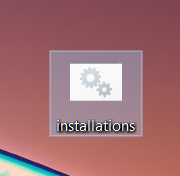
**DO NOT SUBMIT THIS FILE AS PROJECT REPORT**

**INSTALLATIONS**

1. Run the batch file(windows):



Download and open “installations.bat”

It will take care of the required libraries

Takes time as it downloads the libraries and requires data connection

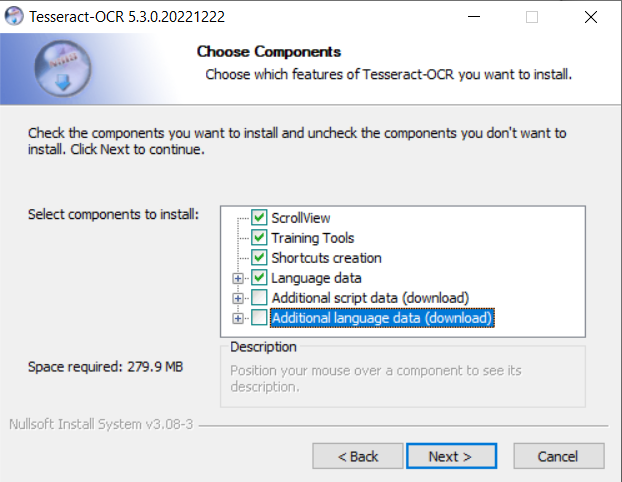
1. Install Pytesseract:

Follow this video till 2 minutes

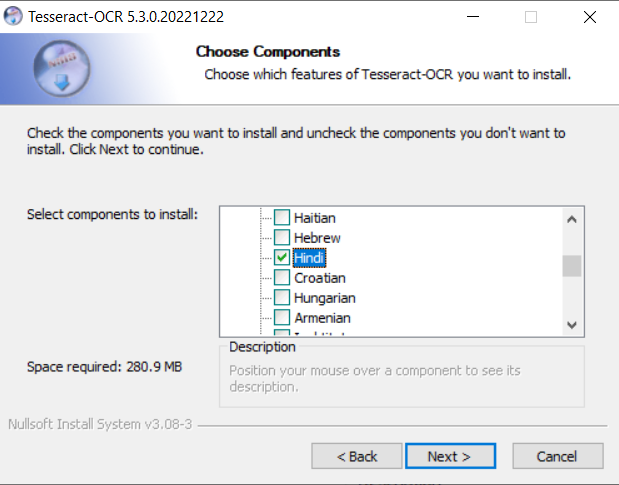
<https://www.youtube.com/watch?v=DG5D8A3zi4o>

Open the tesseract installer file you just downloaded and complete the installation considering the below points

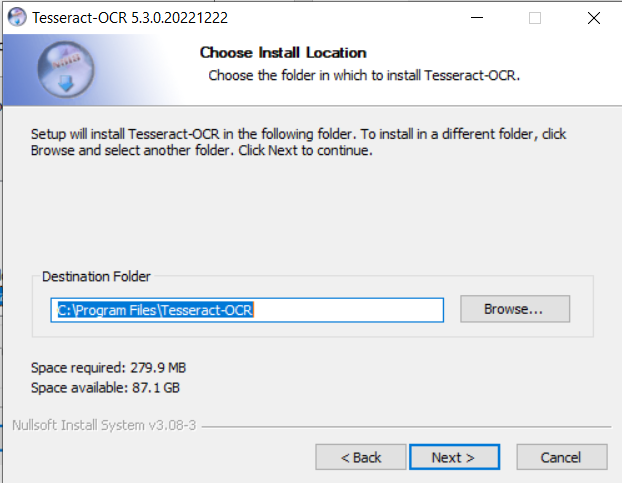
* Expand the additional language data (download) option



