Volume: 10 Issue: IX Month of publication: September 2022

DOI: https://doi.org/10.22214/ijraset.2022.46839

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue IX Sep 2022- Available at www.ijraset.com

# Document Automation Using Artificial Intelligence

Ajay Bhatt
*Data Scientist (vinnovate Technologies)*

***Abstract: Documents such as Invoices, receipts, bills, ID cards, and passports are basic and very common documents around the world which are used for accounting, tax records, payments, financial history, performing activities like data analytics, digitizing company's records etc. All these documents are available in different formats such as images, PDFs and hard copies (paper). To retrieve the data from these documents, a person needs to manually write or type the data from documents to the table, which is time-consuming as well as irritating. We provide a simple, unique and easily implementable end-to-end approach that uses AI and Deep Learning models to automate the above tasks by just uploading the document of any format image, PDF or Docs, and the software will extract the data and save it in the required structural format. Our approach eliminates the need to manually enter data in an Excel or database record with no limit on the amount of work while companies are facing problems because of their limited workforce and limited work hour for manual data entry, but our software can run 24x7.***

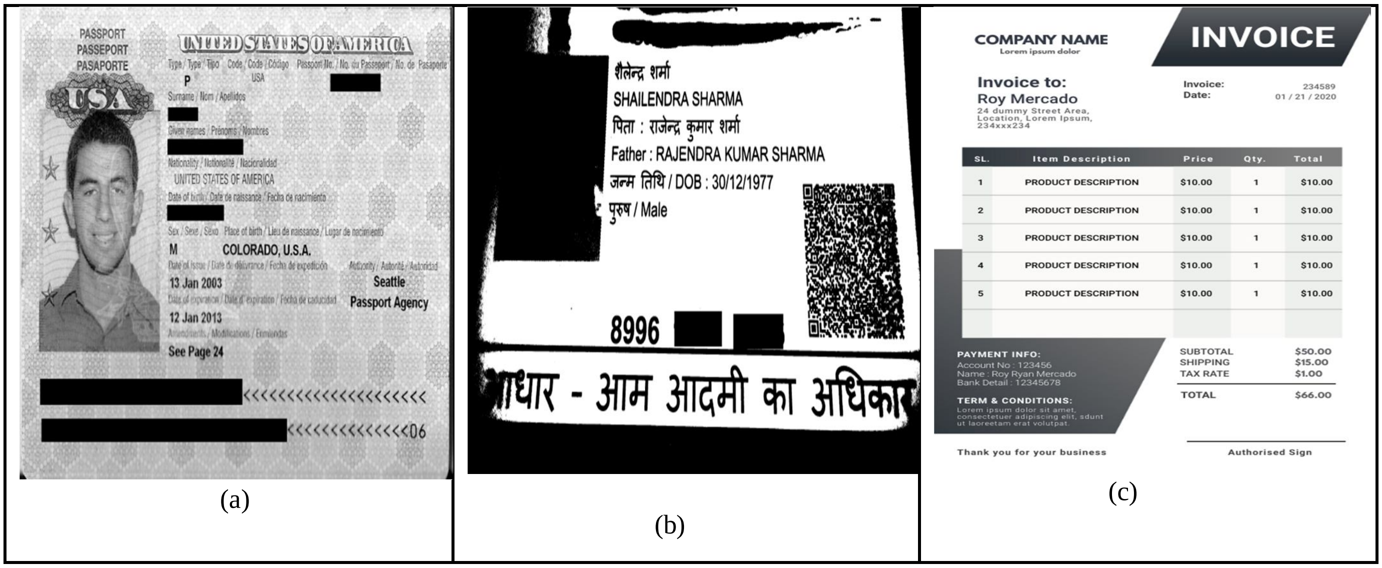
***Keywords: Deep learning, Artificial Intelligence, Model training, Labelling, Data Extraction, OCR***

## I. INTRODUCTION

Manual work is always time-consuming, irritating and also costly when done on a large scale or for a long time. As technology is getting better daily, all the work is shifting from labour work to computer-based automation. Large accounting firms must take care of all their client's financial and personal documents and keep track of their records. Every day they get thousands of such documents and a team of data entry staff keep records of every document by entering its data into their database system. We have developed software which is capable of entering all the data into the database by itself and saving all the labour costs of entering the data manually and speeding up the task by more than 50 times.

We have implemented and integrated three different modules for developing the complete software. One of the modules is manage labels to manage the classes. The second module is model training. It is something which has already been done in the past, but we have added an extra model to make it more robust and accurate. The third module is the data extraction module, which is used to extract the data from the documents using the OCR engine, where it processes embedded text in the scanned documents to the parsable textual form. The last module is the data parser, where we parse the data and convert it to the user-required format.

