MINI PROJECT-II

(2020-2021)

Text retrieval from Image using CNN Features

FINAL REPORT

Institute of Engineering and technology

Submitted By-

Lav Varshney (181500350)

Mukund Agrawal (181500401)

Shruti Sharma (181500692)

Supervised By-

Mr. Mandeep Singh

(Technical Trainer)

Department of Computer Engineering & Applications

Declaration

We hereby declare that the work which is being presented in the Mini Project **“Text Retrieval from Image using CNN Features”,** in partial fulfillment of the requirements for Mini project is an authentic record of our own work carried under the supervision of **Mr. Mandeep Singh, Technical Trainer.**

Lav Varshney

Mukund Agrawal

Shruti Sharma

Date of Submission:- 20-04-2021

CONTENTS

ACKNOWLEDGEMENT

We take this opportunity to acknowledge all the people who have helped us wholeheartedly in every stage of this project.

I also extend our sincere thanks to all other faculty members of the Computer Science & Engineering Department.

Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Abstract

Text detection is the method of locating areas in a picture wherever, text is present. Text detection and classification in natural pictures is very important for several computer vision applications like optical character recognition, distinguish between human and machine inputs and spam removal. Currently the challenge in text identifying is to detect the text in natural pictures due to many factors like, low-quality image, unclear words, typical font, handwritten text, image having a lot of color stroke than the background color, blurred pictures due to some natural problems like rain, sunny, snow, etc. The main aim of this work is to identify and classify the text in natural pictures with handwritten text. Here system detects the text and finds the connected regions, chain them together in their relative position.

INTRODUCTION

Text Extraction is a process by which we convert Scanned page or image in which text content is available to ASCII character that a computer can recognize.

Today’s world is digital with the appearance of many devices that are used in image acquisition. Nowadays, it becomes easy to store huge number of images by using image processing techniques. Since the collection of images and databases is fast and is increasing day by day, there is a need for new image retrieval techniques that should be efficient and fast.

CNN specifically designed to deal with the variability of 2D shapes, are shown to outperform all other techniques. Recognition systems are composed of multiple modules including feature extraction, classification and paradigm learning. They are allowing such multimodal systems to be trained globally using gradient-based methods so as to optimize an overall performance measure.

We tend to extract the options of the image of the text and apply the correct algorithmic rule to enhance accuracy to acknowledge the text. We tend to apply here a straightforward algorithm rule that works properly. The image of the text is captured through a camera. The hidden options of the text are highlighted within the actinic radiation. Currently, process on the image is completed thereon non-inheritable image exploitation ideas like image segmentation, edge data of image and characteristics feature extraction.

OBJECTIVE

The main objective of this project of Text Retrieval from Image with some extra features of text to speech and text translation to other language is to provide the facility to the user to take out the data from an image directly without read-write manually and can directly translate it to another language if needed he /she can listen to the text for easy recognition.

It can be more useful to the students or to the employees to take out the data from the image they want to.

As of Now we will try to cover all the English fonts and the handwritten text. We will try to add on more fonts to it further.

Requirements

Hardware and Software Requirements

Hardware Requirements

- **Processor:** - Intel(R)Core i5 - 7500 3.8Ghz or above
- **Ram: -** 4gb or above

Software Requirements

- Jupyter Notebook/ Google collab
- Python 3

Technologies Used

- Digital Image Processing
- Machine Learning

Dataset used

- https://drive.google.com/drive/u/0/folders/15L14KyFoUHAqV_f1Jn-u20YYEYVH5u8Z

Technologies and Modules Used

EasyOCR is a python package that allows the image to be converted to text. It is by far the easiest way to implement OCR and has access to over 70+ languages including English, Chinese, Japanese, Korean, Hindi, many more are being added. EasyOCR is created by the Jaided AI company.

The Reader class is the base class for EasyOCR which contains a list of language codes and other parameters such as GPU that is by default set to True. This needs to run only once to load the necessary models. Model weights are automatically downloaded or can be manually downloaded as well.

gTTS (*Google Text-to-Speech*), a Python library and CLI tool to interface with Google Translate's text-to-speech API. Writes spoken mp3 data to a file, a file-like object (bytestring) for further audio manipulation, or stdout. It features flexible pre-processing and tokenizing.

Methodology

The system proposed here work here on the image of text under ultraviolet light acquired by a digital camera. The algorithm which is applied here is as follows

