A Project Report on

A. I. Base Document Digitization

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering

in

Information Technology

by

Sujoy Dev(17104036)
Priya Naik(17104021)
Rashmi Shetty(17104070)

Under the Guidance of

Dean Sameer Nanivadekar
Prof. Kiran B. Deskpande

Department of Information Technology
NBA Accredited
A.P. Shah Institute of Technology
G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615
UNIVERSITY OF MUMBAI

Academic Year 2020-2021

Approval Sheet

This Project Report entitled ***"A. I. Base Document Digitization"*** Submitted by ***"Sujoy Dev"(17104036),"Priya Naik"(17104021),"Rashmi Shetty"(17104070)***is approved for the partial fulfillment of the requirenment for the award of the degree of ***Bachelor of Engineering*** in ***Information Technology*** from ***University of Mumbai***.

(Name)
Co-Guide

(Name)
Guide

Prof. Kiran Deshpande
Head Department of Information Technology

Place:A.P.Shah Institute of Technology, Thane
Date:

CERTIFICATE

This is to certify that the project entitled ***"A. I. Based Document Digitization"*** submitted by ***"Sujoy Dev" (17104036), "Priya Naik" (17104021), "Rashmi Shetty" (17104070)*** for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Information Technology***,to the University of Mumbai,is a bonafide work carried out during academic year 2020-2021.

(Name)
Co-Guide

(Name)
Guide

Prof. Kiran Deshpande
Head Department of Information Technology

Dr. Uttam D.Kolekar
Principal

External Examiner(s)

1.

2.

Place:A.P.Shah Institute of Technology, Thane
Date:

Acknowledgement

We have great pleasure in presenting the report on **A.I. Based Document Digitization** We take this opportunity to express our sincere thanks towards our guide **Dean Sameer Nanivadekar** & Co-Guide **Prof. Kiran Deshpande** Department of IT, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Prof. Kiran B. Deshpande** Head of Department,IT, APSIT for his encouragement during progress meeting and providing guidelines to write this report.

We thank **Prof. Vishal S. Badgujar** BE project co-ordinator, Department of IT, APSIT for being encouraging throughout the course and for guidance.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Sujoy dev
17104036

Priya Naik
17104021

Rashmi Shetty
17104070

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

_____

(Signature)

_____

(Sujoy Dev 17104036)
(Priya Naik 17104021)
(Rashmi Shetty 17104070)

Date:

Abstract

One of the crucial challenges faced by every industrial organization is the maintenance of records, mainly non-digitized type, i.e. hard copies and prints.

While a few documents can be obtained in a pre-digitized format, majority of the documents which are crucial in nature remain non-digitized.

Such documents are majorly billing invoices, which contain all the information about the seller, consumer, products, prices/taxes etc.

Extraction of such data requires manual labour which is prone to human error and expensive. Hence data extraction systems are becoming increasingly important for cost cutting, efficient data processing and analysis.

Our aim is to build an application that can be built using modern software technologies that can be used to automate the process of data extraction from invoices (single/bulk) by performing image processing,character/pattern recognition.

This application will provide notifications to it's users about the metadata extracted from the documents such as the Invoice ID, amount, etc. as a reminder with the help of date parameters, if any.

This application will also allow training of a centralized model to extract newer fields which were previously unidentified

Contents

List of Figures

List of Tables

List of Abbreviations

NER:	Named Entity Recognition
NLP:	Natural language processing
OCR:	Optical Character Recognition
openCV:	Open Computer Vision

Chapter 1

Introduction

In today's fast paced world every thing has moved to the "Cloud". Everything from reports to credit cards are available online, yet a majority of the documents are still processed via manual paperwork. There are various methods of storing these documents, but none provide an efficient knowledge-base. Organizations both, small and large, process a large number of documents on a daily basis with an lot of incorrect input. Some choose to work with similar softwares that cost a fortune to run, thus pulling small scale organizations out of the race. Yet these softwares lack important features such as insight collection and reporting. Also they fail to recognize document which are of an unknown type.

Businesses, both large and small, receive a large number of document invoice and bills in printed format. Manual data extraction is a slow and cumbersome process which still revolves around the copy-paste methodology. It requires a moderately large workforce and is not error free. Frequent errors such as incorrect data and spelling errors lead to data inconsistency. Furthermore it is a time consuming process, this is one of the leading cause of business failures. A manufacturing unit with slow document processing accompanied by incorrect data evaluation will give poor results, which may drive the business away, hence incurring loss.

By automating data entry, OCR and data capture tools can lower costs, eliminate keying errors, and streamline the approval and payment cycle [6]. According to Wikipedia, ' Intelligent Data Capture (IDC) or learning systems enable end users to extract content from invoices without the system having to learn the layout of the invoice.Some intelligent engines are able to correctly sort batches on the fly, locate data fields such as invoice and PO number, as well as line item information, and then extract the desired content from those data fields. Intelligent solutions do not require the coding of rules or design form templates. Rather the system learns by reviewing a relatively small number of invoice samples. This helps the system scale to large invoice volumes and widely varying document layouts without requiring a human operator'[7].

With the advent of machine learning, image and language processing it is now possible to scan any image and extract just about everything. We can leverage the existing libraries, developed over time, to implement an automated invoice scanning and extraction pipeline. Computer vision can scan, eliminate noise, and extract localized text areas. Natural Language Processing (NLP) can extract meaningful data from arbitrary sentences, i.e., it mimics

the human lexical capabilities. The combination of both the said technologies in an application with notification features will can prove to be a cost effective solution for document digitization.

1.1 Problem Statement

Document maintenance is a difficult and cumbersome task. Everyday huge chunks of data are misplaced due to improper handling of documents during and after their need. This must be very disruptive since there may be loss of confidential or important data which my potentially be harmful to any organization.