* Select Hindi only



* Copy the installation path that appears while installing

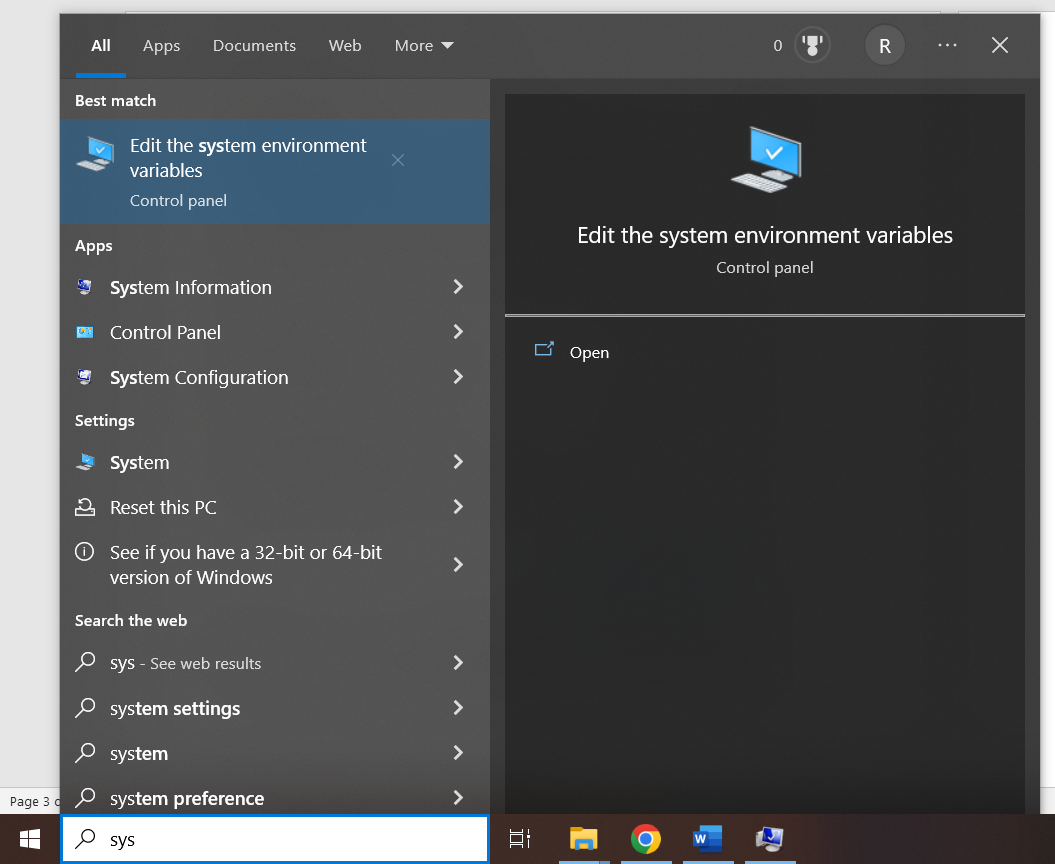


Finish the installation

It will take some time and requires data connection

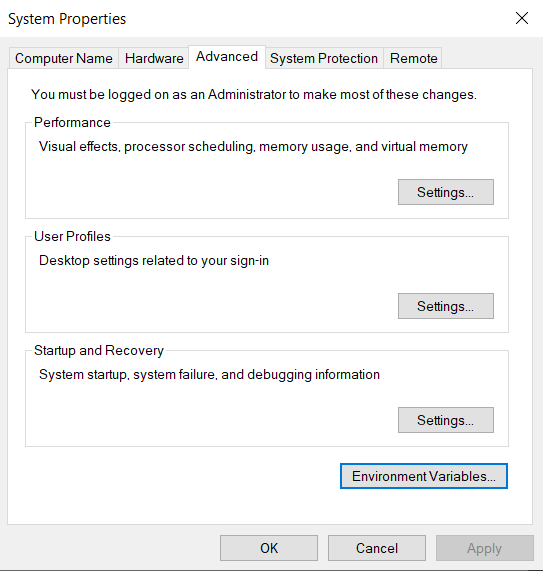
1. Adding Tesseract to path

* Search for Edit the system environment variables

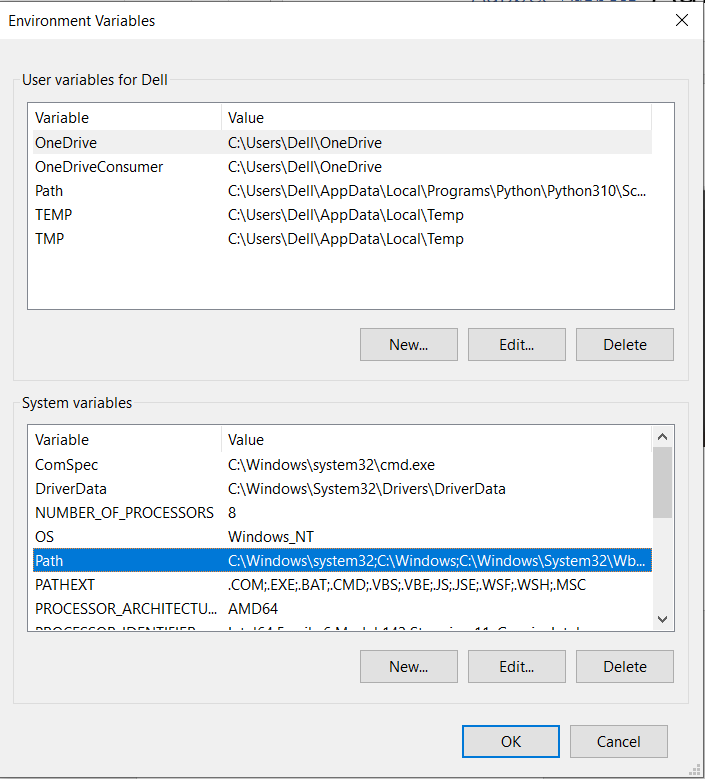


* The below window will appear

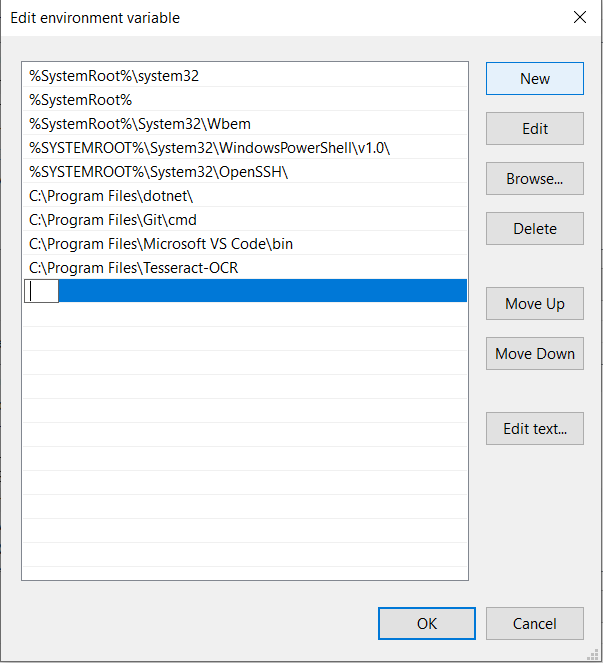
Click on Environment Variables



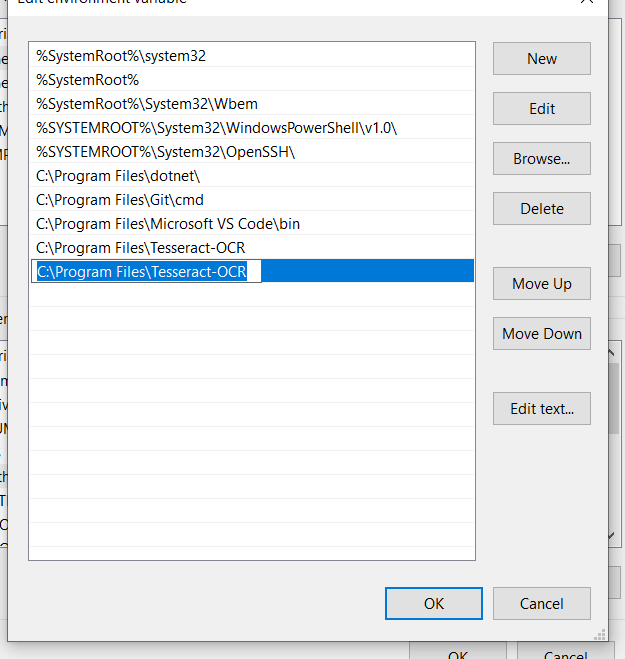
* Double click on “PATH” under system variables



* Click on New



* Paste the tesseract installation path that you copied during the installation here like this (ONLY ONCE)



Click OK -> OK->OK

You can close it now

**RUNNING THE FILE**

* Files required to run
  + Dataset\_san\_en.csv (This is the small dataset)
  + translator.py
  + small\_translator.h5 (This holds trained Model weights for smaller dataset)
  + 2.jpg (This is the background image. You can change it to some other image yourself if you want by changing the file name in translator.py file ka - GUI section)

TO RUN IT WITH LARGER DATASET IN FUTURE, CHANGE THE FILE NAMES IN translator.py

I AM YET TO SHARE THE MODEL TRAINED ON LARGER DATASET

CANNOT GUARANTEE ABOUT LARGER DATASET IT’S STILL TRAINING AND I AM SEEING A SATURATION AT 53% ACCURACY IT WILL TAKE MORE TIME FOR ME TO CONFIRM