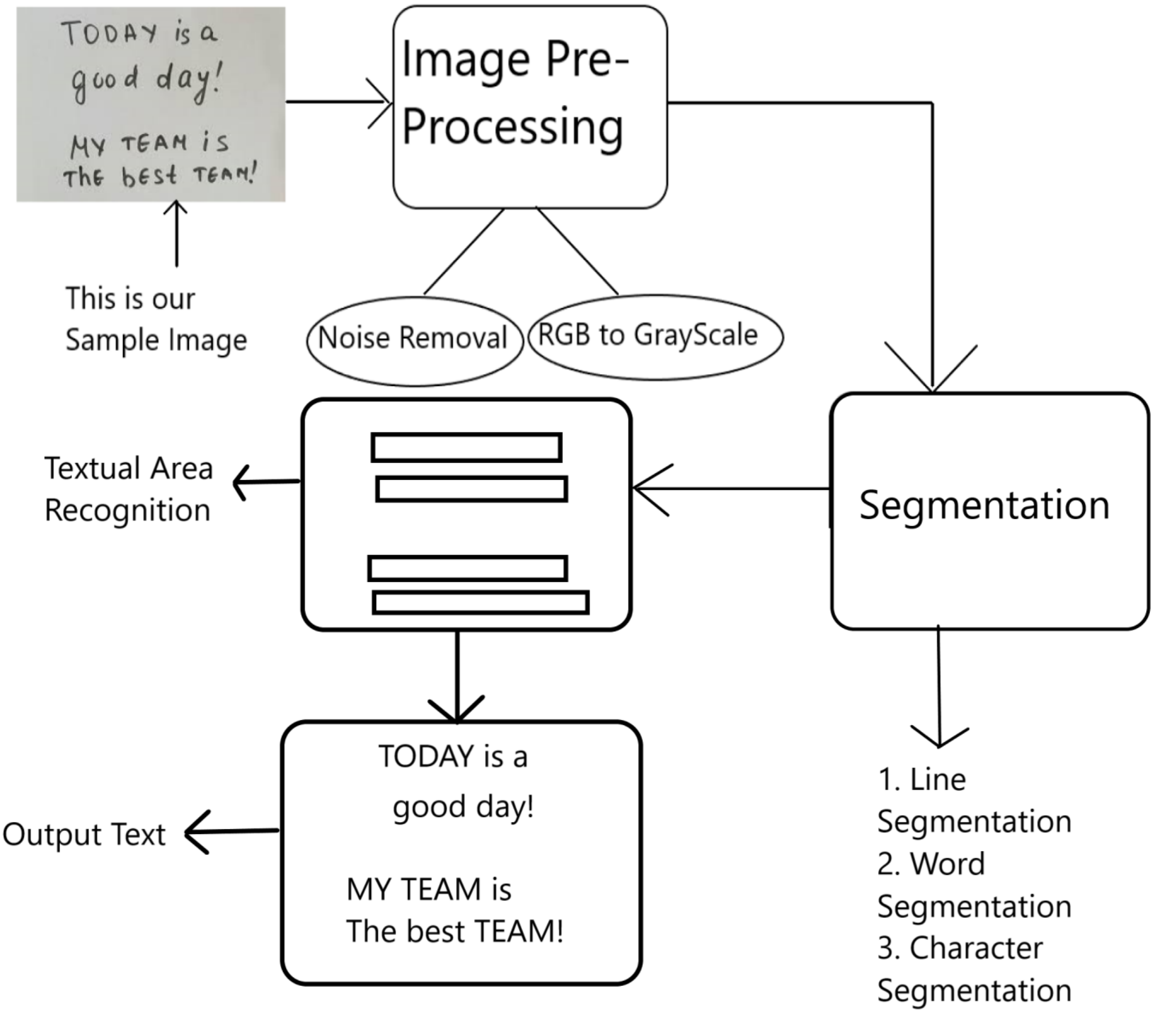
a. Firstly, we will take input as an image which contains some text which is to be retrieved.

b. Then, we will pre-process the image to reduce the noise and convert the RGB image to grayscale.

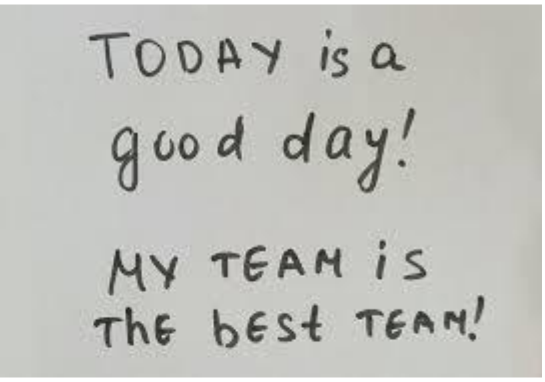
c. Then, we will perform segmentation and recognize the textual area.

d. Then, we will recognize the text content from that segmented area.

e. Then, finally we get the result printed on the output screen.


TODAY is a
good day!
MY TEAM is
The best TEAM!
This is our
Sample Image
Image Pre-
Processing
Noise Removal
RGB to GrayScale
Segmentation
1. Line
Segmentation
2. Word
Segmentation
3. Character
Segmentation
Textual Area
Recognition
TODAY is a
good day!
MY TEAM is
The best TEAM!
Output Text

Goal Of Project



TODAY is a
good day!

MY TEAM is
ThE bESt TEAM!

Today is a good day!

MY TEAM is The best TEAM!

APPLICATIONS

- Banking (To read Credit Card)
- Libraries (To convert Scanned Page to Image)
- Govt. Sector (Form Processing)
- Used in Car Number Plate Recognition System
- Undesirable Text removal from images.

