International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 11 | Nov 2020 www.irjet.net p-ISSN: 2395-0072

Image to Text and Speech Converter

Nivetha.S[1], Kameshwari.S[2]

[1]*Student, Department of Computer Science and Engineering, Sri Ramakrishna Engineering College, Coimbatore, Tamil Nadu, India*

[2]*Student, Department of Computer Science and Engineering, Sri Ramakrishna Engineering College, Coimbatore, Tamil Nadu, India*

Abstract - *Vision is the most beautiful gift that living beings have. Vision allows people to perceive and understand the environment. Vision is needed for the enjoyment of visual and literary mediums. However, people who are deprived of vision are able to obtain information from their surroundings using their hearing capability. Currently majority of the visually impaired people use Braille for reading documents and books, which are difficult to be made and are less readily available. Additionally, Braille requires time for learning and practicing.*

***Key Words*:** Visually impaired, Text to Speech, OCR, Translator

1. INTRODUCTION

The main purpose of the project is to recognize the text from an image and convert it to text and speech and translate it. The system will be a useful tool for students and visually impaired people. Through this system students can extract data from text documents such as circulars, notices, books, etc. and store it as an editable text file format. Visually impaired people can use this system to hear the contents of text documents they come across in their day-to-day life and translation of the text.

1.1 Optical Character Recognition Module

First conversion of the image document into text takes place. The system should detect only the text from the image and convert it into text which can later be manipulated as needed. It must also be able to detect the text in various languages and fonts. For this purpose, we are using the Tesseract with additional trained data for handwritten texts and other languages. The user can select any one of the desired languages and the text from the image in that language will be extracted and will be ready to be translated into any other language or to be converted into an audio file. It can also be converted into editable text document format.

1.2 Text to Audio Conversion Module

Second the text extracted from the image is converted into an audio file which can be played by the user whenever needed.

1.3 Translation Module

Then, the text extracted from the image is translated into any desired language and the translated text can be then converted to speech or document.

1.4 Text to Document Module

Finally, the text is converted to editable text document which can be used to do any editing in the text.

2. SYSTEM IMPLEMENTATION

In neural network, training is a very important process. There are two types of training used with a neural network. They are namely:

1. Un-Supervised Training

2. Supervised Training

Supervised training provides the neural network with training sets and the anticipated output. Unsupervised training supplies the neural network with training sets, without any anticipated output provided.

In our System, we have used tesseract with trained data for handwritten text for the Image to text conversion. The training procedures are as follows:

- Generate Training Images and Box Files
- Run Tesseract for Training
- Generate the unicharset file
- The font_properties file
- Clustering
- Dictionary Data
- The unicharambigs file
- Putting it all together

2.1 Generate Training Images and Box Files

The first step in the process is to determine the full character set to be used and then prepare a text or word processor file with multiple examples. The points to remember when creating a training file are:

- Make at least a minimum number of samples of each character. Ten examples are good, but if it is only five samples, then it not ok for rare characters.
- Create more samples of the more recurring characters - minimum 20. Print and scan to create an image of your training page. Maximum 64 training files with multiple pages can be used.
- It is better to create a mix of fonts and styles, including italic and bold.

```
s 734 494 751 519 0
p 753 486 776 518 0
r 779 494 796 518 0
i 799 494 810 527 0
n 814 494 837 518 0
g 839 485 862 518 0
t 865 492 878 521 0
u 101 453 122 484 0
b 126 453 146 486 0
e 149 452 168 477 0
r 172 453 187 476 0
d 211 451 232 484 0
e 236 451 255 475 0
n 259 452 281 475 0
```

Fig -1: Produced Text File

2.2 Run Tesseract for training

For each training image, boxfile pairs, then run Tesseract in training mode:

```
tesseract [lang].[fontname].exp[num].tif [lang].[fontname].exp[num] box.train
```

or

```
tesseract [lang].[fontname].exp[num].tif [lang].[fontname].exp[num] box.train.stderr
```

2.3 Generate the unicharset file

Tesseract's unicharset file holds information of each symbol (unichar) and the Tesseract OCR engine is trained to recognize it.