Open CMD in the folder where the above files are located

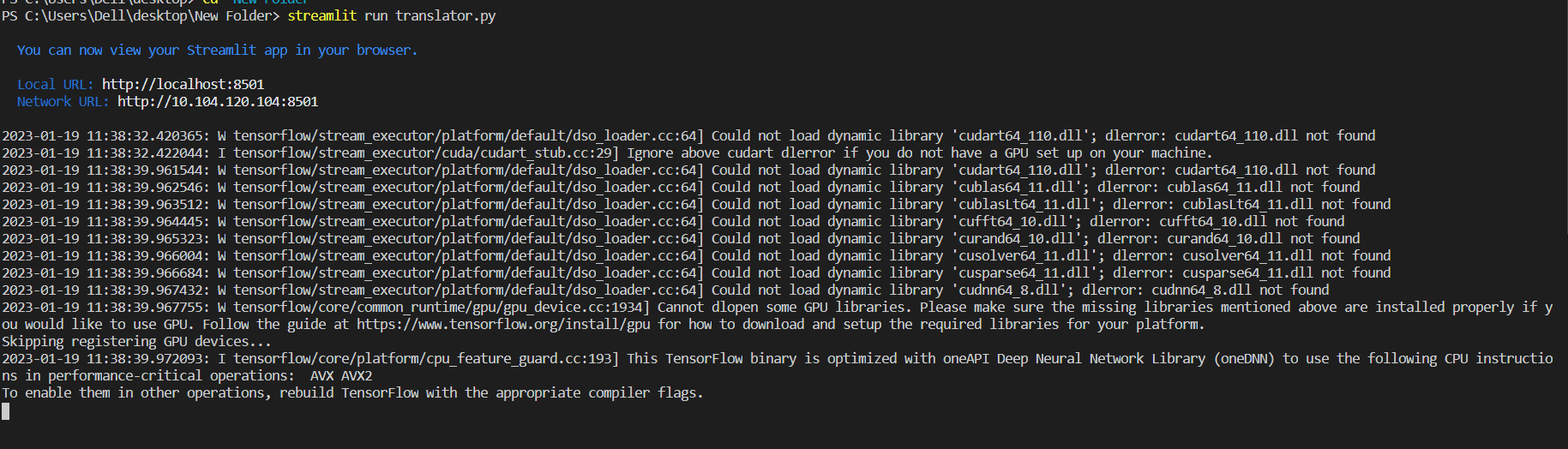
(I am using VS code terminal. Works same does not matter)

Type the following command

streamlit run translator.py

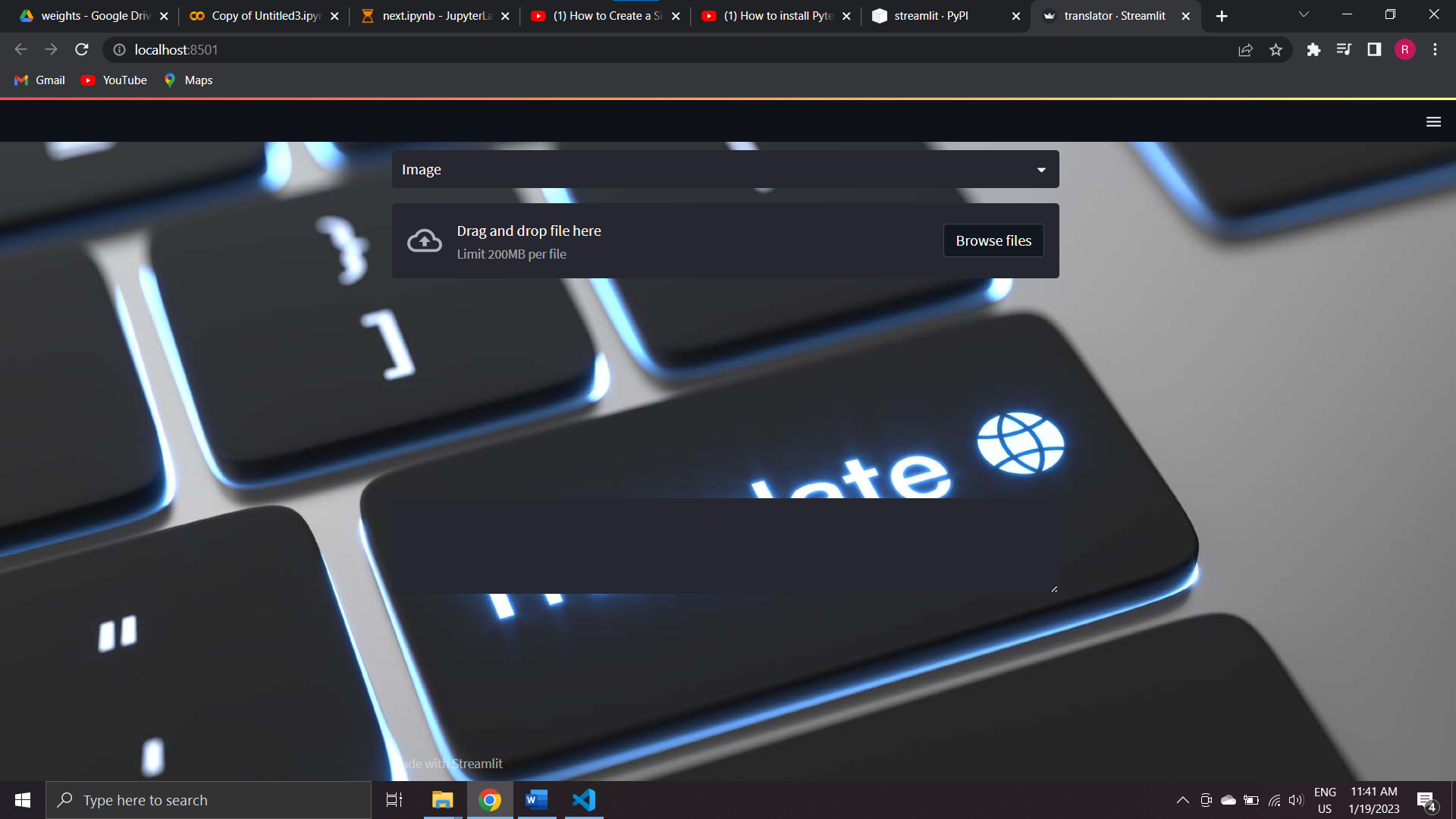


The above command will result in something like this



This will open a new tab in your default browser go check that

The new tab looks like this



CANNOT DO MUCH WITH UI AS ALL OF IT COMES INBUILT (NOT FLEXIBLE AS HTML CSS PHP)