CODE

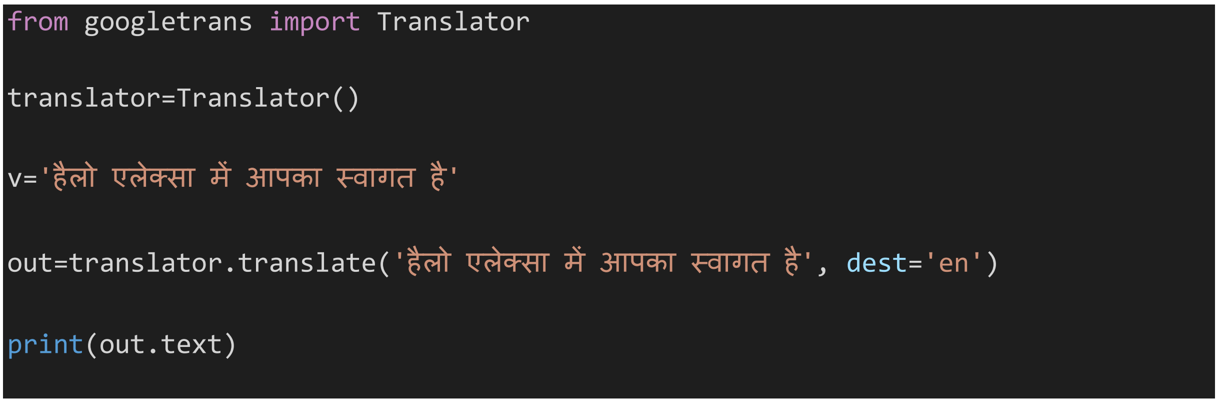
textretrieval.py

```
import matplotlib.pyplot as plt
import cv2
import easyocr
from pylab import rcParams
from IPython.display import Image
rcParams['figure.figsize'] = 8, 16

reader = easyocr.Reader(['en','hi'])

Image("/content/p10.jpg")

output = reader.readtext('/content/p10.jpg')

output

l=len(output)
for i in range(0,l):
  print(output[i][1])
```

textvoice.py

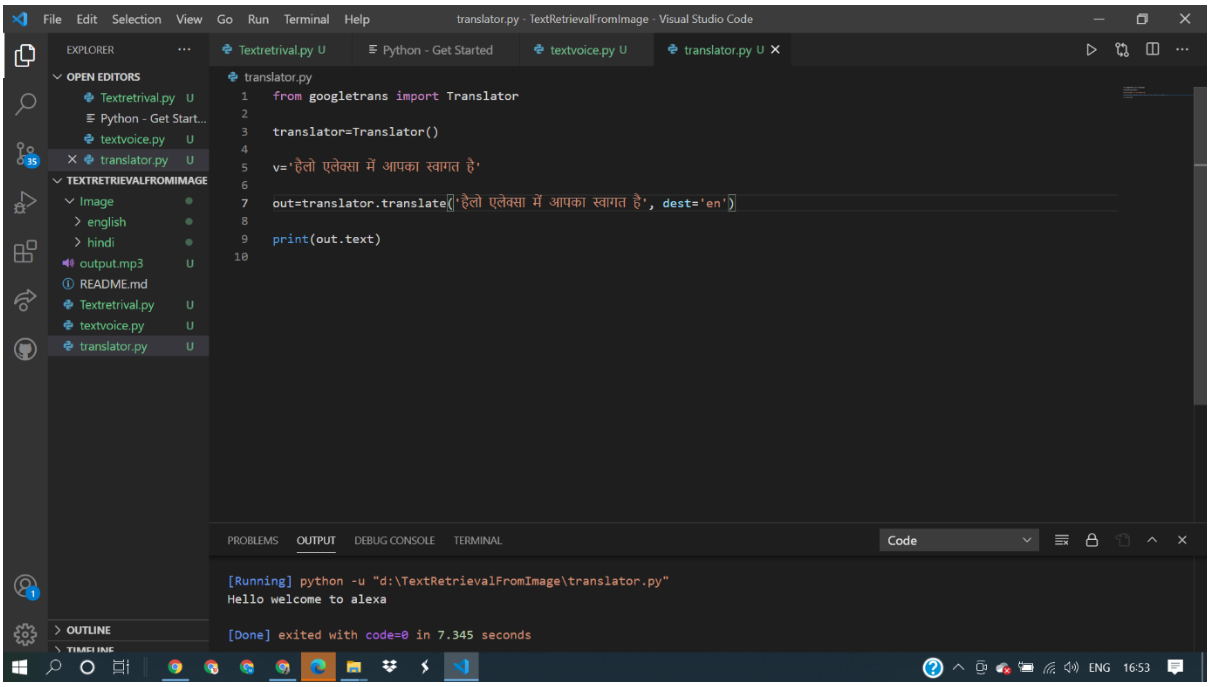
```python
from gtts import gTTS

import os

mytext = 'Hello this project is presented by lav, shruti and mukund on text retrieval and text translation with pronounciationnnnn .'


myobj = gTTS(text=mytext, slow=False)


myobj.save("output.mp3")

os.system("start output.mp3")
```

