1. **INTRODUCTION**
   1. **Project Overview**

The project is to convert the image to speech. An image is processed and segmented to identify the characters in the image. Then the characters are combined to form words and save it as a text file. This text file is converted to speech. We have divided the project into four sub parts : image is pre-processed, segmented to extract the images of characters, then characters are recognized and combined , then the text is translated then converted into speech.

* 1. **Project Objective**

Objective of our project is to reduce the complexity in reading and understanding the information’s in text format. This is done by identifying the text content in the captured image and then the text is converted to speech.

* 1. **Relevance of the project**

The image to text conversion has many levels of application nowadays. We can implement this project as solution for many applications. One of the important relevancy of this topic is in the development of devices like an automatic reader to blind people and people those who have visual impairments. The second primary application is in tourism.

1. **LITERATURE SURVEY**

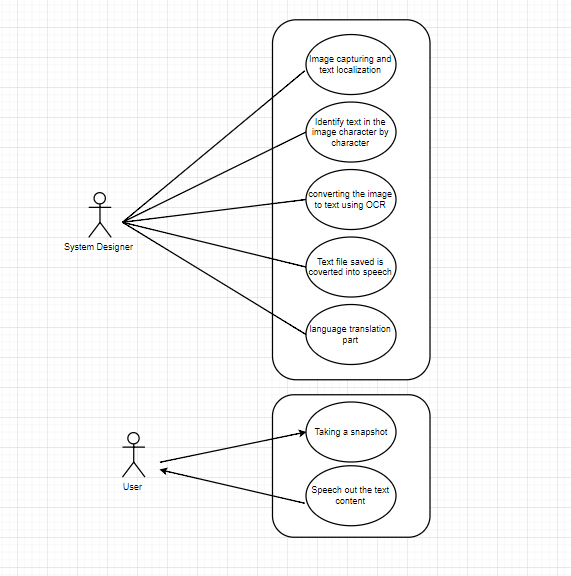
**Paper 1:** is based on the application of an algorithm for text detection. Here they mention about colour reduction technique, method fore edge detection and localization of text regions. Implementation of OCR engine done for the character recognition.

**Paper 2:** they say that the main problem in communication is language bias between communicators. It is based on a prototype which help user to hear the contents of the text images in the desired language and involves extraction of text from the image and converting the text to translated speech in the user desired language. Here they use Raspberry Pi and camera module by using the concepts of Tesseract OCR engine and Google API.

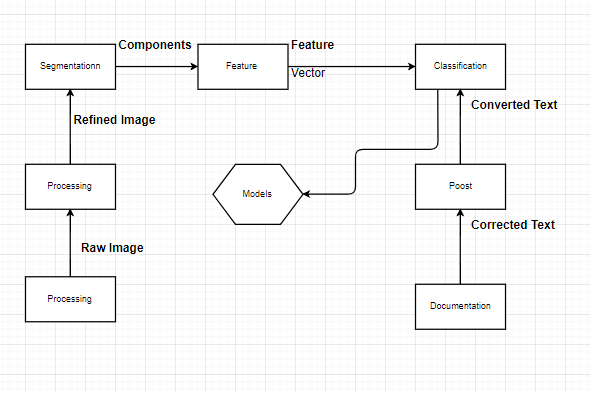
**Paper 3:** consist of an innovative, efficient and realtime cost beneficial technique which can be used to hear the content of a document or text image that combines the concept of OCR and TTS. It is used to develop a system which extracts text data from images containing horizontally aligned text using Optical Character Recognition (OCR) and speaks out the extracted text data using Text to Speech synthesizer (TTS).

**Paper 4:** enable user to hear the content of text instead of reding through them. Combines the concept of OCR and TTS. Challenging task is Text extraction from colour images. Steps include acquiring of character image and reading it, pre-processing, character extracted and resized. Text file is created and make the speech of the text file.

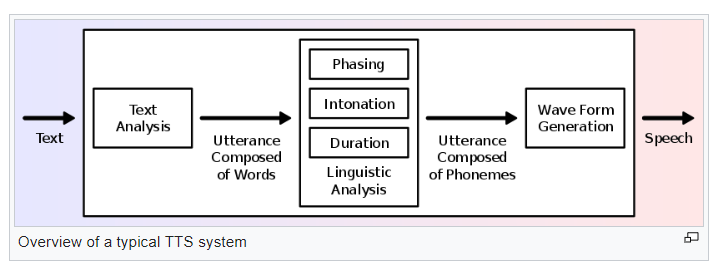
1. **PROJECT DESIGN**



**Fig 1**: Use Case Diagram



**Fig 2**: OCR Block Diagram



**Fig 3**: TTS Block Diagram

1. **IMAGE TO SPEECH**
   1. **Introduction to problem**

The problem is to make a device which convert the text in to an image to speech. As a future advance also, we need to make the speech in any desired language.

