**Introduction:**

Document Identifier harnesses the power of GCP API's along with Machine learning to sort various documents into relevant folders.

Legal documents can include Passport, Driving Licence, Document of Incorporation etc. for any country.

**Input**

Currently the program supports jpg images and docx files.

**Output**

Internal working which includes calls to gcp (a diagram to make it more clear)

**Machine learning component**

Machine learning is implemented using tensorflow.

**Setup:**

* Download and install Python 3.6 version of **Anaconda** from <https://www.anaconda.com/download/>
* Download and install **git** from <https://git-scm.com/downloads>
* Clone repository from github : <https://github.com/gskishan004/Document-Identifier>
* Save GCP key as "key.json" in resources folder of the repository.
* Clone tensorflow repo from: <https://github.com/tensorflow/models> and extract it in the **ml** directory of **Document-Identifier**
* Download faster\_rcnn\_inception\_v2\_coco\_2018\_01\_28 <http://download.tensorflow.org/models/object_detection/faster_rcnn_inception_v2_coco_2018_01_28.tar.gz>
* Extract the faster\_rcnn\_inception\_v2\_coco\_2018\_01\_28 folder to the \**ml\models\research\object\_detection** folder
* Copy labelmap.pbtxt from resouces\files to copy folder and paste it in the C:\tensorflow1\models\research\object\_detection\training dir.
* Copy the faster\_rcnn\_inception\_v2\_pets.config file from resouces\files into the \object\_detection\training directory

**Run in cmd from the repo dir:**

* pip install -r requirements.txt
* python -m spacy download en\_core\_web\_lg
* conda install -c anaconda protobuf
* Change directories to the **\ml\models\research directory** and run the following:

protoc --python\_out=. .\object\_detection\protos\anchor\_generator.proto .\object\_detection\protos\argmax\_matcher.proto .\object\_detection\protos\bipartite\_matcher.proto .\object\_detection\protos\box\_coder.proto .\object\_detection\protos\box\_predictor.proto .\object\_detection\protos\eval.proto .\object\_detection\protos\faster\_rcnn.proto .\object\_detection\protos\faster\_rcnn\_box\_coder.proto .\object\_detection\protos\grid\_anchor\_generator.proto .\object\_detection\protos\hyperparams.proto .\object\_detection\protos\image\_resizer.proto .\object\_detection\protos\input\_reader.proto .\object\_detection\protos\losses.proto .\object\_detection\protos\matcher.proto .\object\_detection\protos\mean\_stddev\_box\_coder.proto .\object\_detection\protos\model.proto .\object\_detection\protos\optimizer.proto .\object\_detection\protos\pipeline.proto .\object\_detection\protos\post\_processing.proto .\object\_detection\protos\preprocessor.proto .\object\_detection\protos\region\_similarity\_calculator.proto .\object\_detection\protos\square\_box\_coder.proto .\object\_detection\protos\ssd.proto .\object\_detection\protos\ssd\_anchor\_generator.proto .\object\_detection\protos\string\_int\_label\_map.proto .\object\_detection\protos\train.proto .\object\_detection\protos\keypoint\_box\_coder.proto .\object\_detection\protos\multiscale\_anchor\_generator.proto .\object\_detection\protos\graph\_rewriter.proto

Incase of any error while running above commands, consider running terminal with Admin privileges

**Running the code**

python main.py <optional flags>

**Optional flags:**

**-m** : predict only using ML

**-t** : train the model

After the model is trained press Ctrl + C and run the following command from object\_detection dir (where XXXX is highest checkpoint number)

python export\_inference\_graph.py --input\_type image\_tensor --pipeline\_config\_path training/faster\_rcnn\_inception\_v2\_pets.config --trained\_checkpoint\_prefix training/model.ckpt-XXXX --output\_directory inference\_graph

**-i** : use another dir for input (By default input dir is taken as input)

**-w**: Sort images from word documents in input directory

**Some common error and the resolution :**

* ValueError: Tried to convert 't' to a tensor and failed. Error: Argument must be a dense tensor: range(0, 3) - got shape [3], but wanted [].

1. Go to the ml\models\research\object\_detection\utils folder. Find the learning\_schedules.py file. Go to the line 167. And replace the line 167 with below  
     
   rate\_index = tf.reduce\_max(tf.where(tf.greater\_equal(global\_step, boundaries),  
    list(range(num\_boundaries)),  
    [0] \* num\_boundaries))
2. Then run the following from ml\models\research directory:  
   python setup.py build  
   python setup.py install