translator.py



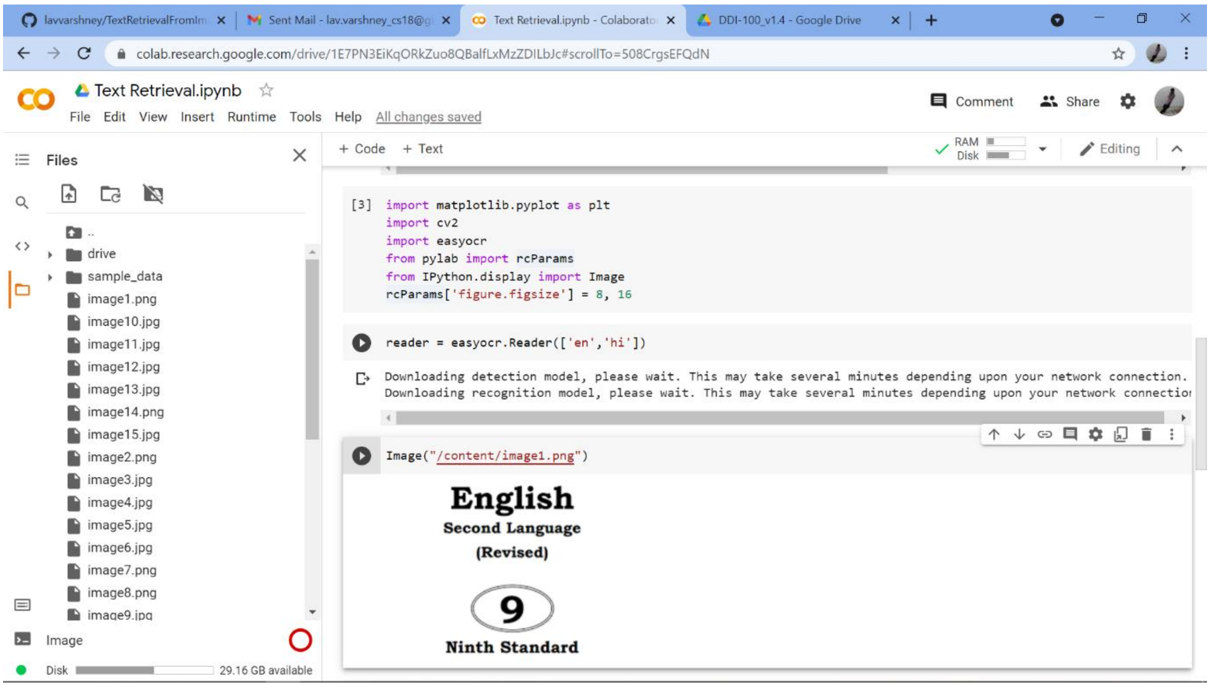
```python
from googletrans import Translator

translator=Translator()

v='हैलो एलेक्सा में आपका स्वागत है'

out=translator.translate('हैलो एलेक्सा में आपका स्वागत है', dest='en')

print(out.text)
```

Some of the Output Samples



```
from googletrans import Translator

translator=Translator()

v='हैलो एलेक्सा में आपका स्वागत है'

out=translator.translate('हैलो एलेक्सा में आपका स्वागत है', dest='en')

print(out.text)
```

```
[Running] python -u "d:\TextRetrievalFromImage\translator.py"
Hello welcome to alexa

[Done] exited with code=0 in 7.345 seconds
```



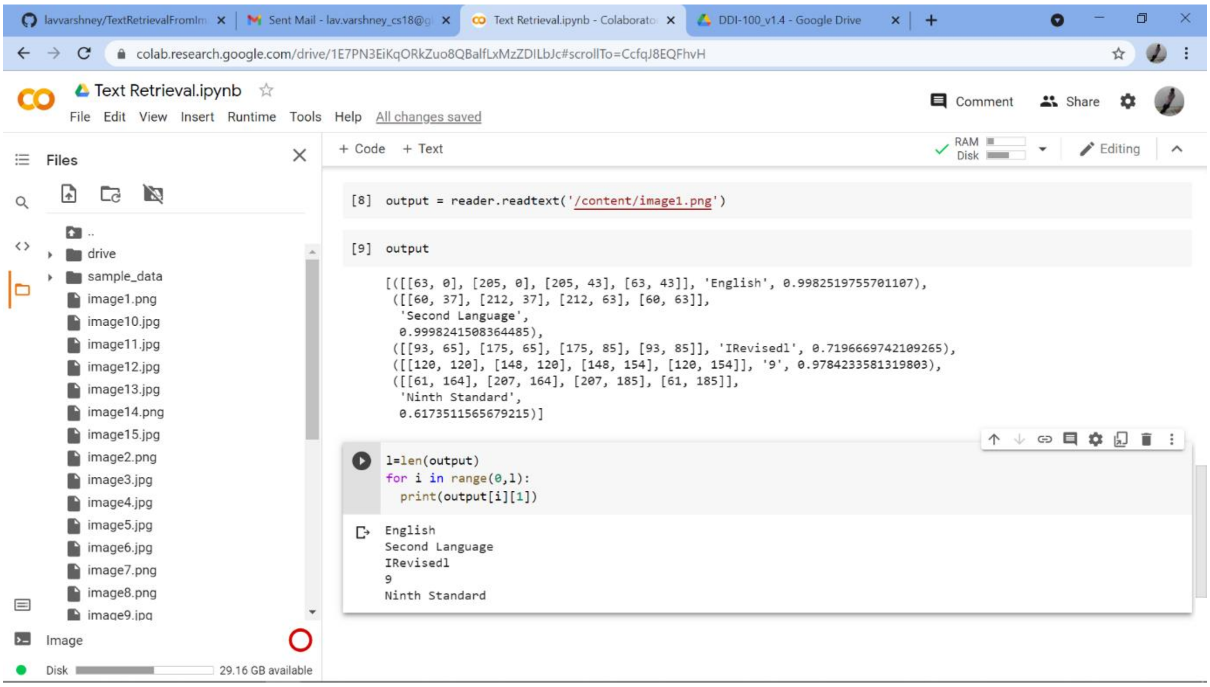
colab.research.google.com/drive/1E7PN3EiKqORkZuo8QBalfLxMzZDILbJc#scrollTo=508CrgsEFQdN

```
[3] import matplotlib.pyplot as plt
    import cv2
    import easyocr
    from pylab import rcParams
    from IPython.display import Image
    rcParams['figure.figsize'] = 8, 16
```

```
reader = easyocr.Reader(['en','hi'])
```

```
Downloading detection model, please wait. This may take several minutes depending upon your network connection.
Downloading recognition model, please wait. This may take several minutes depending upon your network connectio
```

```
Image("/content/image1.png")
```



