IMAGE TO TEXT CONVERSION IN FOREIGN LANGUAGES USING DOCUMENT IMAGE PROCESSING TECHNIQUE

by Rishabh Jain, Satyam Gupta And Shahzeb Rizvi

MINOR-2

PROJECT

Final Report

on

IMAGE TO TEXT CONVERSION IN FOREIGN LANGUAGES USING DOCUMENT IMAGE PROCESSING TECHNIQUE

Submitted by: -

Name	Roll No	Branch
Rishabh Jain	R110216125	CSE-CCVT
Satyam Gupta	R110216138	CSE-CCVT
Shahzeb Rizvi	R110216143	CSE-CCVT

Under the guidance of
Mr. Nitin Arora
Assistant Professor
Department of Informatics

School of Computer Science
UNIVERSITY OF PETROLEUM AND ENERGY STUDIES Dehradun-248007

2018-19

Approved By

(Mr. Nitin Arora)
Project Guide
Department of Informatics

(Dr. Deepshikha Bhargava)
Department Head
Department of Virtualization

DECLARATION

I hereby declare that the project entitled "***Image to text conversion in foregin languages using document image processing technique***" is a Bonafide and authentic record of work done by our team under the supervision of Mr. Nitin Arora during academic session 2018-2019. The work presented here is not copied from any external website or source and is also not registered earlier for any degree/diploma to any other educational institution. I understand that any such unfair means is liable to be punished in accordance with UPES rules and regulations.

Place: University of Petroleum and Energy Studies, Dehradun

Department of Virtualization

University of Petroleum and Energy Studies

Certificate

It is to certify that the works contained in the project titled **"***Image to text conversion in foregin languages using document image processing technique***"** by following students has been carried out under my/our supervision and that this work has not been submitted elsewhere for a degree or certification programme.

Student Name	***Roll Number***	***Signature***
1. Rishabh Jain	***R110216125***	
2. Satyam Gupta	***R110216138***	
3. Shahzeb Rizvi	***R110216143***	

Signature

Mr. Nitin Arora
Mentor
Department of Informatics
School of Computer Sciences, U.P.E.S.
Dehradun, Uttarakhand
India-248007

Signature

Dr. Deepshikha Bhargava
Head of the Department
Department of Virtualization,
School of Computer Sciences, U.P.E.S.
Dehradun, Uttarakhand
India- 248007

Abstract

Every country has their own native languages such France has French, Japan has Japanese, India has Hindi and other local languages (Gujrati, Marathi, Telugu, Tamil, Bengali, etc.). When a person who don't know the English Language travel to a country whose local language is English will face many problems, understand road sign texts, shop's name, Instructions Boards etc. So, we are going to design a software with the help of which user can take snap of the scene containing te3 of which he/she want to translate it, upload that photo and software, choose the language and the software will give the output of text of photo in User's native language. We are using CNN (Convolutional Neural Network) to identify the characters, EAST (Efficient and Accurate Scene Text Detector) for detecting the text in the image, Digital Image Processing methods are used to segment the text in the detected text, googletrans for translating the text to other language and Tkinter for creating GUI (Graphical user Interface) for software.

Keywords: GUI, CNN, EAST Detector, googletrans

Acknowledgement

It gives us immense pleasure to extend our feelings of gratitude to each and every person and the almighty who have helped in achieving this feat. We express our heartful thanks owe a deep sense of gratitude to our teacher and Project Mentor, Mr. Nitin Arora University of Petroleum and Energy Studies for their sincere guidance and inspiration in completing this project.

We are extremely thankful to **Mr. Pravin Dagdee, Mr. GL Prakash, Mr. Nitin Arora** and all the faculty members of University of Petroleum and Energy Studies for their immense guidance and support in the development of this project.

We would also like to thank all the friends of ours who have helped us in times when the project was in a condition from where project completion would have been a difficult task.

The in-depth knowledge attained for developing this project is immense and would explore more areas to come up with more such solutions. The knowledge gaining process was priceless and would definitely apply this knowledge in the near future.

Name of Students: -

Rishabh Jain

Satyam Gupta

Shahzeb Rizvi

Contents

List of Figures and Tables

List of Figures

List of Abbreviations

Abbreviations

CNN	Convolutional Neural Network
GUI	Graphical User Interface
EAST	Efficient and Accurate Scene Text Detector
GOOGLETRANS	Google Translator
THRESH	Threshold

Introduction:

Early years, Human Translators are used to set the communication between different country people. Having human translator also have various cons such as turnaround time is longer, a good income should be given to human translators, human translators are restricted to certain number of languages, etc.

3

Image translation is a term related to machine translation services for mobile devices. Image translation refers to an extra service provided by mobile translation applications wherever the user will take a photograph of some written text (menu list, road sign, document etc.) apply optical character recognition (OCR) technology to it to extract any text contained in the image, and then have this text translated into a language of their choice.

Optical character recognition (OCR) is that the mechanical or electronic conversion of pictures of typewritten, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo (for example the text on signs and billboards in a very crowded photo) or from subtitle text superimposed on a picture (for example from a tv broadcast). It is a typical methodology of digitizing written texts so they'll be electronically altered, 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 could be a field of analysis in pattern recognition, artificial intelligence and computer vision.

2

Early optical character recognition is prerequisite to technologies involving telegraphy and making reading devices for the blind. In 1914, Emanuel Goldberg developed a machine that scan characters and turn them into quality telegraph code. Concurrently, Edmund Fournier d'Albe developed the Optophone, a handheld scanner that when moved across a printed page, produced tones that corresponded to specific letters or characters.

In the late 1920s and into the 1930s Emanuel Goldberg developed what he called a "Statistical Machine" for searching microfilm archives using an optical code recognition system. In 1931 he was granted USA Patent number 1,838,389 for the invention. The patent was acquired by IBM.

Literature Review

Here is the conclusion of some of the reference paper that we review to make our project better and to know more technologies that we can use in our system.

[1] EAST: An Efficient and Accurate Scene Text Detector – This methodology is used to detect the text in a natural scene image with the help of 2 output layers.

- First layer is used for output probabilities
- Second layer is used to find the bounding box coordinates

Limitation of this algorithm is that text is detected only on the basis of confidence score. If we set confidence score high then we face problem in detecting the whole text and if decrease it then it will detect those regions that do not contains text.

[2] Understanding of a convolutional neural network- This paper provides the overall understanding of what is neural network, what is ANN, what is CNN, complete methodology how to use the CNN, how to learn the dataset and how to map. Convolution Neural Network uses various layers for image feature extraction and converting them into feature vectors for further be classified by the Artificial Neural Network Layers.