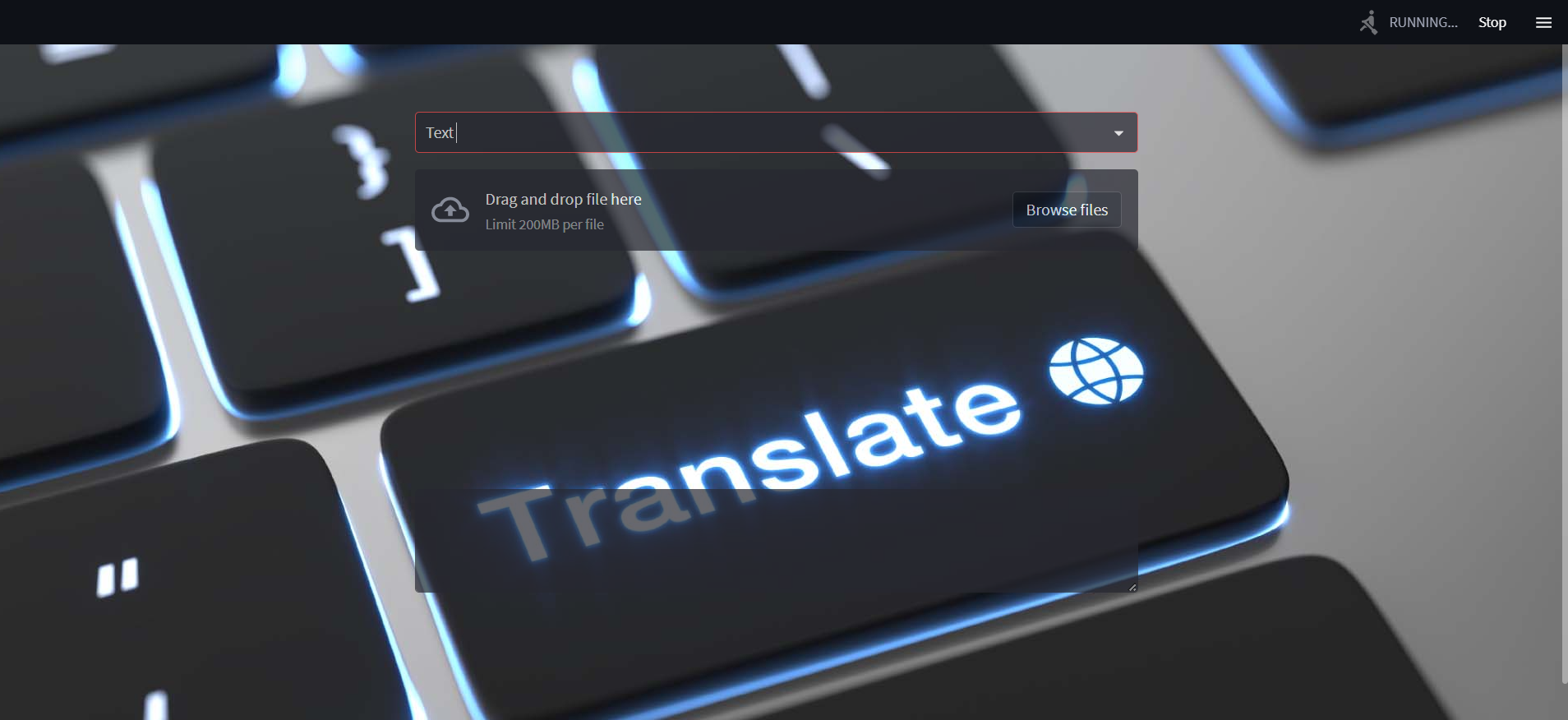
YOU CAN CHANGE THE BACKGROUND AS MENTIONE EARLIER

PLEASE ADJUST WITH THE POSITIONS OF THOSE WIDGETS

If in case the new tab does not open or you closed it by mistake,

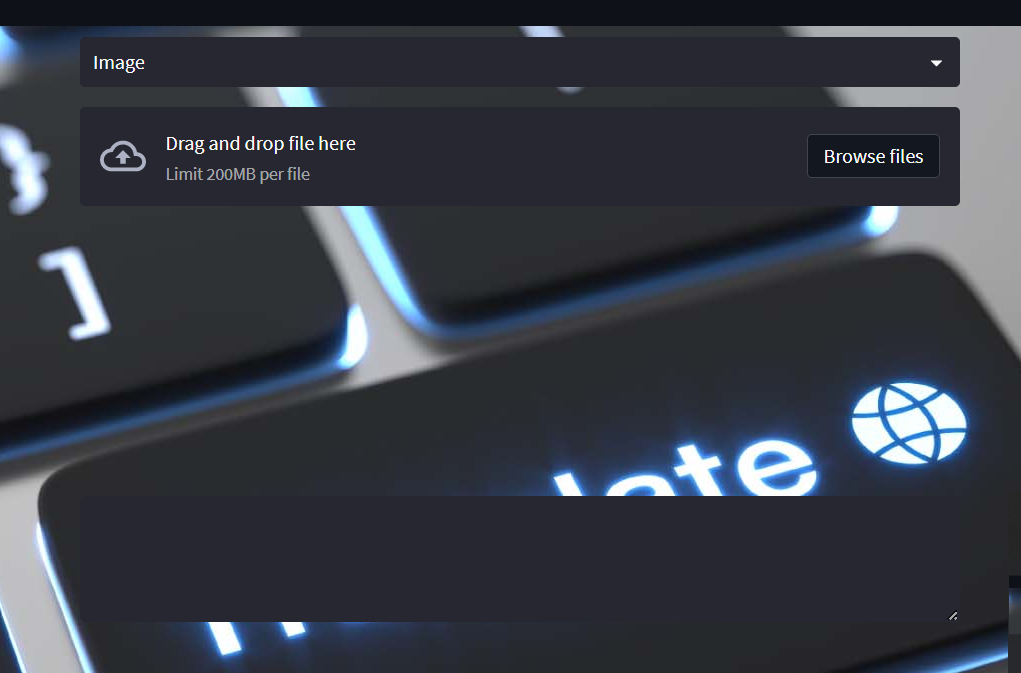
http://localhost:8501/

You can change your input preferences as Text or Image by clicking here (Highlighted in orange box below) but the switching will take a few seconds

