INCREDIBLE INDIA

A PROJECT REPORT

Submitted by

KEVIN PAUL 2018115049

SRIJEYARANKESH 2018115109

VISHWA KUMAR S 2018115134

submitted to the Faculty of

INFORMATION AND COMMUNICATION ENGINEERING

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY

COLLEGE OF ENGINEERING, GUINDY

ANNA UNIVERSITY

CHENNAI 600 025

MONTH YEAR

ABSTRACT

India has 22 constitutionally recognised languages written in 13 different scripts. An average traveller, on a business or pleasure trip, often gets confused by the various signboards written in an unfamiliar language in a new region. This often spoils the experience of visiting a new place and the traveler goes back with not-so-fond memories. The resulting unfortunate, unpleasant and unproductive bitterness can easily be avoided by building better Apps. The field of Artificial Intelligence and Mobile Application platforms have made it possible to offer an efficient and robust solution to those problems.

The Project focuses upon automatically detecting and recognizing the text inside an image and translate it from the corresponding language into English. The above tasks require the knowledge of python, state-of-the-art deep learning algorithms, flutter, flask API and principal experience in android app development.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Technology has been evolving at a rapid pace since the dawn of the 21st century. Humans are being replaced by machines whose computational capacity and efficiency is exponentially greater. Artificial Intelligence has pervaded every aspect of our lives. Mobile phones have also seen tremendous improvements over the past few decades. From analog communication channels in the 1980s to the introduction of smartphones in the late 2000's, mobile devices have shown great potential. In the last few years, many advancements have been made to integrate machine learning with mobile applications to provide a next-gen experience for the users. Adopting these techniques, many creative and innovative solutions can be provided to the public by the developers.

1.1 CONVOLUTIONAL RECURRENT NEURAL NETWORK

Neural networks are a subset of machine learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another. Neural Networks can learn the parameters of their respective functions through various techniques. The CRNN module used here is a combination of the ResNet architecture and Bi-directional LSTM. When a dataset tagged with corresponding word inside the signboard is given as input, the CRNN module learns the parameters by using a suitable loss function. After training the module and obtaining a reasonable accuracy, it can be integrated with the application server hosted remotely to serve the client mobile application. The user is abstracted from the underlying algorithms of the project by presenting an interactive UI and appealing output with swiftness.

1.2 PROBLEM STATEMENT

The project's aim is to identify the text within a signboard and translate it to English without losing the context and meaning which the original wants to convey. This is achieved by using CRAFT algorithm to identify characters with close affinity such that they form a word. Co-ordinates of the words identified are fed into the CRNN module which learn the features of the word. The translation API later translates the word into English by maintaining it's context and meaning. The mobile application captures the image of a sign board using a camera and uploads the image onto the flask server. The server containing the trained models, performs the above mentioned techniques and returns the answer back to the user.

1.3 INCENTIVE

India's tourism has been booming in the recent years. As a result, many foreigners visit India's monument and learn about the tradition. One hurdle they face is language. Owing to the low literacy rate as compared to the rest of the world, many can't speak English. The pre-existing applications are not suitable for use in Indian languages. So we were inspired to provide a solution by building an application which enables the tourists to understand local sign boards. The project covers a wide range of languages so that any place of India can be visited without hassle. Such an application would give tourists a pleasant experience and to travel seamlessly without worrying about communication.

1.4 PROJECT DETAILS AND PLATFORMS

1.4.1 Flask

Flask is a micro web framework written in Python. It is classified asa microframework because it does not require particular tools or libraries. It hasno database abstraction layer, form validation, or any other components wherepre-existing third-party libraries provide common functions. However, Flasksupports extensions that can add application features as if they were implementedin Flask itself. Extensions exist for object-relational mappers, form validation,upload handling, various open authentication technologies and several commonframework related tools. We use flask to connect the localhost of flutter web tothe yolo model.

1.4.2 Python Notebook

We use deep learning algorithms and corresponding architectures to yield better results with much higher accuracy. They are trained on jupyter notebooks using a GPU and then checkpoint models are used in the project. Advantages of using a interactive python notebook is that only a portion of a code in a program can be run instead of the entire program. This enables easy debugging and faster development of applications by developers.

1.4.3 Android

We have mnay platforms available for mobile app development. Java and Kotlin are native to Android whereas swift is native to iOS. To overcome this inherent non-compatibility of an app built in either of the above languages, we use Flutter and React Native, two of the most popular cross platform frameworks used extensively in mobile app development. In this project though Java and Kotlin are used as the target audience is android users. The application is also intended to work with devices with android version higher than 9.0.

CHAPTER 2

EXISTING WORK

Optimal Character Recognition is the mechanical conversion of typed, handwritten or printed text into machine encoded text which is widely used to reduce the error of human data entry by creating a well trained sufficient model to handle the complexities of our work. It is common method of digitizing printed texts so that they can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as cognitive computing, machine translation, text-to-speech, key data and text mining. OCR is a field of research in pattern recognition, artificial intelligence and computer vision.