In this research paper we propose to create automated system for secure digitization and maintenance of documents. Institutions/users must be able to upload documents(expense bills/ invoices etc.) and the important data from these documents must be extracted and stored for later use. If the data is not extracted from the documents the users must be able to select the data themselves. Send the users insights about their document/invoices ,i.e., expiration, validity, etc. The application will store the data extraction model in a centralized repository. This centralized model will retrain at regular intervals to keep the models updated. This will increase the model accuracy over time as well as the number the entities that can be recognized. A web based user interface will be provided to upload the invoices(single or bulk) and view the extracted information in unstructured format.

In order to extract data from invoices the application will make use of Named entity Recognition(NER) which is a subset of natural language processing and openCV image processing. With the help of openCV we will perform optical character recognition on the document/invoice, furthermore we will extract tabular data using custom algorithms. The OCR data will be passed to the NER service for extraction of important Invoice metadata, which may be used later for analysis.

Chapter 2

Literature Review

Lu, H. et al.[1], have conceived a method to for better preprocessing of shadowed text images, for which the character recognition performance of Tesseract drops significantly. In this paper, we propose a new method to process the shadowed text images for the Tesseract's optical character recognition engine. First, they performed a local adaptive thresholding to transform the document to gray-scale image into a binary image to capture the contours of texts. Now in order to get rid of the salt-and-pepper noise in the shadow areas they applied a double-filtering algorithm, in which a vertical and horizontal projection method is method is used to remove the noise between texts and after that median filter removes the noise within characters.This type of preprocessed data when provided to Tesseract OCR produces much better result.

Sidhwa et al.[2], are using OpenCV to extract the bill or invoice document from an image buy substracting the background after finding the largest rectangular contour. Next they perform line segmentation in which they scan the document horizontally, then they perform line segmentation. However due to the absence of any NLP algorithm their process of data extraction only returns the test but no lexical definition.

K.M. Yindumathi et al. [3], have studied and compared the various methods,metrics and algorithms for text extraction from bills and invoices. According to their findings OpenCV, Tessseract are able to extract text from most of the images but when it comes to handwritten documents they fall short.

Zhang, J et al. [4], have extracted textual data from VAT invoices by preforming image preprocessing, OCR and localizing the data of a particual region of interest (ROI). In the preprocessing stage they performed skew-correction by applying local adaptive thresholding and Hough transformations. The ROI was segmented and extracted with the help of the projection method. They remove stamps from the VAT invoice with by elimination of the red color channel, i.e., the color of the stamps.

Wei Ruan and Won-sook Lee [5], built a Named Entity Recognition medical imaging procedure recognition system based on conditional random fields (CRF) model with word-based, part-of-speech. The NER model has been trained on a custom annotated dataset of medical notes from I2B2 with the F1 score up to 0.923 for recognizing medical imaging procedure entities. This system can be used to add and recognize new entities by simply

creating new medical entities to extract data from the medical notes/documents.

Chapter 3

Project Design

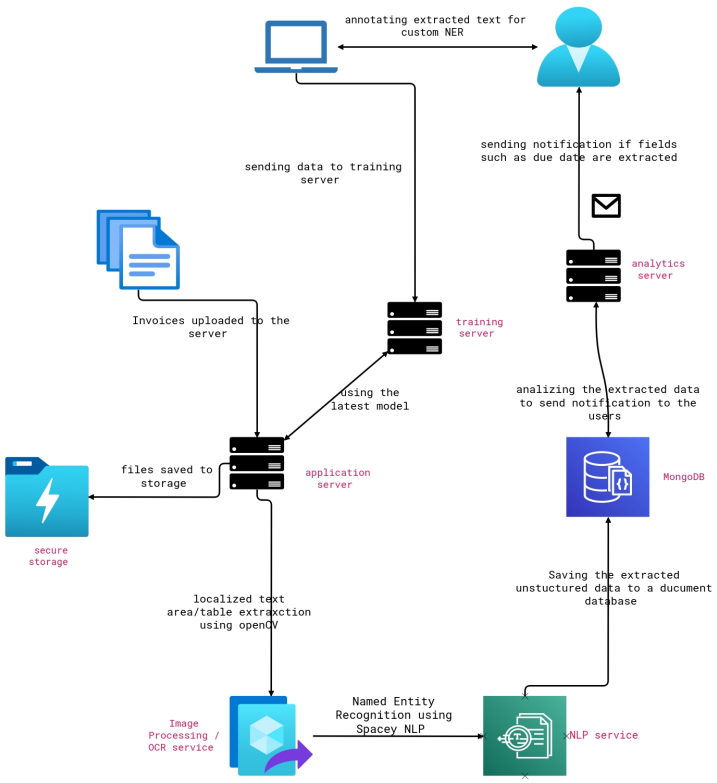

Figure 3.1: Block Diagram of Proposed System Architecture

The following architecture describes a highly scalable NER service which can be deployed on the cloud. Separation of the application, training and the analytics server some level of distributed computing. The training server is responsible for retraining itself and providing the latest model to the application server as well as the nlp service. Since all the information exchange happens across multiple servers a central NoSQL database is used to record all the events. The data stored in NoSQL format on the database can be used to run analytics on the users and the documents to find out purchase habits, etc. Documents that are uploaded by the user are stored onto a secure storage for review later, at the same time the documents which will be submitted during retraining will also be stored on a separate location in a similar manner. Database transactions are made for each record

