A
Project Report on

Speech Translator With OCR Reader

By

Nai Om Rajeshkumar (CE072) (20CEUBS141)
&
Parikh Vedant Jasminkumar (CE080) (20CEUOG024)

B. Tech CE Semester-VI
Subject: Software Development Practices

Guided by:
Dr. Brijesh S. Bhatt
Professor
Computer Engineering Department

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University, Nadiad

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University, Nadiad

CERTIFICATE

This is to certify that the project work carried out
in the subject of
Software Development Practices
is the bonafide work of

Nai Om Rajeshkumar (CE072) (20CEUBS141)
&
Parikh Vedant Jasminkumar (CE080) (20CEUOG024)

of B. Tech Semester **VI** Computer Engineering during the academic year **2022 - 2023**.

Dr. Brijesh S. Bhatt
Professor
Computer Engineering Department
Faculty of Technology
Dharmsinh Desai University, Nadiad

Dr. C. K. Bhensdadia
Professor & HoD
Computer Engineering Department
Faculty of Technology
Dharmsinh Desai University, Nadiad

Contents

Abstract

- Speech translator provides the flexibility to translate English language text to Gujarati language and vice versa.
- Also, it provides the set of functionalities to give voice as an input and translate to the targeted language.
- Thus, it provides a bridge among the people of two different dialects.
- On addition to that this also have a feature of OCR – Reader which translate the English text to Gujrati just by providing the image as an input.
- Thus, it helps the people to connect with other.

Introduction

About Project :-

Our main goal behind this project is to provide just a service which can serve day-to-day need of any person who is willing to communicate but cannot only because of the language problem. Also, tourist from the various country need to communicate with people with local dialect. So, for that purpose we provide “Speech Translator” helps people to communicate and get connect to them and it will help them to get translation of ENGLISH language into GUJARATI language and vice versa.

Also, in addition to that we have feature of OCR-Reader which just takes an image and translate the text into ENGLISH text to GUJARATI text which can also provide the flexibility of text reading.

Technology Used :-

- Languages and Module:
    - Python programming language
    - Google translate (version :3.0)
    - Open CV
    - Play Sound
    - GTTS
    - Tesseract
    - PyTesseract
- Version Control:
    - Git
    - GitHub
- Development Environment (Tools Used):
    - Google Collaboratory
    - PyCharm (Community Edition)
    - Microsoft Visual Studio Code

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

Speech Translator With OCR Reader

Functional Requirements

R 1: User Side

Description: The main interactive and minimalistic landing page which provides the functionalities of translating text and speech from ENGLISH to GUJRATI and vice a versa. Also, by providing the image with ENGLISH text it translates to the GUJRATI language.

R 1.1: Select the input language

Input: User selects input language ENGLISH or GUJARATI
Process: Selection.
Output: Selected language is going to take inputs and remaining is the language where we want to get translation.

R 1.2: Provide the speech as an input text

Input: User enters or dictates input (via button "Click to Speak") in selected language ENGLISH or GUJARATI.
Output: All text format of dictation is printed in text box and ready to get translated.

R 1.3: Click to hear text generated for output lang

Input: After translation of our provided input text, we get translated text in text box and user must click over the button of "Click to Listen".
Output: We can clearly listen the dictation of our translated output text.

R 1.4: Upload English text input image

Input: On user(admin) click of delete button.
Output: The selected student will be deleted from the system.

R 1.5: Input the text

Input: Users must upload a picture containing ENGLISH text to get extracted text and its translation in GUJARATI.
Output: Text is extracted from the image and translation is also printed.

Non-Functional Requirements:

1. Reliability

The ability of the system to behave consistently in a user-acceptable manner when operating within the environment for which the system was intended.

2. Maintainability

Easy to maintain as if minimalistic UI.

3. Security

Secure access of confidential data (User Information).

4. User friendly

System should be easily used by the user with minimalistic UI.

5. Performance

Performance should be fast.

6. Efficient

System should be efficient for the managing notes.

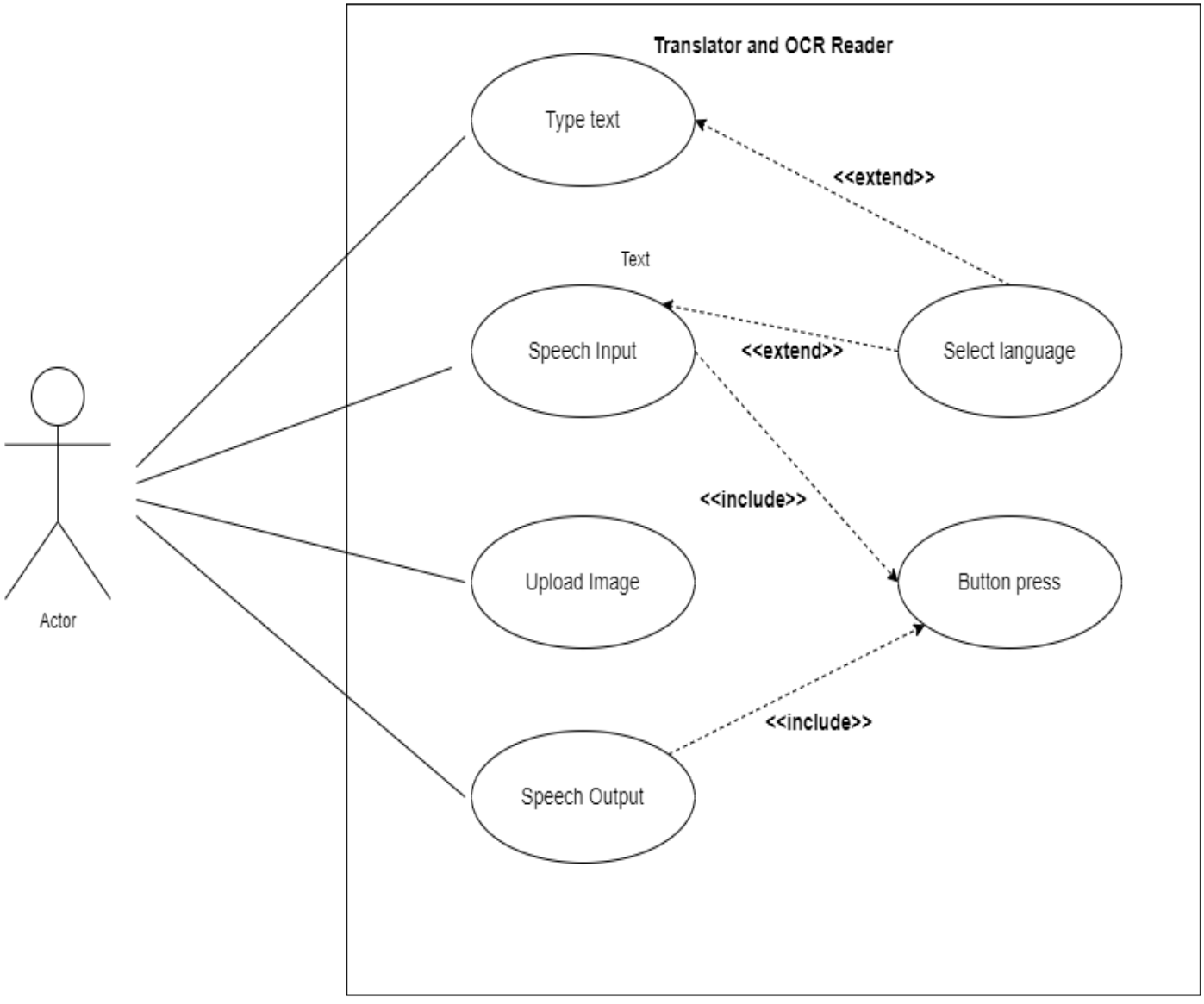
7. Privacy

Personal data of the system should not disclose to anyone.