For implementing all these modules, we are using python as it provides all the necessary frameworks required for AI and Deep Learning. Our software can be easily deployed on any machine or cloud.



PASSPORT
PASSEPORT
PASAPORTE
UNITED STATES OF AMERICA
USA
Type / Type / Tipo: P
Code / Code / Código: USA
Passport No. / No. du Passeport / No. de Pasaporte: [redacted]
Surname / Nom / Apellidos: [redacted]
Given names / Prénoms / Nombres: [redacted]
Nationality / Nationalité / Nacionalidad: UNITED STATES OF AMERICA
Date of birth / Date de naissance / Fecha de nacimiento: [redacted]
Sex / Sexe / Sexo: M
Place of birth / Lieu de naissance / Lugar de nacimiento: COLORADO, U.S.A.
Date of issue / Date de délivrance / Fecha de expedición: 13 Jan 2003
Authority / Autorité / Autoridad: Seattle Passport Agency
Date of expiration / Date d' expiration / Fecha de caducidad: 12 Jan 2013
Amendments / Modifications / Enmiendas: See Page 24
[redacted]<<<<<<<<<<<<<<<<<<<<<
[redacted]<<<<<<<<<<<<<<<06

(a)

[redacted]
[redacted]
शैलेन्द्र शर्मा
SHAILENDRA SHARMA
पिता : राजेन्द्र कुमार शर्मा
Father : RAJENDRA KUMAR SHARMA
जन्म तिथि / DOB : 30/12/1977
पुरुष / Male
8996 [redacted] [redacted]
ाधार - आम आदमी का अधिकार

(b)

COMPANY NAME
Lorem ipsum dolor

INVOICE

Invoice to:
Roy Mercado
24 dummy Street Area,
Location, Lorem Ipsum,
234xxx234

Invoice: 234589
Date: 01 / 21 / 2020

| SL. | Item Description | Price | Qty. | Total |
|---|---|---|---|---|
| 1 | PRODUCT DESCRIPTION | $10.00 | 1 | $10.00 |
| 2 | PRODUCT DESCRIPTION | $10.00 | 1 | $10.00 |
| 3 | PRODUCT DESCRIPTION | $10.00 | 1 | $10.00 |
| 4 | PRODUCT DESCRIPTION | $10.00 | 1 | $10.00 |
| 5 | PRODUCT DESCRIPTION | $10.00 | 1 | $10.00 |

PAYMENT INFO:
Account No : 123456
Name : Roy Ryan Mercado
Bank Detail : 12345678

TERM & CONDITIONS:
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sdunt ut laoreetam erat volutpat.

| | |
|---|---|
| SUBTOTAL | $50.00 |
| SHIPPING | $15.00 |
| TAX RATE | $1.00 |
| TOTAL | $66.00 |

Thank you for your business

Authorised Sign

(c)

Fig. 1: Scanned images of documents with different layouts and formats

## II. PROBLEM DEFINITION

As technology is getting better day by day, automation using AI and ML is making our lives easier. We are trying to develop an automated document parsing system using AI which can reduce the human effort in managing the documents, saving time on the manual work, can be easily implemented, and gives higher or competitive accuracy than any other paid document OCR software in the market with no API request limits (can be self-hosted) and no constraints on the document size or the number of documents processed. We are taking the use case of invoices, as it is the most complex and popular document in the OCR domain. If the software can perform well for invoices, then most of the other use cases should be covered under it too. This use case is applicable in any industry. It will help reduce labour costs as well as human error this labour can be utilized in a complex tasks instead of repetitive work which automation can do.

## III. LITERATURE SURVEY

Joseph Redmon [2] (2016) introduced YOLO, a game-changing model and a new approach to object detection, before other object detection models used two steps for object detection: first detection and then classification. But YOLO did both in a single step. It can process images in real-time at 45 frames per second (smaller architecture can go up to 155 frames per second) which makes it faster than any other model. Even in accuracy, YOLO outperforms all the other detection models.

When YOLO was introduced, it was implemented with its custom framework called Darknet. But as of today, it is also available in PyTorch. It was one of the most used models for getting the coordinates of the text in OCR.

Yiheng Xu [1] (2019) of Harbin Institute of Technology, Minghao Li of Beihang University and several other researchers from Microsoft Asia published a research paper named “LayoutLM: Pre-training of Text and Layout for Document Image Understanding” introducing a model which is capable of understanding the layout of a document and the content in the document successfully. They used BERT (Bi-directional Encoder Representational of Transformer) [5] which is a multi-layer attention-based bidirectional Transformer encoder Architecture as a backbone for the LayoutLM model [14].