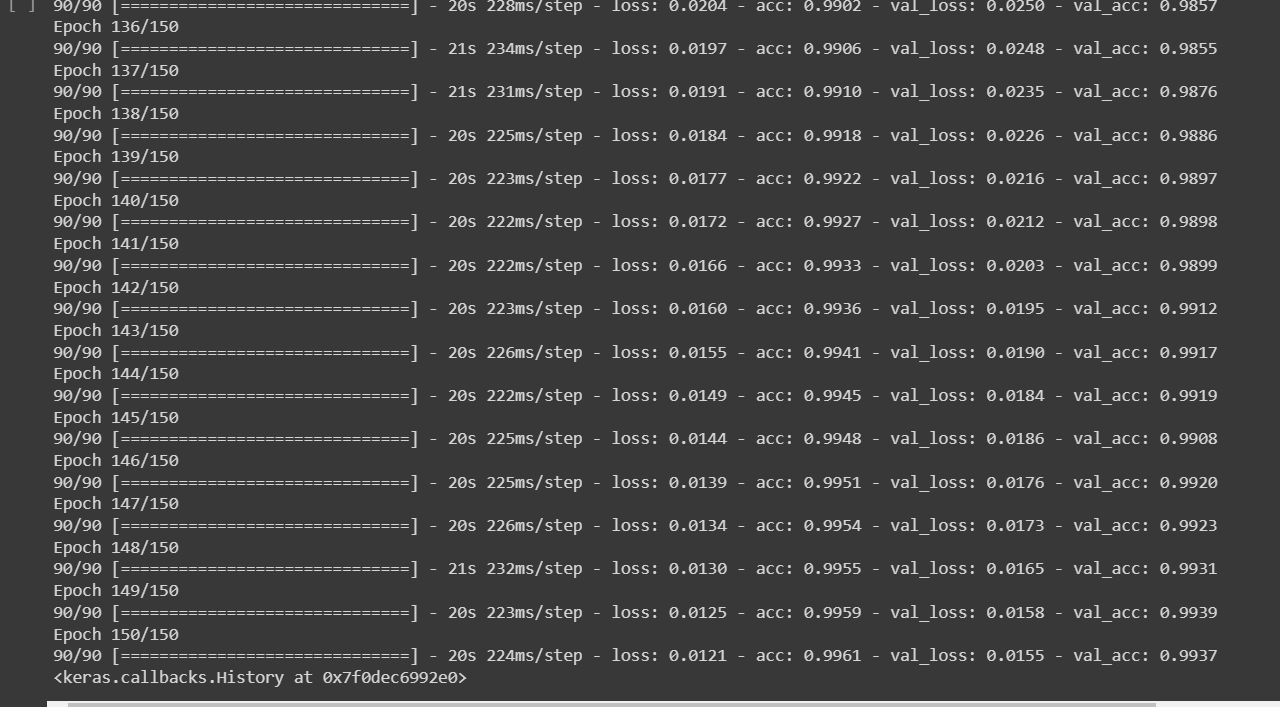


Whenever you change the input type or give some input, the execution status can be seen on the top right corner as seen in the screenshot above

Wait for it to Finish



* In case of Image as input
  + Click on browse files and select the image with Sanskrit text (does not work well with handwritten texts. Works well with computer texts only)
  + It will take some seconds to generate the output
  + Output will be displayed in the black box below
  + In case there are some new words in the inputs that are not already present in the dataset on which it was trained, the output will be “Couldn’t find an appropriate translation”
* In case of text input wait the text box to appear
  + Then you can paste your textual input there
  + It will take some seconds to generate the output
  + Output will be displayed in the black box below
  + In case there are some new words in the inputs that are not already present in the dataset on which it was trained, the output will be “Couldn’t find an appropriate translation”



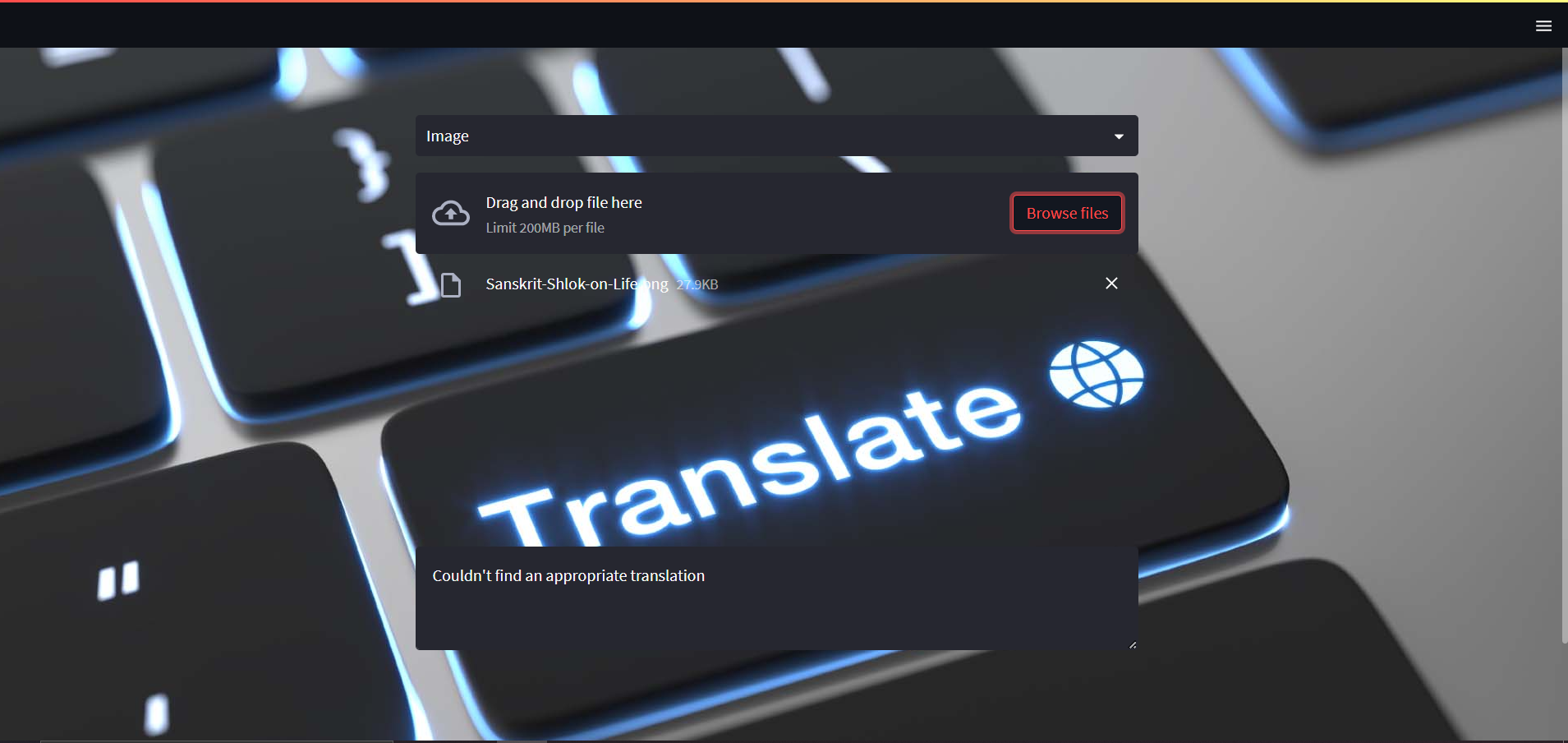
Accuracy of the smaller dataset is 0.9961 and other parameters can be checked from the above screenshot