lavvarshney/TextRetrievalFromIm
Sent Mail - lav.varshney_cs18@g
Text Retrieval.ipynb - Colaborato
DDI-100_v1.4 - Google Drive
colab.research.google.com/drive/1E7PN3EiKqORkZuo8QBalfLxMzZDILbJc#scrollTo=CcfqJ8EQFhvH
Text Retrieval.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code + Text
RAM
Disk
Editing
..
drive
sample_data
image1.png
image10.jpg
image11.jpg
image12.jpg
image13.jpg
image14.png
image15.jpg
image2.png
image3.jpg
image4.jpg
image5.jpg
image6.jpg
image7.png
image8.png
image9.jpg
Image
Disk
29.16 GB available

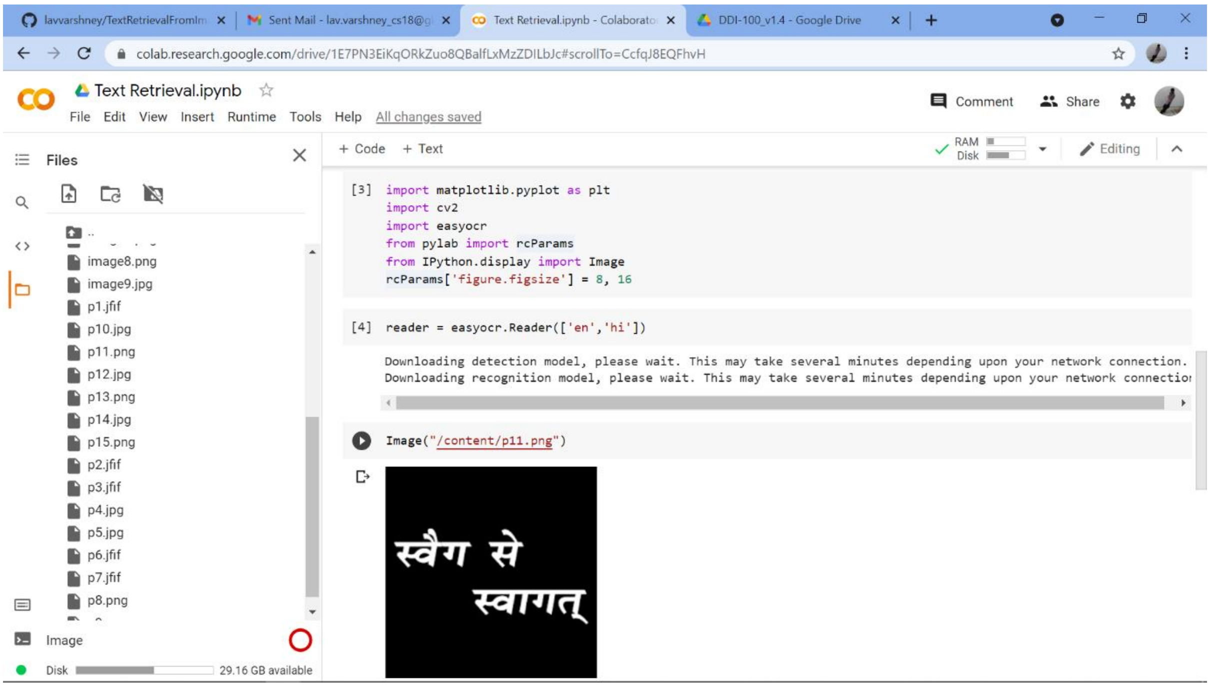
```
[8] output = reader.readtext('/content/image1.png')
```

```
[9] output
```

```
[([[63, 0], [205, 0], [205, 43], [63, 43]], 'English', 0.998251975570107),
 ([[60, 37], [212, 37], [212, 63], [60, 63]],
  'Second Language',
  0.9998241508364485),
 ([[93, 65], [175, 65], [175, 85], [93, 85]], 'IRevisedl', 0.7196669742109265),
 ([[120, 120], [148, 120], [148, 154], [120, 154]], '9', 0.9784233581319803),
 ([[61, 164], [207, 164], [207, 185], [61, 185]],
  'Ninth Standard',
  0.6173511565679215)]
```

```
l=len(output)
for i in range(0,l):
  print(output[i][1])
```

```
English
Second Language
IRevisedl
9
Ninth Standard
```