The layers used in CNN are:

1. Convolutional Layer – for crossing the image with various feature detectors.
2. ReLu Layer – for converting the image into favorable pixel values
3. Max Pooling Layer – to reduce the image size for further processing
4. Flattening Layer – for converting the images to vector values
5. Full-Connection – connecting the flattening layer to the ANN layers

This method of classification can only be used for a single character recognition and detection and it cannot be used for text recognition as it would make it a very complex process.

[3] Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning:

This article is used to exploit and extensively evaluate CNN (Convolutional Neutral Network) in three important factors

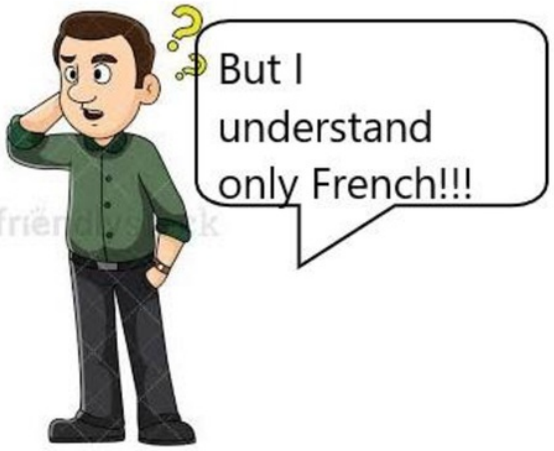
1. Dataset characteristics
2. Architecture
3. Transfer Learning

This evaluates CNN performance on two different computer-aided diagnosis applications:

1. Thoraco-abdominal lymph node detection
2. Interstitial lung disease classification.

Problem Statement

Many times, when you are travelling to a new country or area where the language or text is not same as your own then it is hard or sometimes impossible to understand the information conveyed in the symbols (menu list, road sign, document etc.) and texts. The main objective of this project is to make a reliable image to text translator using the help of various technologies and techniques like, OpenCV (For Digital Image Processing), Deep Learning, traditional Artificial Neural Network (ANN), Google translator, etc., to help the user to understand the native language in text or hand-written from any country by just loading the image into our program and finding the result in any language they understand. All of this process would be done through a pipe line which would contain the following steps i.e. Image Acquisition, Image Pre-processing, Segmentation, Classification and Reorganization, Character Pre-processing, Translation of text. Hand-written character recognition is hard, as people have different hand writing. Many techniques are being used in ANNs to recognizing the characters and symbols, choosing from various training methods and pattern recognition techniques would be challenging. We would provide more effective image translator that is versatile in understanding various symbols and texts in different languages.



Work scope

As earlier shown in the literature review section about the shortcomings of various research in the respective field of our project, we here list the shortcomings that we will be overcoming in our project-

CNN model with high accuracy -

Problem- Previous CNN models have less accuracy over the character recognition so they are unable to detect the text accurately.

Our solution- Our solution provides better trained CNN model of high accuracy of character recognition in natural scene images.

Building Text Translation module

Short comings- Previous project is not having any text translation module to convert in different languages.

Our solution- We introduced this module of text translation of natural scene images text detection and translation

Building Text Segmentation module

Short comings- Previous model has very less segmentation accuracy

Our solution- We have provided better segmentation module using sliding window technique

Constraint- Characters in text must not be linked.

Objectives

Our main objective is to build a software with the help of which user take the snap, upload in the software, user will select its native language, then software will convert into user's native language. It will very useful for the user to read the road instruction board, road signs, Information of any product etc. We have to make a CNN model of more than 90% accuracy for character recognition in natural scene images.

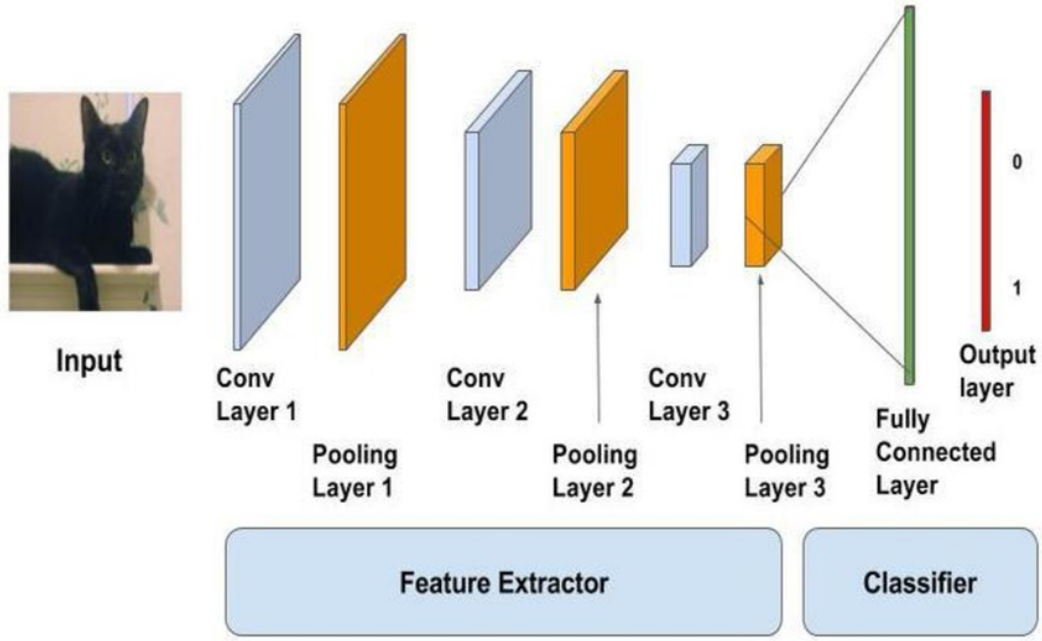
Following are the main goals that should be achieved in order for better productivity of tool:

- Detect maximum text present in the image
- Proper segmentation of a character in detected text
- Providing a tool for translating the detected text into user's native language

Motivation

When we travel to a country whose native language is English but the user's native language is some other. So, our motivation is to build a software with the help of which user take the snap, upload in the software, user will select its native language, then software will convert into user's native language. It will very useful for the user to read the road instruction board, road signs, Information of any product etc.

Methodology



In this section, the methods that are going to be used in this project are described in details.

The process of character and symbol recognition is generally described in a pipeline:

1. **Image acquisition:** In this process the image containing the text or symbol is acquired by taking a snap of the surrounding.

2. **Image pre-processing (processing also includes Text Detection):** The image is then pre-processed by making it ready for operations. It is done by cleaning the image from noise, various filters are added, morphological operations are applied, etc. Then text is detected within the rectangular window.

3. **Character Segmentation:** In this step the image is segmented to fetch the required characters from it. Segmentation technique is used from digital image processing and individual characters in the image are uniformly resized into 30x20 pixels to make the processing more efficient.

4. **Character recognition:** This is the decision-making part of the pipeline where neural networks are trained and applied by various techniques like feed forward back propagation, re-enforcement learning algorithms etc.