**Attempt 1:** We try with the R-PI 3 and the PY-CAM. We implemented the required libraries and with the help of pytesseract and GTTS (google text to speech) we made a system which takes the photo and identify the text parts. The text was saved into text file and then converted to a speech file.

**Problem faced:** The limitation of pytesseract is only images with white background and black text can be identified

**Attempt 2:** Language translation with recurrent neural network

**Problem Faced:** The large amount of data was making the computer to hang. Secondly, the French output was difficult to interpret.

**Attempt 3:** Converting images to sub-images of characters was a big task

**Problem Faced:** Noise behind the text, connected letters, special characters and skew images. Misinterpretation of (I, 1, L), (0, O), (S, 5) etc.

**4.2 Platform and technologies**

**Platform:** We used Jupyter for running the python codes. FTPs used were Putty.We developed our code on Keras and Tensorflow.

**Technologies Used:** NVIDIA DGX-1 V100, the first system builds with ground breaking NVIDIA PascalTM Powered Tesla V100 accelerators (world’s first purpose-built system for deep learning and accelerated analytics, installed in Bennet University).

Microsoft Azure Notebook, free lifetime cloud service provided to the interns.

**4.3 Dataset Collection**

We use two different datasets:

**Language translation:** <https://machinelearningmastery.com/prepare-french-english-dataset-machine-translation/>

**Image to text:** <http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/>

* 1. **Classifiers and models**

**Language translation:** For language translation, we used model built on RNN (Recurrent Neural Network). As the final model, we used encoder- decoder implemented with the help Embedding layer as the top most layer and then the accuracy was enhanced using Bidirectional LSTM.

**Image to text:** For image to text, we used CNN (Convoluted Neural Network).

Through the results we analyzed that our model was highly biased. Thus, we used tanh with Leaky Relu as our activation function. Our model comprised of 3 Fully convoluted layers, then 2 hidden layers and then a softmax for classifying 62 classes.

**Technical terms:**

Image to text

* **Convolution 2d**: This layer creates a convolution kernel that is convolved with the layer input to produce a tensor of outputs. If use\_bias is True, a bias vector is created and added to the outputs. Finally, if activation is not None, it is applied to the outputs as well.
* **Max Pooling:** This is a form of non-linear down-sampling. There are several non-linear functions to implement **pooling** among which **max pooling** is the most common. It partitions the input image into a set of non-overlapping rectangles and, for each such sub-region, outputs the maximum.
* **Activation Function** (Tanh, Relu, Sigmoid, Leaky Relu): We have used the combination of Leaky Relu and Tanh because our model was very highly biased.
* **Flatten:** We need to convert the output of the convolutional part of the CNN into a 1D feature vector, to be used by the ANN part of it. This operation is called flattening. It gets the output of the convolutional layers, flattens all its structure to create a single long feature vector to be used by the dense layer for the final classification.
* **Dropout:** It is a [regularization](https://en.wikipedia.org/wiki/Regularization_(mathematics)) technique for reducing [overfitting](https://en.wikipedia.org/wiki/Overfitting) in [neural networks](https://en.wikipedia.org/wiki/Neural_networks) by preventing complex co-adaptations on training data. It is a very efficient way of performing model averaging with neural networks. The term "dropout" refers to dropping out units (both hidden and visible) in a [neural network](https://en.wikipedia.org/wiki/Neural_network).