The unicharset file must be regenerated each time when inttemp, normproto and pffmtable are generated (i.e. they should be recreated by the time when the box file is changed) as they have to sync each other.

```
110
NULL 0 NULL 0
N 5 59,68,216,255,87,236,0,27,104,227 Latin 11 0 1 N
Y 5 59,68,216,255,91,205,0,47,91,223 Latin 33 0 2 Y
1 8 59,69,203,255,45,128,0,66,74,173 Common 3 2 3 1
9 8 18,66,203,255,89,156,0,39,104,173 Common 4 2 4 9
a 3 58,65,186,198,85,164,0,26,97,185 Latin 56 0 5 a
```

Fig -2: Unicharset File

2.4 The font_properties file

A font_properties text file is created. The main purpose of this file is to give font style information which appears in the output when the font is recognized.

Each line of the font_properties file is formatted as follows:

fontname italic bold fixed serif fraktur

```
1809_Homer 0 0 0 0 0
Aachen_Std_Bold 0 1 0 0 0
Aachen_Std_Medium 0 0 0 0 0
aakar_Medium 0 0 0 0 0
Abadi_MT_Std_Bold 0 1 0 1 0
Abadi_MT_Std_Bold_Italic 1 1 0 1 0
Abadi_MT_Std_Condensed 0 0 0 1 0
Abadi_MT_Std_Light 0 0 0 1 0
Abadi_MT_Std_Light_Condensed 0 0 0 1 0
Abadi_MT_Std_Light_Italic 1 0 0 1 0
Abadi_MT_Std_Medium_Italic 1 0 0 1 0
Abadi_MT_Std_Ultra-Bold 0 1 0 1 0
Abadi_MT_Std_Ultra-Bold_Italic 1 1 0 1 0
Abaton_ITC_Std_Light 0 0 0 0 0
Aboriginal_Sans 0 0 0 0 0
Aboriginal_Sans_Bold 0 1 0 0 0
Aboriginal_Sans_Bold_Italic 1 1 0 0 0
Aboriginal_Sans_Italic 1 0 0 0 0
```

Fig -3: Font_Properties File

2.5 Clustering

Once when the character, features of all the training pages that have been extracted, we should cluster them to create prototypes.

The shapeclustering, mftraining and cntraining programs can be used to cluster the character shape features.

2.6 Shapeclustering

Since we are using Indic languages too, shapeclustering is a necessary part of the training.

Shapeclustering –F font_properties –U unicharset lang.fontname.exp0.tr lang.fontname.exp1.tr

shapeclustering creates a master shape table. Once this shape clustering is done it is then written it to a file named shapetable.

2.7 Mftraining

```
mftraining -F font_properties -U unicharset -O lang.unicharset lang.fontname.exp0.tr lang.fontname.exp1.tr
```

The -U file is the unicharset generated by unicharset_extractor and lang.unicharset is the output unicharset that will be given to combine tessdata.

mftraining will create two data files as output: inttemp (the shape prototypes) and pffmtable (this lets the user know the number of expected features for each character).

2.8 Cntraining

```
cntraining lang.fontname.exp0.tr lang.fontname.exp1.tr
```

This will output the normproto data file which is the character normalization sensitivity prototypes.

2.9 Dictionary Data

In general Tesseract uses up to 8 dictionary files for each language. The dictionary data files are all optional, and they help Tesseract to decide the possibility of different character combinations. In eight files, seven of the files are coded as a Directed Acyclic Word Graph (DAWG), and the other is a plain UTF-8 text file.

To make the Directed Acyclic Word Graph (DAWG) dictionary files, one should need a wordlist (is formatted as a UTF-8 text file with one word per line) for any language.

2.10 The unicharambigs file

The unicharambigs file is a text file which is used to describe the possible ambiguities between characters or sets of characters. This file is usually manually generated.

The unicharset must include each separate character. That is, all of the characters used must be part of the language that is being trained.

3. PUTTING IT ALL TOGETHER

Collect together all the files (shapetable, normproto, inttemp, pffmtable, unicharset) and rename them with an extension lang. prefix (example eng.), and then run combine_tessdata on the files as follows:

combine_tessdata lang.