1. Text detection

2. Character segmentation

3. Character classification

At last when the characters are recognized the translator detects the language and asks for favorable language to translate to. In this step we would just be using google translator for fast translation.

ALGORITHM:

Convolutional Neural Network Model

Step 1) Import keras packages for layers of Convolutional Neural Network
Step 2) Import model_from_json package for saving and loading model from keras.models

Step 3) Initialize layers of model with Sequential

Step 4) Add layers:

1. First Convolution layer with 64 feature detectors of 3x3 dimension, Input size of images is 32x32x3 and activation function is relu (rectifier function)
2. First Max-Pooling layer with size 2x2
3. Second Convolution layer same as first layer with forwarded input size of images
4. Second Max-Pooling with size 2x2
5. Flattening layer for flattening the image
6. 3 Dense Layers of ANN with node units 256, 160, 26 and activation function relu, relu and softmax respectively

Step 5) Complie model with optimizer, loss function, and accuracy metrics.

Step 6) Image processing with the help of ImageDataGenerator

Step 7) Scale the pixel values of the input image for processing

Step 8) Enter the training_set and test_set location with target size, batch size and class mode

Step 9) Run the model for training using fit_generation function by setting steps of epoch, number of epocs and validation steps

Step 10) After training process save the model into an json and h5 file for later loading of model.

```
# serialize model to JSON
model_json = classifier.to_json()
with open("model_2.json", "w") as json_file:
   json_file.write(model_json)
# serialize weights to HDF5
classifier.save_weights("model_2.h5")
print("Saved model to disk")
```

Step 11) End.

EAST DECETION:

Step 1) Start.

Step 2) Image Acquisition with the image address

Step 3) Initialize constant variables like min_confidence, width and height for scaling the image.

Step 4) Resize the image to the scaling size for text detection process and get its height and width.

Step 5) Make a list of layerNames to be used in EAST for getting the output probabilities and deriving the coordinated of bounding box

Step 6) Read the binary model file of frozen_east_text_detection.pb for deep learning network

Step 7) Construct a blob for the image using the image

Step 8) Input the blob to the network

Step 9) Set the layers of the network from layerNames and get scores and geometry list from the network output

Step 10) Loop over the rows in score list

Step 11) In the loop get the score data for each geometric rectangle with its dimenstions and angles

Step 12) Loop over the columns

Step 13) Check if scoreData[x] < min_confidence

Step 14) Compute offset factor with 4*times value

Step 15) Set sin and cosine angles if the text is rotated with some angle

Step 16) Compute Starting and Ending X and Y coordinates

Step 17) Save rectangle and confidence score in a list

Step 18) Exit column and row loop when done

Step 19) Apply non_max_suppersion to array of rectangle with confidence scores as probabilities

Step 20) End.

CROPPING OF CHARATERS WITH THE HELP OF MORPHOLOGICAL OPERATIONS AND THRESHOLDING

Step 1) Start.

Step 2) Initialize crop_img list to store all character img

Step 3) Make a function to detect if a rectangle r1 contains rectangle r2.

Step 3) Loop throw all the rectangles

Step 4) Scale the coordinated to the ratio of scaling

Step 5) Crop the original image with the coordinates

Step 6) Change the crop image to gray-scale image

Step 7) Set the threshold constant to the mean value of the gray-scale pixel values.

Step 8) Make a binary image using threshold constant

Step 9) perform dilation and erosion on the image using a kernel of 2x2 ones size

Step 10) Find contours in the binary image

Step 11) Get polygons and rectangle list of all contours

Step 12) Sort the rectangles in increasing starting x coordinate order

Step 13) Initialize char_crop list to store all the charaters cropped of a text.

Step 14) Take all the rectangles that are not inside of another rectangle.

Step 15) Crop the characters using rectangles in rectangle list

Step 16) Store the char_crop of a text in crop_img.

Step 17) Exit the loop

Start 18) End.

LOADING CNN MODEL AND PREDICTION

Step 1) Start.

Step 2) Load model from json and weights from h5 file

```
# load json and create model
    json_file = open('model.json', 'r')
    loaded_model_json = json_file.read()
    json_file.close()
    loaded_model = model_from_json(loaded_model_json)
    # load weights into new model
    loaded_model.load_weights("model.h5")
```

Step 3) Loop through all text in crop_img

Step 4) Loop through all character images in text

Step 5) Resize the image to the target image of the model

Step 6) Expand the image dimension at axis = 0

Step 7) Exit the character image loop

Step 8) Add spacing in the text.

Step 9) Exit the text loop.

Step 10) Print the prediction result

Step 11) End.

LANGUAGE TRANSLATION USING GOOGLETRANS

Step 1) Import Translator from googletrans package

Step 2) Initialize Translator()

Step 3) Translate the predicted result into any language in the Language dictionary