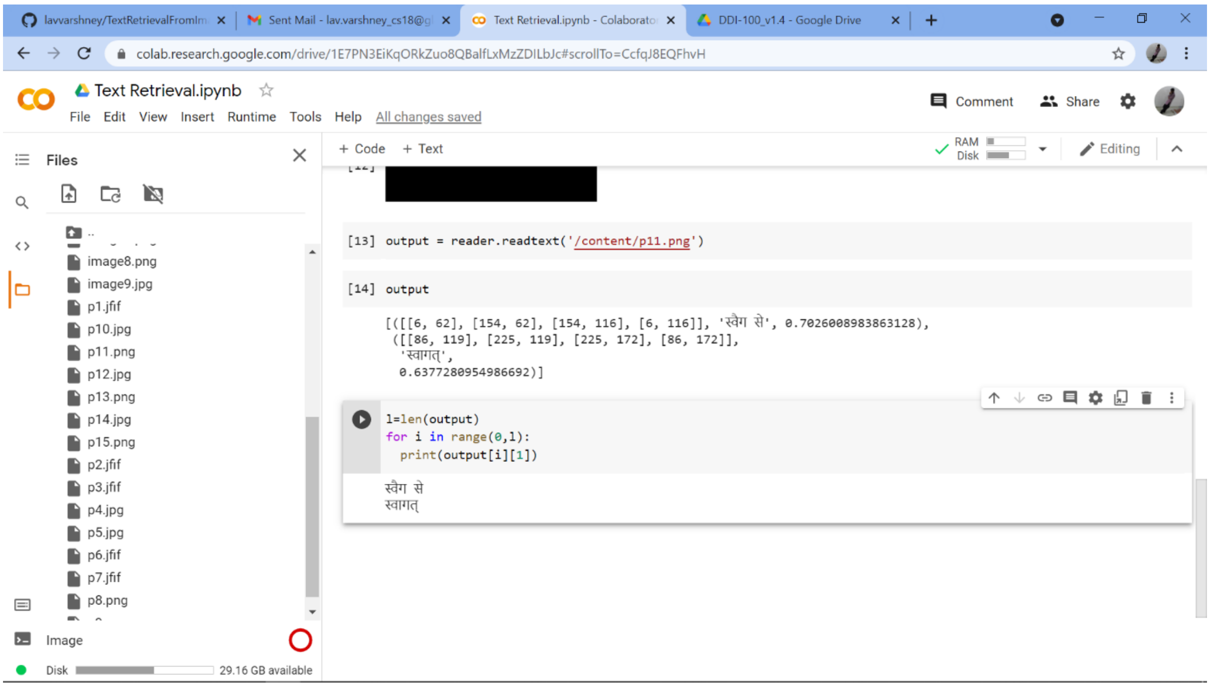
lavvarshney/TextRetrievalFromIm
Sent Mail - lav.varshney_cs18@g
Text Retrieval.ipynb - Colaborato
DDI-100_v1.4 - Google Drive
colab.research.google.com/drive/1E7PN3EiKqORkZuo8QBalfLxMzZDILbJc#scrollTo=CcfqJ8EQFhvH
Text Retrieval.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code + Text
RAM
Disk
Editing
..
image8.png
image9.jpg
p1.jfif
p10.jpg
p11.png
p12.jpg
p13.png
p14.jpg
p15.png
p2.jfif
p3.jfif
p4.jpg
p5.jpg
p6.jfif
p7.jfif
p8.png
Image
Disk
29.16 GB available
स्वैग से
स्वागत्

```
[3] import matplotlib.pyplot as plt
    import cv2
    import easyocr
    from pylab import rcParams
    from IPython.display import Image
    rcParams['figure.figsize'] = 8, 16
```

```
[4] reader = easyocr.Reader(['en','hi'])
```

```
Downloading detection model, please wait. This may take several minutes depending upon your network connection.
Downloading recognition model, please wait. This may take several minutes depending upon your network connectio
```

```
Image("/content/p11.png")
```


lavvarshney/TextRetrievalFromIm
Sent Mail - lav.varshney_cs18@g
Text Retrieval.ipynb - Colaborato
DDI-100_v1.4 - Google Drive
colab.research.google.com/drive/1E7PN3EiKqORkZuo8QBalfLxMzZDILbJc#scrollTo=CcfqJ8EQFhvH
Text Retrieval.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code + Text
RAM
Disk
Editing
image8.png
image9.jpg
p1.jfif
p10.jpg
p11.png
p12.jpg
p13.png
p14.jpg
p15.png
p2.jfif
p3.jfif
p4.jpg
p5.jpg
p6.jfif
p7.jfif
p8.png
Image
Disk
29.16 GB available

[redacted]