The resulting lang.traineddata goes in your tessdata directory. Tesseract can then recognize text in your language with the following: tesseract image.tif output -l lang

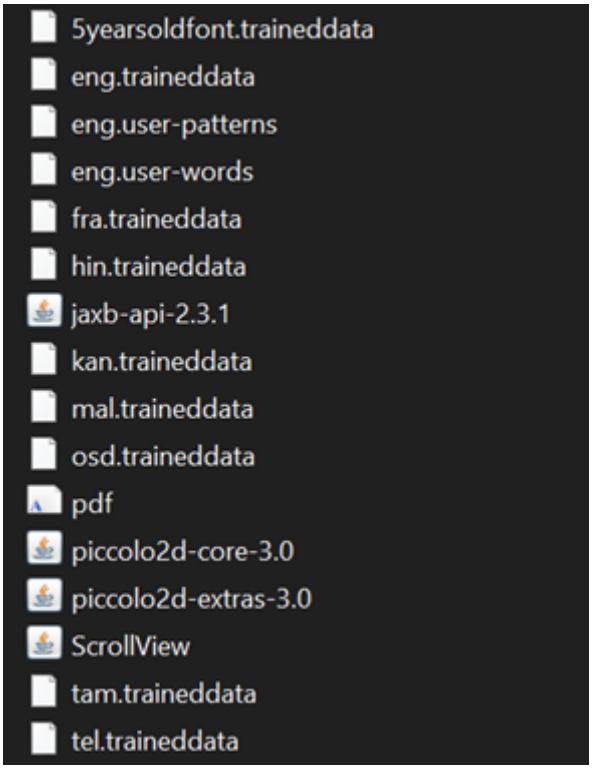

FIG-4: The output trained data Files created

4. RESULTS

The above modules can be used to perform the following functions and the output will be as follows.

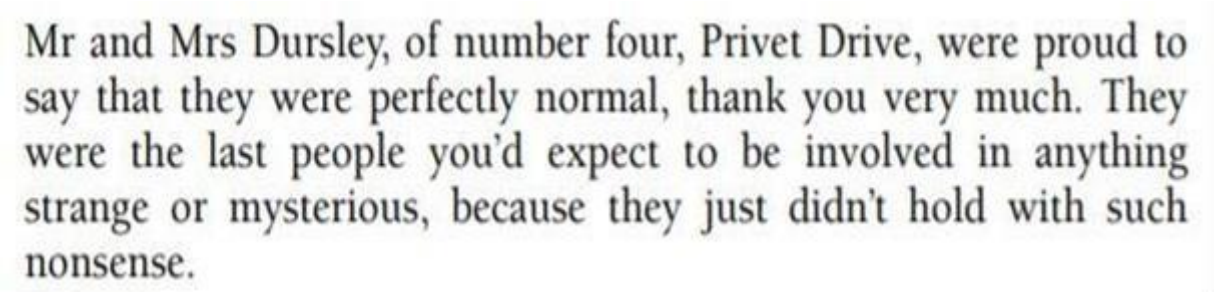
Mr and Mrs Dursley, of number four, Privet Drive, were proud to say that they were perfectly normal, thank you very much. They were the last people you'd expect to be involved in anything strange or mysterious, because they just didn't hold with such nonsense.

FIG-5: Sample Image with English Text

FIG-6: Image converted to Text [English]

The text from the image FIG 5 is extracted and the result is displayed as shown in FIG 6.