What we aim to do is create and deploy our custom version of this model by manually training said model to our dataset to develop a highly accurate model which recognises the text and also translate it with our very own custom developed API which invokes the Google Translate servers and finally integrate it all to a mobile application which we developed.

2.1 METHOD OF CITATION

Following are a few methods of citation.

2.1.1 Citation of Journals

Method for citing journals can be seen in References [1], [2] and [3].

2.1.2 Citation of Proceedings

Proceedings should be cited as given in the References [4] and [5].

2.1.3 Citation of Websites

https://cloud.google.com/translate/docs [6].
GToken key:
https://translate.google.com/translate/releases/twsfew20170306RC00/r/js/desktopmodulemain
[6].
Clickable Links Below : -
Research Paper on CRAFT Algorithm
Research Paper on CRNN Architecture
Beam Search Approach
CTC Loss function explanation

2.2 DETECTION, RECOGNITION, TRANSLATION

2.2.1 Detection

The Detection module has already been implemented using models such as SSD(single shot) model and CTD(Curve Text detection) model. These models are however specific to the problems they address i.e SSD is mainly used for linear text and CTD is mainly used only for curved text. What if a situation arises where need to use both? Thus, we want to develop an application which is independent of the nature of the image and universally acceptable.

2.2.2 Recognition

Recognition has undergone tremendous changes in the last few years. Only conventional recurrent neural networks were being used with some using LSTMs and GRUs. However those memory cells cannot hold information of a hidden state for a very long time. Moreover, as the input size gets bigger we have vanishing gradient descent problem which has to be fixed. Thus we wanted build a recognition model which overcomes these faults and which is scalable.

2.2.3 Translation

Our model is enhanced to reverse engineer a GToken key as given in the citations [6] which was released by Google Cloud servers to manually use a custom modified version of Google Translate for characteristic usage. We have used this GToken key and studied and deployed a Translate API to take the recognition text as input and deploy an output which is extremely accurate given Google's consistency with Machine Translation.

CHAPTER 3

DESIGN OF YOUR WORK

3.1 VIEW OF FIGURES

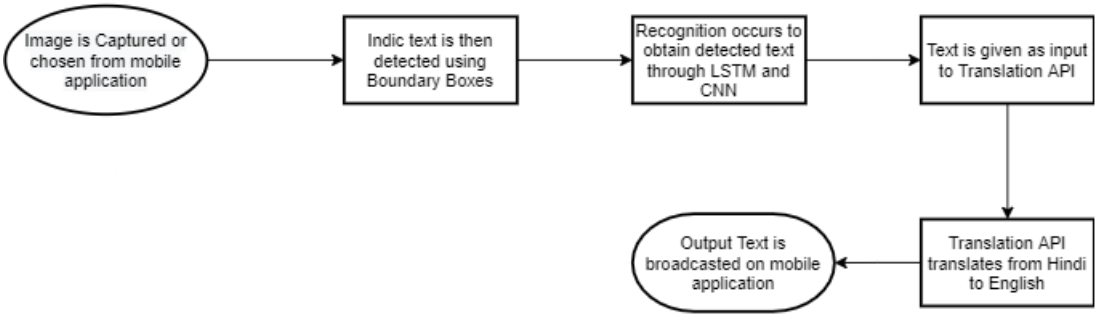

Figure 3.1: Flowchart of working model

```
translator = google_translator()
translations = {}
for column in df.columns:
    unique_elements = df[column].unique()
    for element in unique_elements:
       translations[element] = translator.translate(element)
translations
```

```
{'आदमी ': 'man ',
 'इंसान ': 'human ',
 'एक्सपर्ट्स': 'Experts ',
 'खिलाड़ी': 'Player ',
 'गरीब': 'Poor ',
 'चिकित्सक': 'Doctor ',
 'चिकित्सा विशेषज्ञ': 'medical specialist ',
 'जवान ': 'young ',
 'जीवनसाथी': 'Spouse ',
 'डाक्टर': 'Doctor ',
 'डेन्टिस्ट': 'Dentist ',
 'डॉक्टर': 'doctor ',
 'ततामह': 'Engrossed ',
 'दंत चिकित्सक': 'Dentists ',
 'दद्दा': 'Dadda ',
 'दादा': 'grandfather ',
 'दादा-दादी': 'Grandparents ',
 'दादी': 'grandmother ',
 'दुश्मन ': 'enemy ',
 'दोस्त ': 'friend ',
 'धावक': 'Runners ',
 'नवजात': 'Newborn ',
 'नाना': 'maternal grandfather ',
 'नाना-नानी': 'Maternal grandparents ',
 'नानी': 'Granny '
```

Figure 3.2: Output of Translation API

The Modules in the Project are : -

- Android Client application
- Detection Module
- Recognition Module
- Translation Module

As we notice from Figure 3.1, our model is designed with an interactive front end which provides the user an option to either capture an image on the

spot or choose a manual media option as the raw input for our process.
Each process performs its role and we obtain a highly accurate output as a result of the systematic processing of our raw input.