Character Segmentation

* **C# (.exe):** In c# we have made an exe file to segment the characters

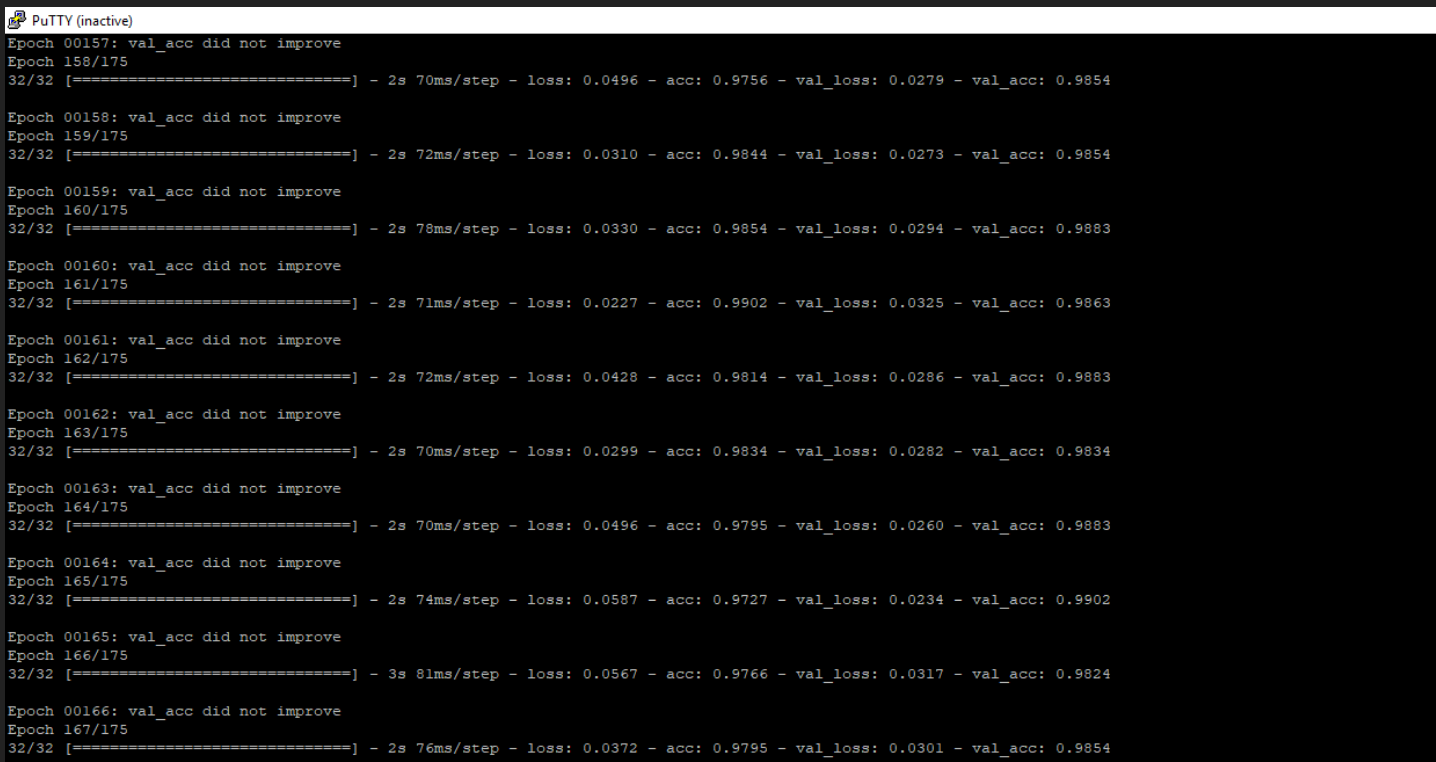
Language Translation

* **LSTM:** LSTMs are explicitly designed to avoid the long-term dependency problem.
* **GRU:** Gated recurrent units (GRUs) are a gating mechanism in [recurrent neural networks](https://en.wikipedia.org/wiki/Recurrent_neural_networks) and their performance on polyphonic music modeling and speech signal modeling was found to be similar to that of [long short-term memory](https://en.wikipedia.org/wiki/Long_short-term_memory) (LSTM).
* **Bi-directional RN**N: BRNNs were introduced to increase the amount of input information available to the network but have limitations on the input data flexibility, as they require their input data to be fixed.
* **Embedding layer:** embedding layers can be used to embed many more things than just words.
* **Encoder and Decoder:** The encoder-decoder architecture for recurrent neural networks is the standard neural machine translation method that rivals and, in some cases, outperforms classical statistical machine translation methods.

