**Document Classifier & Text Extractor**

**Atharva Kousadikar, Debpriyo Roy, Harsh Sindhwa, Praveen Pandey**

**Technology Used:**

* Tesseract – Batch files,
* Google Vision API- Single files
* FNN, ResNet50, VGG15.
* Tkinter

**Benefits:**

* By testing many samples, we discovered that Google Vision API is much better to use for text extraction.
* Gives Bounding box, by which we can get coordinates around text, to retrieve it.
* Can access information extracted from image in the csv file.
* User can easily get attributes from csv file.
* User can get all the text extracted neatly on notepad.

**Drawbacks:**

* With tesseract the text extraction wasn’t up to par level.
* Tesseract fails to give the bounding box, which is essential in getting text back to append into the csv file.
* With our project we are successfully extracting text and appending the text with respect to their categories into the csv file.
* For to append there is a only limitation that, user has to select attributes in decided format only, otherwise the function will append wrong attributes.

**Challenges:**

* Faced challenges in appending the attributes extracted with the help of Google Vision API- Bounding Boxes.

**Working Example:**

In our Project, we have developed UI, which will classify your document and will extract the text from given file. You will be given options to extract texts from multiple images or single image. We are taking image from user and with the help of tesseract/ Goole Vision API, we are building building boxes around the texts we want to extract and then we are passing the coordinates of building box in our pre built function, which will take coordinates and will return the word and then we are appending those information according to their attributes.(e.g., name, age) in the csv file. The result will be the csv file with all the attributes extracted as per user’s requirement.