The project is deployed as a mobile application such that the user can capture an image and upload it from their mobile device onto a remote server which hosts the deep learning model. Furthermore, our application offers an interactive user interface along with a feedback system so as to enhance the user experience. The culmination of all the above mentioned makes our app unique and productive for users.

The application consists of a splash screen which opens and closes after 4 seconds i.e the latency between the time of clicking the application and it's actual opening. The user is then taken to the dashboard which gives an overall idea of our app and it's goal and purpose. A navigation drawer helps the user to switch between various activities depending upon their interest. In the "Upload and Translate" section, an image has to be uploaded from the SD card. A pop notification saying "1 Image(s) has been selected" can be seen. If user tries uploading image without selecting any image then correspondingly "No image has been selected" can be seen. The "Connect to server and Upload" button can be be pressed after the above mentioned steps. Finally, the image will be sent to server and the most probable answer will be returned.

On the server side, the image is parsed through detection, recognition modules and a translation API. The detection and recognition modules have already been trained separately by using Google Colab with a Nvidia Tesla 4 GPU (12 GB VRAM). The Translation API done using G-token and reverse engineering. The below attached screenshots of training phase of the modules will give a good insight into the training phase.

- Accuracy for detection = 84.1
- Accuracy for recognition = 78.8

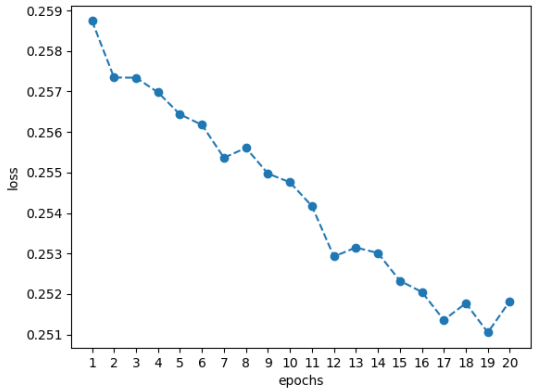

Figure 3.3: Training Phase(Detection)

Figure 3.4: Testing phase(Detection)

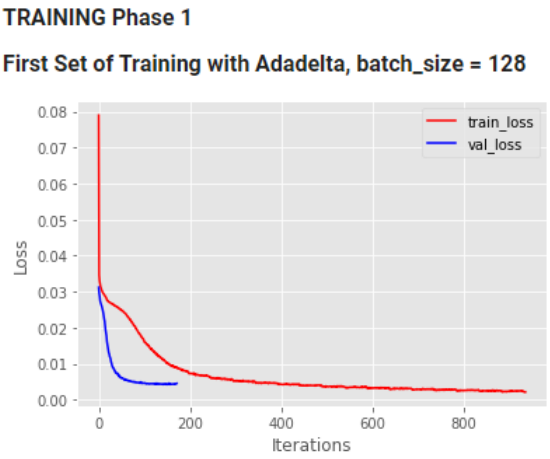

Figure 3.5: Training Phase(Recognition)

Figure 3.6: Testing phase(Recognition)

CHAPTER 4

IMPLEMENTATION OF YOUR WORK

Our functioning model works on step by step procedural execution of fundamental blocks which function our application.

4.1 DETECTION

For detection we use the CRAFT(Character Region Awareness for text Detection) algorithm.

The CRAFT algorithm localizes the individual character regions and link the detected characters to a text instance. CRAFT adopts a fully convolutional network architecture based on VGG-16 as its backbone. In simple words, VGG16 is essentially the feature extracting architecture that is used to encode the network's input into a certain feature representation. The decoding segment of the CRAFT network is similar to UNet. It has skip connections that aggregate low-level features. CRAFT predicts two scores for each character:

- Region Score: As the name suggests, it gives the region of the character. It localizes the character.
- Affinity Score: 'Affinity' is the degree to which a substance tends to combine with another. So, an affinity score merges characters into a single instance (a word).

Finally, the affinity and region scores are combined to give the bounding box of each word. The coordinates are in the order: (left-top), (right-top) (right-bottom), (left-bottom), where each coordinate is an (x, y) pair.