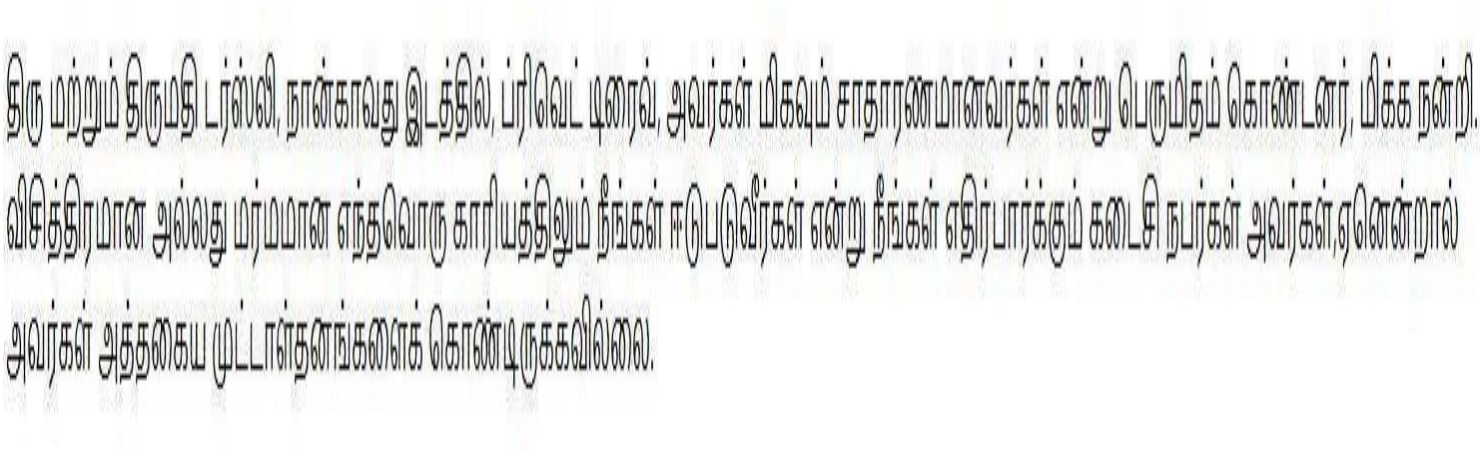

FIG-7: Image with Tamil Text

FIG-8: Image to Text [Tamil]

Tamil text in the FIG 7 is extracted and the output is displayed in the system as shown in FIG 8. Likewise, text of various languages can be extracted and processed.

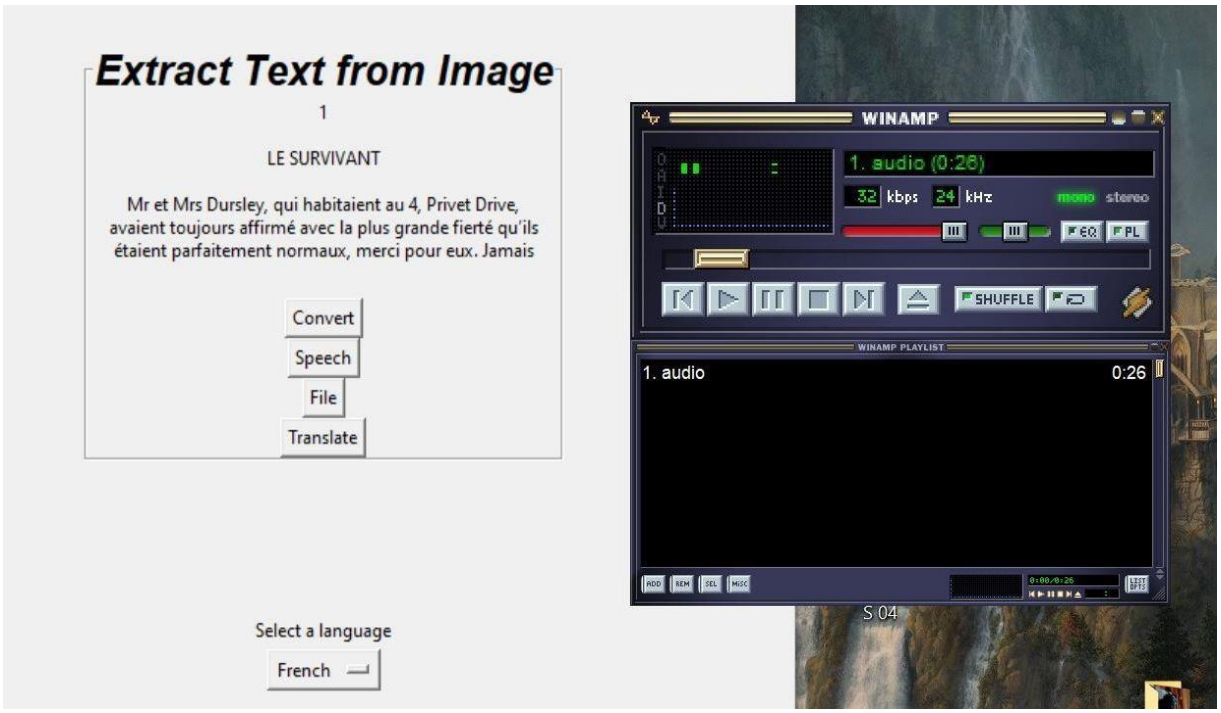

FIG-9: Text to Speech

Text extracted from the images can be converted to speech and this is shown in FIG 9.