THE BELOW TABLES ARE ONLY FOR THE SMALL DATASET

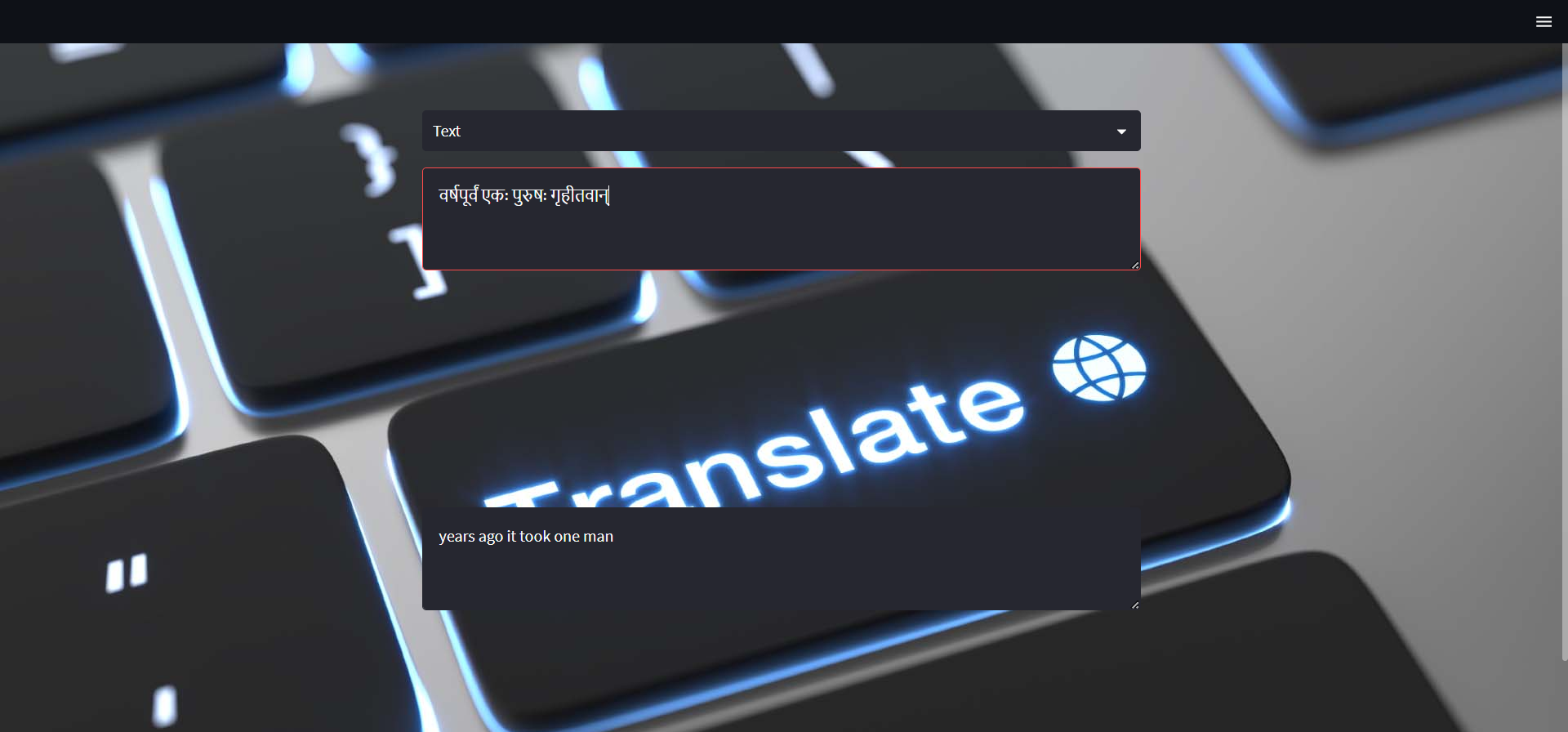
|  |  |
| --- | --- |
| Total number of epochs ran | 150 |
| Loss | 0.0121 |
| Accuracy | 0.9961 |
| Validation loss | 0.0155 |
| Validation accuracy | 0.9937 |

**Demo screenshots:**

* Image input



* Textual input



To terminate the process

CTRL+C in the CMD where you typed the command “streamlit run translator.py”

**CODE COMPONENTS**

**Tried methods**

1. OCR using Pytesseract
2. Denoising of Images
3. Blur Image detection

* Random Forest Classifier
* SVM
* K-Means clustering

1. Sharpening of Images
2. Encoder Decoder Model for translation

**Useful methods**

1. OCR using Pytesseract
2. Encoder Decoder Model for translation

**Justification**

1. Denoising of Images was done using Auto encoders

<https://www.youtube.com/watch?v=N9gER1axl5E>

The dataset used for this purpose was taken from Kaggle which consisted of English texts only but the model was seen to work well with Hindi/Sanskrit texts as well.

The results obtained were satisfactory but with the denoising, performance of the Pytesseract used for OCR was seen to degrade.

1. Blur image detection

The dataset used for this purpose was hindi text images obtained from google by web scraping. Dataset has been further populated using data augmentation technique. Blur images have been generated using cv2.blur().

<https://www.youtube.com/watch?v=rfM4DaLTkMs>

Dataset size obtained through web scraping: 200

Useful data: around 30%

Final dataset size including augmented data: 111(sharp) + 111(blur)

Two models were trained and compared for this purpose.

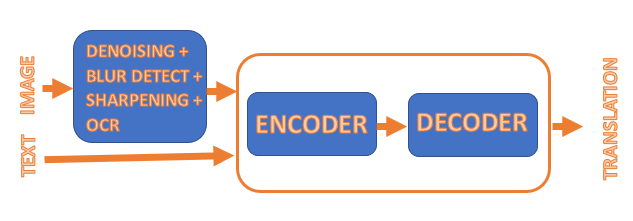
* Random Forest Classifier
  + Trains in few seconds but the accuracy was found to be 15.5555%
  + <https://www.youtube.com/watch?v=WkFtIqWmX9o>
* SVM
  + Accuracy was found to be 48.8888%
  + <https://www.youtube.com/watch?v=xLkk6MUrvrw>
* K-Means clustering
  + Accuracy was found to be 50%
  + <https://www.youtube.com/watch?v=CLKW6uWJtTc>

Conclusion: None of them are accurate enough to be used for the purpose

1. Sharpening of Image:

* Sharpening kernel was used for the purpose
* Results obtained didn’t work well with Pytesseract OCR

**INITIAL ARHITECTURE**

****

**INITIAL APPROACH**

* **Case: Image input**
  + Denoise the input image
  + Check if the image is blur
  + Sharpen if found blur from above step
  + Extraction of text from the image using pytesseract OCR
  + Extracted text is fed as input to the encoder-decoder model
  + Model gives appropriate translations
* **Case: Textual input**
  + Text is directly fed as an input to the encoder-decoder model
  + Model generates appropriate translations