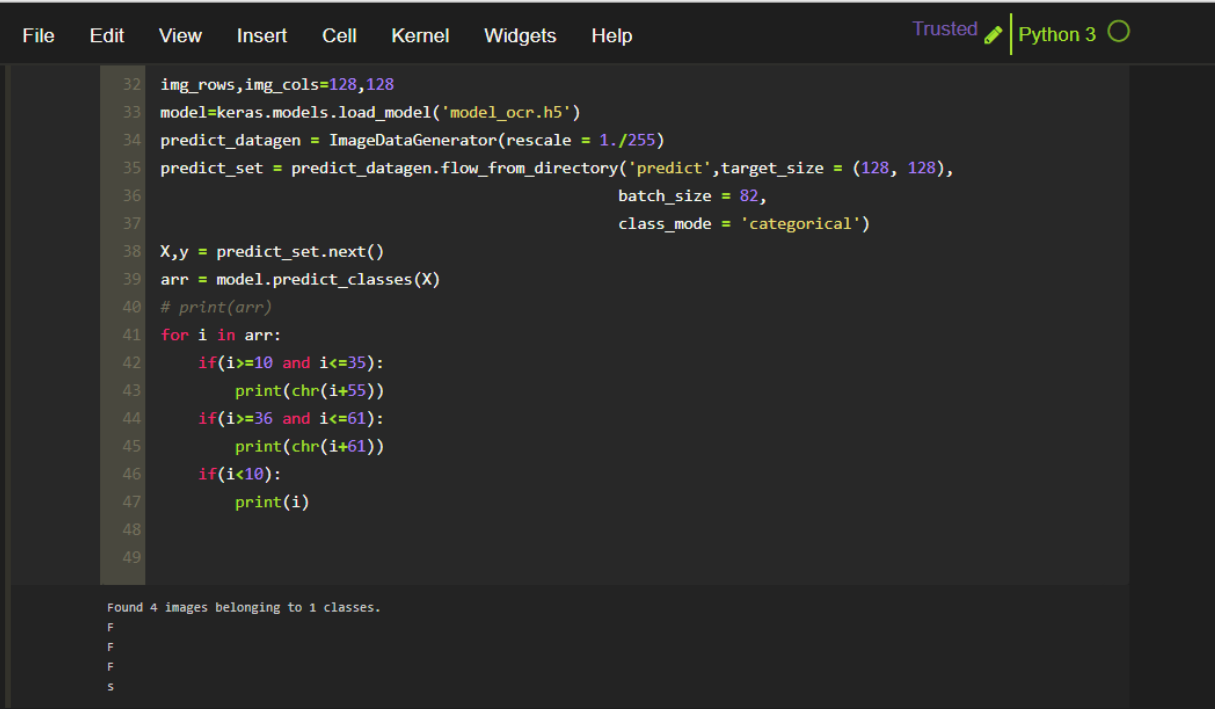
Text to Speech

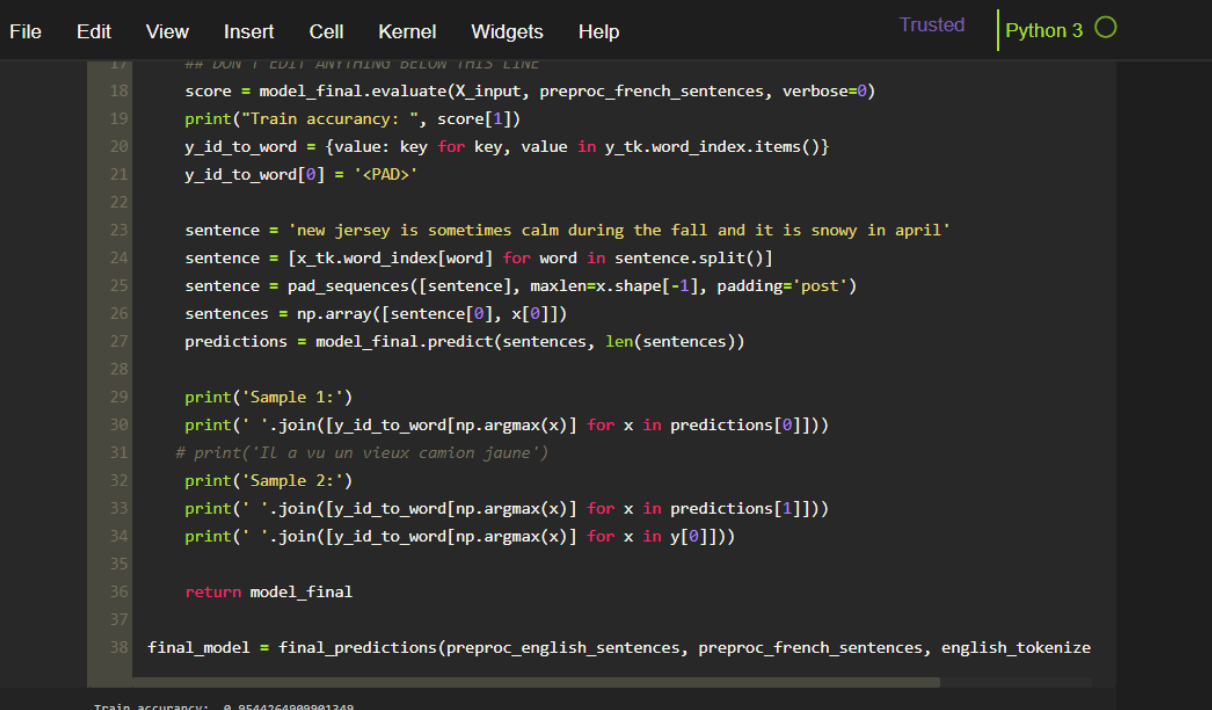
* **GTTS:** This is google-text-to-speech which is used to convert the text to speech.
* **PYgame:** Pygame is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) set of [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) modules designed for writing [video games](https://en.wikipedia.org/wiki/Video_game). It includes [computer graphics](https://en.wikipedia.org/wiki/Computer_graphics) and sound libraries designed to be used with the Python [programming language](https://en.wikipedia.org/wiki/Programming_language).