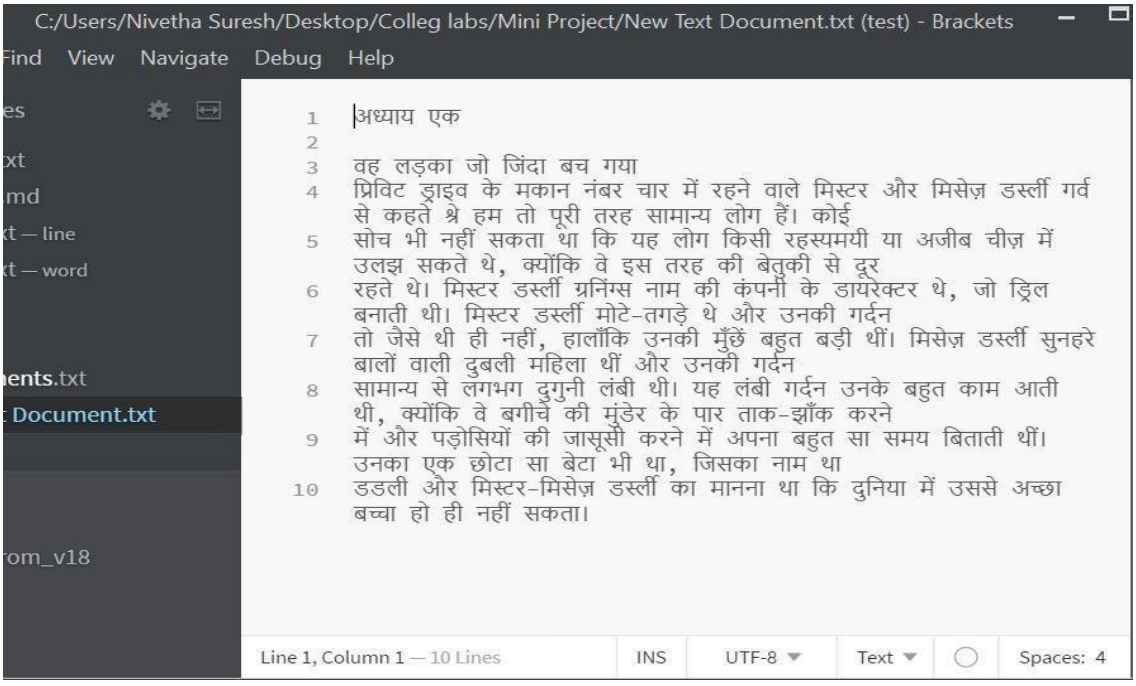

FIG-10: Text to Word

Text extracted from the image can be put together in an editable document type such as word document as shown in FIG 10.