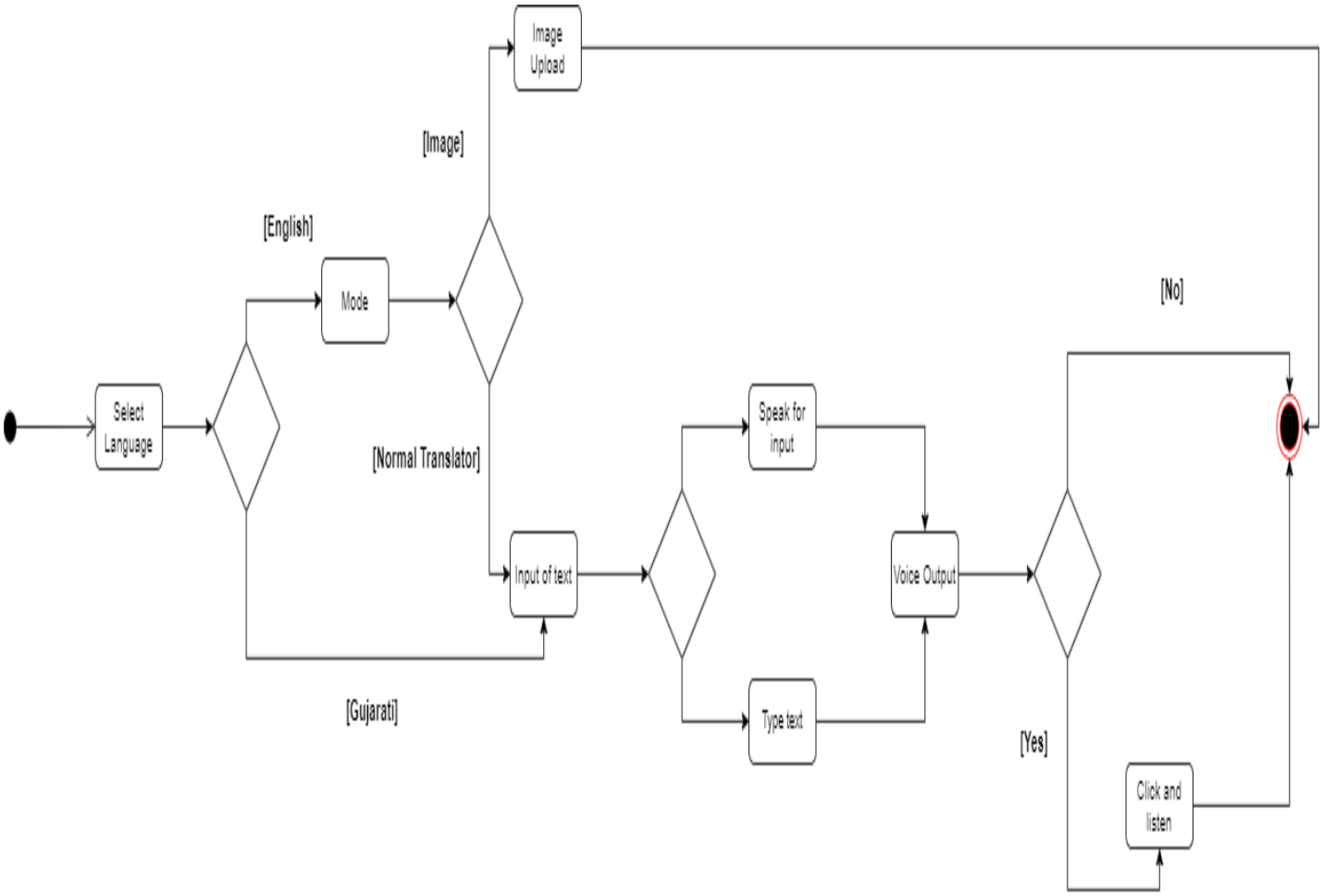
End Of SRS

DESIGN

Use Case Diagram



Activity Diagram



Implementation Details

Modules created and Brief Description of each module: -

This project consists of **4** *major set of basic functionalities*.

1) Select the Input Language

- User is allowed to select the input text language.
- On the selection of the user input language the corresponding output language is changed.
- The text will be served over to the Google Translate model which will detect the language and corresponding output language is served as the translated text.

2) Voice Input

- For that click on the button of "Click to Speak."
- Give Speech to translate it in Input Language.
- After dictation it will print all text in that text box and ready to translate it so only you must click on the "Translate" button.

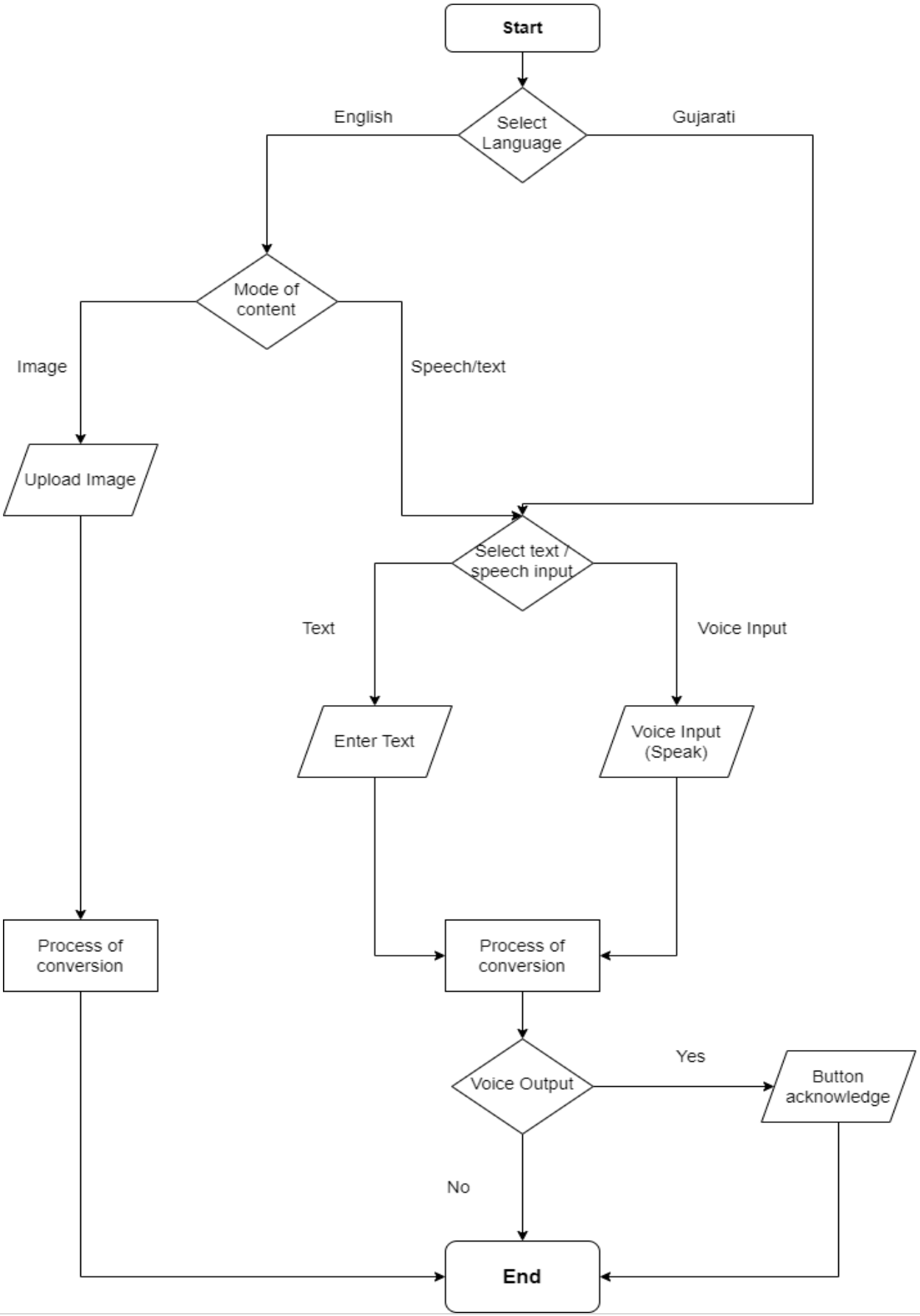
3) Voice Output

- Translated text is printed over the text box and if we want to listen that translated text all, we just need to do is click on the "Click to Listen" button.
- After that all dictation of that translated text is going to play.

4) Upload the File

- Click on the button "Choose File" to upload the picture containing ENGLISH text to get its extracted text and its translation in GUJARATI.
- After uploading just press on "Submit" button and it will redirect you to another webpage where extracted text and its translated text is printed of the text provided in the uploaded picture.

Flow Chart :-


Start
Select Language
English
Gujarati
Mode of content
Image
Speech/text
Upload Image
Select text / speech input
Text
Voice Input
Enter Text
Voice Input (Speak)
Process of conversion
Process of conversion
Voice Output
Yes
Button acknowledge
No
End

Algorithm and Logic Implemented: -

a. Normal language conversion.