**4.5 Results**



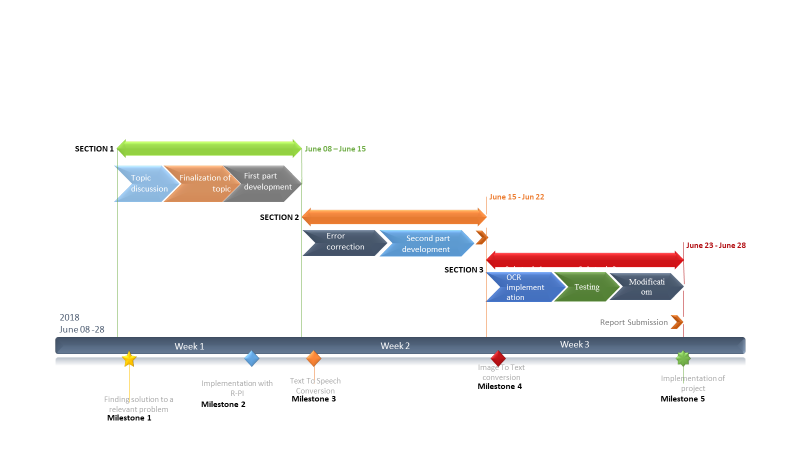
**Fig 4**: OCR Accuracy

 **Fig 5**: OCR Result



**Fig 6:** Translation Accuracy

**4.6 Project Development Time Schedule**



**Fig 7:** Project Development Time Schedule

**4.7 Learnings and reflections from the project**

Through the project we developed advanced knowledge of machine learning. Our concepts strengthened both theoretically and practically.

We learned dataset preprocessing, gained enhanced knowledge in convolutional neural network and

Recurrent Neural Network.

We learned languages namely Python and C#.

We developed deep understanding of Keras and Tensorflow, on which our model was built.

Other than technical knowledge we are taking back home ample amount of tolerance and patience

Which we inculcated within us while doing this project.

Second most important thing we learned was team work.

1. **CONCLUSION AND LIMITATIONS**

**5.1 Conclusion**

We successfully converted image into sub images of characters and then recognized the characters using 62 classes classifier. Nextly the characters recognized were successfully combined to words and sentences.

Then by translating their language using RNN’s ,we converted the text to speech .

Every thing (except speech) was coded from scratch(all thanks to MIT for keras)

Hence,our conclusion, as we got pretty satisfying results this project can be really enhanced and worked upon for developing an economic product which might be really very useful.

**5.2 Limitations**

It has some underlying limitations:

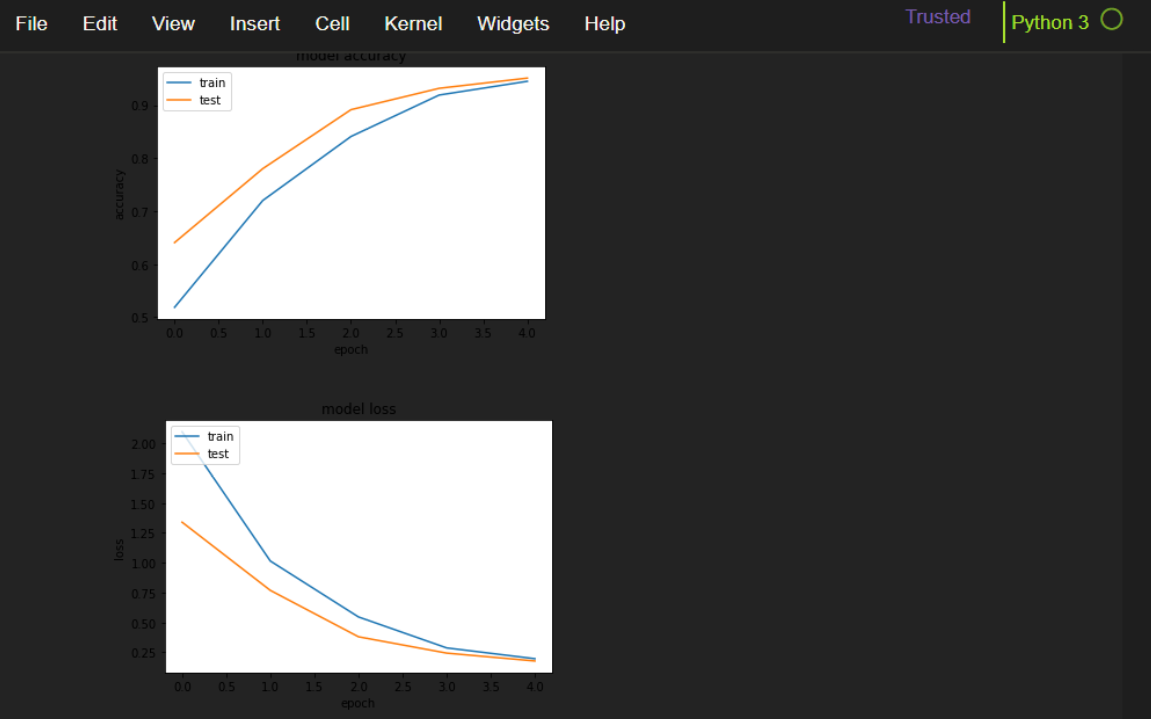
1. It can’t detect handwritten images
2. Images with lot of noise in the background can’t be detected.(Our model needs little enhancement for that.)
3. It fails for regional languages like telugu(lot of curves) and Urdu(the letters are connected).
4. **FUTURE SCOPE**

This project can be further worked upon for many economical applications.