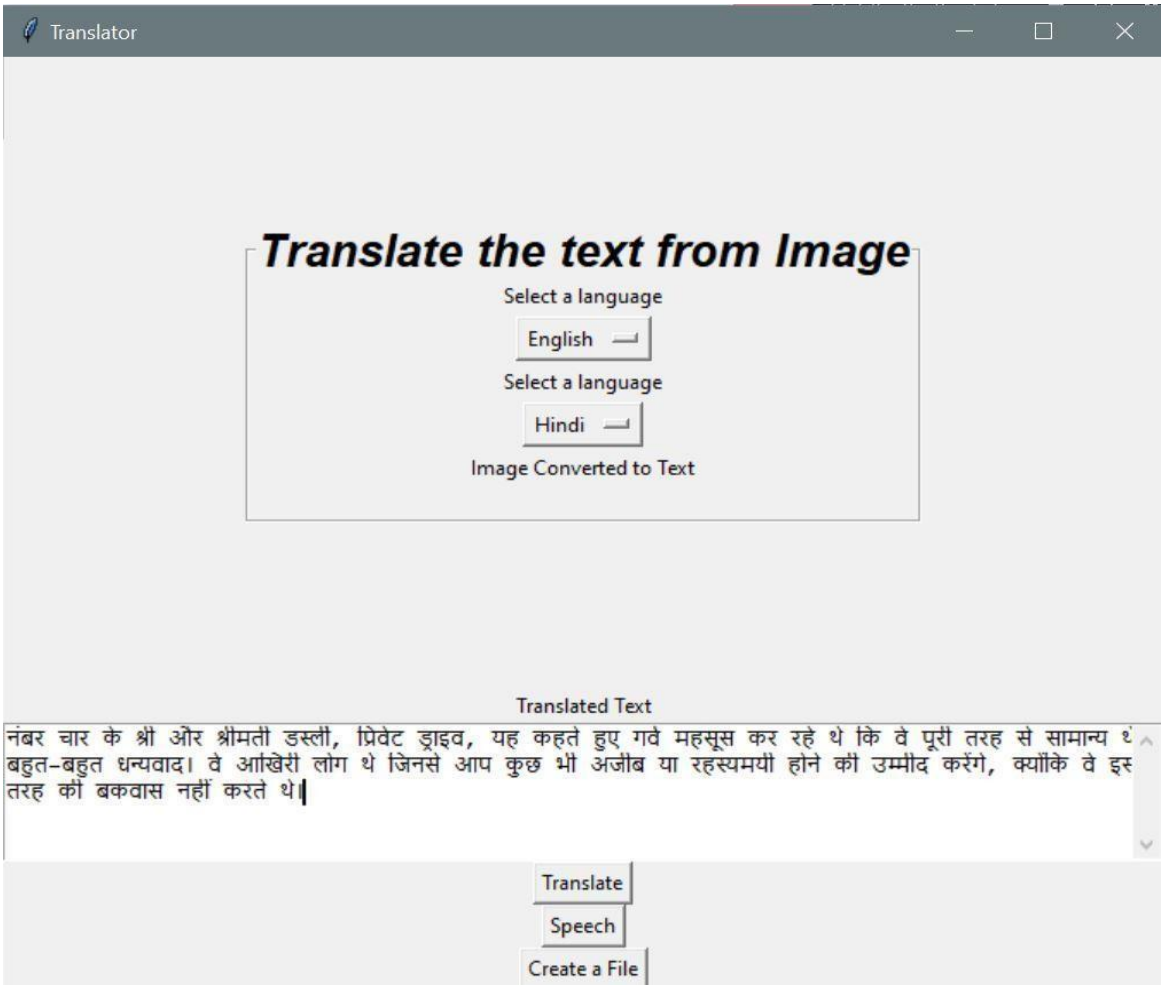

FIG-11: Text Translation

Likewise, the text recognized can be translated into various languages and can be processed in the selected language into speech or document.

FIG-12: Image with Uneven Text with Different Font

The trained data is created for all available fonts and handwritten texts in English so that the OCR will be able to convert any text available in the image intotext.

