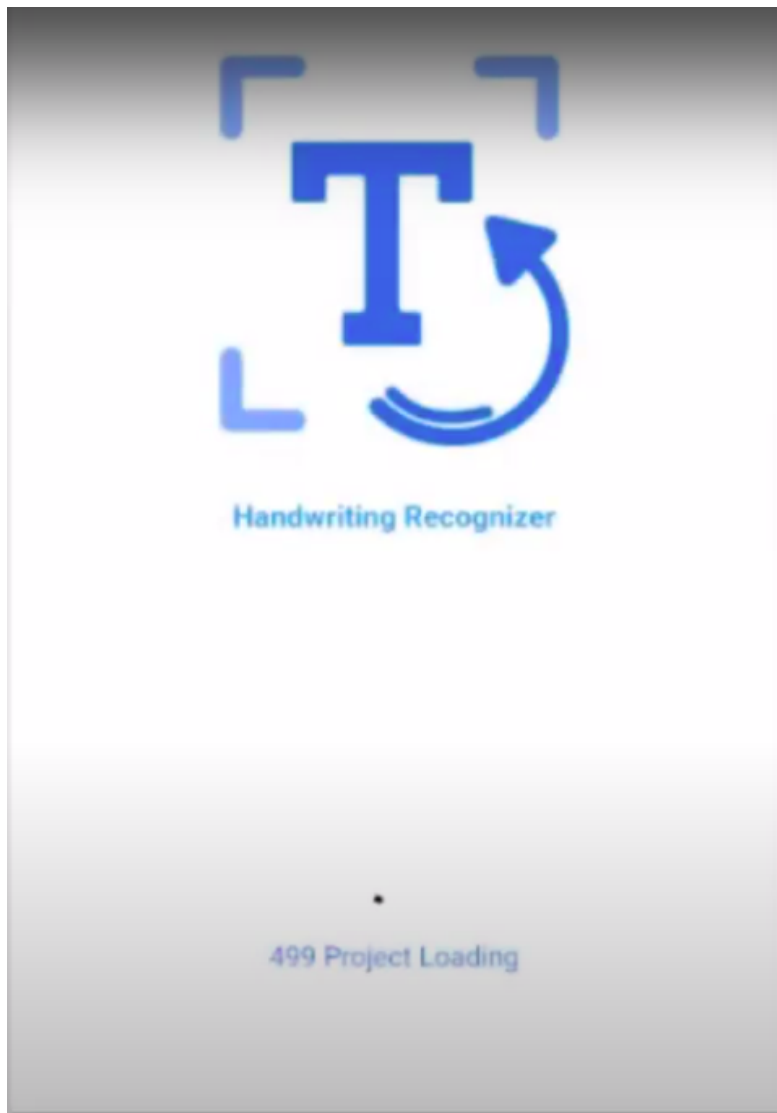
Figure 5: Train set images with labels after data processing