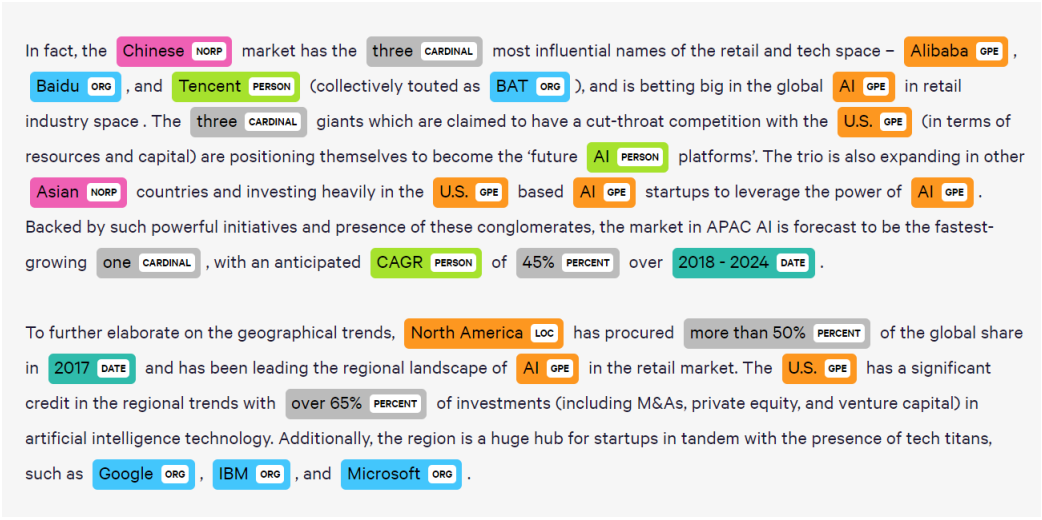

Figure 3.2: Principal NER

The default in NER model can detect labels from a huge scope of words / entities. This acts as an advantage since terms such as ' money',' time', currency' along with various other data can easily be extracted. But at the same time this model is unable to detect entities that may be noticeable visually on an invoice.

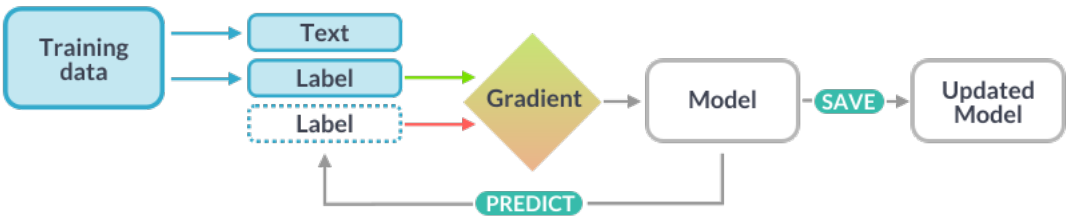

Figure 3.3: NER retraining and update

In order to overcome this challenge we dive a little deeper into NLP and build our own custom named entity recognition model on top of existing model. Hence we keep on adding to the weight of the model and gradually increase its accuracy in an incremental fashion. Training requires two primary inputs namely the extracted content with its start and end coordinates along with the custom label assigned to it.

Chapter 4

Project Implementation

```
# nlp.create_pipe works for built-ins that are registered with spaCy
if 'ner' not in nlp.pipe_names:
    ner = nlp.create_pipe('ner')
    nlp.add_pipe(ner, last=True)


# add labels
for _, annotations in TRAIN_DATA:
     for ent in annotations.get('entities'):
        ner.add_label(ent[2])

# get names of other pipes to disable them during training
other_pipes = [pipe for pipe in nlp.pipe_names if pipe != 'ner']
with nlp.disable_pipes(*other_pipes):  # only train NER
    optimizer = nlp.begin_training()
    for itn in range(10):
        random.shuffle(TRAIN_DATA)
        losses = {}
        for text, annotations in TRAIN_DATA:
            nlp.update(
                [text],  # batch of texts
                [annotations],  # batch of annotations
                drop=0.2,  # dropout - make it harder to memorise data
                sgd=optimizer,  # callable to update weights
                losses=losses)
```

Figure 4.1: NER retraining and update Code Snippet

The following snippet of code (Fig. 4.1) is used to update the existing NLP spacy model with the new labels that were custom created by the users Building on top of the existing models helps in increasing the accuracy and scope of a use case and machine learning in general. We randomise the dataset after every epoch by shuffling it in order to get various occurrences of the data at different positions this is particularly useful when the size of the dataset is rather small and near under-fitting.

```
+ Text                                                                    RAM Disk
with open(dataturks_JSON_FilePath, 'r') as f:
    lines = f.readlines()
    for line in lines:
        data = json.loads(line)
        text = data['content']
        entities = []
        annotations=[]
        if data['annotation'] == None:
            continue
        for annotation in data['annotation']:
            point = annotation['points'][0]
            label = annotation['label']
            annotations.append((point['start'], point['end'] ,label,point['end']-point['start']))
        annotations=sorted(annotations, key=lambda student: student[3],reverse=True)
        seen_tokens = set()
        for annotation in annotations:
            start=annotation[0]
            end=annotation[1]
            labels=annotation[2]
            if start not in seen_tokens and end - 1 not in seen_tokens:
                seen_tokens.update(range(start, end))
                if not isinstance(labels, list):
                    labels = [labels]
                for label in labels:#dataturks indices are both inclusive [start, end] but spacy is not [start, end)
                    if len(labels)==1:
                        entities.append((start, end+1  ,label))
                print(seen_tokens)
        training_data.append((text, {"entities" : entities}))
return training_data
```

Figure 4.2: NER Data Processing

Before we can actually start the training for the NLP model, we must first preprocess (Fig. 4.2)the data in order to remove any discrepancies that may lead to an errored model. This can be done in various ways in our case we simply remove any overlapping text entities since the spacy NLP library cannot work with such overlapping entities. The overlapping entities are pinpointed by looking at the start and end indexes of every individual entity text.

Initially the model starts losing data (Fig. 4.3) and losses are over 21,000 but gradually after 80-90 epochs the data loss reduced to mere 840, which is good considering the fact that we had such small data set to begin with. An ideal loss less than 200 would mean that a data set produce partially correct results.