**FINAL ARCHITECTURE**

****

**FINAL APPROACH**

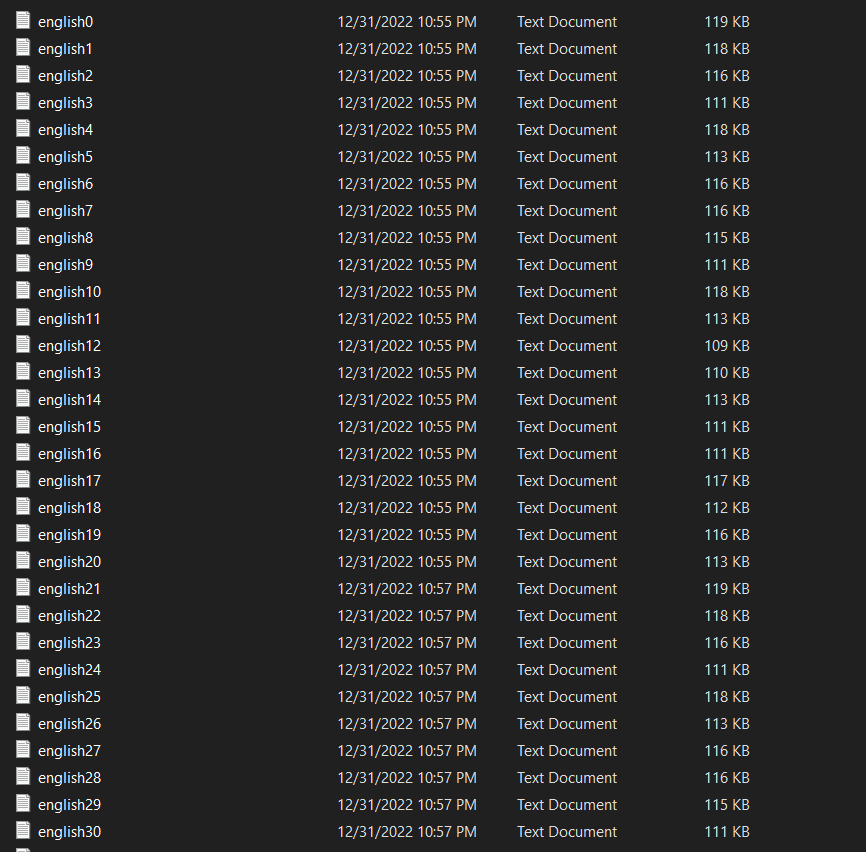
* **Case: Image input**
  + Extraction of text from the image using pytesseract OCR
  + Extracted text is fed as input to the encoder-decoder model
  + Model gives appropriate translations
* **Case: Textual input**
  + Text is directly fed as an input to the encoder-decoder model
  + Model generates appropriate translations

**Assumption is made that the Sanskrit text that will be fed as inputs to the model are written in hindi language**

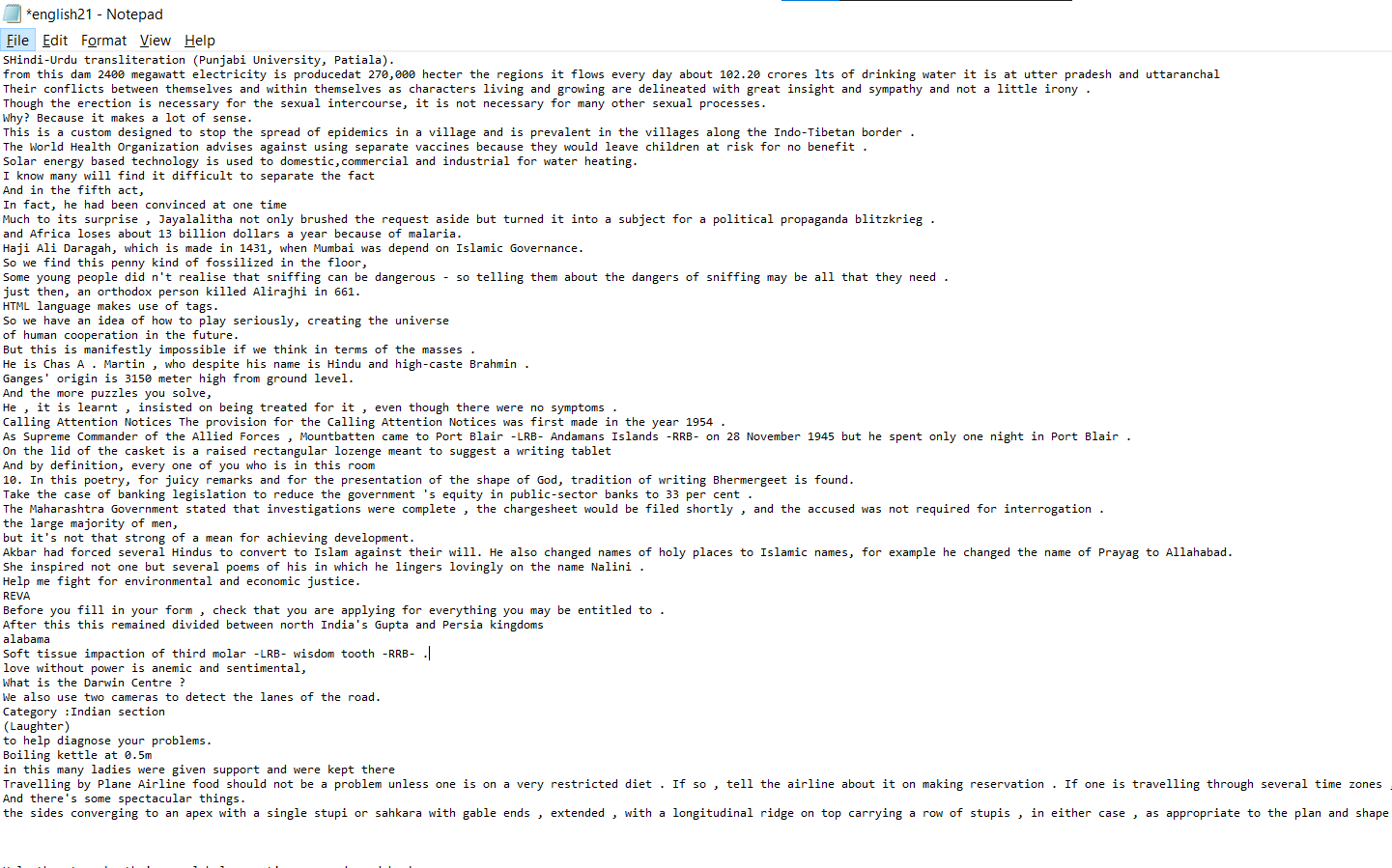
**DATA COLLECTION**

1. **TRANSLATION DATASET**

* **Data source:** [**https://www.kaggle.com/code/aiswaryaramachandran/english-to-hindi-neural-machine-translation/data**](https://www.kaggle.com/code/aiswaryaramachandran/english-to-hindi-neural-machine-translation/data)
  + This dataset contains Hindi sentences and their English translation
  + Total count: 127608 sentences with translations
* **Preprocessing**
  + English texts have been taken from the above dataset and processed into several text file after removing the NULL values



* + Each file contains not more than 28 sections with a max of 4500 characters in each section
  + TOTAL NUMBER OF SUCH FILES 121
  + Below if a glimpse of the text file



* + 3-line breaks mark the end of a section

WHY 4500?