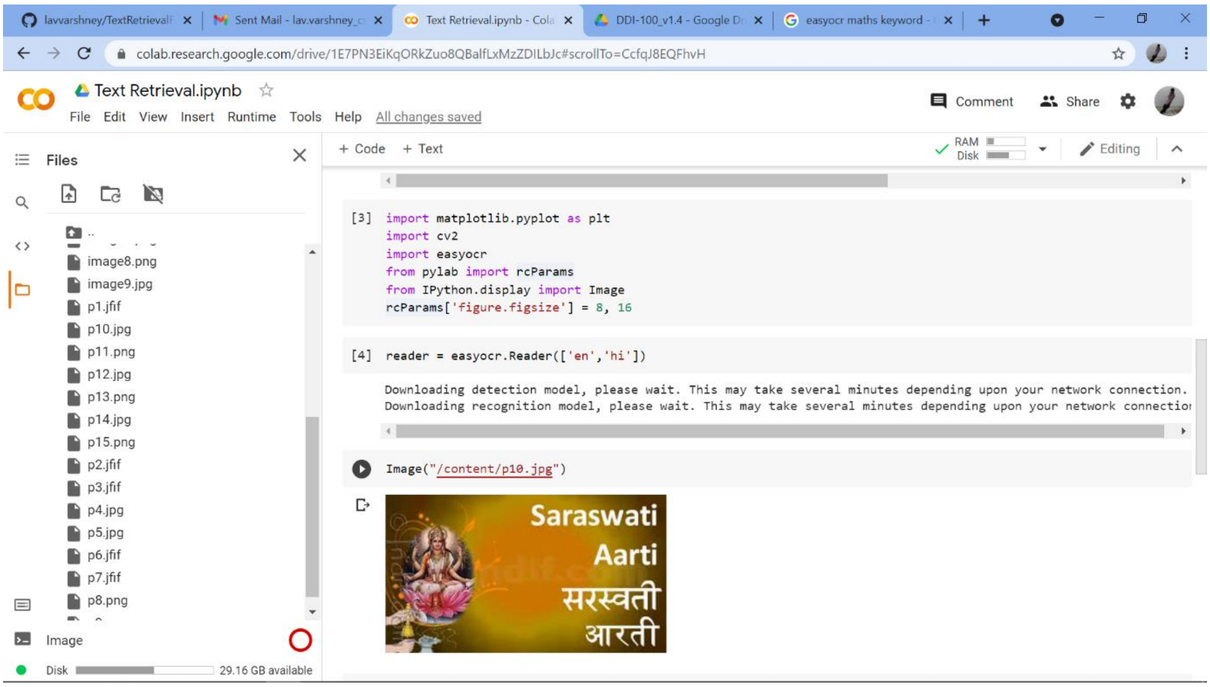
```
[13] output = reader.readtext('/content/p11.png')
```

```
[14] output
```

```
[([[6, 62], [154, 62], [154, 116], [6, 116]], 'स्वैग से', 0.7026008983863128),
 ([[86, 119], [225, 119], [225, 172], [86, 172]],
  'स्वागत्',
  0.6377280954986692)]
```

```
l=len(output)
for i in range(0,l):
  print(output[i][1])
```

```
स्वैग से
स्वागत्
```

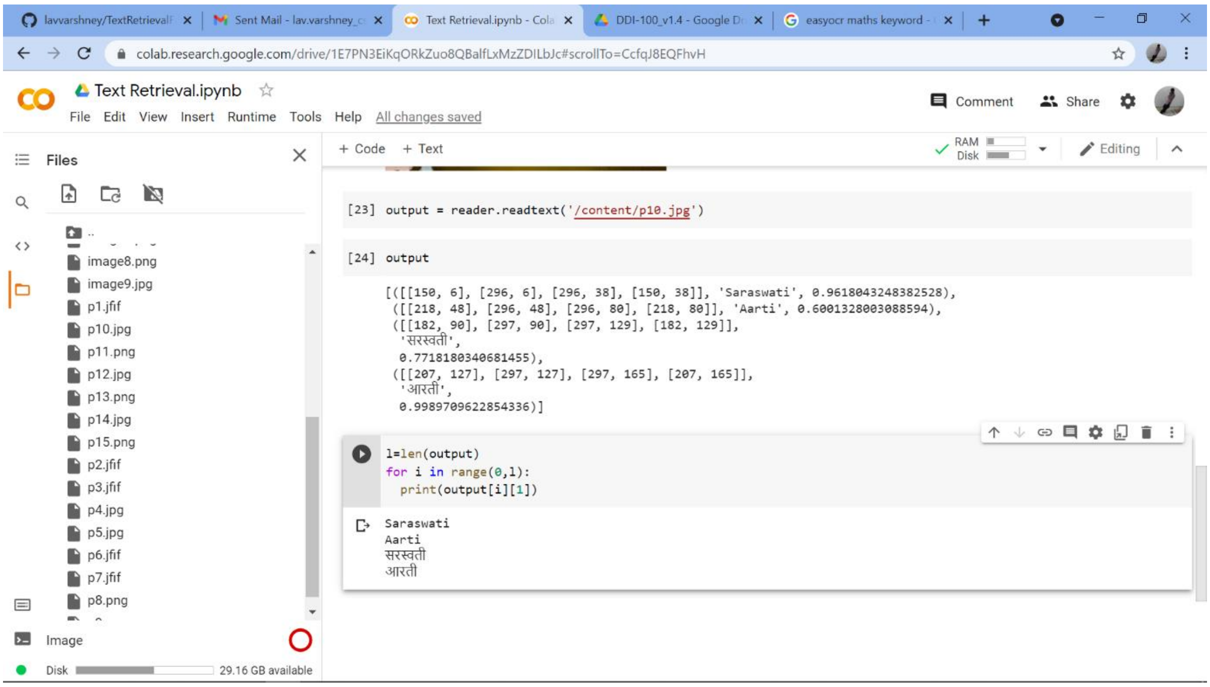


```
[3] import matplotlib.pyplot as plt
    import cv2
    import easyocr
    from pylab import rcParams
    from IPython.display import Image
    rcParams['figure.figsize'] = 8, 16
```

```
[4] reader = easyocr.Reader(['en','hi'])
```

```
Downloading detection model, please wait. This may take several minutes depending upon your network connection.
Downloading recognition model, please wait. This may take several minutes depending upon your network connectio
```