- It uses the large set of neural networks for the conversion of the language.
- Here Google Trans python module (version :3.0) is used for processing the language.
- Automatic source language detectors are used to identify the source input language.
- For the purpose of language translation Encoders and Decoders are used which is the integral part of Natural Language Processing (NLP).

b. Speech as an input text.

- Here user needs to click the button which uses the browser embedded functionalities to capture the voice of the user.
- Browser will ask for the permission of the microphone access to the user.
- This is totally handled by the java script and the browser provided functionalities.

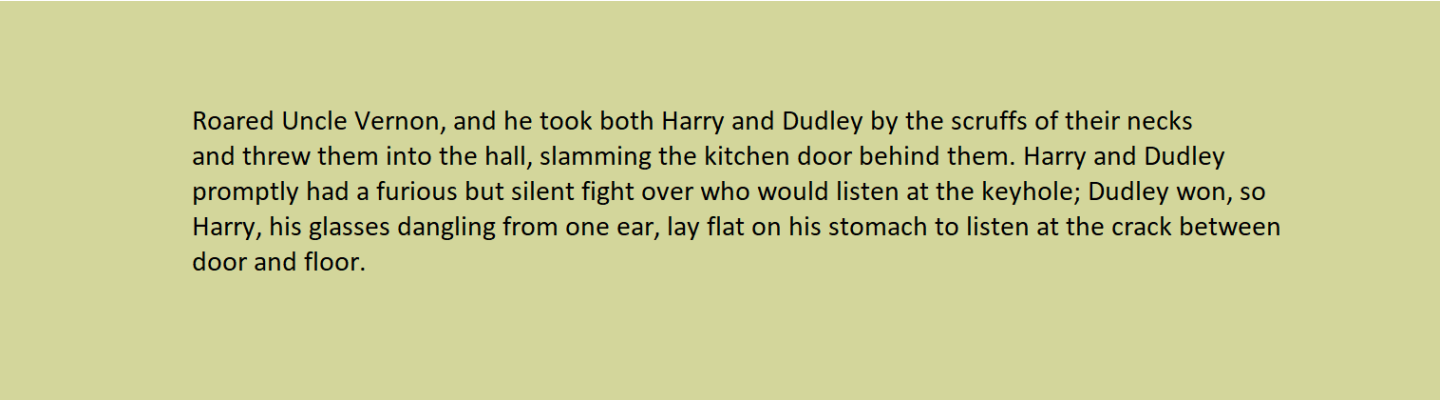
c. OCR – Reader

- Here user will provide the input image to the system.
- This is limited to converting source language **English to targeted Gujarati language only**.
- The OCR – function works in various stages.

    1. Binarizing the image.
    2. Noise removal from the image.
    3. Removing the border.
    4. Extracting the text.

- Brief of each stage is provided below.

- For the consideration the provided image is as under.
- Provided image contains the yellowish background and the text.



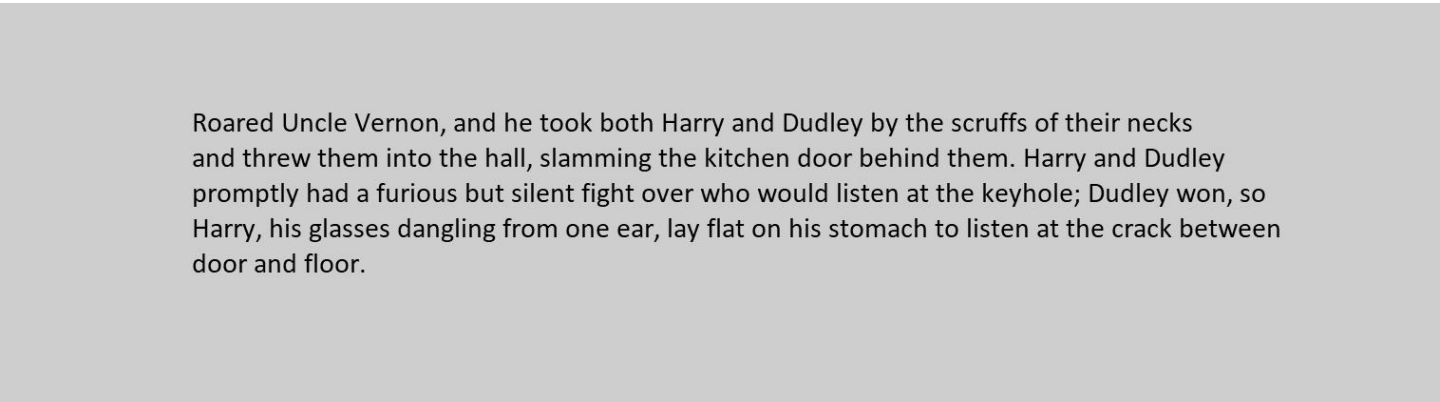
(Fig - 1)

1. **<u>Binarizing the image.</u>**

- Now as the provided image contains large set of the input information, so for the better result and for efficient processing it is binarized.
- This is the combination of two step processes.

I. Converting To grey scale (Using Open CV).

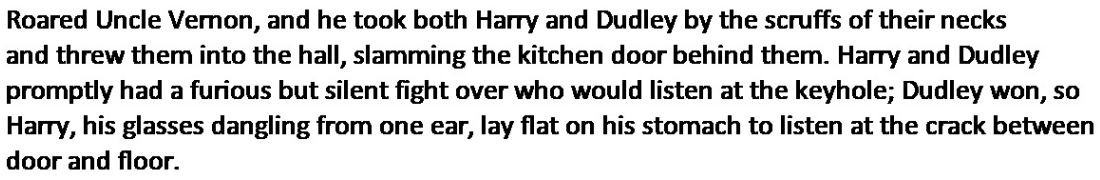
- Grayscale is a pre-processing layer that transforms RGB images to Grayscale images.
- Input images should have values in the range of [0, 255].
- This is done because the grey scaled images are compressed to minimal pixels.
- Thus, it enhances visualization.
- Following is the output generated.
- The background is scaled to grey.



(Fig - 2)

I. Converting To Black & White (Using Open CV).

- This transformation is useful in detecting blobs and further reduces the computational complexity.
- Threshold for every pixel is applied.
- The function thresholds works on the following mathematical imputation, dst(x,y) = { (maxValue, if src(x,y)>thresh), (0, otherwise)}.
- Thresholding is simple having two values of white or black.
- THRESH_BINARY is using the simple thresholding of an image.
- The image is converted to black and white.



(Fig - 3)

2. Noise Removal from the image.

- This the important part of the image processing.
- As if image contains some useless data because of some external factors and some generated due to internal pre-processing stage.
- Also, some data might data also be lost.
- So, for that purpose it is done.
- **Kernal** is set here for the certain pixel range to work upon.

I. Dilution Stage (Using Open CV).

- It increases the thickness of the characters based on the morphological criteria (neighbourhood) provided the given kernel.
- The erosion operation is: dst(x,y)=max(x′,y′):element(x′,y′)≠0src(x+x′,y+y′)

(Fig - 4)

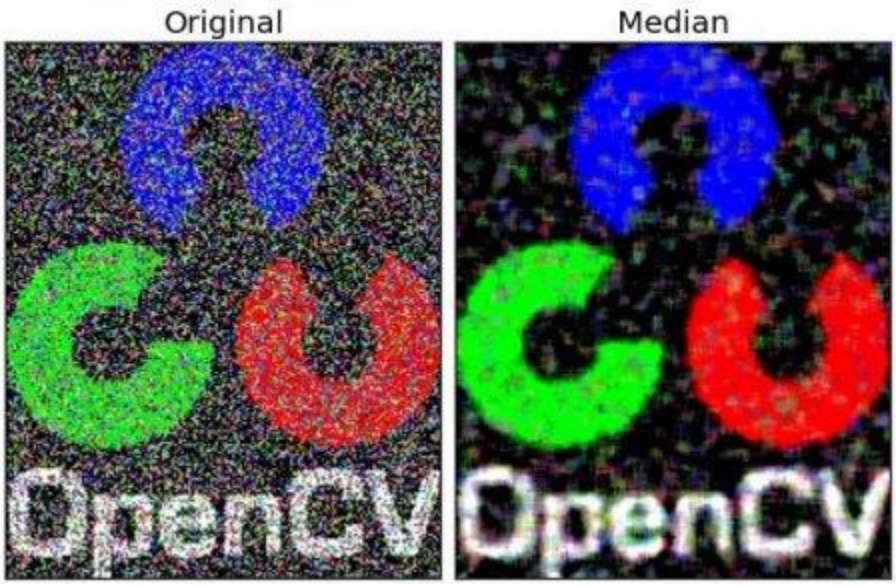
II. Erosion (Using Open CV)