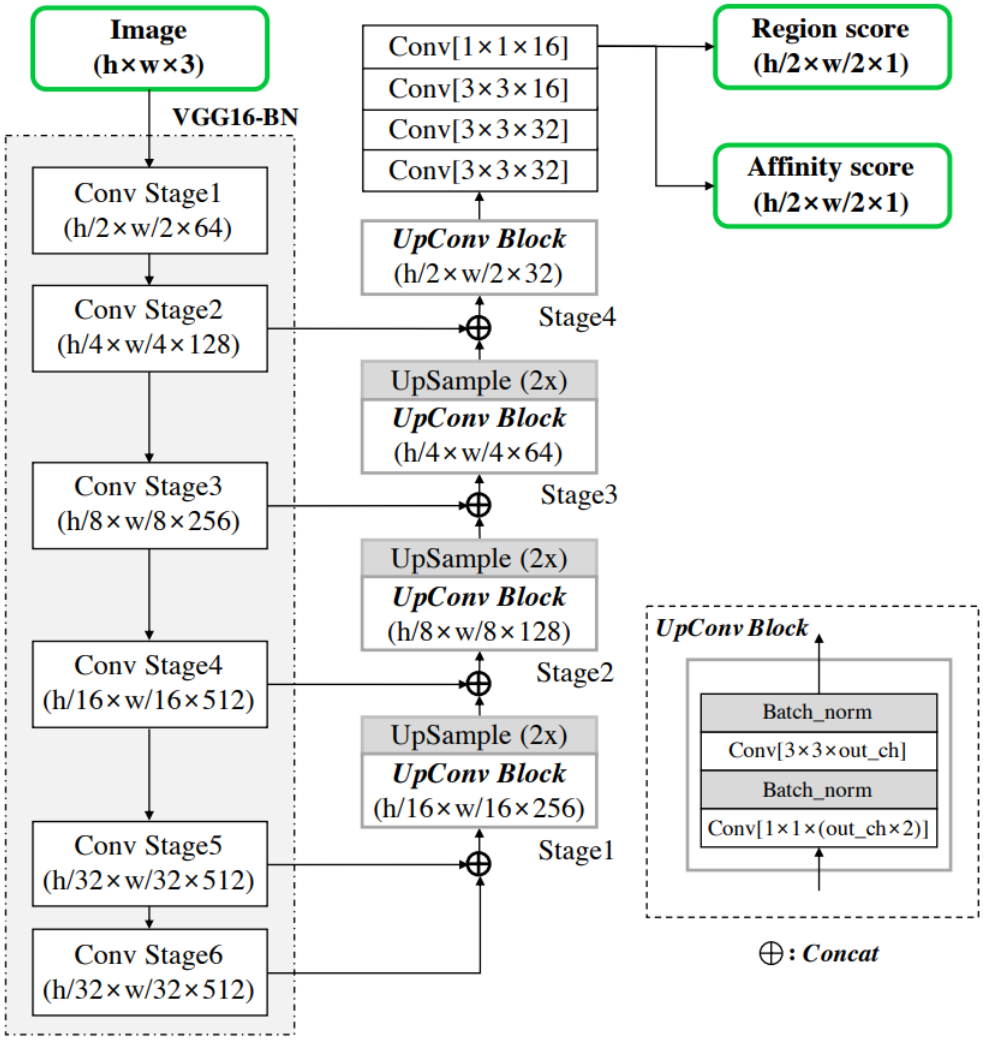

Figure 4.1: CRAFT Architecture

4.2 RECOGNITION

For the Recognition module, we use the CRNN(Convolutional Recurrent Nueral Network) architecture. The CRNN architecture consists of ResNet for feature extraction, a Bi-directional LSTM to remember the context of the preceding and succeeding words and finally CTC function for loss calculation and decoding output.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 2, 256, 40)	0
conv2d_1 (Conv2D)	(None, 128, 256, 40)	1152
batch_normalization_1 (Batch	(None, 128, 256, 40)	512
activation_1 (Activation)	(None, 128, 256, 40)	0
max_pooling2d_1 (MaxPooling2	(None, 128, 256, 8)	0
dropout_1 (Dropout)	(None, 128, 256, 8)	0
conv2d_2 (Conv2D)	(None, 128, 256, 8)	65664
batch_normalization_2 (Batch	(None, 128, 256, 8)	512
activation_2 (Activation)	(None, 128, 256, 8)	0
max_pooling2d_2 (MaxPooling2	(None, 128, 256, 4)	0
dropout_2 (Dropout)	(None, 128, 256, 4)	0
conv2d_3 (Conv2D)	(None, 128, 256, 4)	65664
batch_normalization_3 (Batch	(None, 128, 256, 4)	512
activation_3 (Activation)	(None, 128, 256, 4)	0
max_pooling2d_3 (MaxPooling2	(None, 128, 256, 2)	0
dropout_3 (Dropout)	(None, 128, 256, 2)	0
permute_1 (Permute)	(None, 256, 128, 2)	0
reshape_1 (Reshape)	(None, 256, 256)	0
bidirectional_1 (Bidirection	(None, 256, 64)	55488
bidirectional_2 (Bidirection	(None, 256, 64)	18624
time_distributed_1 (TimeDist	(None, 256, 32)	2080
dropout_4 (Dropout)	(None, 256, 32)	0
time_distributed_2 (TimeDist	(None, 256, 10)	330
strong_out (Activation)	(None, 256, 10)	0

Total params: 210,538
Trainable params: 209,770
Non-trainable params: 768

Figure 4.2: CRNN Architecture

4.2.1 ResNet (Residual Network Architecture)

There were several feature learning networks before ResNet such as AlexNet and VGGNet. However, the inherent problem they all faced was that

whenever a deep learning module has too many layers then : -

- The network has to learn too many parameters.
- The vanishing gradient problem which occurs as a side-effect due to repeated multiplication of relatively smaller gradients as we pass through each layer.