Once the model is trained it can be exported to the disk and can be used for converting data from raw text to meaningful text Input can be single page PDF or Image. For image pre-processing we use Google vision API, which takes care of skew correction, thresholding etc. and extracts the data using OCR. this extracted data is then provided as input to the NLP model in response to which the NLP model returns an unstructured JSON whit the labels as keys and extracted information as values, which can be integer array or string. (Fig. 4.4)

In case the users feels that the model does not recognise enough labels for a custom document, have a choice to train the model using the custom document annotation data. They can select the document and only extract the raw text. Now the user can start labelling the text with the existing labels or can create new labels dynamically this must be done for multiple documents in order for the model to recognise the extraction pattern. Once the user has annotated sufficient number of documents upon retraining the model should be able to text data from other similar documents. (Fig. 4.5)

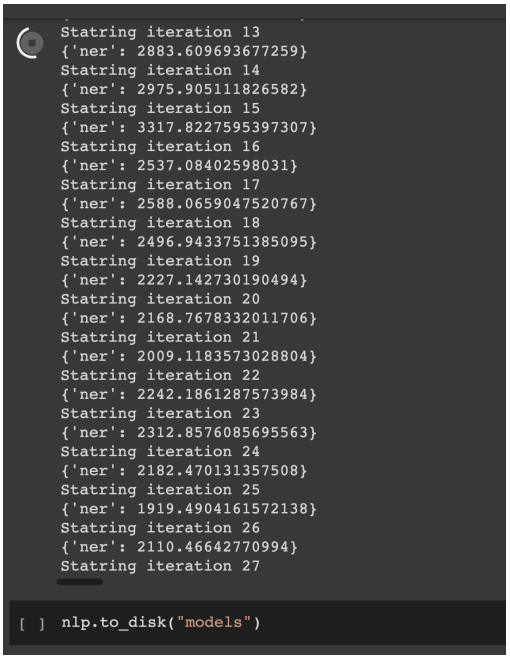

Figure 4.3: Model Training

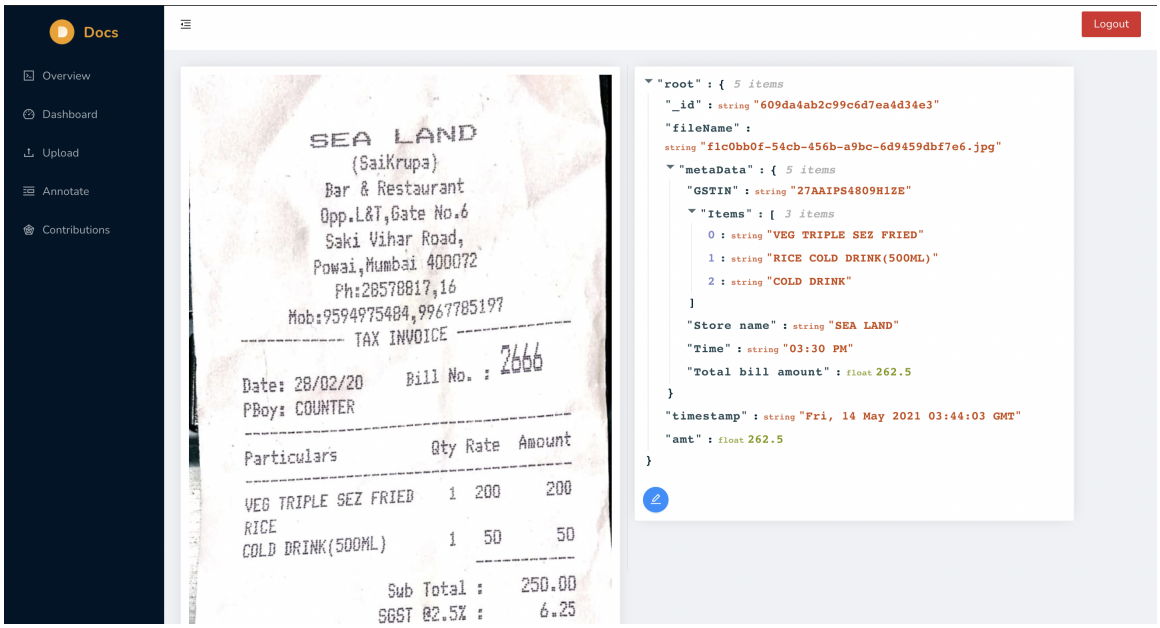

Figure 4.4: NER Data Extraction

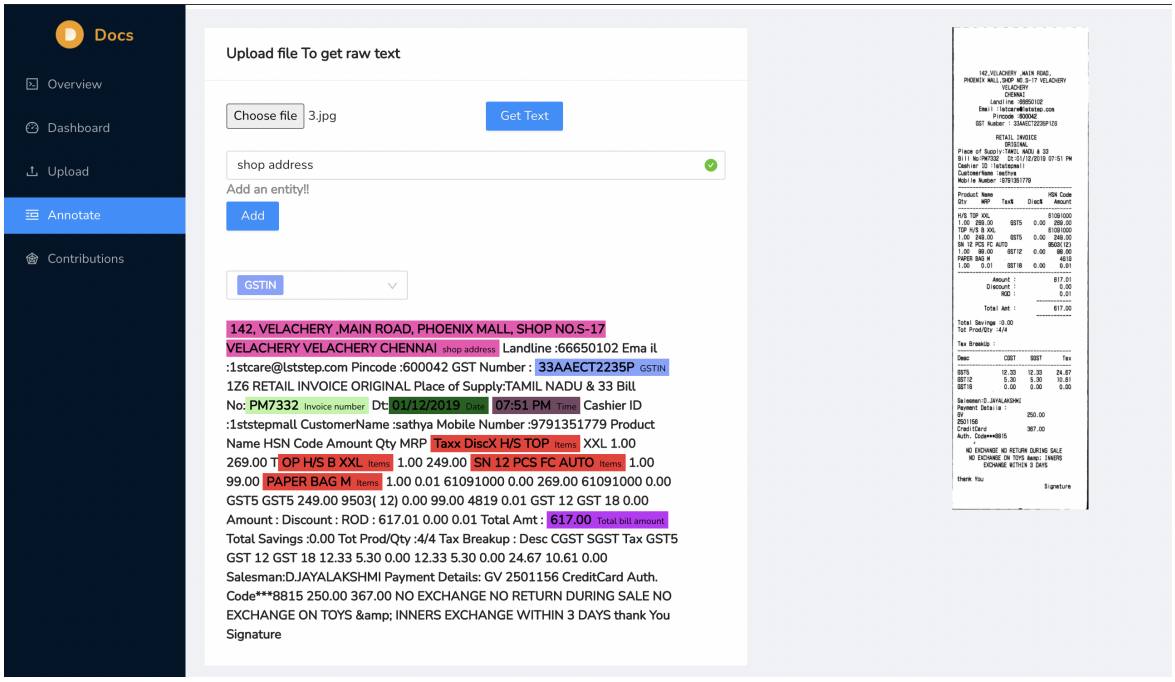

Figure 4.5: NER Custom Data Annotation For Training

Chapter 5

Testing

For the given project we would choose manual and automated testing plans. In Manual Testing once the application has been developed a user or a set of users test the application to find out bugs and put it under extreme stress so that it reaches its breaking point in automated testing in order to test a certain set of codes write additional code to either perform mock or integration tests via runtime script. Manual tests are time consuming but easy to perform, they gave us a good idea about User experience on the other hand automated testing is complex and but at the same time we develop fault resilient code since if any set of codes don't pass a test, the entire build pipeline stops unless the failed cases have been fixed.