1. Most important thing is this will be very useful to blind people and those with visual impairments to identify their environment.
2. Also, this will reduce the human effort in reading big documents.
3. This can be your child’s visual friend, by implementing this as a story teller to the children.
4. This can also be implemented for a helping hand for travelers who faces the problem of language.
5. This can be further refined into voice food menus

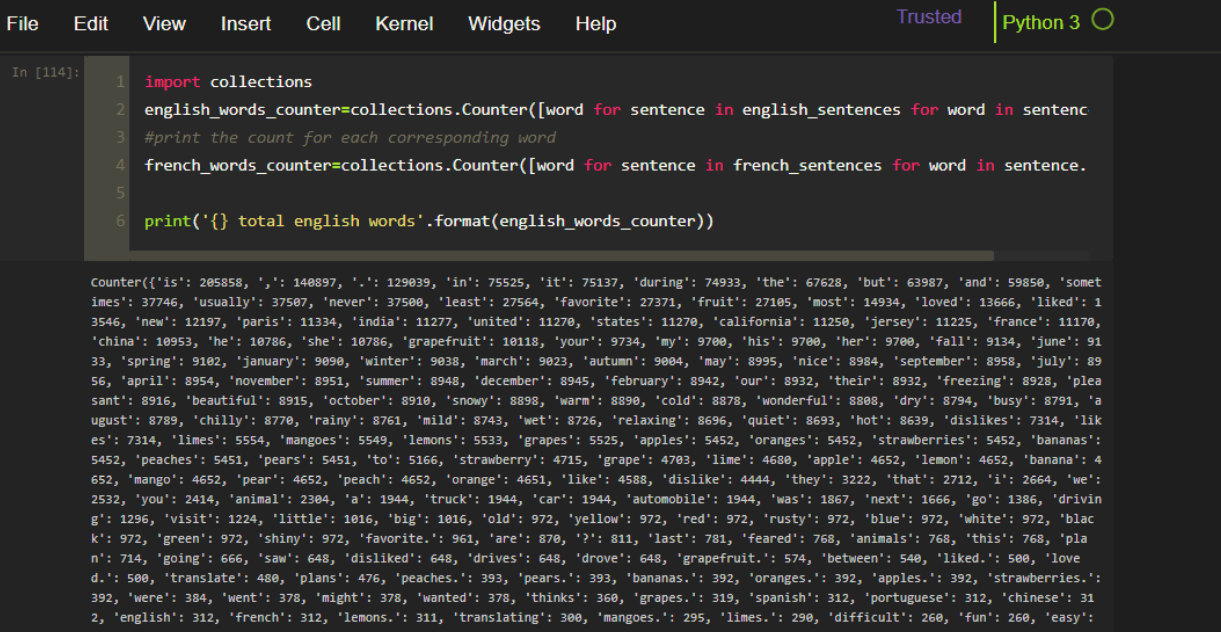
**7.APPENDIX**

**Appendix 1: Output Snapshots**

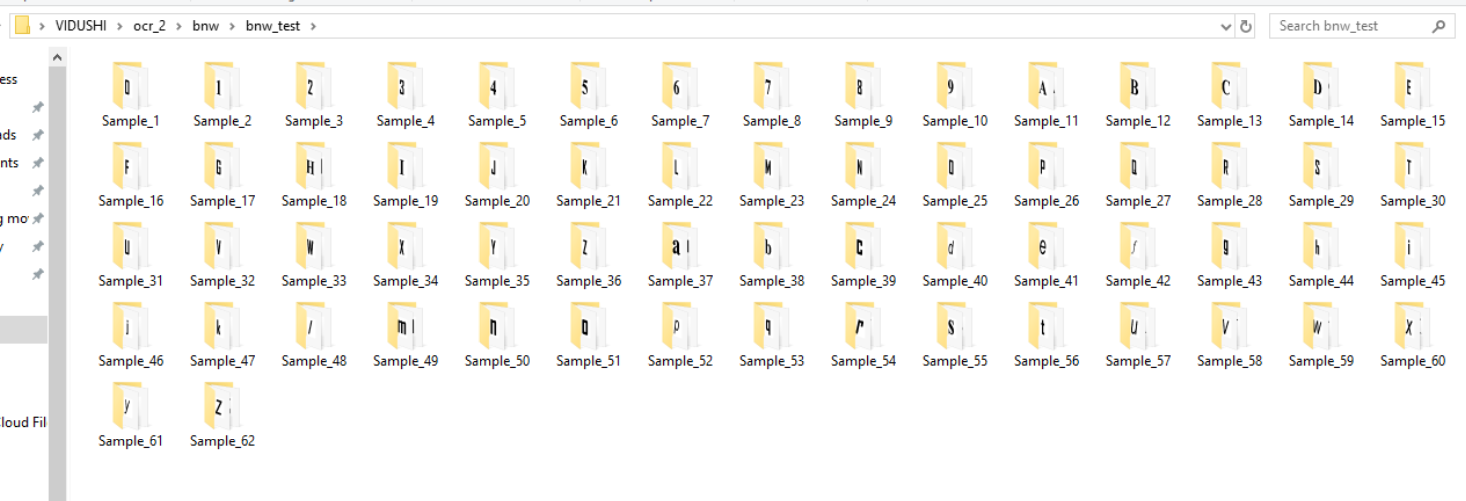


**Fig 8:** Graph for language translation

**Appendix 2: Dataset Snapshots**

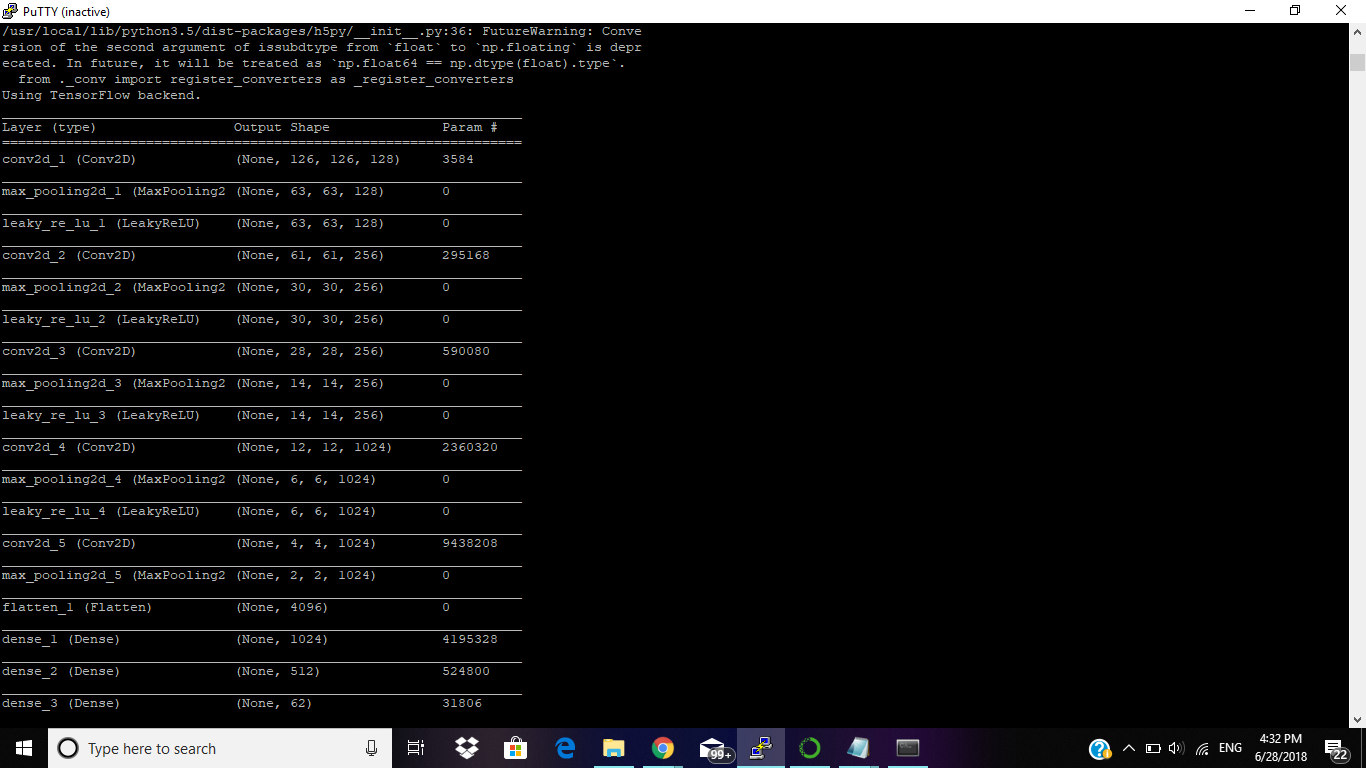