To minimize these, Microsoft proposed the ResNet architecture which can overcome these difficulties. The ResNet architecture exploits the idea of shortcuts i.e layers of CNN can be skipped as required so that the network doesn't get congested resulting in over fitting and considerably reduce vanishing gradient descent problem. Below is the diagram of a ResNet network.

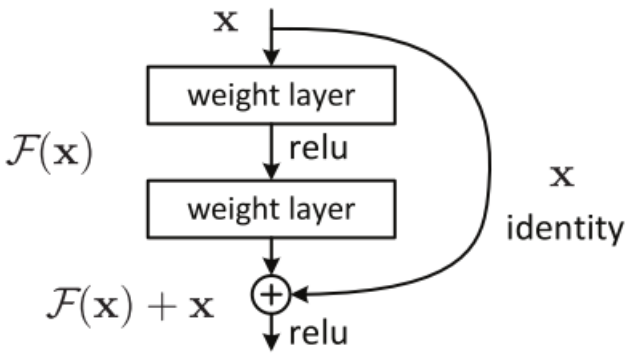

Figure 4.3: ResNet Architecture

4.2.2 Bi-Directional LSTM

LSTM in its core, preserves information from inputs that has already passed through it using the hidden state. Unidirectional LSTM only preserves information of the past because the only inputs it has seen are from the past.

Using bidirectional will run your inputs in two ways, one from past to future and one from future to past and what differs this approach from unidirectional is that in the LSTM that runs backwards you preserve information from the future and using the two hidden states combined you are able in any point in time to preserve information from both past and future.

The bi-directional LSTM diagram below will give a good idea as to how the concept works

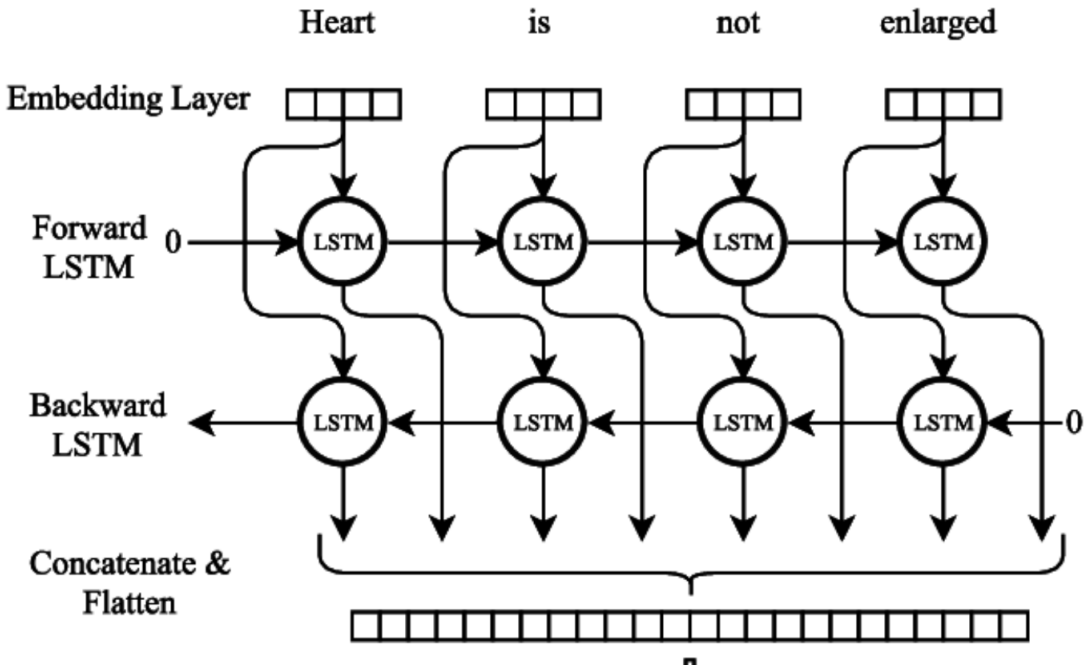

Figure 4.4: Bi-directional LSTM architecture

4.2.3 CTC Loss Function

CTC(Connectionist Temporal Classification) is a loss function used to calculate the alignment of text in a given image. Instead of having the annotations of each and every text in image(cumbersome process), we can rather tell the model to learn the proper alignment by itself by just giving the input and

output of the image. By calculating the score we get by aligning all possible outcomes sequentially in a given timestep, the model can predict the output by choosing the one with highest probable score.