for the scope of the project client side manual testing is important since we have to constantly perform the task of uploading the document and extract the data from them which may keep on changing again and again Hans automated test cases in this case would result in higher rate of area on the other hand since the entire back end is made up of API Endpoints the entire backend can be put under integration tests or mock tests. This would help us to ensure that our APIs are fault tolerant, load bearing and at the same time fast. automated test Can Run all for it is just like an application consuming each one of them individually and very fine the request and the responses.

Chapter 6

Result

6.1 Python Flask NLP

This shall form the penultimate chapter of the report and shall include a thorough evaluation of the investigation carried out and bring out the contributions from the study. The discussion shall logically lead to inferences and conclusions as well as scope for possible further future work.

Upon being trained on a dataset of only 400 documents, with varying data, our model managed to get an F-score of 0.63, which is not as high as expect it to be our model is handling huge chunks of data filter out matching/similar data which is just a very small portion of the actual data. This is one of the major reasons that the model fails to be efficient against documents with just a slight change in their appearance. One of the ways by which the application's performance can be improved is by constantly training at against thousands of new documents this will ensure that the predictions are close to accurate if not completely accurate.


```
{
  "Date": "28-Nov-19",
  "GSTIN": "33AATCG73851125",
  "Invoice number": "LTN02B1920003774",
  "Items": [
    "VEG RICE BOWL MEA",
    "CLASSIC LEMONADE",
    "MILD BASTING\nGAL",
    "VEG RICE BOWL MEA",
    "CLASSIC LEMONADE",
    "MILD BASTING\n",
    "VEG RICE BOWL MEA\nGAL",
    "CLASSIC LEMONADE",
    "MILD BASTING GAL",
    "QUARTER CHICKEN M",
    "CORN ON THE COB",
    "CLASSIC LEMONADE",
    "MILD BASTING GAL"
  ],
  "Store address": "Unit No: UG-41,PMC,Old Door.No. 66,
  "Store name": "Calito's",
  "Store name-1": "Galito's",
  "Time": "16:47",
  "Total bill amount": 762.0
}
```


Figure 6.1: High Output for more dataset images

6.2 React Frontend

```
{
  "GSTIN": "33AAECT2235P 1Z6",
  "Invoice number": "PM7332",
  "Items": [
    "H/S TOP XXL",
    "TOP H/S B XXL",
    "PAPER BAG M"
  ],
  "Store address": "142, VELACHERY ,MAIN ROAD,\r
  "Time": "07:51 PM",
  "Total bill amount": 269.0
}
```


Figure 6.2: Moderate Output with slight change in the document


```
{
  "Items": []
}
```