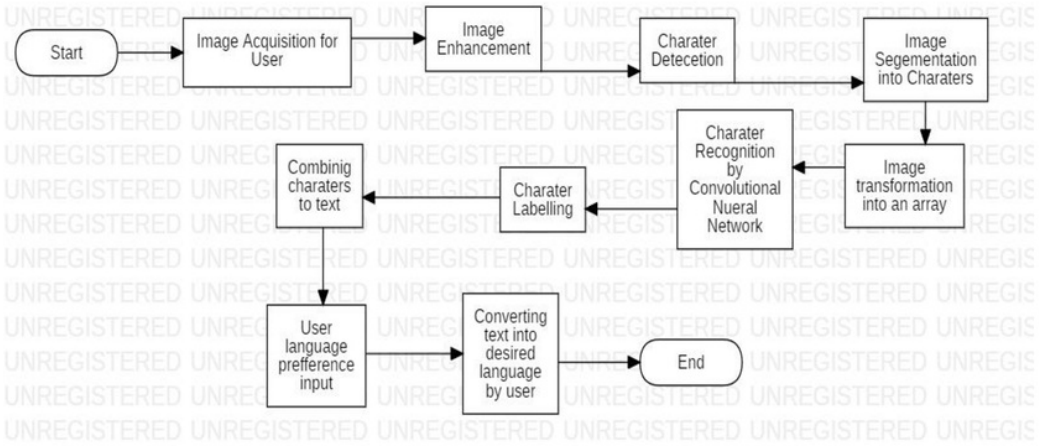
```
LANGUAGES = {
        'af': 'afrikaans',        'sq': 'albanian',        'am': 'amharic',        'ar': 'arabic',        'hy':
'armenian',        'az': 'azerbaijani',        'eu': 'basque',        'be': 'belarusian',        'bn':
'bengali',        'bs': 'bosnian',        'bg': 'bulgarian',        'ca': 'catalan',        'ceb': 'cebuano',        'ny':
'chichewa',        'zh-cn': 'chinese (simplified)',        'zh-tw': 'chinese (traditional)',        'co':
'corsican',        'hr': 'croatian',        'cs': 'czech',        'da': 'danish',        'nl': 'dutch',        'en':
'english',        'eo': 'esperanto',        'et': 'estonian',        'tl': 'filipino',        'fi': 'finnish',        'fr':
'french',        'fy': 'frisian',        'gl': 'galician',        'ka': 'georgian',        'de': 'german',        'el':
'greek',        'gu': 'gujarati',        'ht': 'haitian creole',        'ha': 'hausa',        'haw': 'hawaiian',        'iw':
'hebrew',        'hi': 'hindi',        'hmn': 'hmong',        'hu': 'hungarian',        'is': 'icelandic',        'ig':
'igbo',        'id': 'indonesian',        'ga': 'irish',        'it': 'italian',        'ja': 'japanese',        'jw':
'javanese',        'kn': 'kannada',        'kk': 'kazakh',        'km': 'khmer',        'ko': 'korean',        'ku':
'kurdish (kurmanji)',        'ky': 'kyrgyz',        'lo': 'lao',        'la': 'latin',        'lv': 'latvian',        'lt':
'lithuanian',        'lb': 'luxembourgish',        'mk': 'macedonian',        'mg': 'malagasy',        'ms':
'malay',        'ml': 'malayalam',        'mt': 'maltese',        'mi': 'maori',        'mr': 'marathi',        'mn':
'mongolian',        'my': 'myanmar (burmese)',        'ne': 'nepali',        'no': 'norwegian',        'ps':
'pashto',        'fa': 'persian',        'pl': 'polish',        'pt': 'portuguese',        'pa': 'punjabi',        'ro':
'romanian',        'ru': 'russian',        'sm': 'samoan',        'gd': 'scots gaelic',        'sr': 'serbian',        'st':
'sesotho',        'sn': 'shona',        'sd': 'sindhi',        'si': 'sinhala',        'sk': 'slovak',        'sl':
'slovenian',        'so': 'somali',        'es': 'spanish',        'su': 'sundanese',        'sw': 'swahili',        'sv':
'swedish',        'tg': 'tajik',        'ta': 'tamil',        'te': 'telugu',        'th': 'thai',        'tr':
'turkish',        'uk': 'ukrainian',        'ur': 'urdu',        'uz': 'uzbek',        'vi': 'vietnamese',        'cy':
'welsh',        'xh': 'xhosa',        'yi': 'yiddish',        'yo': 'yoruba',        'zu': 'zulu',        'fil':
'Filipino',        'he': 'Hebrew'

    }
```

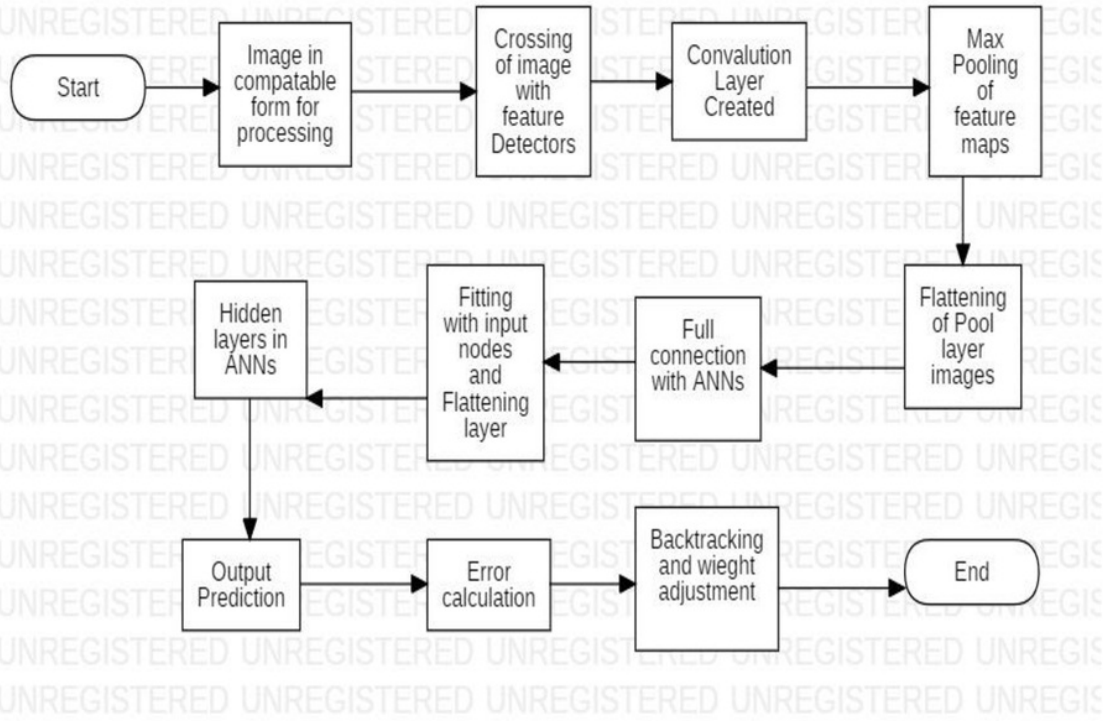
Step 4) Print the translated text with .text object

Step 5) End.

Overall Flow Chart:

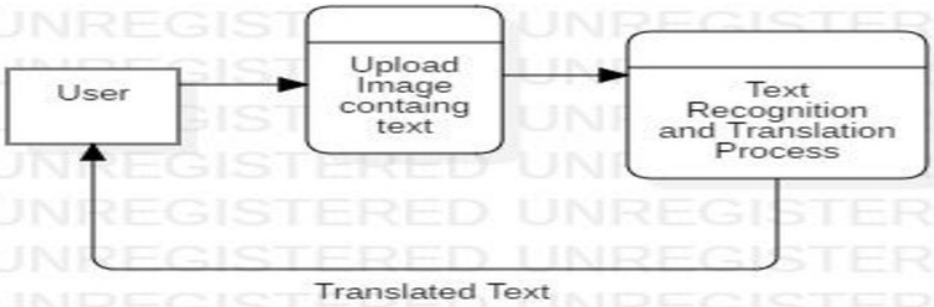


CNN training algorithm flowchart

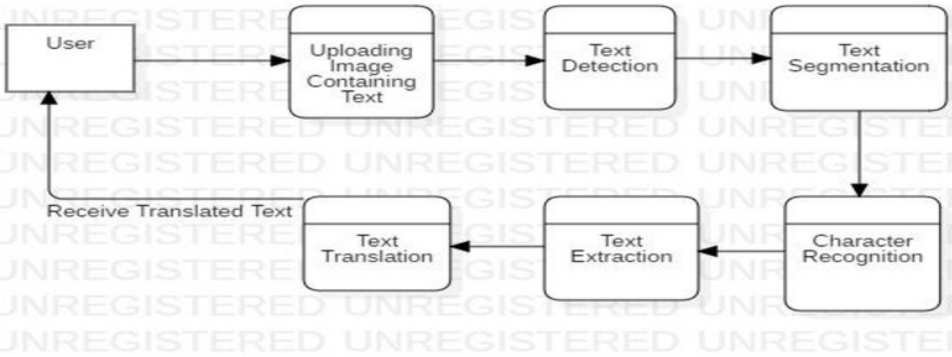


DFD:

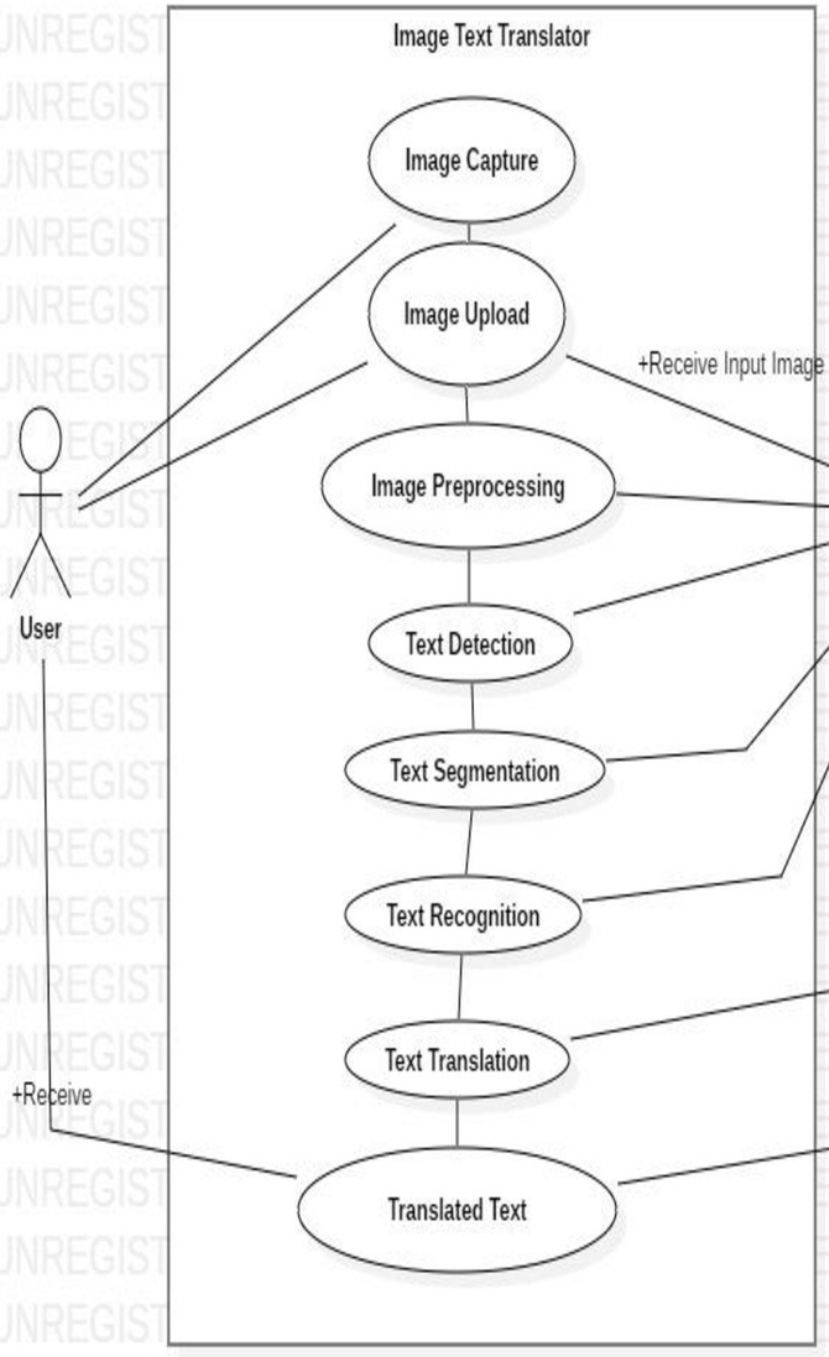
Level 0



Level 1



Use Case Diagram:



Implementation Screenshots

Figure 1

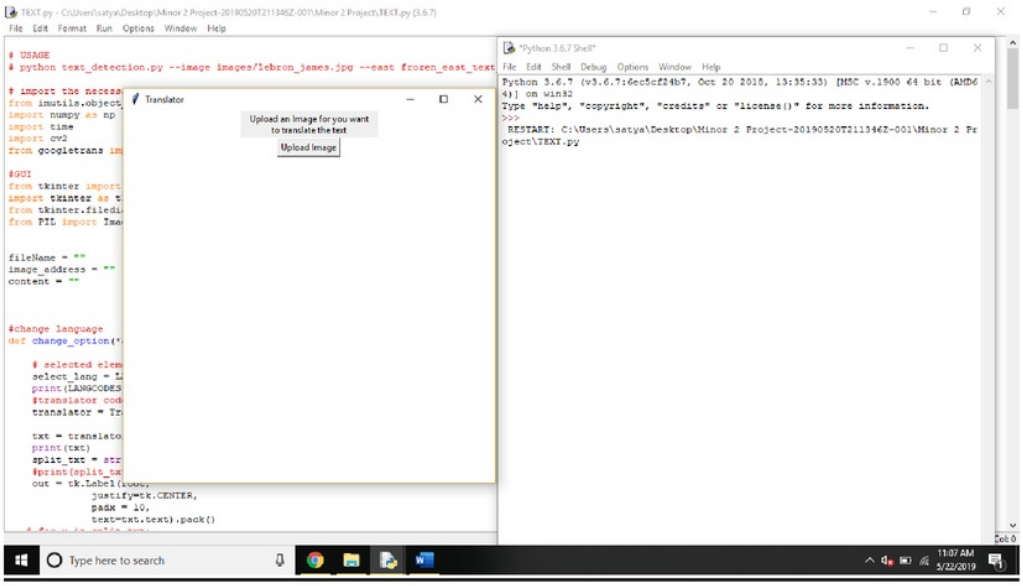


Figure 2

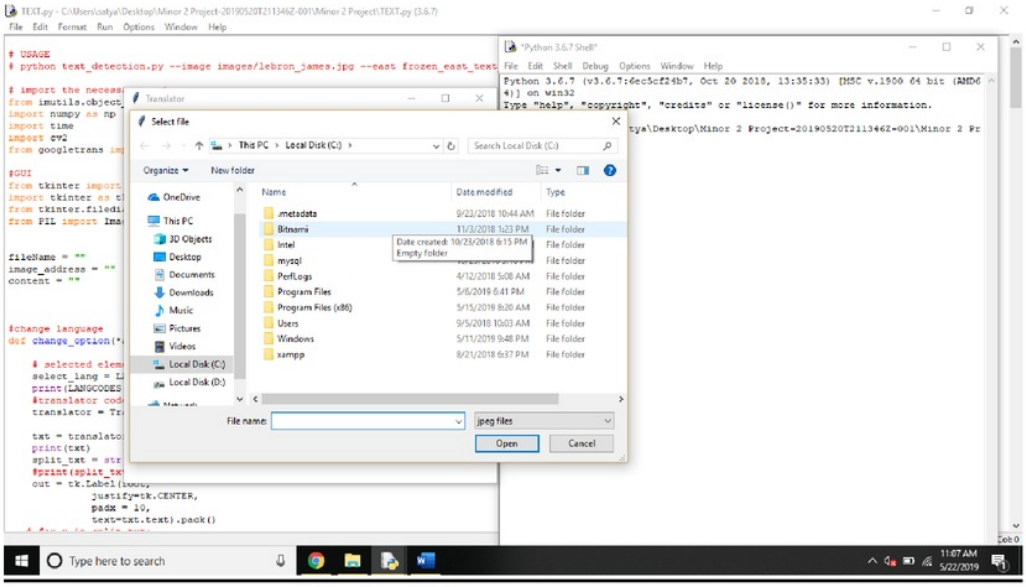


Figure 3

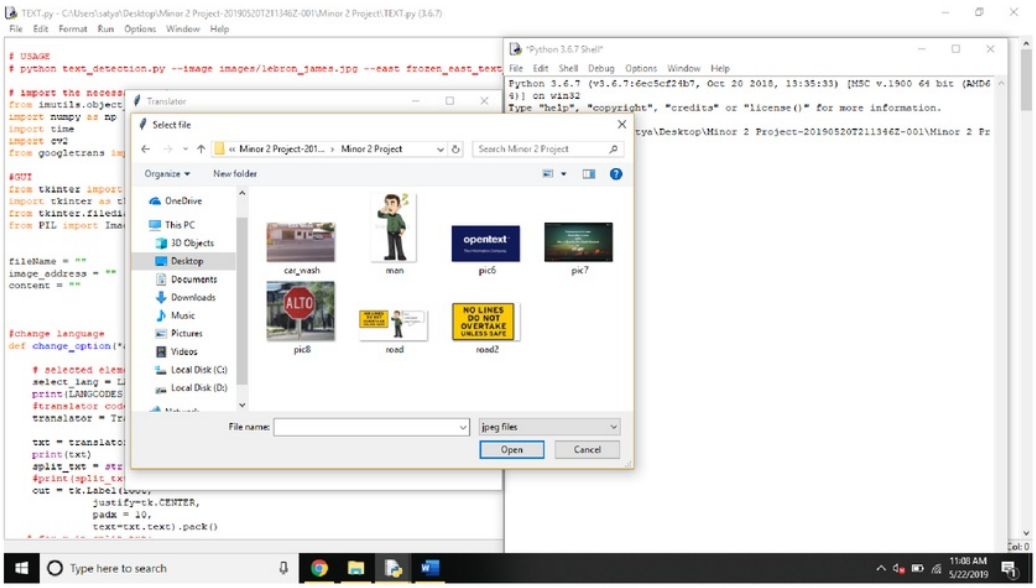


Figure 4

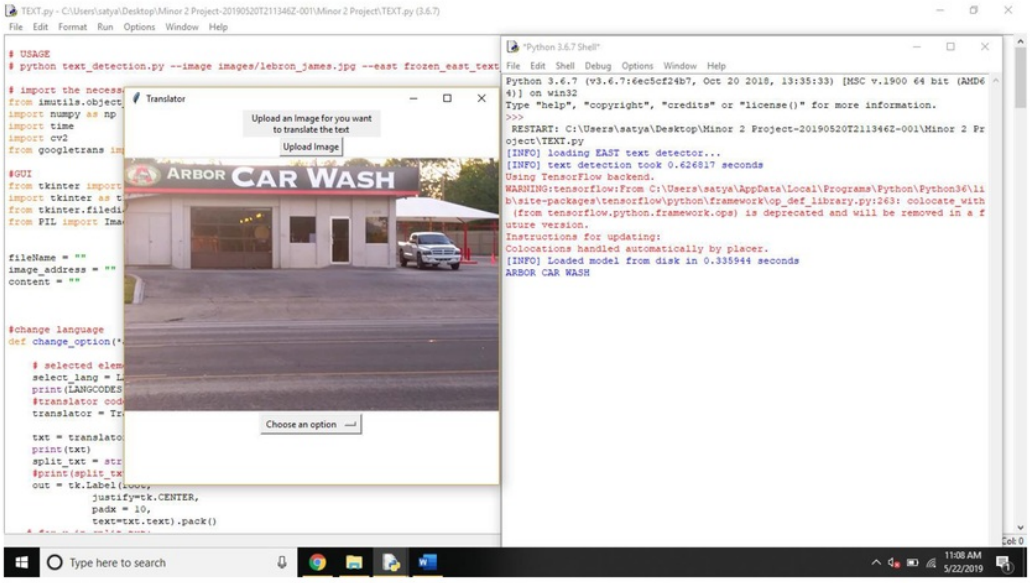


Figure 5