This loss function is used extensively in the field of NLP to predict sequence of texts occurring in any form. Thus, compared to a naive NN loss funtion like cross entropy or mean squared loss, CTC proves to be more efficient.

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

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

Figure 4.5: CTC loss function formula

4.3 TRANSLATION

Scheduled input text is obtained through detection and recognition modules and prepared to be received as an input to our Translation API. This API is developed to use the Google Translate Ajax API with python encoded outputs to make calls to detect and translate. This comes with a combination of python codes to receive input and to detect language in the input and also to translate the input Indic text through reverse engineering a GToken key. This GToken key is provided by Google Cloud as an open source platform available to use for everyone to obtain a translated desirable outcome

4.4 MOBILE APPLICATION

The mobile application in our project acts at the client side by enabling the user to upload images and receive appropriate responses. We used the

Android Studio IDE, developed by IntelliJ. Our application can work on devices whose android version is higher than 9.0. The languages used are Java, Kotlin.

The introductory splash screen appears for about a brief period of 4 seconds i.e the latency between start of the application and then the dashboard open up. The dashboard contains slider screens implemented using liquid drawer. The liquid drawer provides fluid motion while sliding from one screen to another. Then a navigation bar also appears on the side which contains other activities and can be chosen at the behest of the user. They include ”Famous monuments” , ”Upload and translate”, ”Help Section” and ”About us”

- Famous Monuments : India is a country which has been involved in the world’s history for centuries. Many famous monuments like Taj mahal, Qutub Minar, Ajanta and Ellora cave paintings showcase the influence of variopus cultures of Indian’s ancestors. A sliding screen is implemented with ”Ken Burns” view which is the technique wherein the still image is given a motion affect for all round visuals. It’s used extensively in the field of photography.
- Upload and Translate : The user uploads the necessary images for which they require translation. After waiting for a few seconds the answer is relayed to the users from the server and broadcast on the android app.
- Help Section : The help sections contains visual imagery guiding the users about how to upload images onto the server.
- About Us : This page explains the goals of our team. We have also provided our e-mail address to contact us in case they want to give any feedback or to fix any issues if they mail a complaint.

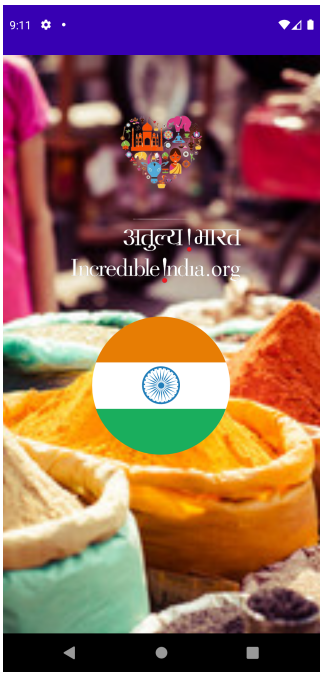

Figure 4.6: Splash Screen

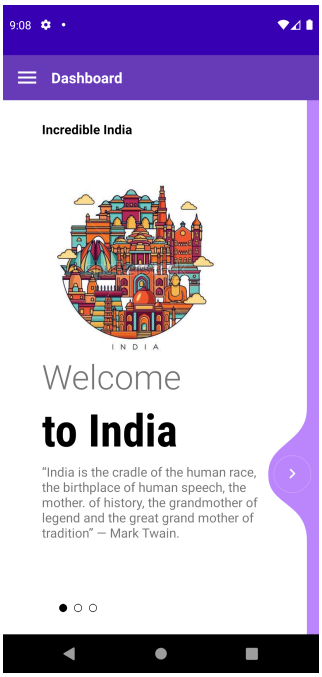

Figure 4.7: Dashboard(1)

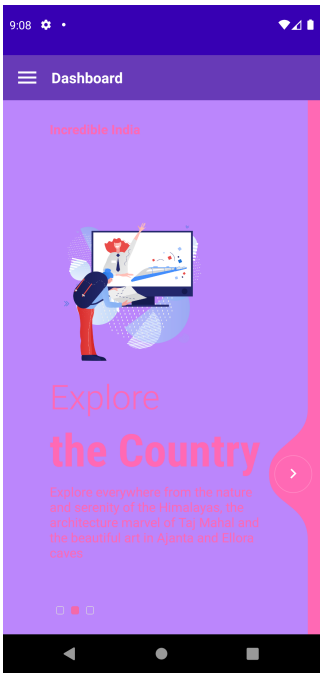

Figure 4.8: Dashboard(2)