They have used two different pieces of information from the documents to train the LayoutLM model.

1) Document Layout Information: This is used to understand the textual data or the format of textual data and how the data is arranged in the document as for most kinds of documents the name of the person will come after the name keyword or below the name keyword (using positional embedding).
2) Visual Information: This is used to understand the structure or the position of the words in the document, for example, the name of a person mostly comes at the start of the document (using image embedding).

LayoutLM replaces all models like YOLO and other approaches for OCR, it's the first model of its kind.

| Researcher | Models | Results |
|---|---|---|
| Joseph Redmon (2016) | YOLO (You only look Once), a real-time object detection model. | Provides decent results on marking the fixed template and table structure. But fails in understanding the text. |
| Yiheng Xu, Microsoft (2019) | LayoutLM a Bert base architecture model that uses positional and image embedding | Easily understands the whole text and provides amazing results for texts, but fails for table structure. |

Table 1. Comparing Existing Deep Learning Models Structure Detection

## IV. METHODOLOGY

The working of the software is divided into multiple modules, each module has its separate important job. These modules are sequential and processed one after another. These are the following modules along with their working.

1) *Manage Labels:* In manage classes, we are managing the classes or the labels available in the documents example company name, invoice number, shipping address, and ID number. Each label has its two own properties which are used during the time of data extraction, as these properties are user-defined and can be easily changed according to the requirements.

   One of the properties is the data type which defines the type of data that will be returned by the label, while the other property is the position type which tells us whether the label belongs to the table or not. If it belongs to the table, its position type is a Table Item, otherwise, it's a Non-Table item. Some sample labels are in the following Table 2.

| labels | date type | position type |
|---|---|---|
| Invoice Number | Numeric | Non-Table Item |
| Date | Timestamp | Non-Table Item |
| Quantity | Alphanum | Table Item |
| Descriptions | Text | Table Item |
| Amount | Numeric | Table Item |
| Total Amount | Numeric | Non-Table Item |

Table 2. Sample Data Type and Position Type for Labels

2) *Labelling and Training Module:* In this module, the labelling of data and training of two different models is performed. We are using the LayoutLM model which uses BERT [5] architecture model for understanding the text and layout of the documents. While the other model is Detectron 2 [15]. Detectron is a state-of-art object detection model which uses a Caffe 2 deep learning framework, built on top of Faster RCNN [3], RPN (region proposal network), and Resnet [4]. It surpasses all other models in terms of correctly differentiating the classes, while models like YOLO are more accurate in bounding boxes but not in predicting the classes in the bounding box. For labelling, we use makesense.ai [16] on a real-world invoice dataset from one of our clients. The dataset includes more than 5000 invoices belonging to multiple different templates. Both models were trained in different classes LayoutLM is trained on non-table items like company name, bill to, date, invoice number, and total amount while the Detectron 2 model is trained on the table items like description, amount, price, quantity, tax etc.

For training, we use the Google colab pro plus version [17] which comes with 8 core CPU, 52 GB of RAM, background execution and 16 GB VRAM GPU (P100, T4, V100) LayoutLM took 5 hours for 100 epochs with batch size 16 while Detectron 2 took 3 hours with batch size 16.

While testing the models on our dataset we didn't go for any metrics like accuracy, precision, or IOU we make our custom test data and defined our own two custom metrics based on IOU (Intersection Over Union) [18] called min detection precision and max detection precision which are based on the area of the for text. In our testing, we achieved more than 96% accuracy in min_detection_precision and 93% accuracy in max_detection_precision, which is great for our use case.