Chapter 4: Design Specification

4.1 Application Design and Implementation

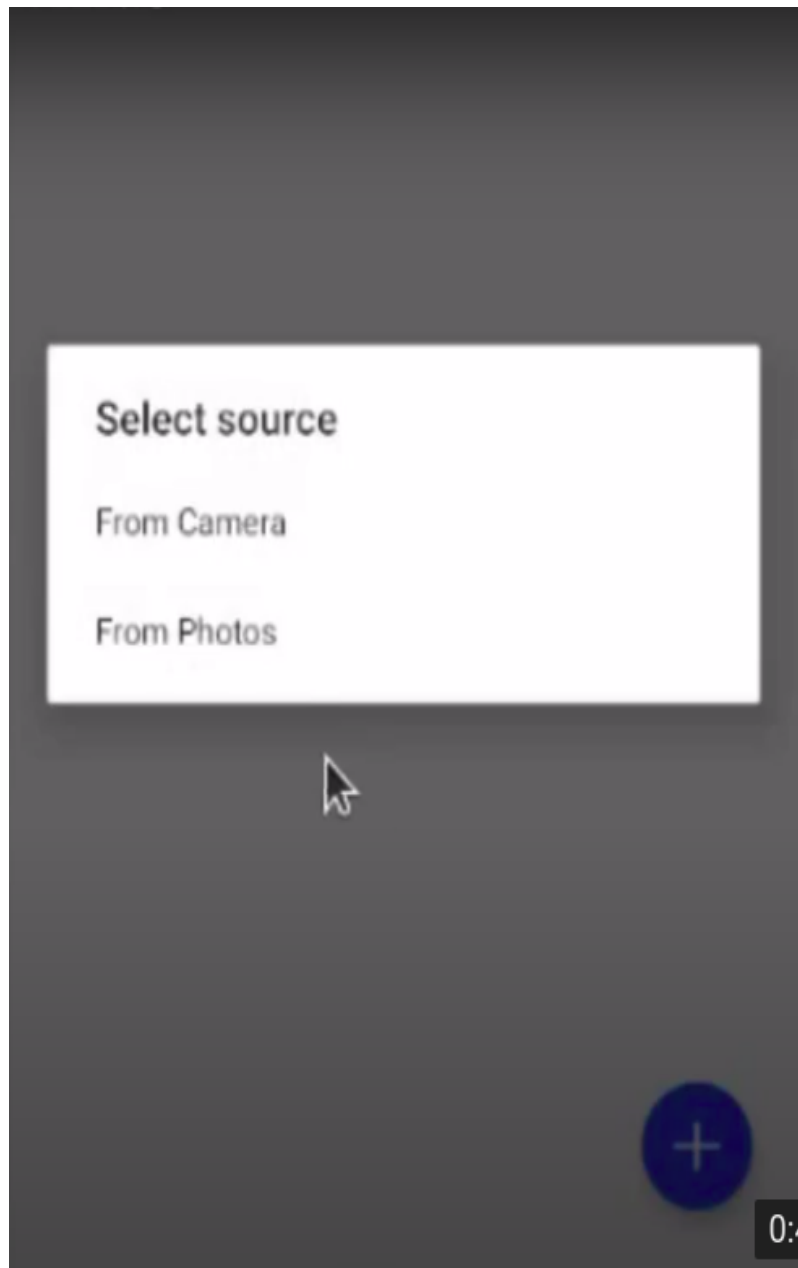
The application design contains the following screens:

Splash Screen

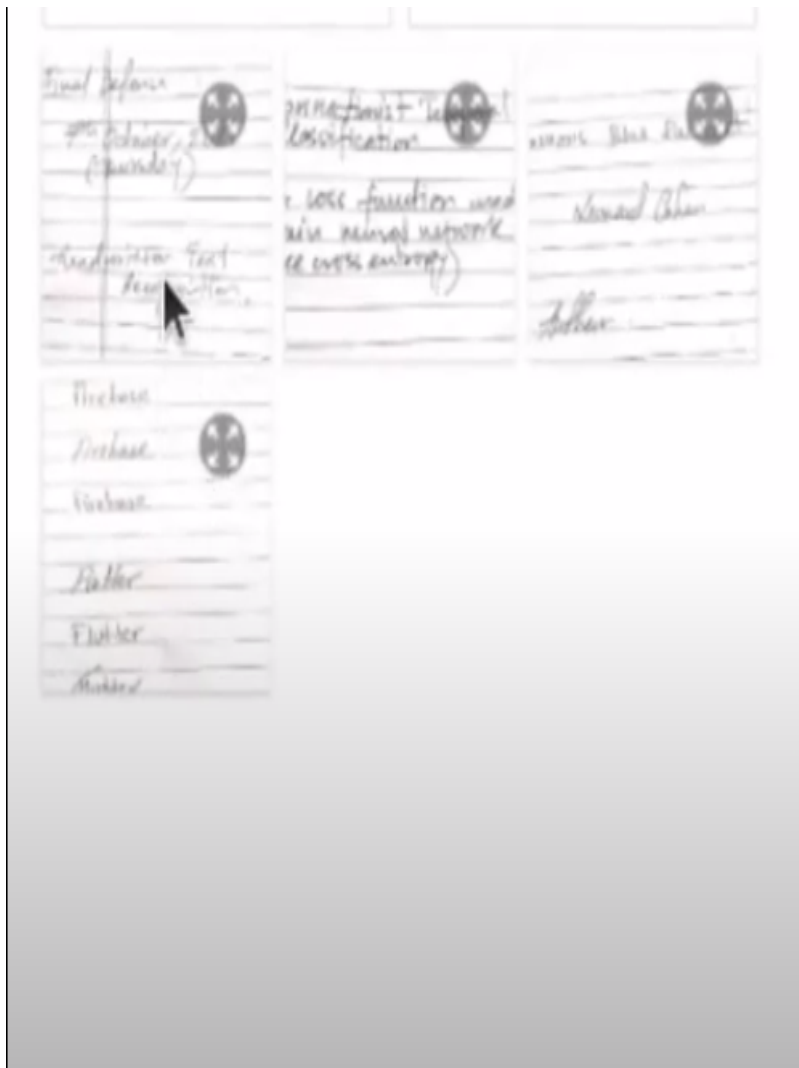


In the homepage, the application allows for access to both the camera and the storage of the android phone emulator:

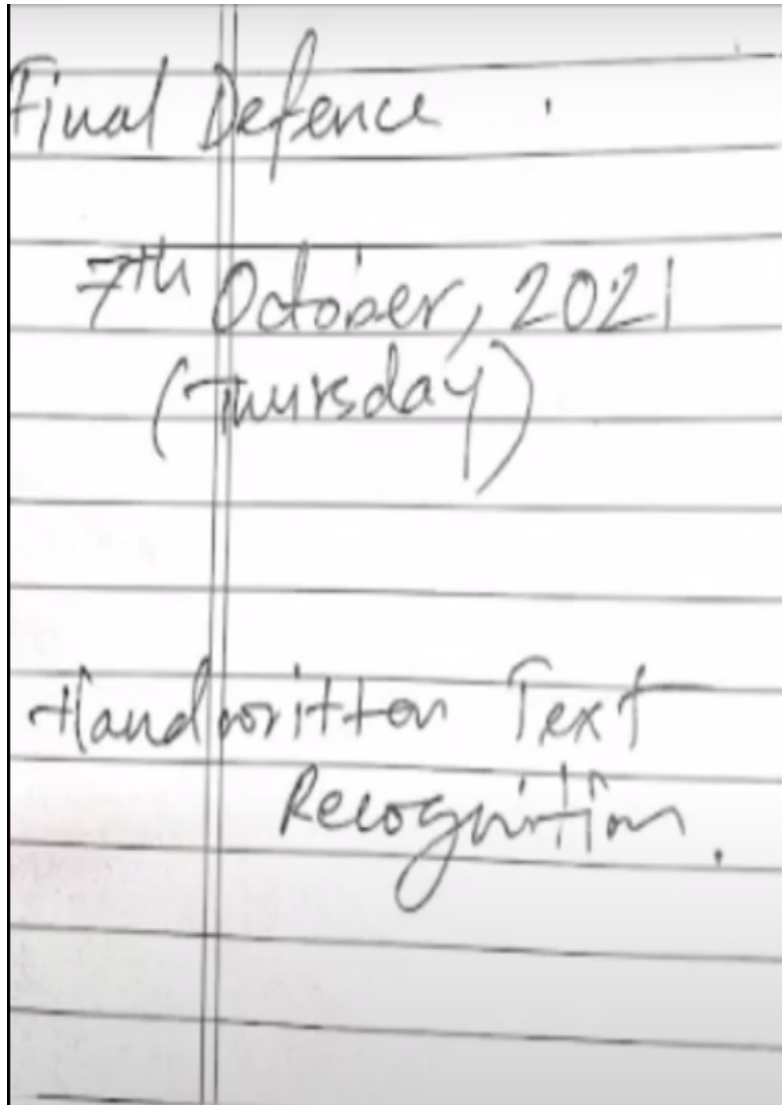
Home Page



The images of handwritten texts can be accessed from the storage of the Android phone emulator:



One of the images is selected, and the image is subsequently cropped to allow for the handwritten text to be displayed.



The results of the model prediction and inference are given below. An editable text box is used to display the predicted output so that the user can edit the text in order to correct it.

Final Defence

Final Defence \$ EDIT

7th October, 2021

th October, 2021 EDIT

(Thursday)

(Thursday EDIT

Handwritten Text

Handwritten Text EDIT

Recognition.

Recognition EDIT

For the storage of the images used in the application, the memory of the Android Application emulator was used.