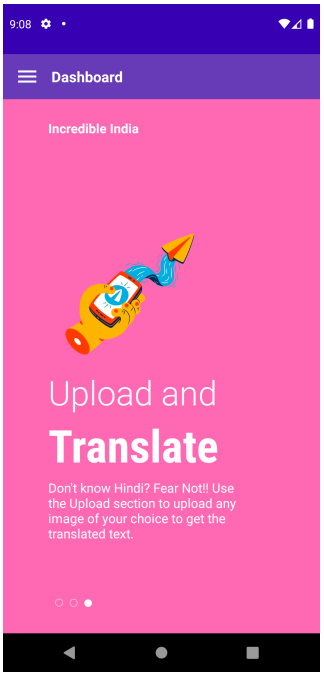

Figure 4.9: Dashboard

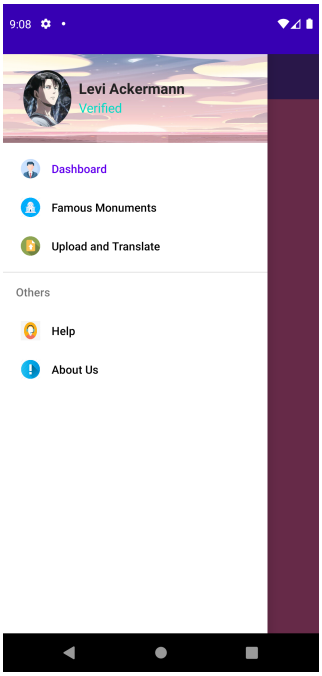

Figure 4.10: Navigation

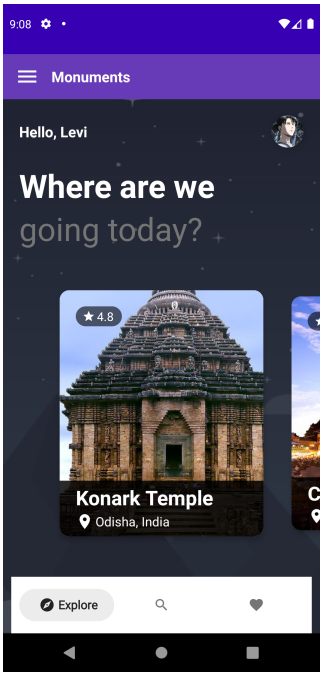

Figure 4.11: Monuments

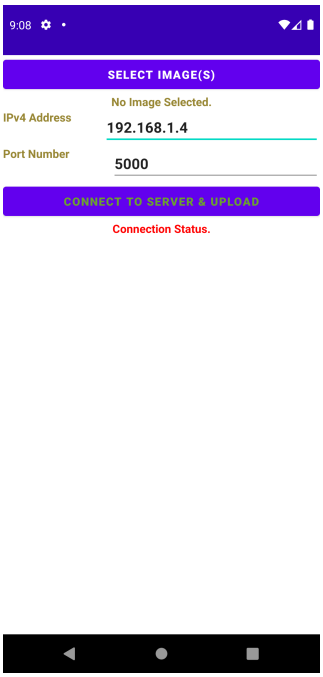

Figure 4.12: Upload

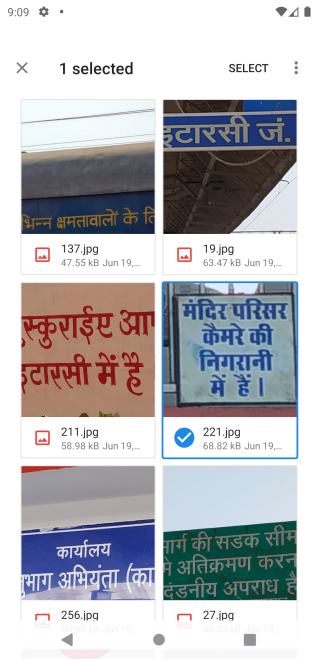

Figure 4.13: Choose image

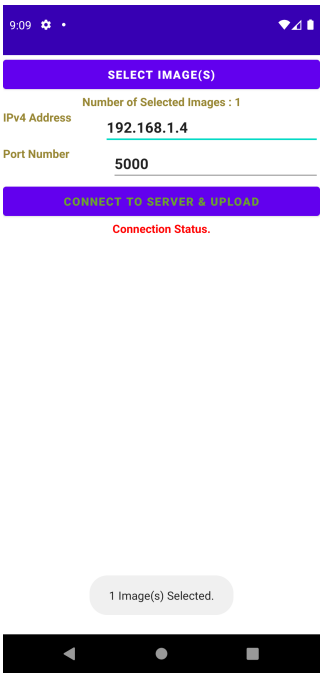

Figure 4.14: Notification

Figure 4.15: Output

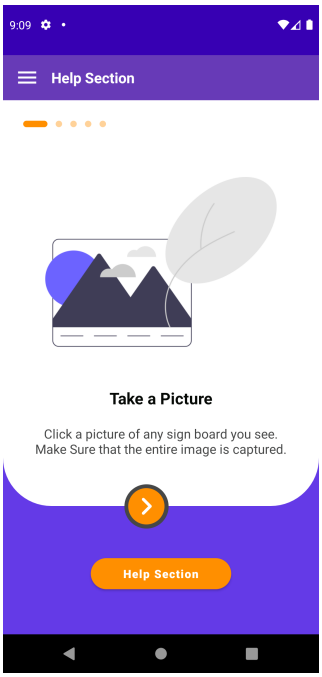

Figure 4.16: Help

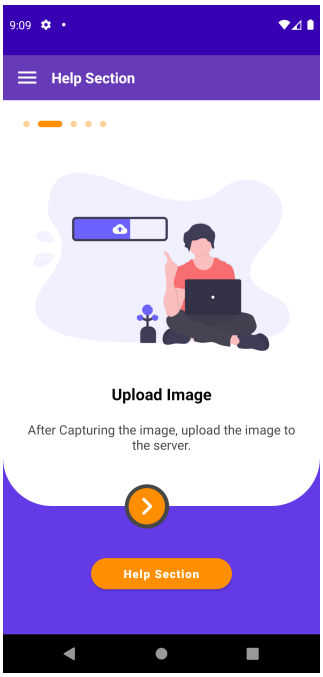

Figure 4.17: Help

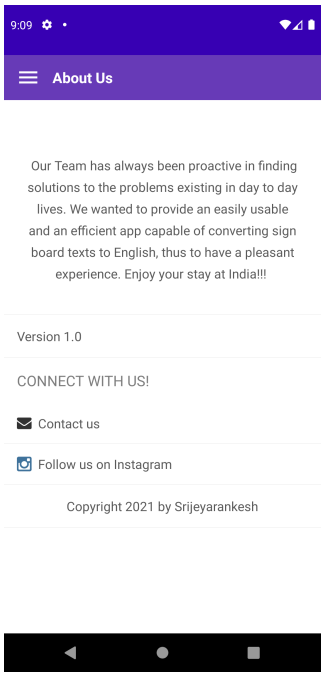

Figure 4.18: About Us

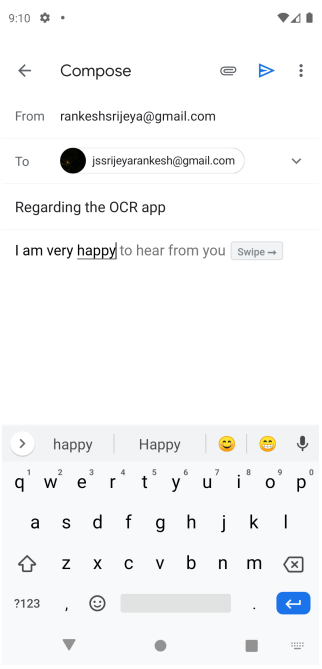

Figure 4.19: Feedback

CHAPTER 5

CONCLUSION

The objective of this work from the beginning is to facilitate the tourists by providing an application that will translate the native language into the international language. The goal has been achieved. The application allows the user to insert a picture and then Machine Learning techniques are used to detect the text area from the image, the text is extracted from the detected module. The text is processed with the reverse translation API using google translator. The result will be shown to the user. The application can also be used as a translation app wherever required.

REFERENCES

[1] K Alishahi, F Marvasti, V A Aref, and P Pad. Bounds on the sum capacity of synchronous binary cdma channels. *Journal of Chemical Education*, 55:3577–3593, 2009.

[2] T G Conley and D W Galeson. Nativity and wealth in mid-nineteenth century. *Journal of Economic History*, 58:468–493, 1998.