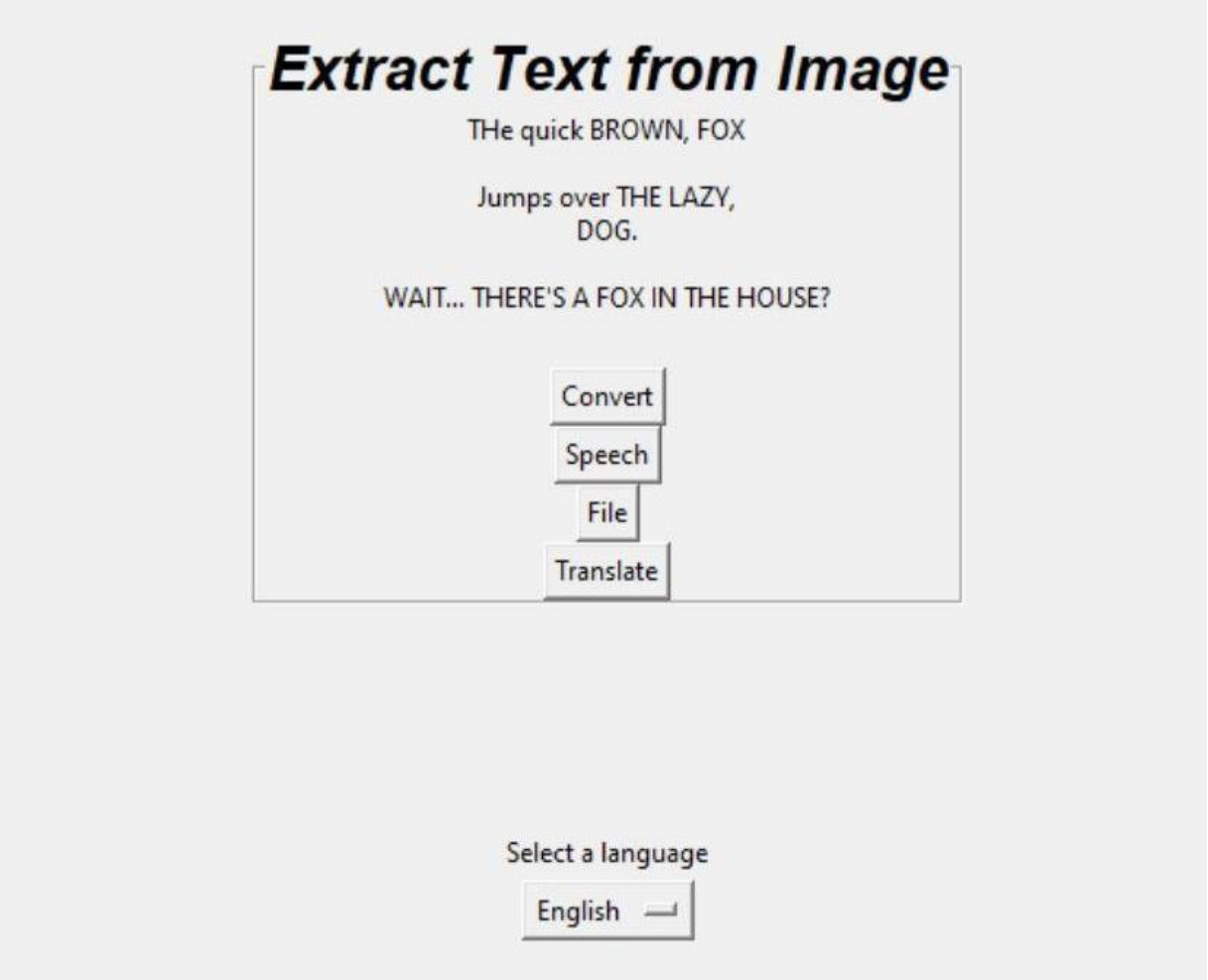

FIG-13: Recognized Text for different font

The system also recognizes text in different styles or fonts and processes it to be available for pre-mentioned functions such as conversion to speech ordocument and also supports translation.

