GROUP 3

FINAL PROJECT DESCRIPTION AND MANUAL

BURIAL RECORD OPTICAL CHARACTER RECOGNITION

EAI6010 APPLICATIONS OF ARTIFICIAL INTELLIGENCE

CRN: 72021

Done By

Anish Nitin Somaiah, NUID: 001894328, somaiah.a@husky.neu.edu

Pratibha Sridhar Iyengar, NUID: 001476161 , iyengar.p@husky.neu.edu

Ankit Arvind Khatavkar, NUID: 001476187, khatavkar.a@husky.neu.edu

Siddhant Sanjiv Sapte, NUID: 001494280 , sapte.s@husky.neu.edu

Srinivasu Narahari, NUID: 001824939, narahari.s@husky.neu.edu

Instructor: Mr. Sergiy Shevchenko



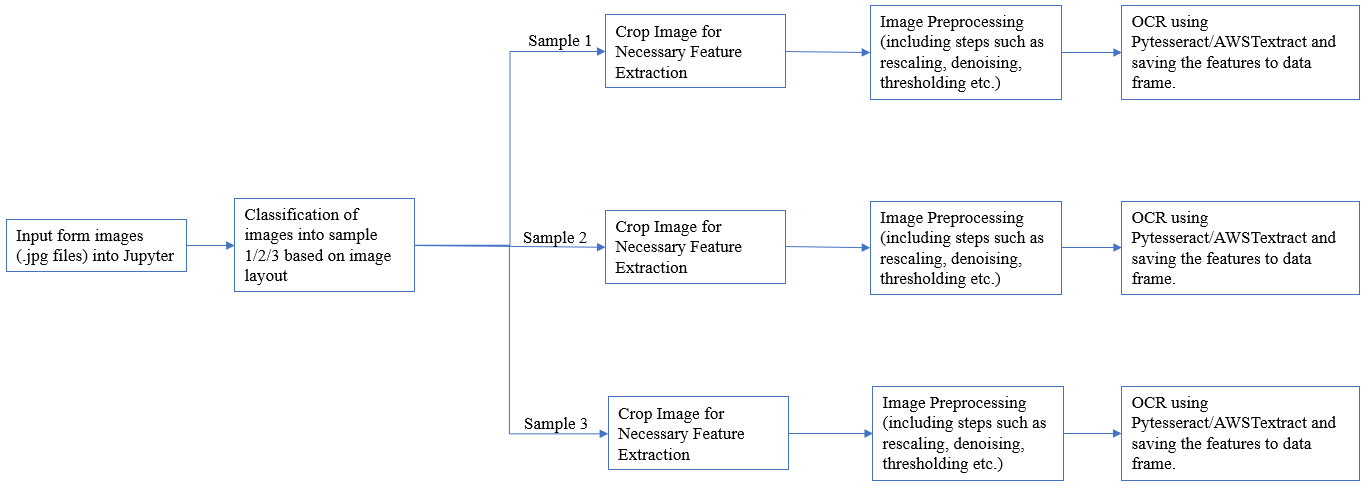
**Project Objective**

To perform Optical Character Recognition over form .jpgs and extract necessary features such as Name, Age and so on from the image given to a .csv file for user to understand features in the image.

**Tools Required**

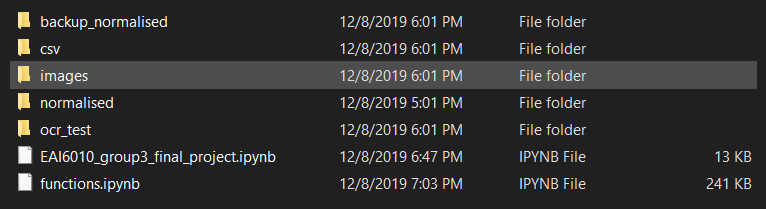
* Jupyter Notebook
* AWS Texteract package (AWS CL2)
* Pytesseract package
* Keras package for CNN
* Pillow, OpenCV packages for image preprocessing.

**Project Workflow**

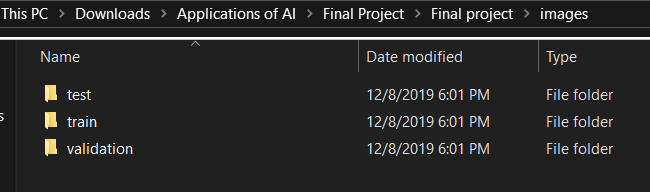
****

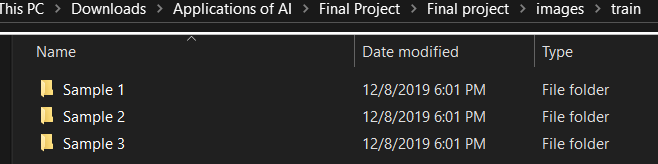
**Steps to implement the Project**

* Entire file management system should be as below



* The 2 Jupyter notebooks
* **Functions.ipynb :** notebook that contains definition of functions to preprocess images based on the respective samples.
* **EAI6010\_group3\_final\_project.ipynb :** notebook that contains our CNN model, commands to call functions to implement normalizing and OCR on the images and the main function to input stream images into our system and store output character data frame as .csv file. **This is the implementation file.**
* **Ocr\_test** folder contains the images that we stream into the machine to perform classification and OCR
* **Normalized** folder contains the images that have been normalized after preprocessing
* **Csv** folder contains the output data frame of the features present in the input forms.
* **Images** folder contains 3 subfolders **Train, Test, Validation**
* Each folder contains 3 subfolders **Sample 1, Sample 2 & Sample 3.**
  + Sample 1: Image layout with column sections
  + Sample 2: Image layout with row sections
  + Sample 3: Image layout with blank sections





* This file management of images is done to train our CNN model for classification of images into the respective samples based on their layout
* After classification of images, preprocessing of image is done by gray scaling, cropping, thresholding using Gaussian Blurring etc.
* Once the image is normalized, separate function is used to perform OCR and extract features of the image into a data frame which is then saved as a .csv output using AWS Texteract.

**BENEFITS OF USING AWS TEXTERACT.**

* Extract data quickly & accurately
* No code or templates to maintain
* Lower document processing costs
* Better accuracy compared to Pytesseract.