- It decreases the thickness of the characters based on the morphological criteria provided the given kernel.
- The erosion operation is: dst(x,y)=min(x′,y′):element(x′,y′)≠0src(x+x′,y+y′)

(Fig - 5)

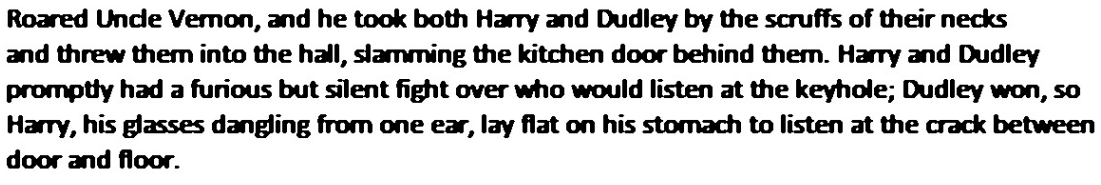
III. Median Blur (Using OpenCV)

- Blurring the image to the given scale.
- Here, the input is passed and them the median of all the pixels under the kernel area and the central element is replaced with this median value. This is highly effective against salt-and-pepper noise in an image. Interestingly, in the above filters, the central element is a newly calculated value which may be a pixel value in the image or a new value. But in median blurring, the central element is always replaced by some pixel value in the image. It reduces the noise effectively. Its kernel size should be a positive odd integer



(Fig - 6)

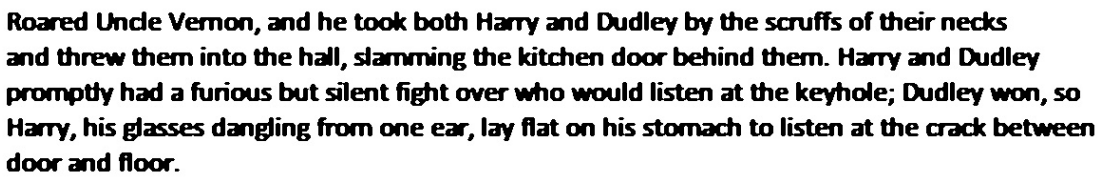
- The final generated output after going through all the function is as under.



(Fig - 7)

3. Remove borders from the image.

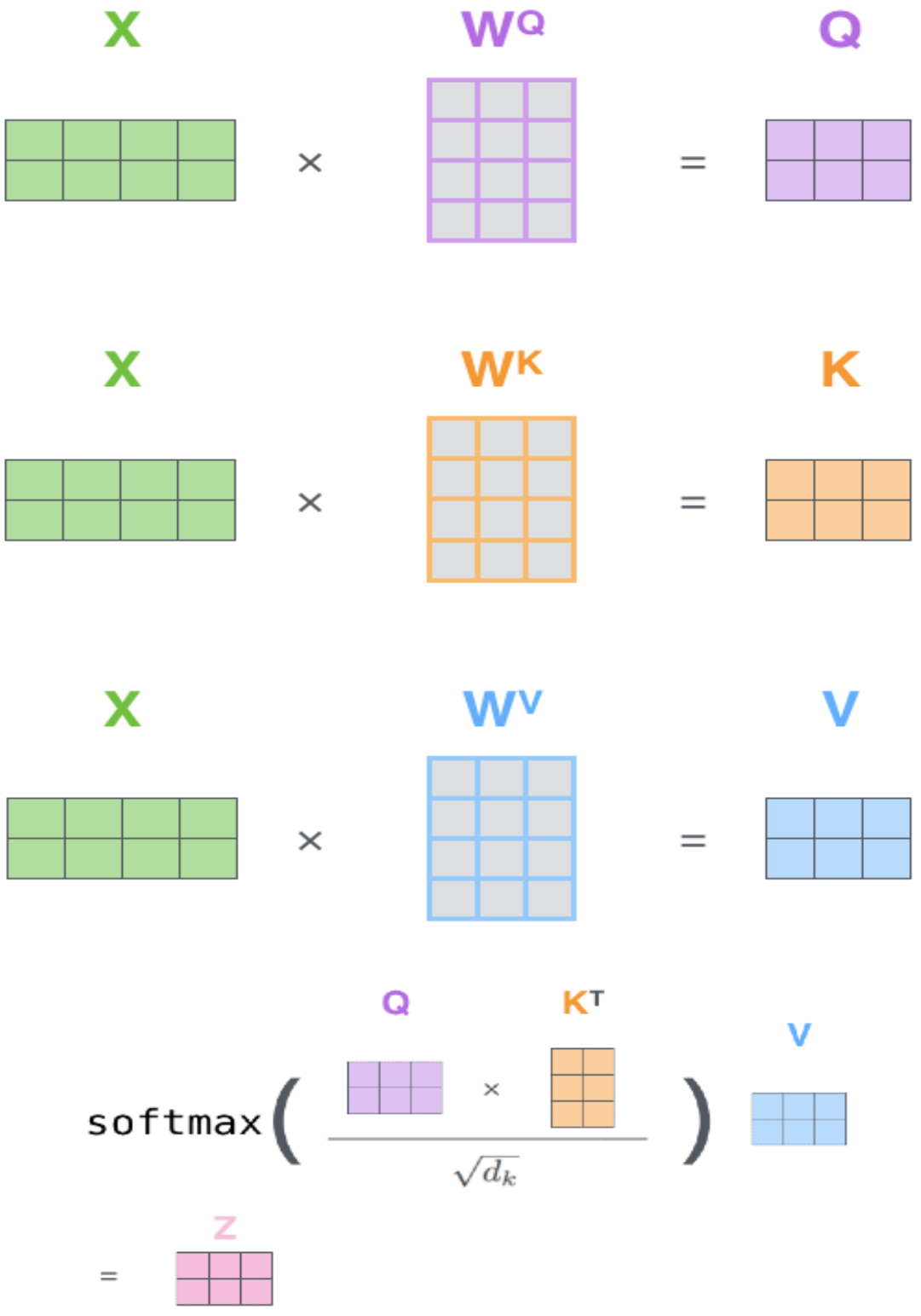
- Remove the image borders in order to allow more accurate OCR detection.
- The output generated is as under.



(Fig - 8)

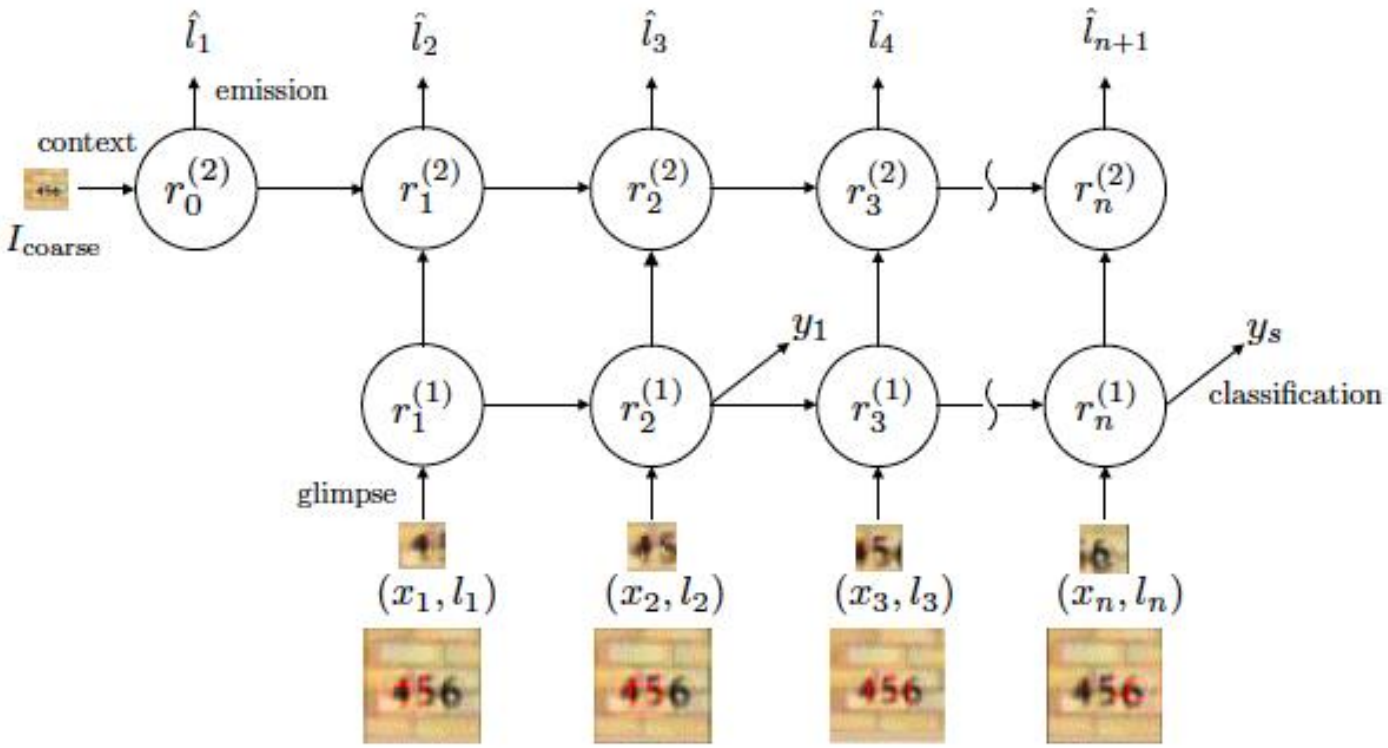
4. Extracting the text (Tesseract and PyTesseract).