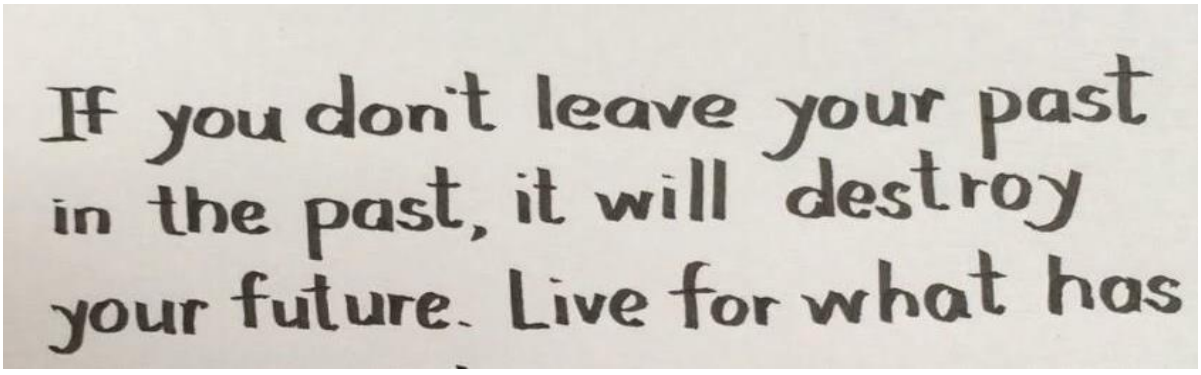

FIG-14: Image with Handwritten Text

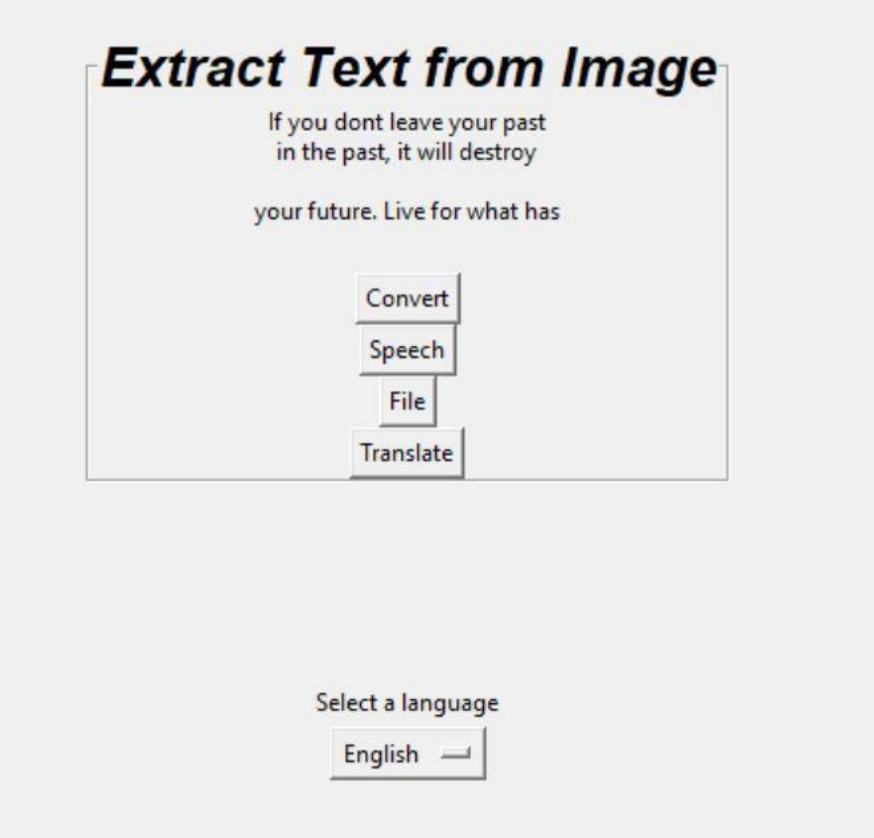

FIG-15: Handwriting Recognition Module

The system also recognizes handwritten text and processes it to be available for pre-mentioned functions such as conversion to speech or document and also supports translation.

5. FEATURES

The system enables its users to recognize text from images and convert it into document and speech. The text can be of various languages and it can also be translated to various languages. The main feature of the system is its ability to convert written text into softcopy which can later be converted to any other language or into audio file.

The conversion of large amount of images into text will make it easier for translation and can be used to convert to audio file.

5.1 Benefits

Presently, the texts in hardcopies are manually converted into softcopies. The OCR converters available do not work accurately for handwritten text. Using this system has the following advantages

- User friendly.
- Highly flexibility
- Highly secure.
- Risk reduction.
- Can be used anywhere
- Highly accurate

5.2 Facilities

- Upload Image from device
- Capture Image
- Select Target Language
- Convert Image to Text
- Convert Text to Audio
- Translate the text
- Convert into Text Document

6. CONCLUSIONS

Our system helps the users to recognize text of various languages from images and process them into document or speech. It also includes the feature of translation of text into various languages. The system will be a great help to visually impaired people and students in their day-to-day life. The system can be used in digitalization of books, records, and scriptures. Many popular written works can be translated into various languages for them to reach different people. Our plan for the future includes incorporating the system into a smart phone application, thus it becomes more portable and easier for access to the users.

REFERENCES

[1] T. Raja Venkatesan, "Intelligent transalate system for visually challenged people", IJSRD - International Journal for Scientific Research & Development| Vol. 3, Issue 12, 2016 | ISSN (online): 2321-0613

[2] S. Muralidharan, "Reading Aid for Visually Impaired People", International Journal of Advance Research, Ideas, and Innovations in Technology

[3] Anusha Bhargava, "Reading Assistant for the Visually Impaired", International Journal of Current Engineering and Technology, Vol.5, No.2 (April-2015)

[4] B. Nithya Santhoshi, "Image Text to Speech Conversion In The Desired Language By Translating With Raspberry Pi", IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)