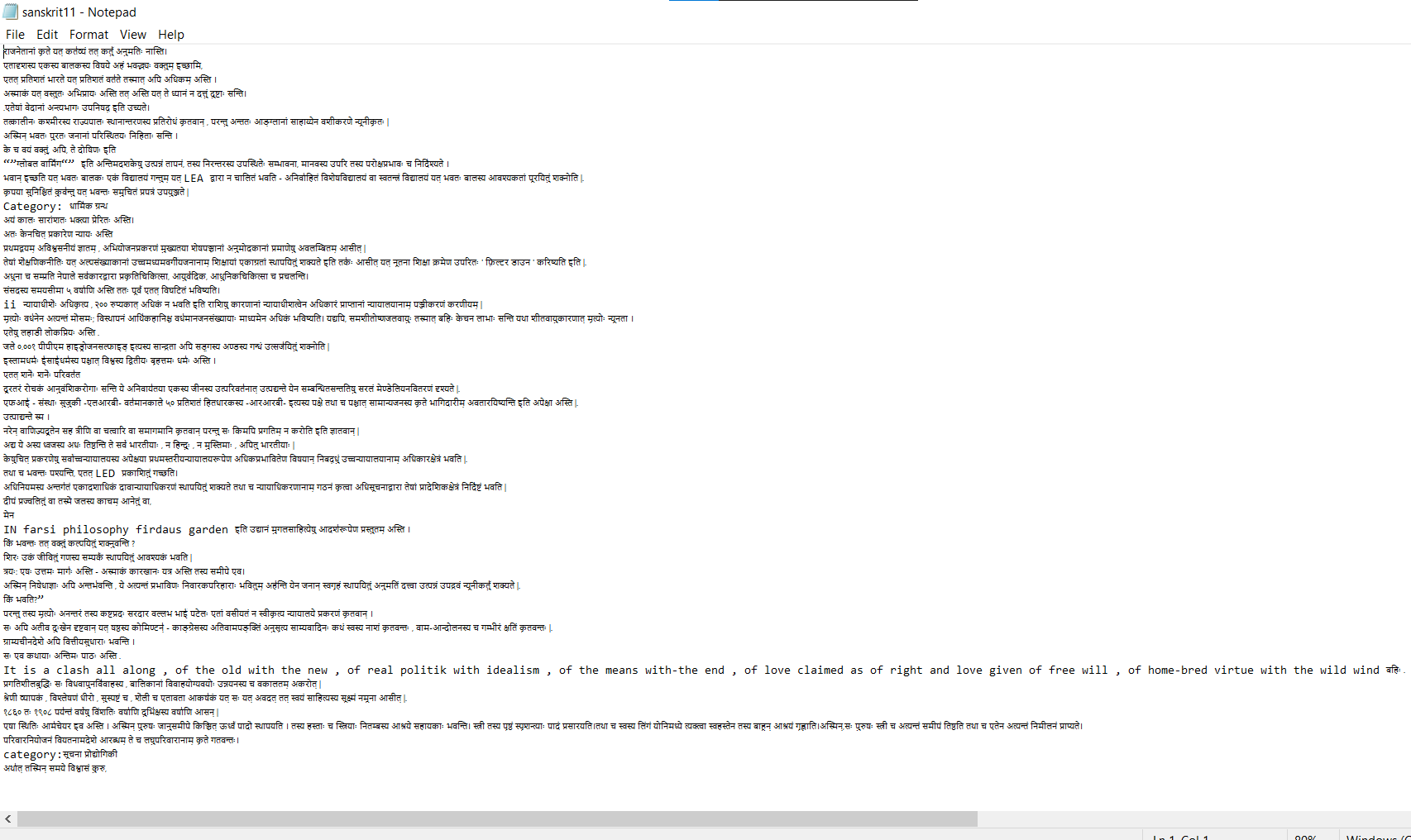
Google translate can translate a maximum of 5000 characters in one go and we are using google translate for generating Sanskrit dataset.

* **Sanskrit data collection**

**Challenge:**

There is no API service for Sanskrit language to automate the translation using scripting languages

* + Each section has been manually copied to the google translate website and the corresponding Sanskrit translation generated has been copied back into another file in the same format

****

* + 3-line breaks mark the end of a section
  + Sanskrit texts and their respective English translations have been processed into a CSV file and this file is the dataset used for this project

1. **DENOISING DATASET**

* **Data source:**

<https://www.kaggle.com/code/rmuhammed97/denoise-images-using-autoencoders>

* + Contains**:** 
    - Train images**:** 144
    - Test images: 72
    - Cleaned trained images(validation): 144

1. **BLUR DETECTION DATASET**

* Images used for this purpose have been scraped through web using Selenium
* Around 200 Sharp images have been scraped initially
* Only around 30% of them were useful
* They have been further augmented to generate more of image data
* All the sharp images have been blurred using cv2.blur()
* Total dataset size post augmentation:
  + Blur images: 111
  + Sharp images: 111

**METHODOLOGY**

1. **PREPROCESSING**

* **Preprocessing of the dataset:**
  + Data has been preprocessed to remove punctuations, digits and extra spaces.
  + Unique words have been processed into a dictionary with an index value allotted to each unique word
  + All the sentences have been converted to lower case.

1. **TEXT EXTRACTION**

* Python’s tesseract library/utility has been used for the purpose
* Tesseract has been configured to read Hindi texts and returns the text it encountered.

1. **TEXT TRANSLATION**

* One hot encoding has been used to represent words as vectors
* Encoder encodes the given text into one hot encoded vectors and the decoder decodes it back to the original word.
* This follows word to word translation.
* In case of image, text will be extracted from image using pytesseract OCR and the extracted text will be fed to the model
* There is also an option to give text as inputs directly
* Model in this way is an encoder-decoder architecture and will produce appropriate translations as outputs

WHY ONE HOT ENCODING and not word2vec or Glove or BERT?

The dataset collected contains a certain level of impurity i.e., as it has been collected from google translate, it is not 100% accurate and the words which google translate couldn’t find an appropriate match for are left in English only. So there is a high possibility of finding a sentence that has both English and Sanskrit text and the stated embeddings (Glove, BERT,etc. ) are not the solution.

**Project Model Explaination:**

[**https://www.youtube.com/watch?v=f-JCCOHwx1c**](https://www.youtube.com/watch?v=f-JCCOHwx1c)

Streamlit library has been used for UI. This library makes it easier to create front end for web apps.