- Now, the image is pre-processing is done.
- Now it is passed to PyTesseract module which does the job of text extraction.
- In a nutshell, attention is a feed-forward layer with trainable weights that help us capture the relationships between different elements of sequences. It works by using query, key and value matrices, passing the input embeddings through a series of operations and getting an encoded representation of our original input sequence.



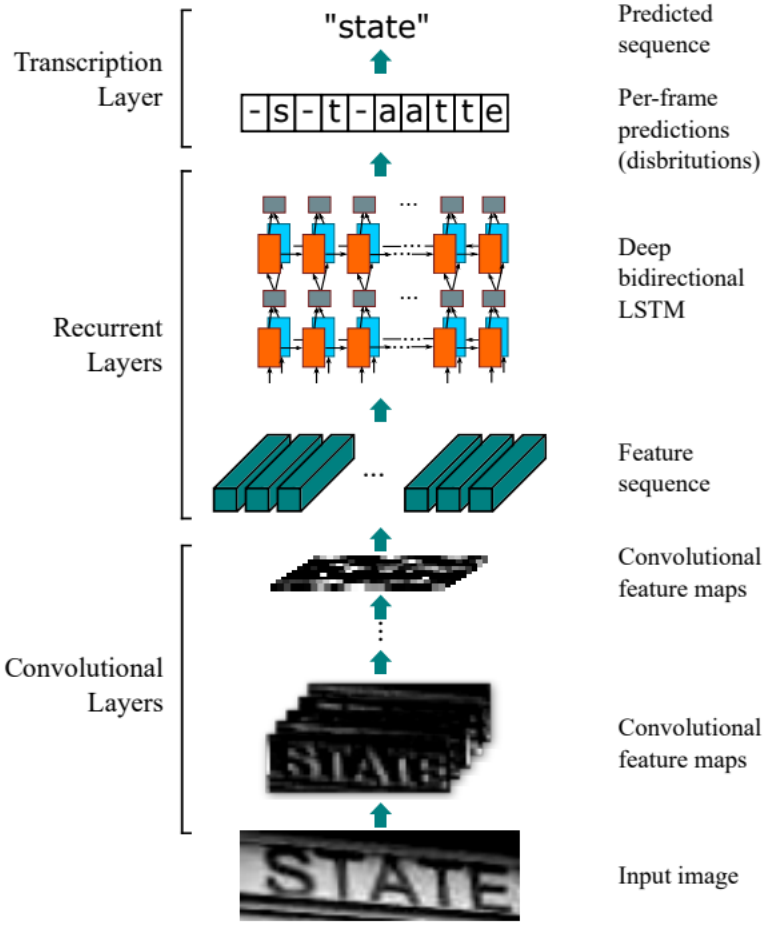
(Fig - 9)

- A location RNN to predict the next glimpse location and another Classification RNN dedicated to predicting the class labels or guess which character is it we are looking at in the text. A context network is used to down sample image inputs for more generalizable RNN states. It also chooses to refer to the location network in RAM as Emission Network. The training is done using an accumulated reward and optimizing the sequence log-likelihood loss function using the REINFORCE policy gradient.



(Fig - 10)

- CRNNs don't treat our OCR task as a reinforcement learning problem but as a machine learning problem with a custom loss. The loss used is called CTC loss - Connectionist Temporal Classification. The convolutional layers are used as feature extractors that pass these features to the recurrent layers - bi-directional LSTMs . These are followed by a transcription layer that uses a probabilistic approach to decode our LSTM outputs. Each frame generated by the LSTM is decoded into a character and these characters are fed into a final decoder/transcription layer which will output the final predicted sequence.
- Then it passed to normal translator function.


Transcription Layer
"state"
Predicted sequence
-s-t-aatte
Per-frame predictions (disbritutions)
Recurrent Layers
Deep bidirectional LSTM
Feature sequence
Convolutional Layers
Convolutional feature maps
Convolutional feature maps
STATE
Input image

(Fig - 11)