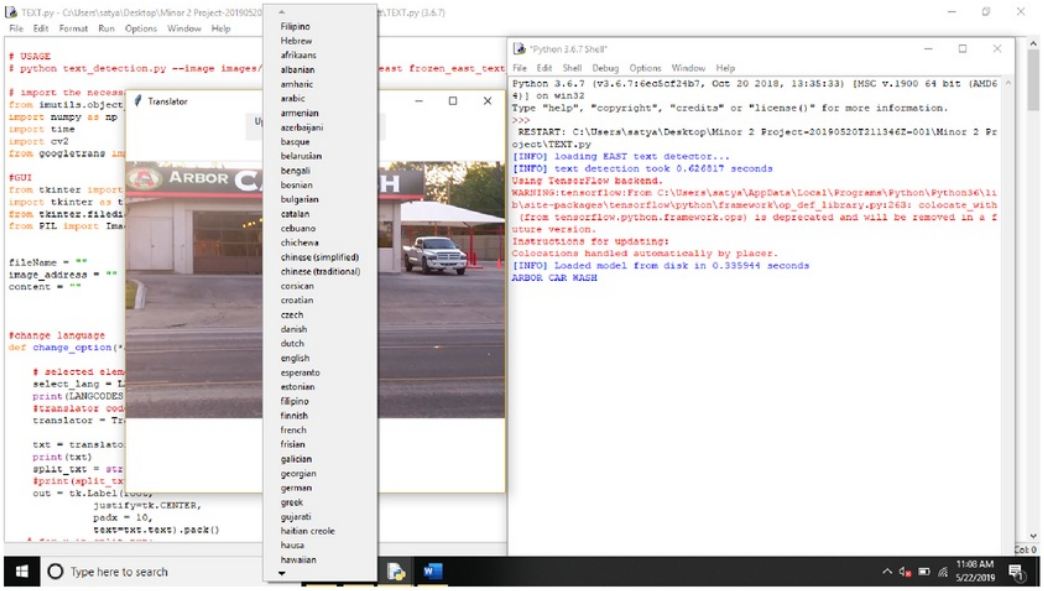


Figure 6

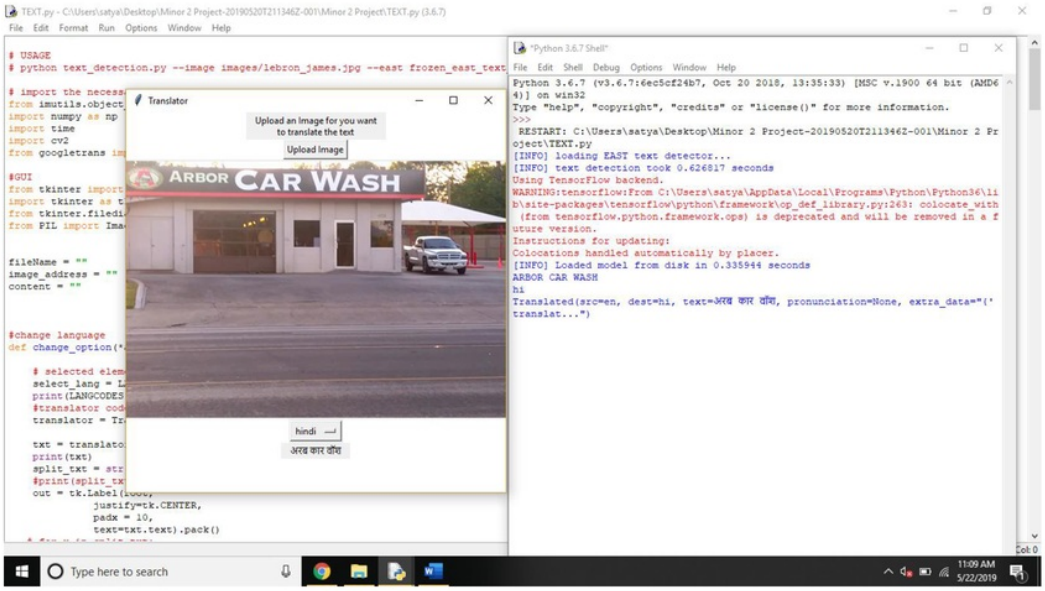


Figure 7

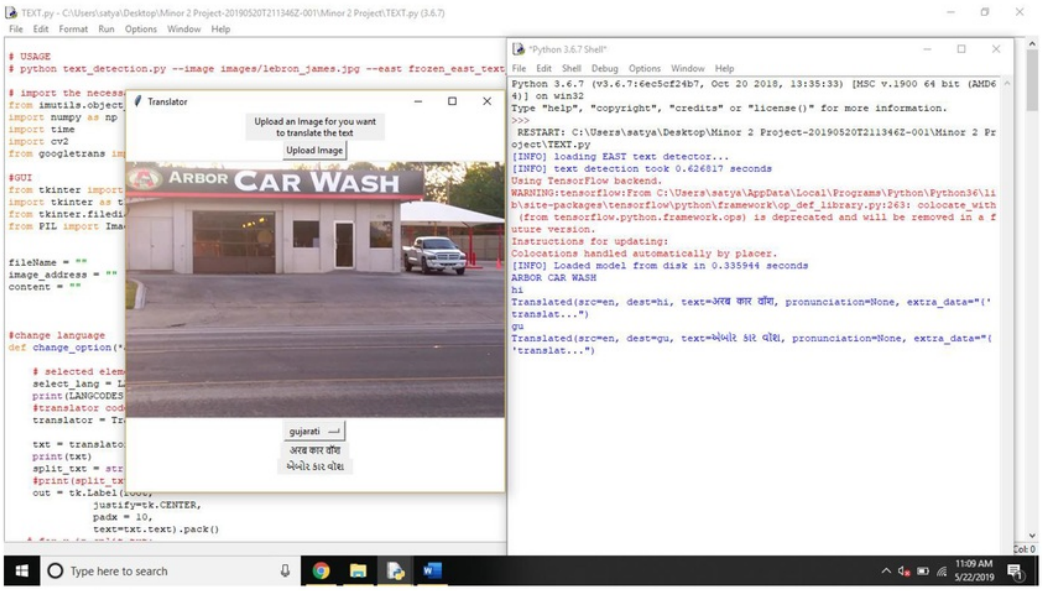


Figure 8

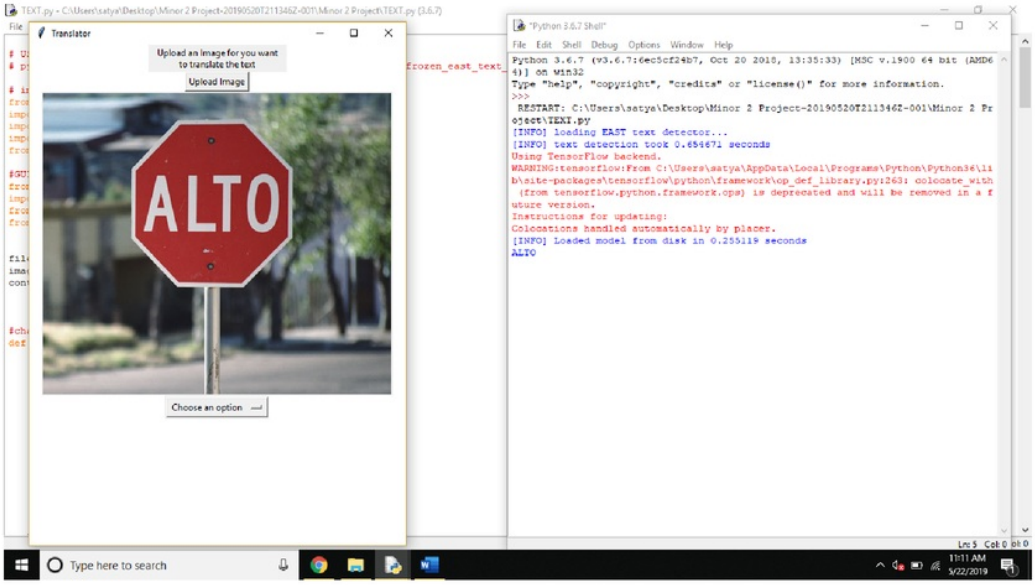
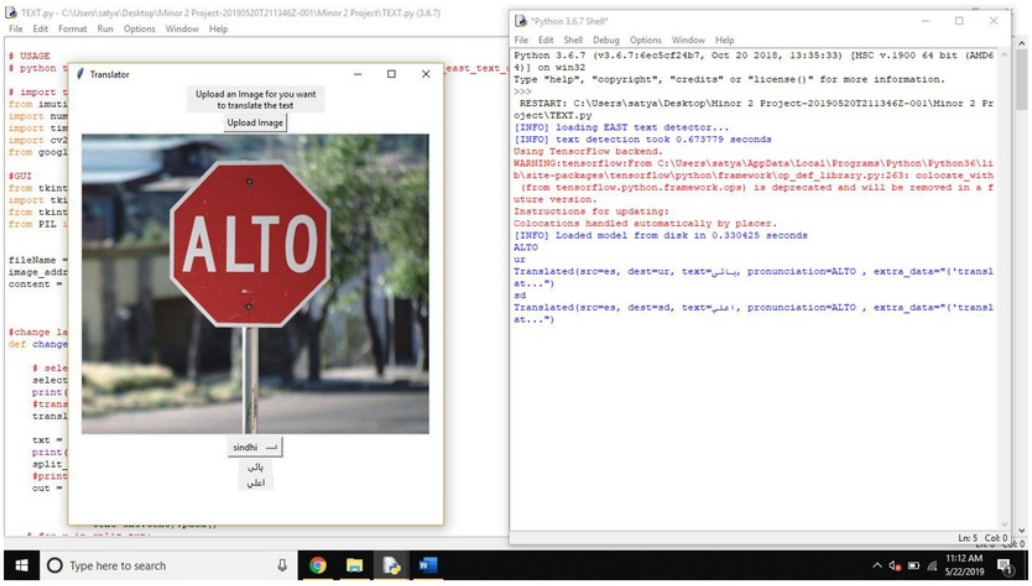


