**Documentation**

**Use Case:**

To parse documents of different formats and extract particular data (For example- Name) from all documents using Deep Learning models. The extracted data is to be populated in a HTML Table.

**Solution:**

All the documents are to be converted to images where each image constitutes one page of the document. These images are the training images for an object detection model where the objects to be detected are the tables in the image. The model is trained such that it can detect multiple tables in a single page along with its location. This location information is collected and the particular section of the image is extracted. This is then split as cells using opencv and OCR (pytesseract) is used to extract the text and saved in CSV format; the required data is obtained by parsing the table data and extracting the required data. The final collection of data is populated in a HTML Table.

This process is repeated to all documents which are to be parsed.

**Model Used:**

Here, the model used is tensorflow Faster RCNN Object detection model. The training images have been trained over 1000 steps. The objects to be detected here are the tables in each page along with their confidence scores and locations which have been stored in a CSV file along with the image name. For better accuracy, the tables which have confidence scores above 80% are only considered. For training the model requires the training image and Ground truth file. Here the ground truth files have been obtained in .xml format using LabelImg. Tensorflow models require their training and testing data to be of the tfrecord format which is also been done. Using the predicted bounding boxes, the table is cropped from the input image into another image which contains only the table.

**Extraction of table data:**

Once the table alone is extracted as an image, the cells of the table are extracted using open-cv. The text in the cells are extracted using Tesseract and saved in csv format according to cell index. These cell images and extracted text are removed for availability of memory. To get the specific data(For example, Name) from all of these tables, each cell is parsed. If it contains ‘Name’ then the required data is written to a final CSV file and converted to a HTML table.

Convert PDF pages to images: pdf2imgpages.py

Detect tables using tensorflow model: detection.py

Extract the tables: extract.py

Extract cells from tables: extract\_cells.py

Extract text from each cell: ocr\_image.py

Parse cells and extract required data: ocr\_to\_csv.py

Main(path of folder containing pdfs as argument): execmain.py

Training data and Model: training/

Example document and result: example/