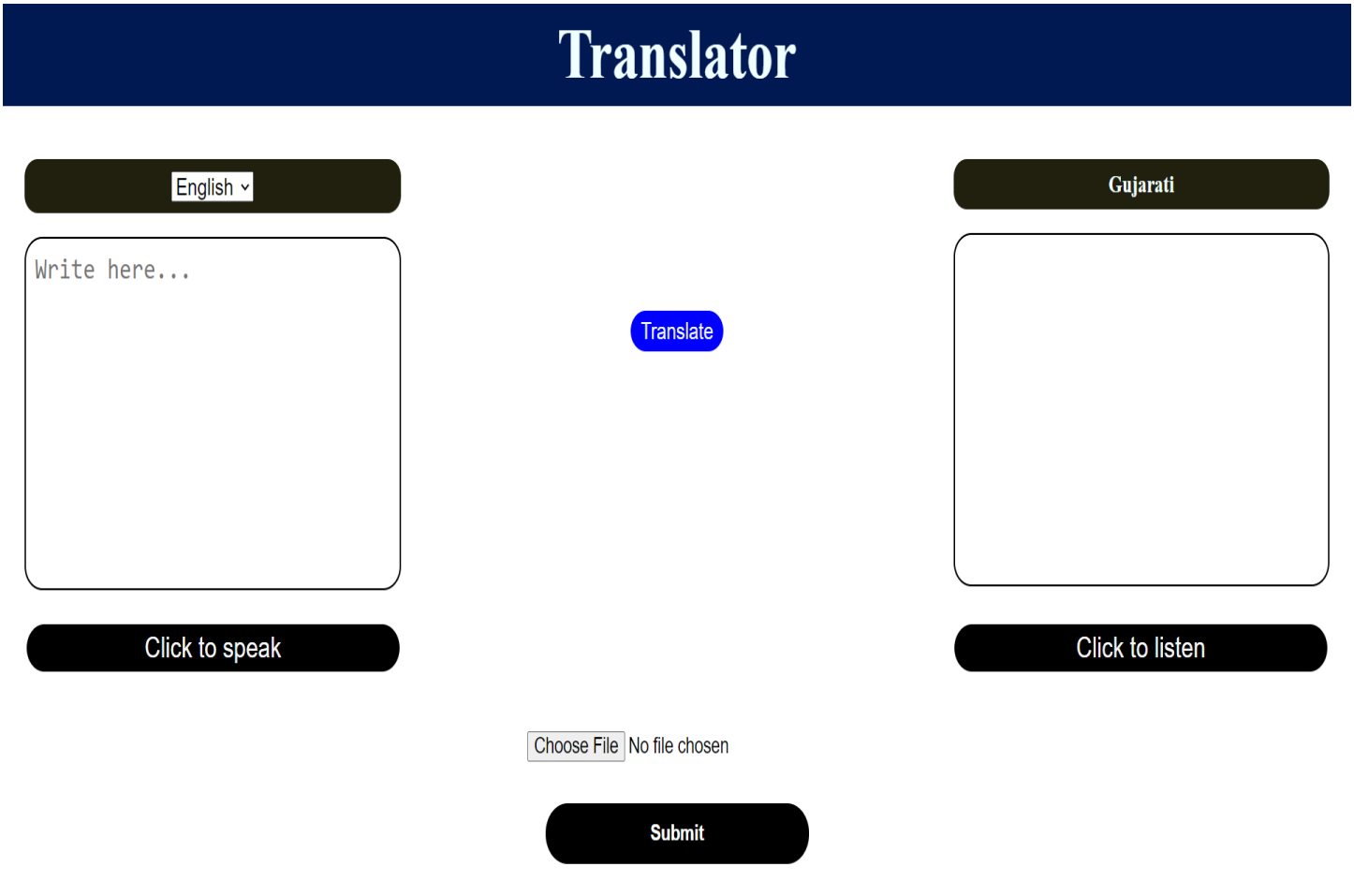
Testing

Test Case Id	Test Case Objective	Input Data	Expected Output	Actual Output	Status
TC_01	Select target input language.	User selects input language.	The targeted language is selected.	Language selection is done.	**Pass**
TC_02	Input to selected language.	Input is taken.	Selected target language text is displayed.	The output in selected language is displayed.	**Pass**
TC_03	Voice input.	Select the input language. Allow the microphone access.	Voice input to be successful.	Output in selected target output language.	**Pass**
TC_04	Click to play voice of text generated.	User Click.	Selected target output language voice output.	Targeted output language voice output.	**Pass**
TC_05	Upload English text image file.	File input.	Original language text(Eng) and translated language text(Guj).	Route to another page and display texts.	**Pass**

(Table - 1)