[3] S Waldron. Generalized welch bound equality sequences are tight frames. *IEEE Transactions on Information Theory*, 49:2017–2309, 2008.

[4] Richard E Fikes and Nils J Nilsson. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. In *Proceedings of the 2Nd International Joint Conference on Artificial Intelligence*, IJCAI'71, pages 608–620, 1971.

[5] Weiguo Fan, Michael D Gordon, and Praveen Pathak. Personalization of Search Engine Services for Effective Retrieval and Knowledge Management. In *Proceedings of the Twenty First International Conference on Information Systems*, ICIS '00, pages 20–34, 2000.

[6] Anna University PhD-Regulations 2015. `https://cfr.annauniv.edu/research/regulation/PhD-Regualtion-2015.pdf`. Accessed: 20 March 2015.

[7] D H Holt. *Management Principles and Practices*. Prentice-Hall, Sydney, 1997.

[8] Philippe Aghion and Steven Durlauf, editors. *Handbook of Economic Growth*, volume 1. Elsevier, 1 edition, 2005.

[9] Dan Riley. *Industrial relations in Australian education / edited by Dan Riley*. Social Science Press [Wentworth Falls, N.S.W.], 1992.

[10] A H Cookson. Particle trap for compressed gas insulated transmission systems, 1985. US Patent 4554399.

[11] J P Hos. *Mechanochemically synthesized nanomaterials for intermediate temperature solid oxide fuel cell membranes*. PhD thesis, University of Western Australia, 2005.

[12] J Ionesco. Federal Election: New Chip in Politics. *The Advertiser*, page 10, 2010.