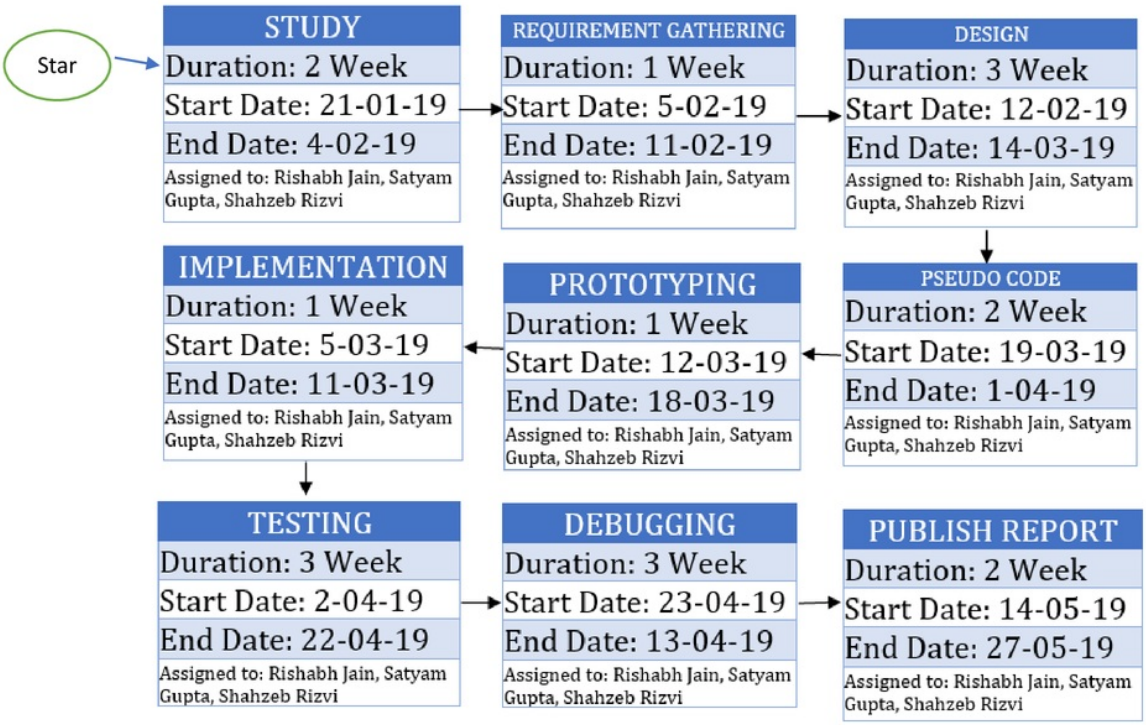
Figure 9



System Requirements (Software/Hardware)

- Hardware Interface:
  - 64 bits processor architecture supported by LINUX
  - Minimum RAM requirement for proper functioning is 4 GB.
  - Required input as well as output devices
  - Webcam required & sufficient graphic card for operation.
- Software Interface
  - Python IDE
  - Tensorflow, keras, pillow, tkinter, tesseract.

Schedule (PERT Chart)



Conclusion:

We have built a model that is used by our system to detect the text from natural scene images, extract the text and then convert into user's native language.

The previous work done is not much accurate for the text segmentation due to which there is difficulty in accurate text recognition.

Our model for text segmentation is somehow better than previous and provide better segmentation and recognition for natural scene text. It is better due to following reasons:

- It uses EAST detector for better text detection.
- It uses better noise removal methodology.
- It uses methodology for better segmentation of detected text.
- High accuracy of our CNN model which recognize the character.
- It uses convex hull, contours, thresholding, bounding box and morphological operations for character segmentation.

The future advancements to our work could be enormous and some of it have been also tried by us. The advancements could be-

1) More accuracy for better segmentation of text because when the characters are very near or linked in our image, our model may fail to segment the text which ultimately fail the whole model.

2) Better noise removal can be done for better recognition of text.

References

[1] Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., & Liang, J. (2017). EAST: An Efficient and Accurate Scene Text Detector. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–1, 2017.

[2] Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. 2017 International Conference on Engineering and Technology (ICET). Doi:10.1109/icengtechnol.2017.8308186

[3] Shin, H.-C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., … Summers, R. M. (2016). Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. IEEE Transactions on Medical Imaging, 35(5), 1298.doi:10.1109/tmi.2016.2528162

IMAGE TO TEXT CONVERSION IN FOREIGN LANGUAGES USING DOCUMENT IMAGE PROCESSING TECHNIQUE

ORIGINALITY REPORT

%13	%11	%4	%9
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	py-googletrans.readthedocs.io Internet Source	%5
2	felicitysmoak.info.tm Internet Source	%3
3	edlib.net Internet Source	%3
4	Submitted to University of Petroleum and Energy Studies Student Paper	%2

EXCLUDE QUOTES	ON	EXCLUDE MATCHES	< 2%
EXCLUDE BIBLIOGRAPHY	ON		