Figure 6.3: Empty output with slight change in the documents

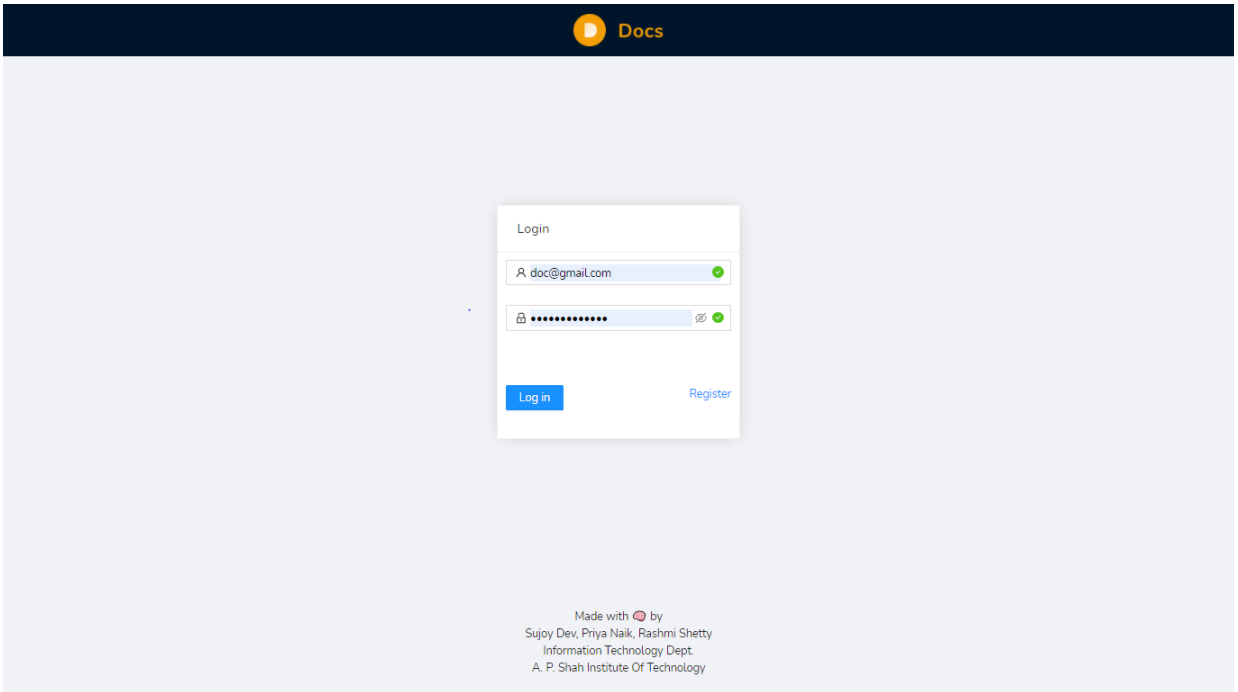

Figure 6.4: Login

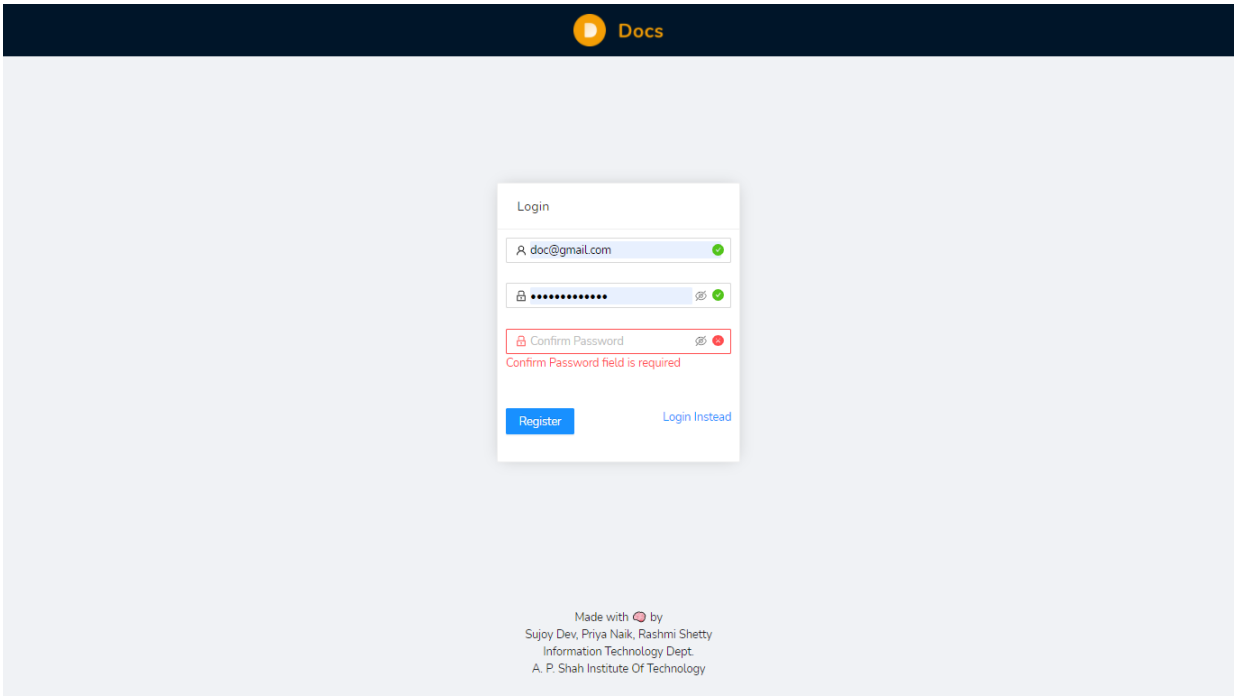

Figure 6.5: Register

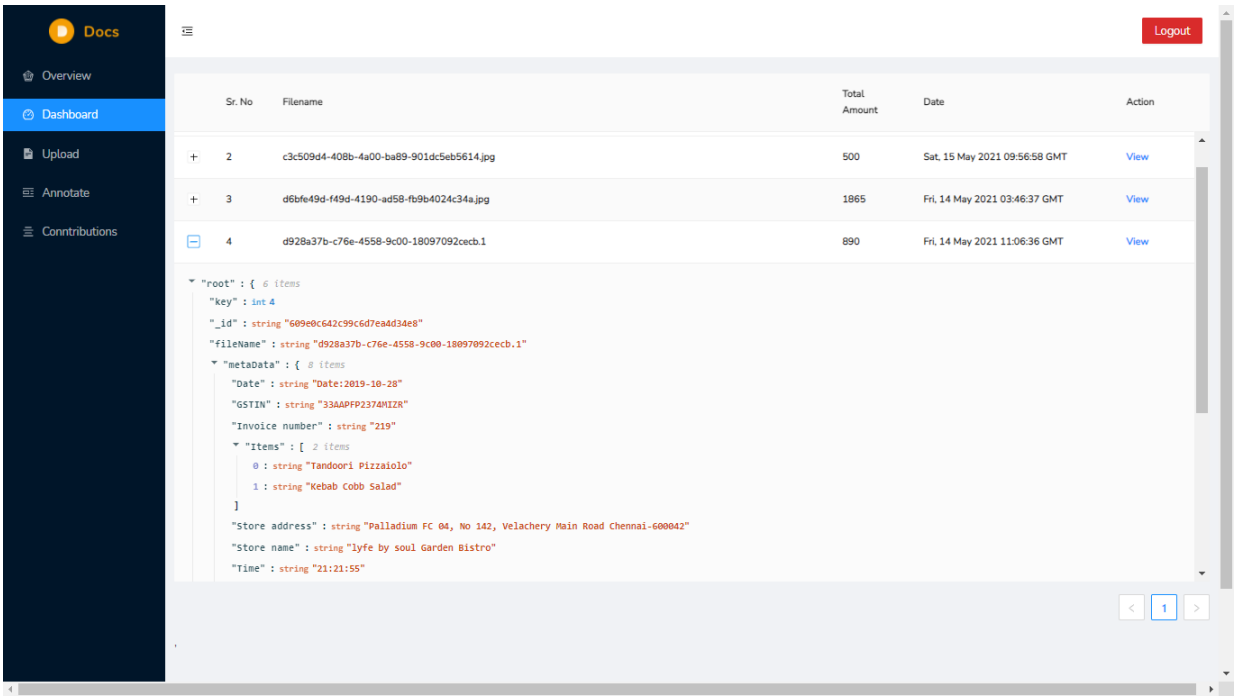

Figure 6.6: Dashboard

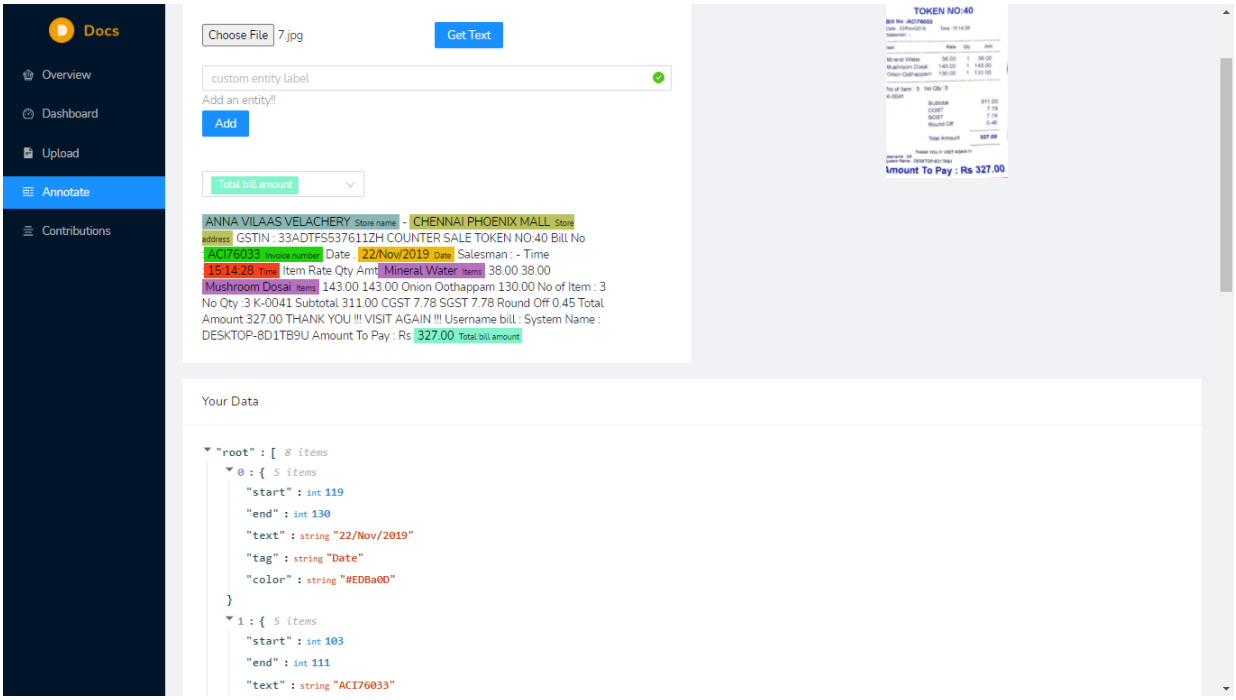

Figure 6.7: Annotate

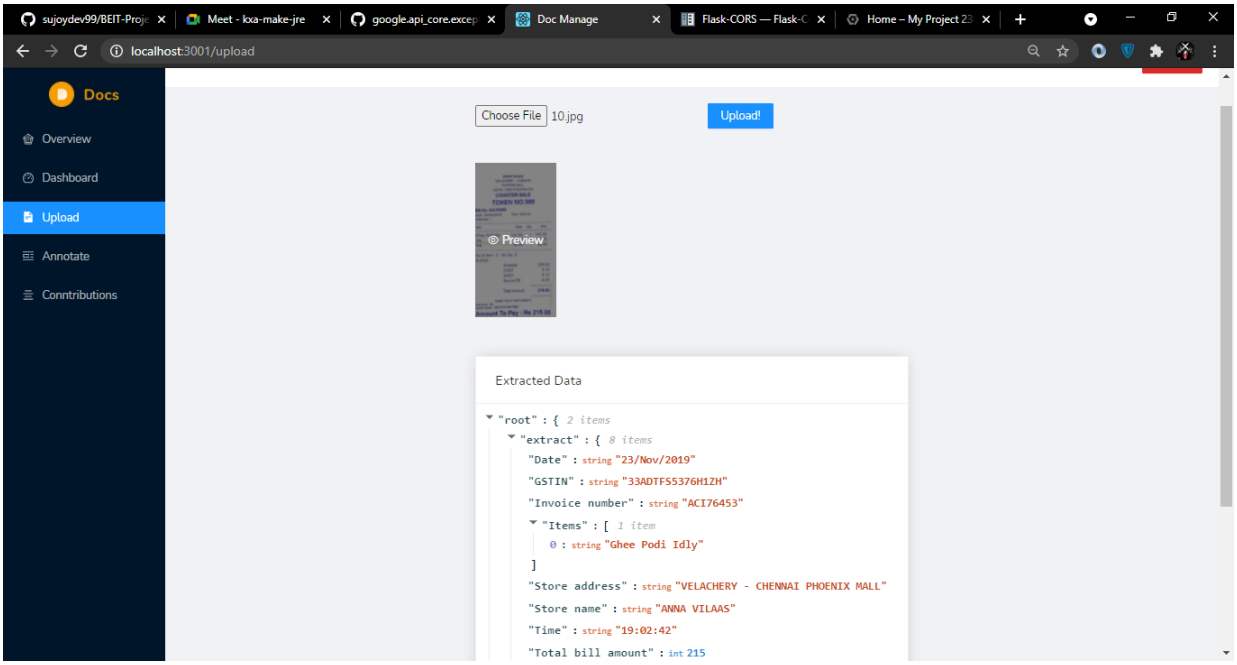

Figure 6.8: Extraction

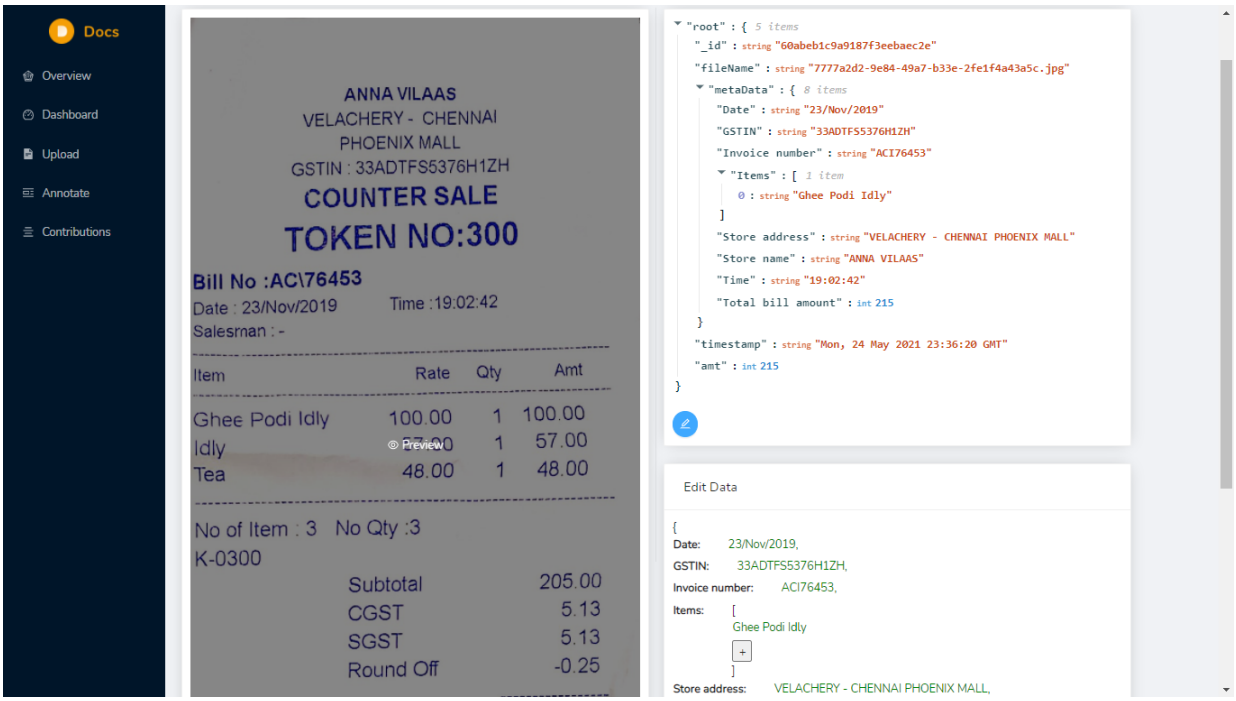

Figure 6.9: Extraction Output + Contribution

Chapter 7

Conclusions and Future Scope

The main objective of building such an application in to increase work efficiency, speed and to reduce manual labour to a minimum. In today's world data is of utmost importance and in order to keep up with the speed at which it is being generated we must move towards automation. Digitization comes with it's own advantages like data persistence, little to no maintenance, everything becomes readily available at our finger tips and most importantly it it has low cost of operation. As the users of the application can generate custom annotations for their own documents they will indirectly help in increasing the data-set size, this will help other users as well. Also