4.2 Machine Learning Design

The machine learning model to be trained and deployed in this application will have the following CRNN architecture, along with CTC as a loss function:

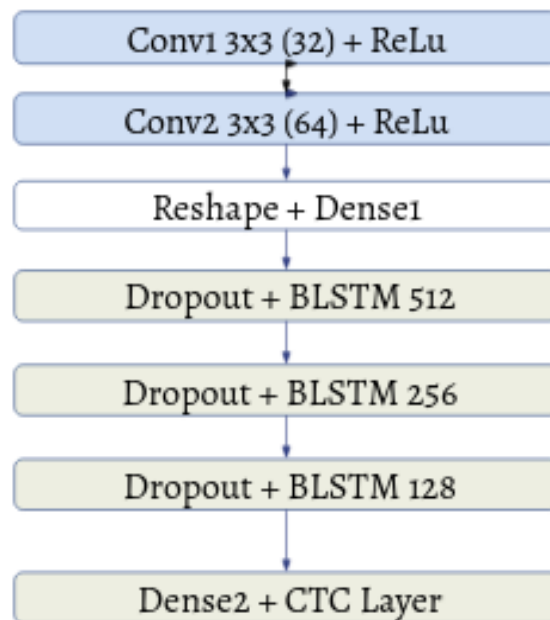


Figure 6 : Deep Learning CRNN Architecture for Handwriting Recognition

The architecture will have two CNN layers - with the ReLu activation function- followed by a reshape and the first dense layer. This is done before feeding the inputs to the RNN layers.

For the RNN layers, three Bidirectional Long Short-Term Memory(BLSTM) layers are used. Finally the output is then fed to a layer containing the second dense layer and the CTC loss function. The Edit Distance and the Character Error Rate are taken as the measurement metric for the model. The loss and validation loss are also measured for the train and validation dataset.

4.3 Implementation Requirements

The training, testing and implementation of the model was carried out in the Google Colaboratory. The hardware specifications of the training environment are as follows:

- Tesla P100-PCIE-16GB
- NVIDIA Cuda Cores → 3584 cores

Chapter 5: Implementation and Testing

5.1 Implementation of Machine Learning Model

The implementation of the machine learning model is illustrated below:

Layer (type)	Output Shape	Param #	Connected to
image (InputLayer)	[(None, 128, 32, 1)]	0	
Conv1 (Conv2D)	(None, 128, 32, 32)	320	image[0][0]
pool1 (MaxPooling2D)	(None, 64, 16, 32)	0	Conv1[0][0]
Conv2 (Conv2D)	(None, 64, 16, 64)	18496	pool1[0][0]
pool2 (MaxPooling2D)	(None, 32, 8, 64)	0	Conv2[0][0]
reshape (Reshape)	(None, 32, 512)	0	pool2[0][0]
dense1 (Dense)	(None, 32, 64)	32832	reshape[0][0]
dropout_1 (Dropout)	(None, 32, 64)	0	dense1[0][0]
bidirectional_3 (Bidirectional)	(None, 32, 512)	657408	dropout_1[0][0]
bidirectional_4 (Bidirectional)	(None, 32, 256)	656384	bidirectional_3[0][0]
bidirectional_5 (Bidirectional)	(None, 32, 128)	164352	bidirectional_4[0][0]
label (InputLayer)	[(None, None)]	0	
dense2 (Dense)	(None, 32, 81)	10449	bidirectional_5[0][0]
ctc_loss (CTCLayer)	(None, 32, 81)	0	label[0][0] dense2[0][0]

Figure 7 : Implementation of the machine learning model

The total number of trainable parameters for this model are 1, 540, 241. The model is trained for a time duration of 70 minutes and for 30 epochs.

5.2 Deployment of Machine Learning Model

For the purpose of deployment of the model, the following steps were followed:

- i. The model was saved as a .h5 file in Keras.
- ii. The model was converted using Tensorflow Lite to a .tflite file in google colab.
- iii. The newly created .tflite file was downloaded and moved in the assets folder of the Flutter application.
- iv. The corresponding dependency of the TF Lite was added in the file (pubsec.yaml file).
- V. The TF Lite library was imported and the model was loaded in the main code.

5.3 Testing Implementation

For testing purposes, different types and texts of handwritten documents were used and tested to see the results of the model's inference. The results are given below:

Test - 1

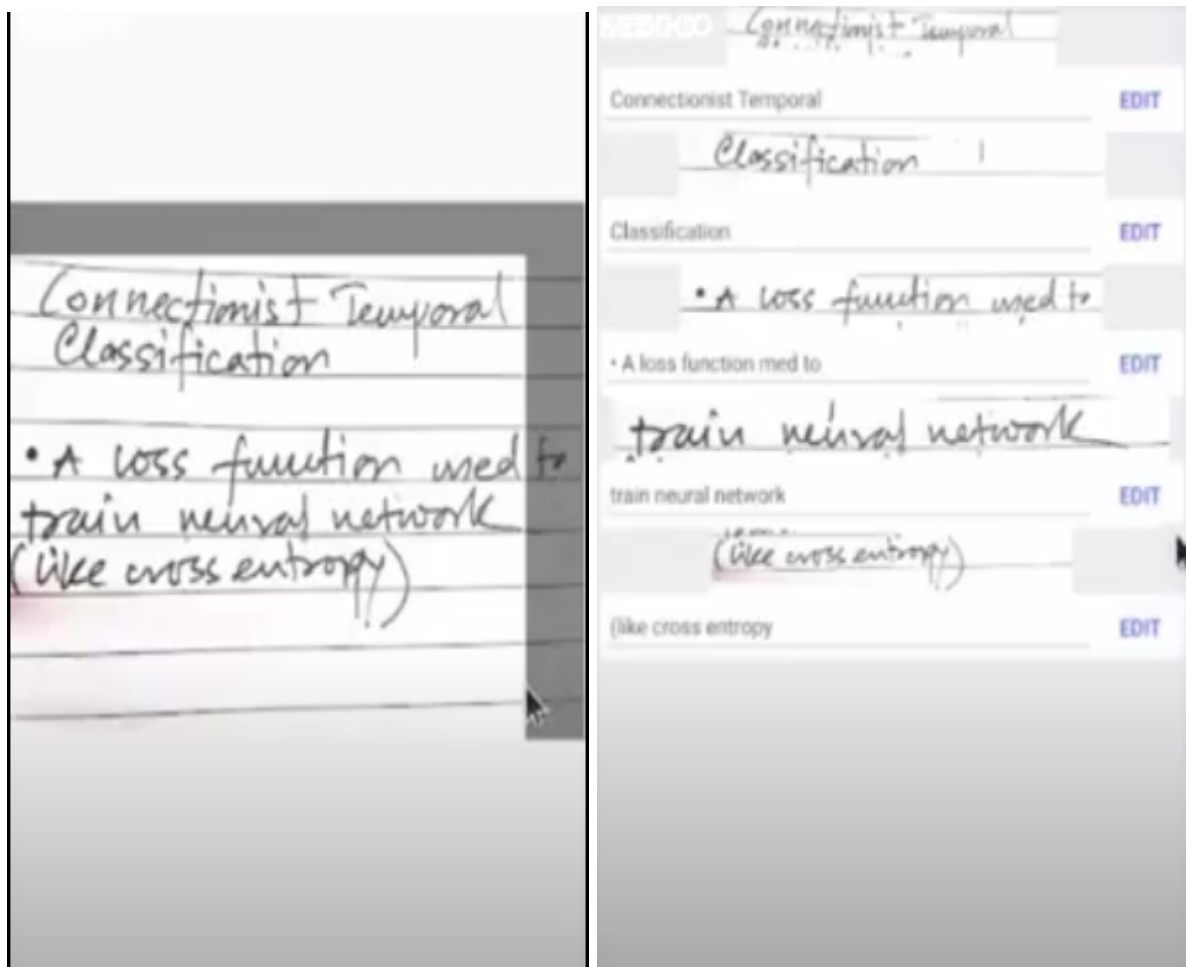


Figure 8: Testing of the application, Test - 1

Test - 2

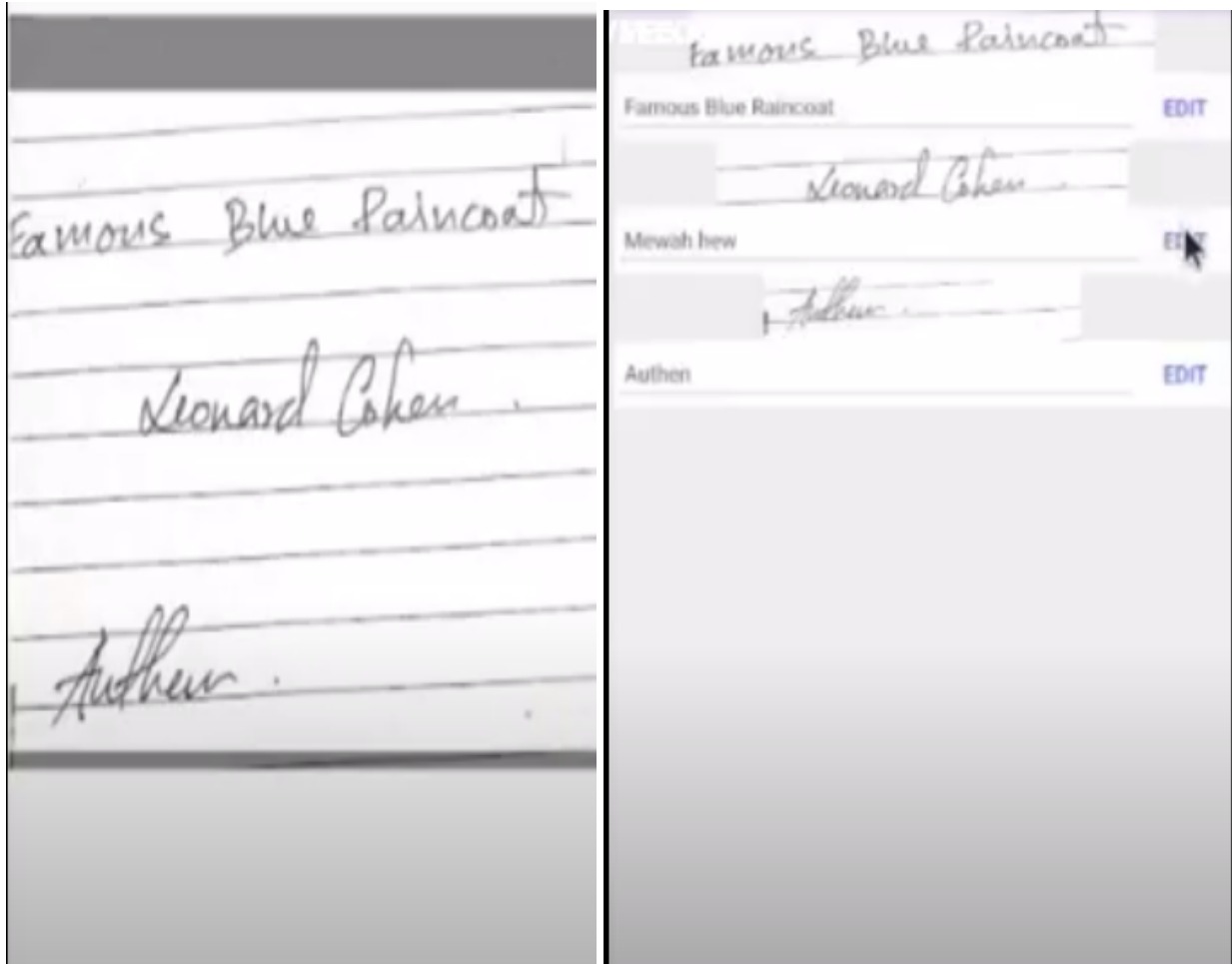


Figure 9: Testing of the application, Test - 2

Test - 3

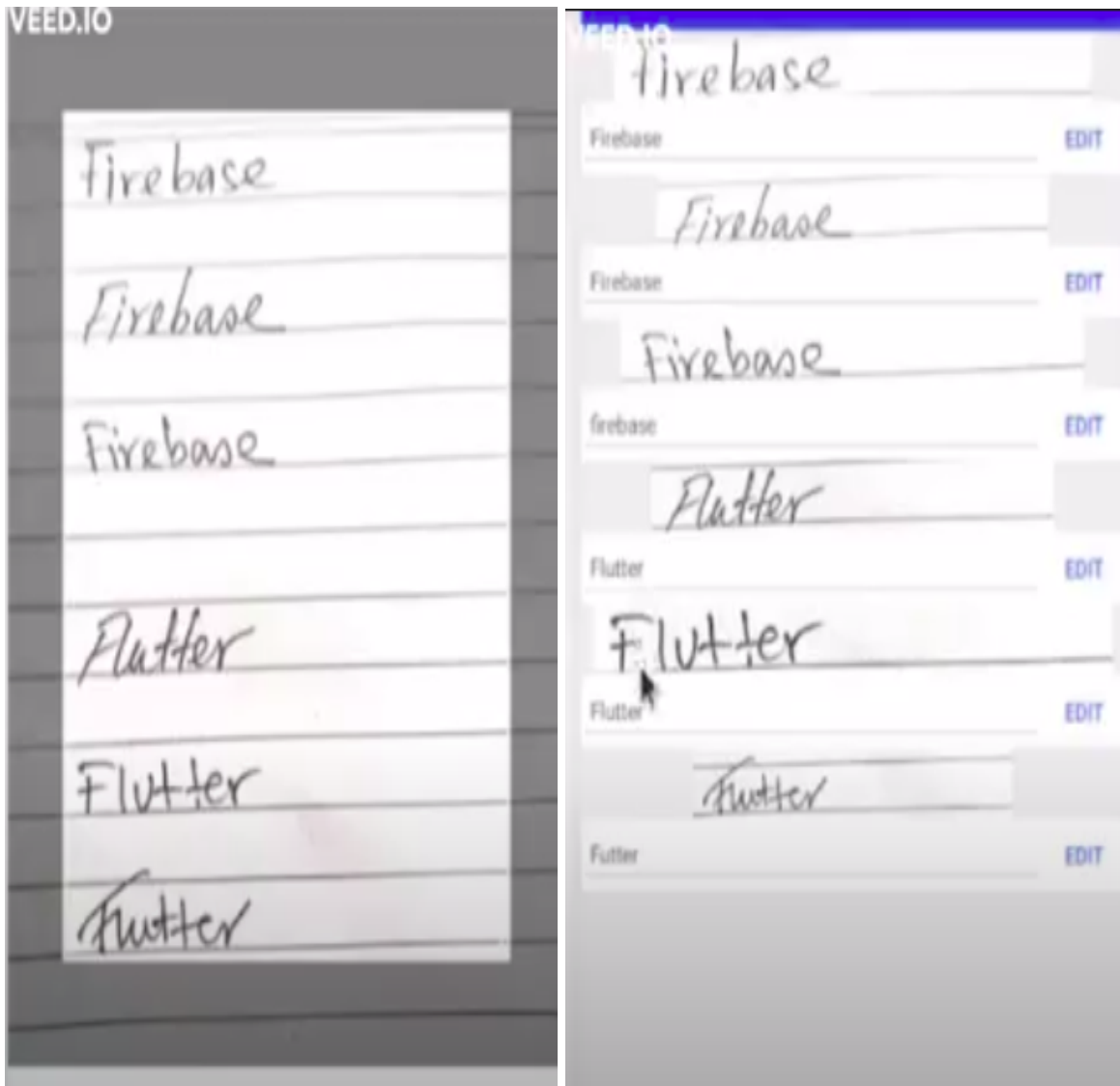


Figure 10: Testing of the application, Test - 3

5.4 Test Results and Evaluation