Screenshots

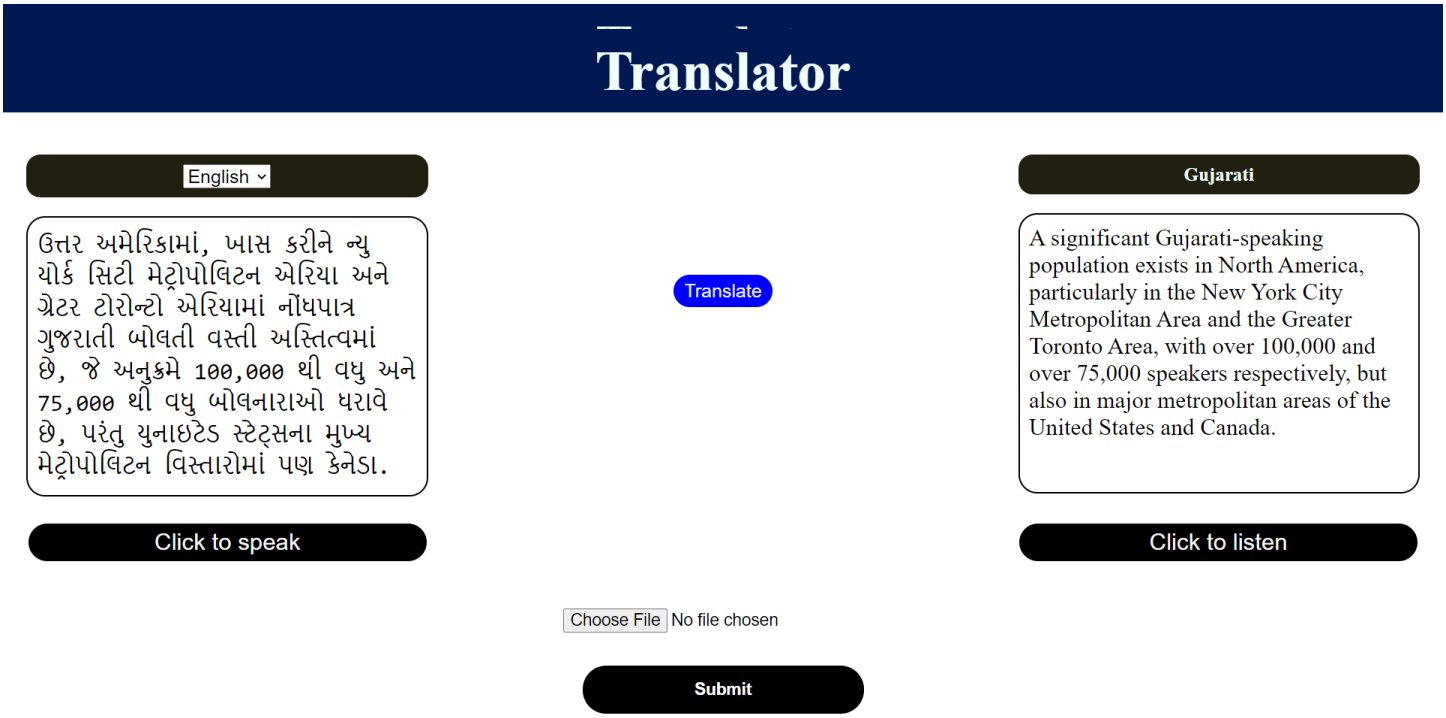
❖ **Landing Screen**



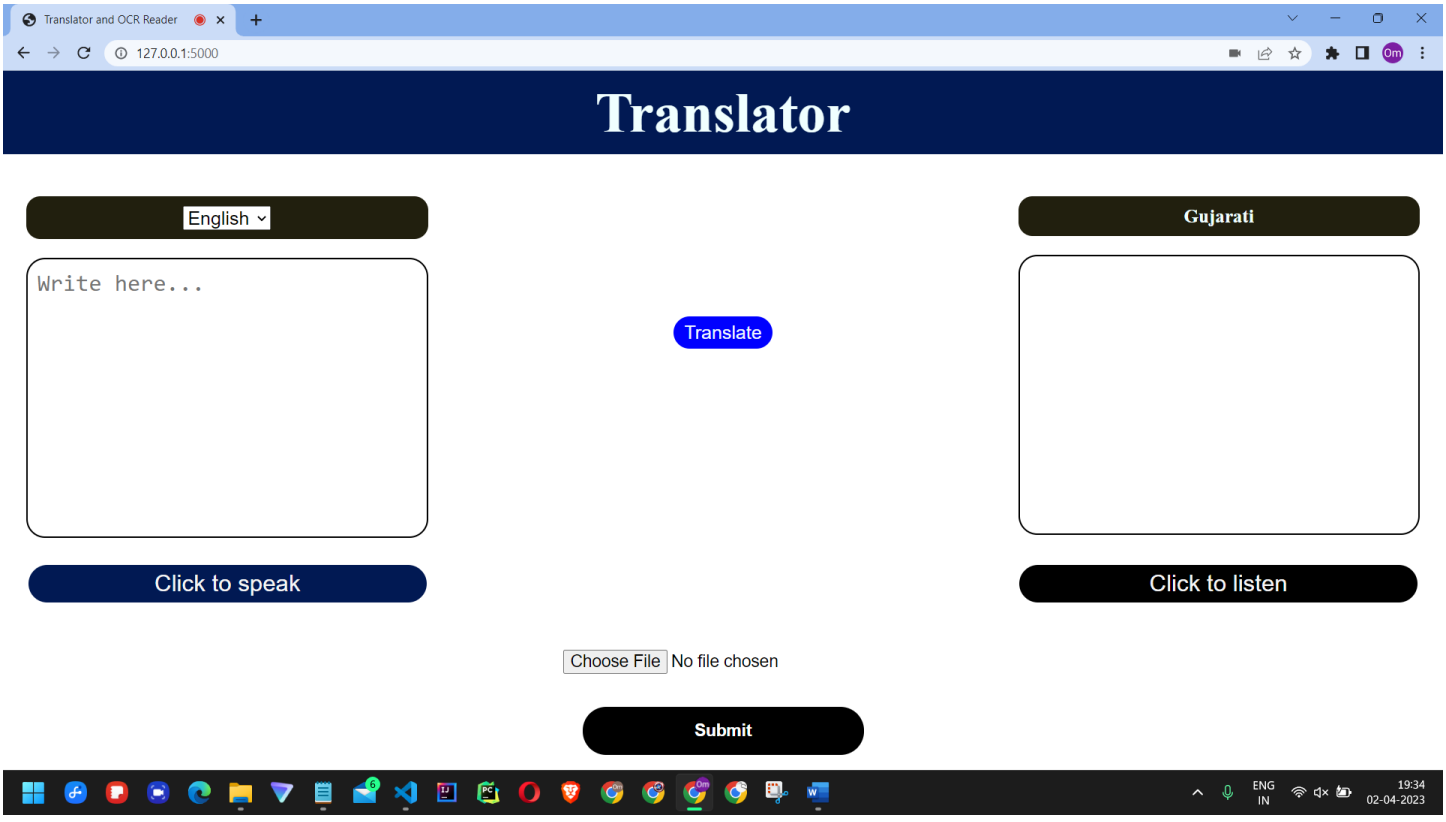
❖ Language Selection

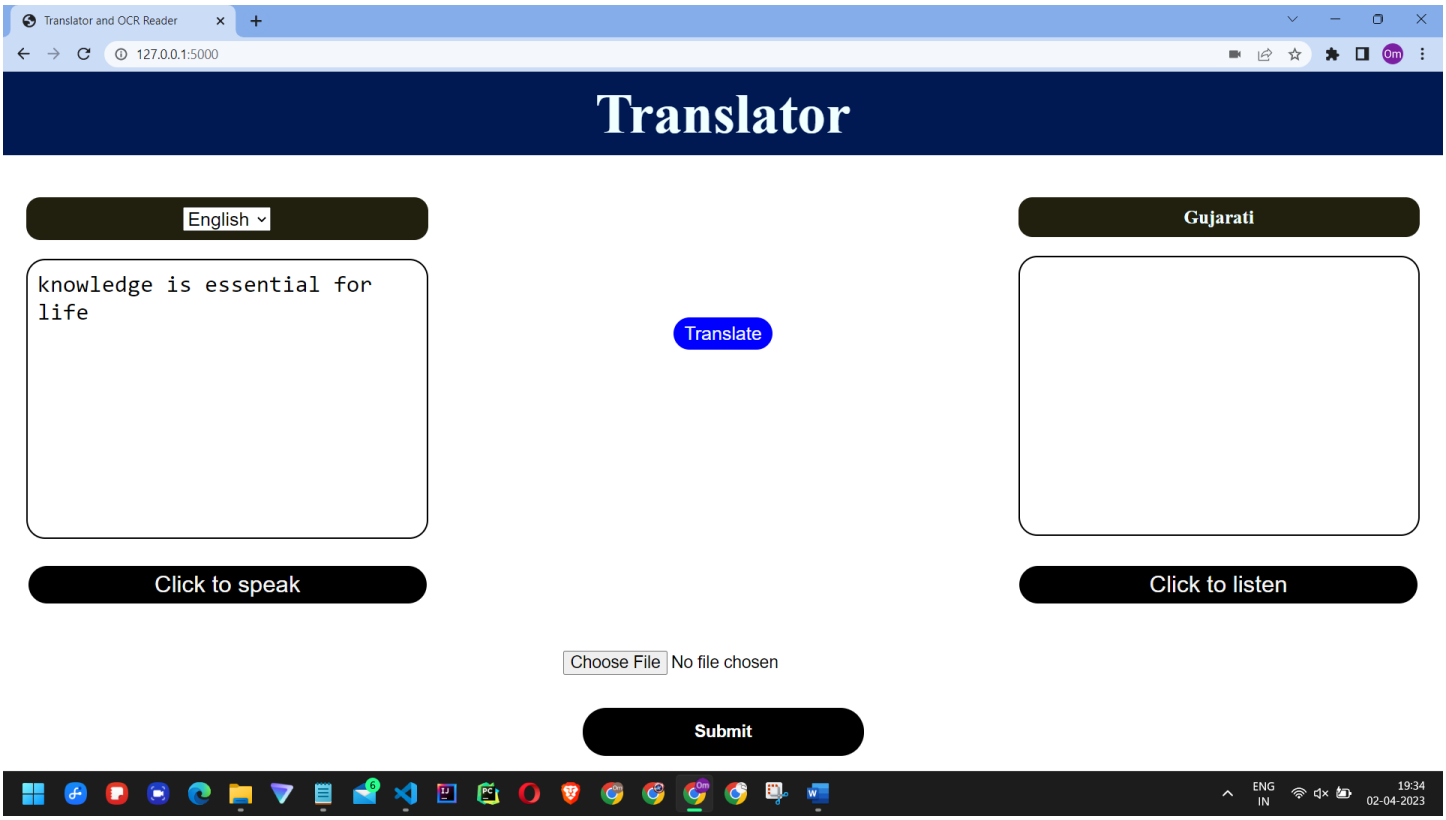
❖ Language Selection (Gujarati to English)