**Fig 9:** Language Translation dataset

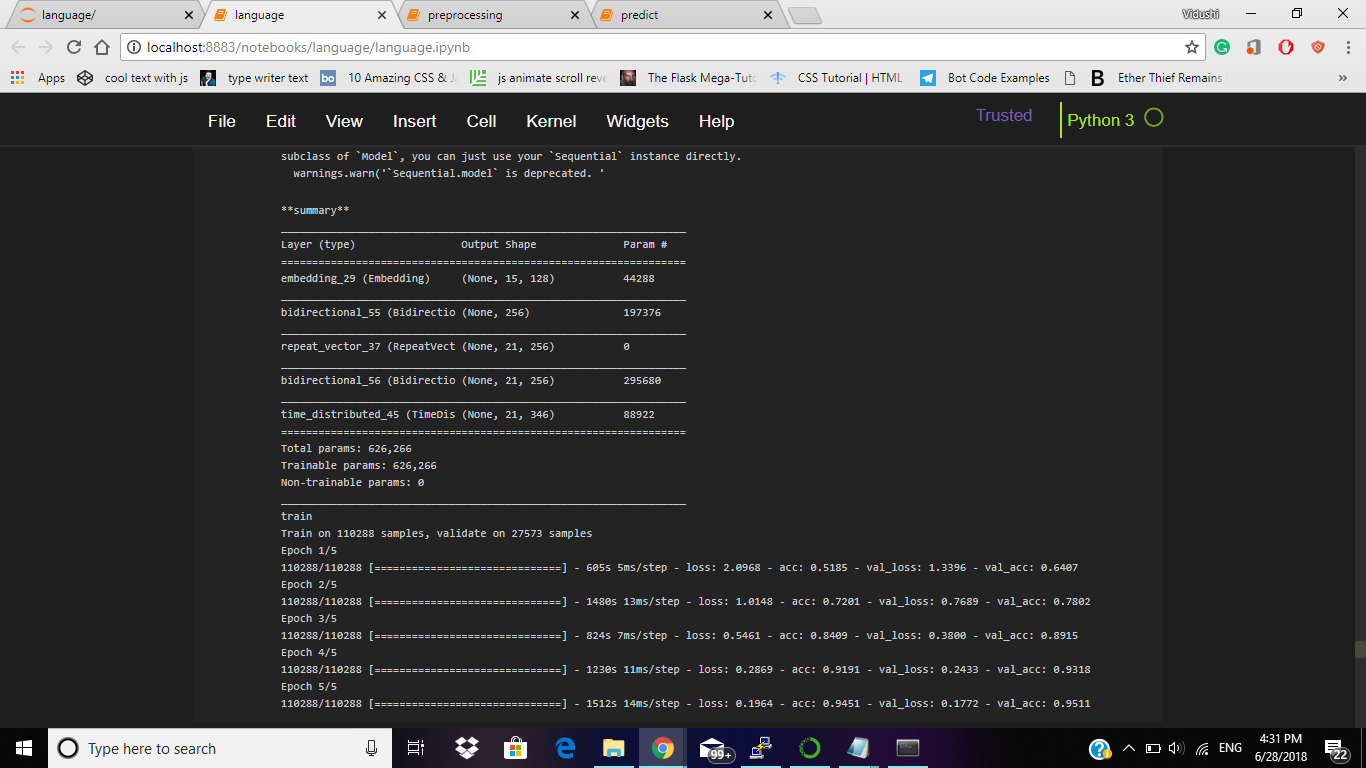


**Fig 10**: OCR dataset

**Appendix 3: Models Snapshots**



**Fig 11:**Model for Image to text

****

**Fig 12:** Model for language translation

**8. References**

[1] Breen A.P., “The future role of text to speech synthesis in automated services”.

[2] Tanprasert, C.; Koanantakool,T., “Thai OCR: a neural network application”

[3] Lucas S.M., „‟High performance OCR with syntactic neural networks”.

[4] https://github.com/aquatiko/Image-Text-Speech-Synthesizer-Converter

[5] https://ieeexplore.ieee.org/abstract/document/1296349/

[6] Morgan Kaufmann, 1990, pp. 612-626. [7] http://code.google.com/p/tesseract-ocr.

[7] https://github.com/prabhakar267/ocr-convert-image-to-text

[8] M. Cai, J. Song and M. R. Lyu. A New Approach for Video Text Detection. In Proc. of International Conference On Image Processing, Rochester, New York, USA, pp. 117-120, 2002.

[9] https://ieeexplore.ieee.org/abstract/document/7919526/

[10] http://ijarcsse.com/Before\_August\_2017/docs/papers/Volume\_7/7\_July2017/V7I7-0117.pdf

[11] A. K. Jain and B. Yu. Automatic Text Location in Images and Video Frames. In Proc. of International Conference of Pattern Recognition (ICPR), Brisbane, pp. 1497-1499, 1998.

[12] <https://www.hackster.io/rgrokett/pitextreader-for-impaired-vision-695331>

[13] https://pdfs.semanticscholar.org/99c9/90ff841da4923771c5d2b6692c3a4656d447.pdf

**Github link:**