After 30 epochs of training the machine learning model, we have the following results:

	Edit Distance	Loss	Validation Loss
1st Epoch	20.5496	13.7544	13.0424
30th Epoch	17.3306	1.4722	1.8272

There has been a sharp reduction in loss and validation loss for the model. A reduction in edit distance is also observed. However, after 30 epochs, the edit distance hovers around 17.33. This may be attributed to overfitting of the model to the training dataset. A graphic illustration of the loss and validation loss is given below:

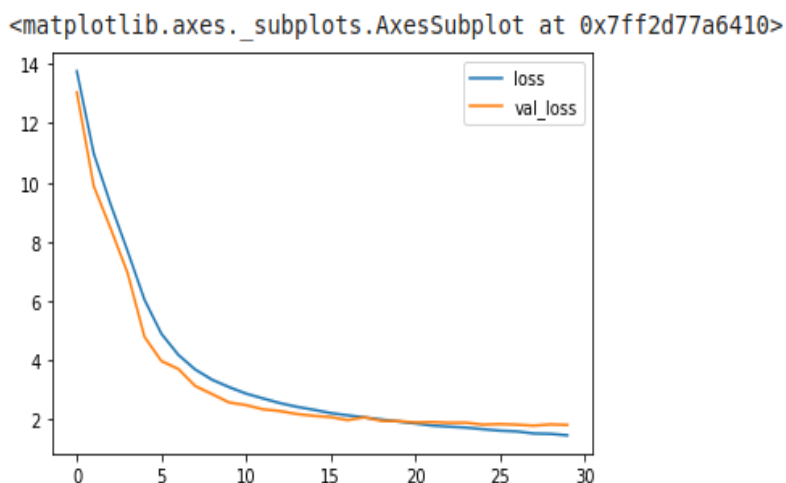


Figure 11: Loss and Validation Loss

An illustration of the inference produced by the model for output predictions:



Figure 12: Output predictions of the Test Set

Evaluation

To evaluate the accuracy of the model we use the **Character Error Rate (CER)**. CER operates on the character level of words, and is relevant when the sentences and paragraphs of words are involved. CER detects the number of characters incorrectly predicted, in comparison to the ground truth.

$$CER = \frac{\text{Edit Distance}}{\text{Average Number of Characters in the reference text}}$$

$$CER = \frac{\text{Edit Distance}}{\text{Total Number of Characters} / \text{Total Number of Words}}$$

$$CER = \frac{17.3306}{4412650 / 12057}$$

$$CER = 0.04735 \text{ or } 4.7353\%$$

Therefore the handwriting recognition rate is **95.26%** accurate on the test dataset.

According to the established benchmark for character error rate is^{[4][5]}:

	Good	Average	Poor
CER	1 - 2 %	2 - 10 %	>10 %
Accuracy	98 - 99%	90- 98%	< 90%

Therefore, the model - with 95.26% accuracy for handwriting recognition prediction- falls within the average performance of the accepted benchmark for character error rate.

Chapter 6: Impact on Society, Environment and Sustainability

6.1 Impact on Society

Optical Character Recognition carries a valuable significance for society as large. Increased digitisation of services and utilities the world will cause OCR in general, and ICR for handwriting recognition in particular, to have significant promise as more and more documents, papers, bills are digitised. This will lead to better accessibility, storage and utility of previously physical documents. Not only will it prove to be convenient and useful to the end users and the business sector capitalizing on this technology- it will also have a major impact on the environment and sustainability. Therefore, this will prove to be a net positive technological advent in individual, social and business life.

As for this project, this is intended to be a step towards the use of offline applications to scan and digitise handwritten text in the service of education in the

period of online learning. This will allow people with low resource technology to scan and digitise text in an effective manner.

6.2 Impact on Environment

By reducing dependence on offline and handwritten methods of documents, the use of paper and their storage can be reduced. Also, by making documents available in a digital format, the effective and efficient transfer of these documents is made possible via the internet. This will reduce the costs - both financial and environmental- associated with the physical storage and transfer of these documents. This will allow for better recycling of unnecessary paper and hardback materials.

6.3 Ethical Aspects

As the digitisation of previously offline documents occurs, security will remain a significant challenge. As more and more individuals and businesses adopt and integrate OCR applications in their life and business activities, the permission to access these documents will prove to be a crucial problem. The hacking and

unauthorized access to these documents can be detrimental to both individuals and businesses.

Chapter 7: Conclusion and Future Scope

7.1 Discussion and Conclusion

In conclusion, the application has proved to be quite effective in doing the following proposed activities:

- i. Scan Handwritten Text
- ii. Extract Handwritten Text
- iii. Digitise Handwritten Text

With a Flutter Application, a deep neural network based machine learning CRNN model was created which was then trained, tested and finally deployed to digitize handwritten texts. The model was accurate about 95.26% of the time in recognizing handwritten texts. Some test cases were made where the inference of the model was seen by feeding custom handwritten texts into the application for digitisation.

7.2 Scope for Further Developments

There are multiple avenues for further development. On the machine learning side, the model can and needs to be improved- in terms of architecture- and consequently to be fine tuned for lesser CER and higher handwriting accuracy. On the application side, the scope for development lies in converting the prototype into a full stack mobile application with robust features that can ultimately have a business value in the market.

References

1. <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
2. <https://gotensor.com/2019/02/28/recurrent-neural-networks-remembering-whats-important/>
3. <https://towardsdatascience.com/an-approach-towards-convolutional-recurrent-neural-networks-a2e6ce722b19>
4. <https://www.scitepress.org/Papers/2015/51786/51786.pdf>
5. <https://towardsdatascience.com/evaluating-ocr-output-quality-with-character-error-rate-cer-and-word-error-rate-wer-853175297510>
6. <https://flutter.dev/docs>
7. <https://sci-hub.se/https://doi.org/10.1109/ESCI48226.2020.9167649>
8. <https://www.efilecabinet.com/what-is-ocr-and-how-will-it-impact-your-life/>

9. <https://www.sciencedirect.com/topics/computer-science/optical-character-recognition>
10. <https://www.forbes.com/sites/forbestechcouncil/2019/09/10/the-future-of-ocr-is-deep-learning/?sh=5b4806626a04>