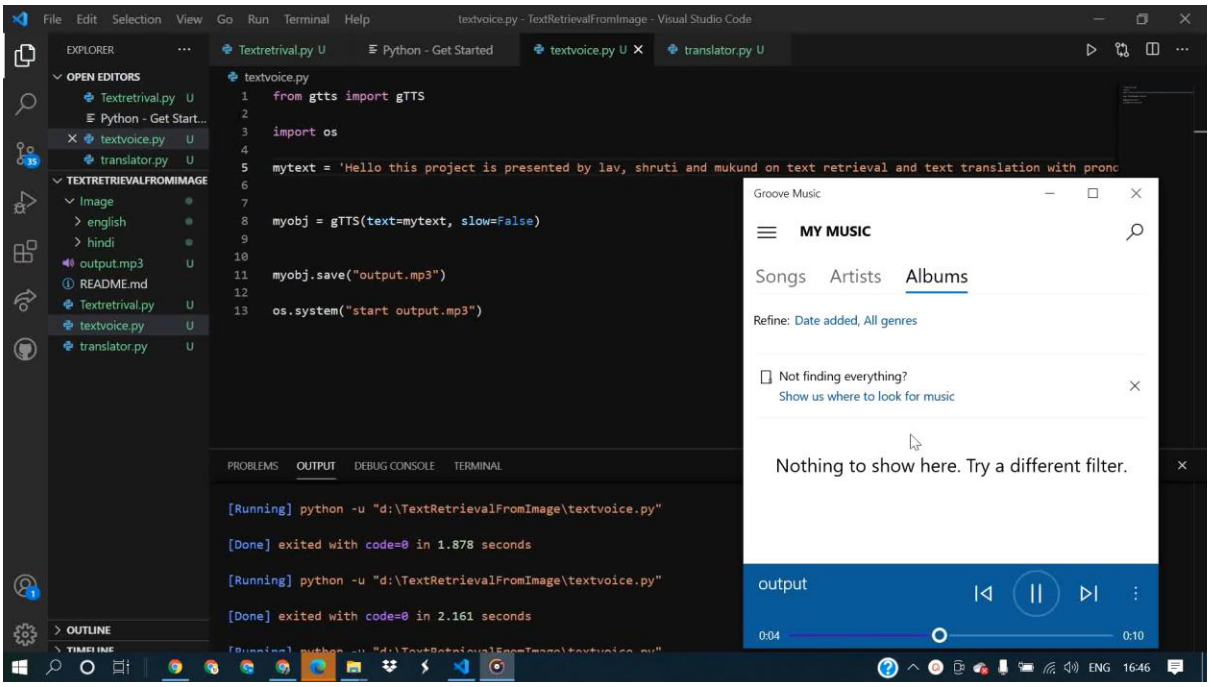
```
Image("/content/p10.jpg")
```



```
[23] output = reader.readtext('/content/p10.jpg')

[24] output

[([[150, 6], [296, 6], [296, 38], [150, 38]], 'Saraswati', 0.9618043248382528),
 ([[218, 48], [296, 48], [296, 80], [218, 80]], 'Aarti', 0.6001328003088594),
 ([[182, 90], [297, 90], [297, 129], [182, 129]],
  'सरस्वती',
  0.7718180340681455),
 ([[207, 127], [297, 127], [297, 165], [207, 165]],
  'आरती',
  0.9989709622854336)]
```

```
l=len(output)
for i in range(0,l):
  print(output[i][1])
```

```
Saraswati
Aarti
सरस्वती
आरती
```



```
from gtts import gTTS

import os

mytext = 'Hello this project is presented by lav, shruti and mukund on text retrieval and text translation with pronc


myobj = gTTS(text=mytext, slow=False)


myobj.save("output.mp3")

os.system("start output.mp3")
```

```
[Running] python -u "d:\TextRetrievalFromImage\textvoice.py"

[Done] exited with code=0 in 1.878 seconds

[Running] python -u "d:\TextRetrievalFromImage\textvoice.py"

[Done] exited with code=0 in 2.161 seconds
```

Future Scope

1. Development of Character Recognition considering the multiple font style needs to be developed in the future.

2. Development of Character Recognition for languages other than English needs to be researched on and developed in the future.

3. There is heavy demand for an OCR system which recognizes handwritten cursive scripts. This avoids keyboard typing and font coding for the image. This method helps in detecting handwritten characters with a precision of
   about 90%.

4. Once we detect languages, we can develop a converter to convert sentences from one language to another through a conversion and translation scheme.

Conclusion

In this project convolutional neural network-based text detection system that learns to automatically extract its own feature set instead of using a handcrafted one. Furthermore, the network learned not only to detect text in its retina, but also to reject multiline or badly localized text. Thus, the exact text localization does not require any tedious knowledge based post-processing. Even though the network was trained with synthetic examples, experimental results demonstrated that it can compete with other methods in a real-world test set. Future work includes the inspection of the text binarization and recognition problems with Convolutional neural networks.

References

- https://www.geeksforgeeks.org/
- https://www.github.com/
- https://www.tutorialspoint.com/
- https://www.google.com/
- https://www.youtube.com/