the entire application is leveraging Open Source technologies the application can itself remain Open Source, free for any one to use. This will encourage application as well as community growth.

Further with the use of templates and visual markers, template base positional data extraction can be implemented. A strong intermediary between the user submitted training data and the training server will be effective in filtering out invalid data, hence aiding in data cleaning and pre-processing. Industrial invoices can be targeted with sufficient training.

Bibliography

[1] Lu, H., Guo, B., Liu, J., Yan, X. (2017). *A shadow removal method for tesseract text recognition. 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI).*

[2] Sidhwa, H., Kulshrestha, S., Malhotra, S., Virmani, S. (2018). *Text Extraction from Bills and Invoices. 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN).*

[3] K.M. Yindumathi, Shilpa Shashikant Chaudhari, R. Aparna. (2020). *Analysis of Image Classification for Text Extraction from Bills and Invoices. 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT).*

[4] Zhang, J., Ren, F., Ni, H., Zhang, Z., Wang, K. (2019). *Research on Information Recognition of VAT Invoice Based on Computer Vision. 2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS).*

[5] Wei Ruan; Won-sook Lee (2018). *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*

[6] Internet Archive, *https://web.archive.org/web/20081223141143/http://www.paystreamadvisors.com last accessed 28.7.20*

[7] Wikipedia, *https://en.wikipedia.org/wiki/Invoice*$_p$*rocessinglastaccessed*28.7.20*towardsdatascience*

[8] Dataset, https://github.com/zzzDavid/ICDAR-2019-SROIE.git

Appendices

Appendix-I: Python Backend, Spacy Setup

1. Install Python 3

2. Install pip.

3. pip3 install requirements.txt

4. zip models.zip

5. /rootappdir/flask run

Appendix-II: Client Setup

1. cd /app-root-dir

2. npm install

3. npm run start

4. open browser http://localhost:3000/

Appendix-III: OCR

1. set up free Google Vision API credentials or Use PyOCR, Tesseract

2. pip3 install OpenCV3 - for image processing

Publication

Paper entitled **"A. I. Based Document Digitizations"** is selected at **'International Conference on Emerging Trends in Engineering and Technology (ICETET-2021)"** by **"Sujoy Dev, Priya Naik and Rashmi Shetty".**