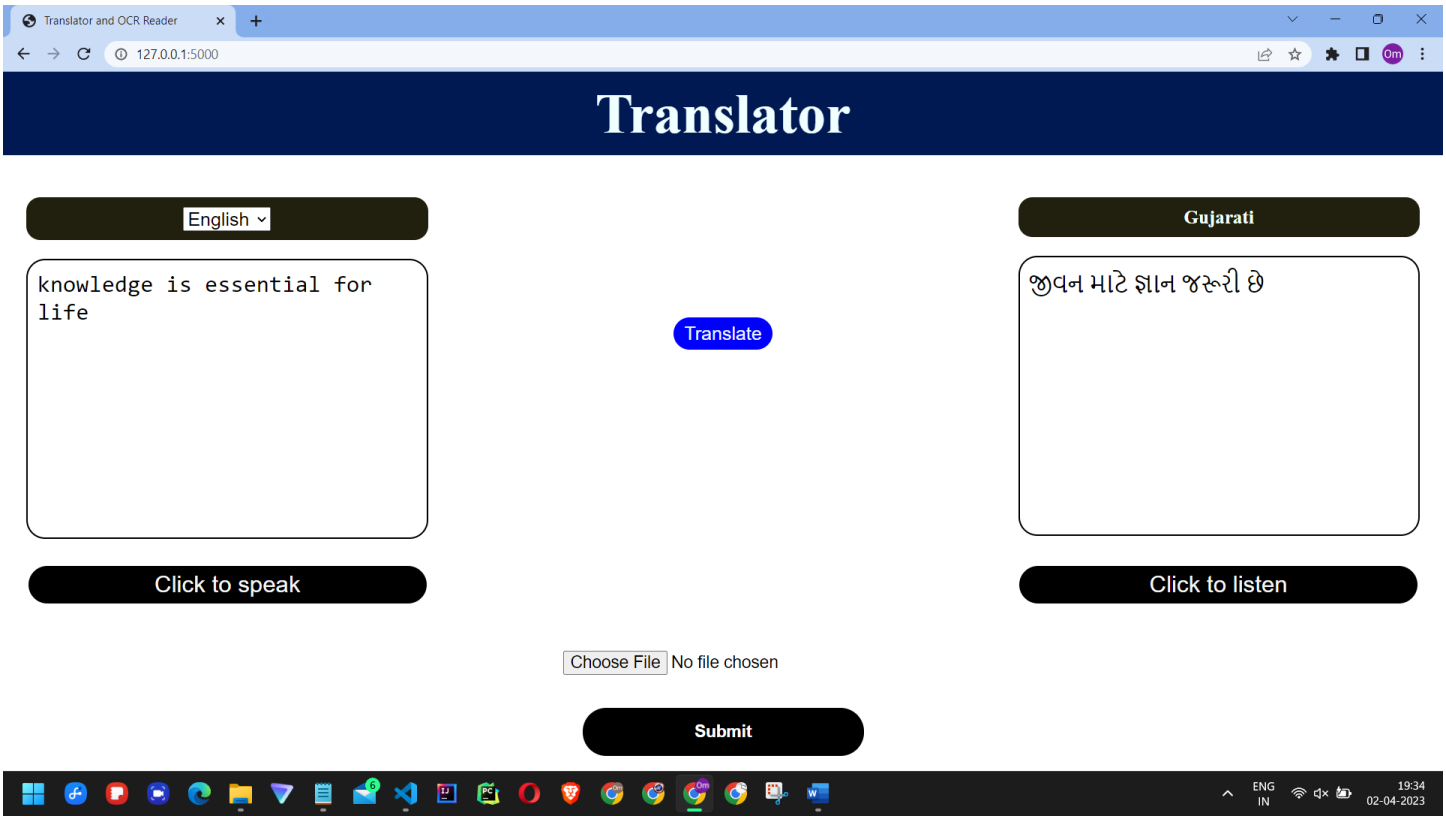
❖ Language Conversion



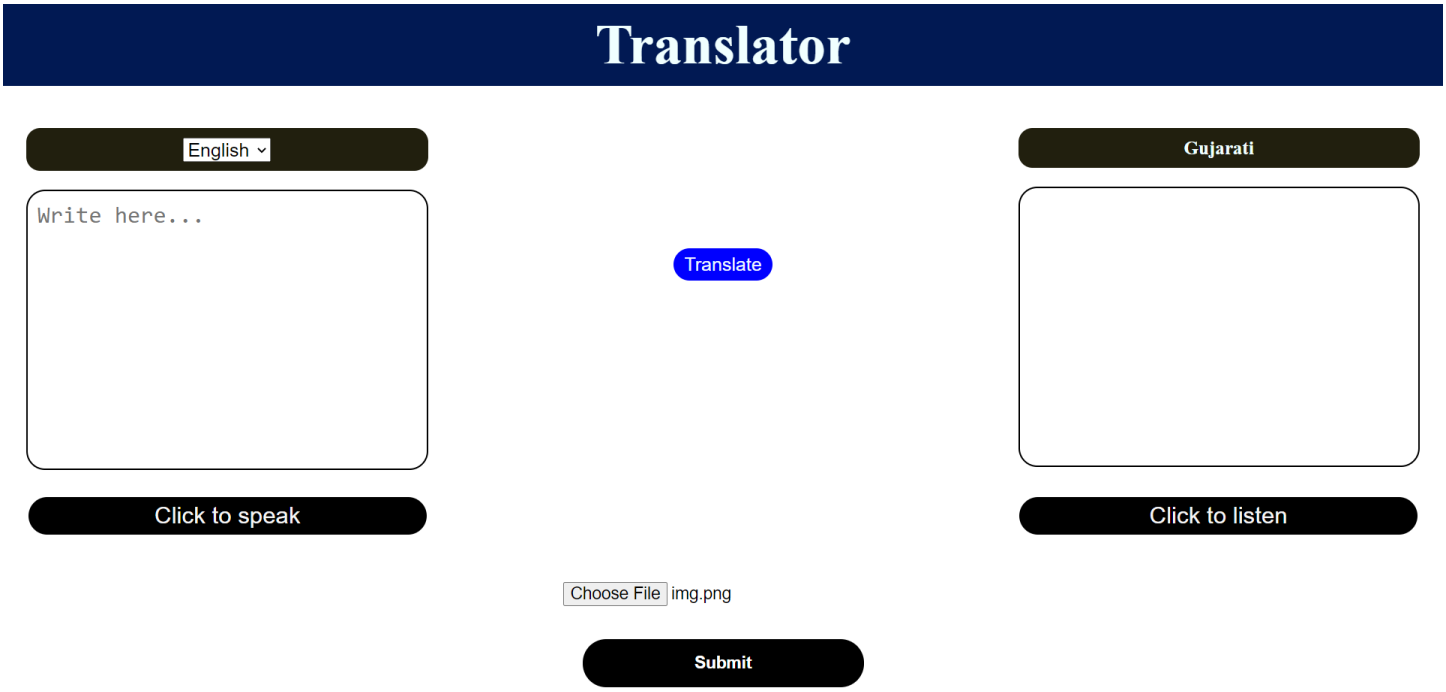
❖ Voice Input



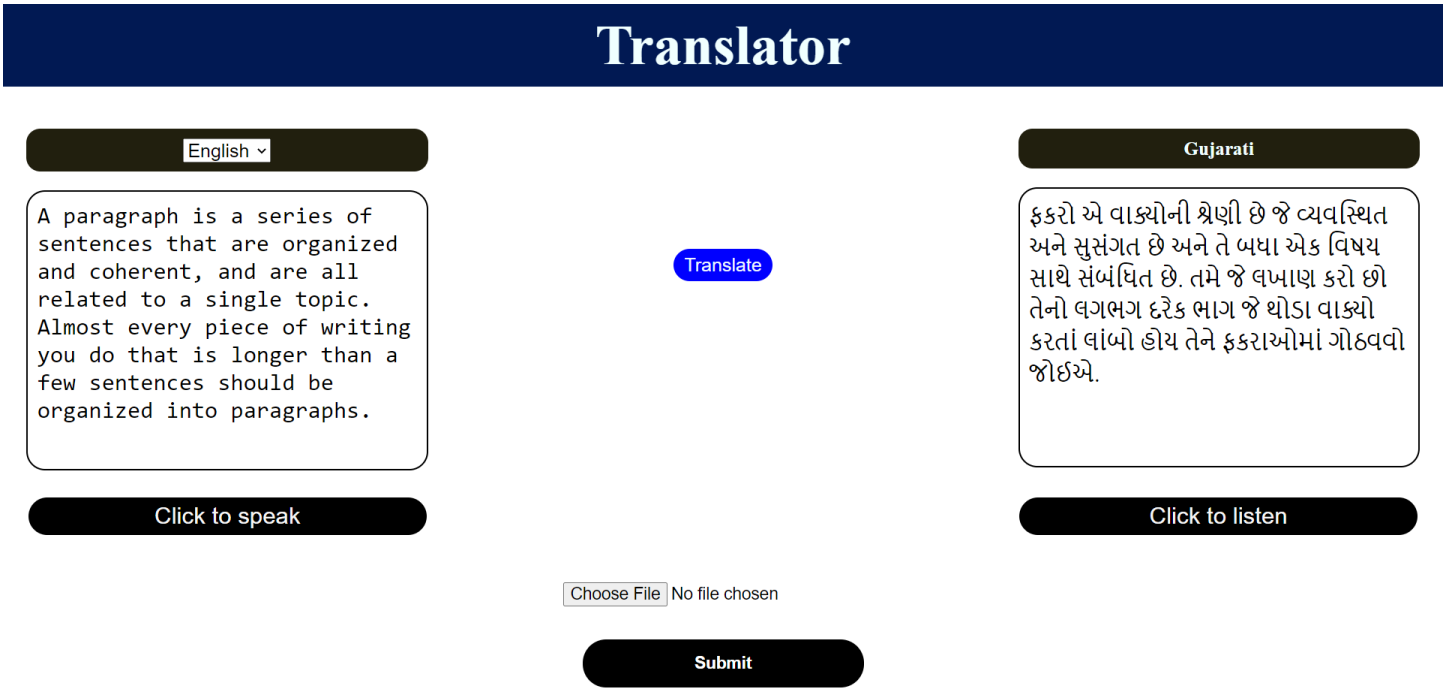

Translator and OCR Reader
127.0.0.1:5000
Translator
English
knowledge is essential for life
Translate
Gujarati
Click to speak
Click to listen
Choose File No file chosen
Submit
ENG IN
19:34
02-04-2023


Translator and OCR Reader
127.0.0.1:5000
Translator
English
knowledge is essential for life
Translate
Gujarati
જીવન માટે જ્ઞાન જરૂરી છે
Click to speak
Click to listen
Choose File No file chosen
Submit
ENG IN
19:34
02-04-2023

❖ OCR – Reader (Input)



❖ OCR – Reader (Output)



Conclusion

Translator and OCR reader is handy tool for users which is robust and provides the quick functionalities of translating text from English to Gujarati and vice versa. All the functionalities are briefly described as under.

- Convert English input text to Guajarati.
- Convert Gujarati input text to English.
- Voice input of both the texts.
- Voice output of both texts.
- Upload image file and translate to Gujarati text.
- Minimalistic UI.

Limitation & Future Expansion

- Limitation
  - Yet, support of Gujarati image file as an input and output as English text is not there.
  - Only typed English text image file can be recognized.
  - Support for voice output for translated text as an image input is not there.
  - Only limited to English and Gujarati languages.

- Future expansion
  - Hand written text image can also be identified by OCR.
  - Support for many other dialects of India.
  - Voice output can be more refined.

Reference / Bibliography

Following links and websites were referred during the development of this project:

- Google Translate Module
- Play Sound Module
- Gtts Module
- Oven CV
- Refrence Reading Material
- Pytesseract
- Dilution and Erosion
- Refrence for ANN
- Stack Overflow

THANK YOU