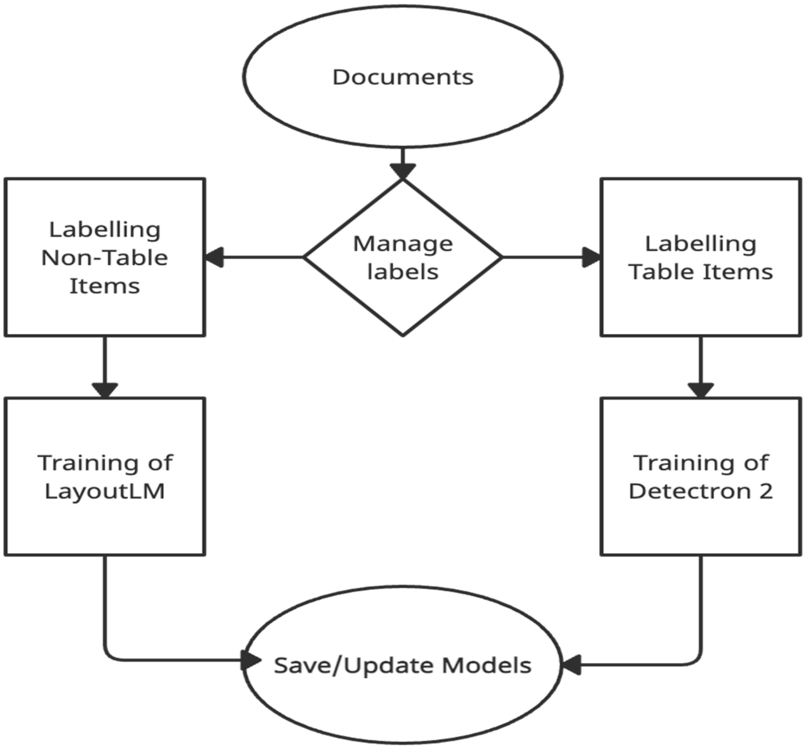


Fig. 2: Workflow of Model Training

3) *Data Extraction and Parsing Module:* Once we get the coordinates of the labels from the LayoutLM and Detectron 2. They are passed to the data extraction and parsing module scripts. This module is divided into two processes, one is data extraction (OCR) and another is data parsing.

Data Extraction: It is a dependent module which depends on the OCR engine used. There are many OCR engines available in the market like Pytesseract [19] and EasyOCR, these are some free OCRs, while Google Vision [20], Azure OCR and OCR Space are paid OCR, able to understand low-quality or blurred text in documents and handwritten text as well.

In our software, we use tesseract (free) and Google Vision (paid) OCR engines so that users can select the OCR engine according to its document quality and financial budget. For each document, we are sending a single request to the OCR engine to keep the inference latency low and the cost minimal (in the case of paid OCR). After getting the data from the OCR engine, we are selecting the data using the output obtained from LayoutLM and Detectron2 model.

Data parsing: we have defined separate parsing functions for each label, using the manage labels for parsing the data. After getting the output from Data Extraction and parsing module, we send the output of each label to their respective functions. Labels which are predicted by LayoutLM have a quite simple parsing function based on the regular expression. While on the other hand, labels obtained from the Detectron 2 model have special logic to deal with the table structure data. Tables consist of rows and columns, In invoices for a single item the rows could be of multiple lines example in descriptions the rows might vary from one to n number of lines which is hard to extract in proper structure because amounts, prices, quantity are always one line but the description is unpredictable which is one of the most complex and difficult part in data parsing.The parsing of the table items is not only based on the regular expression but also on the respective position of the key column of the table, our custom logic to handle the table data is able to retrieve the table data in proper structured form without mixing up any row or column of the table. The complete workflow during inference is shown in Figure 3.

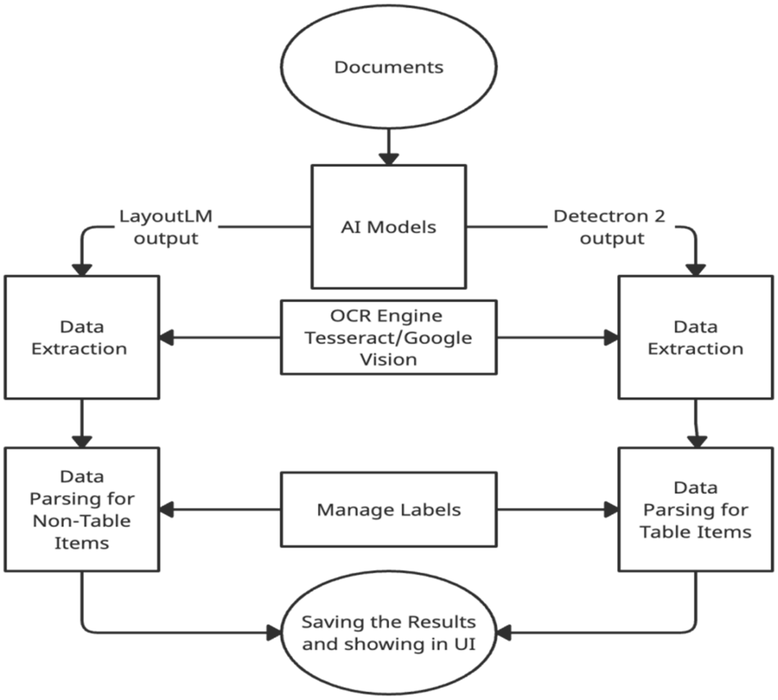

Documents
AI Models
LayoutLM output
Detectron 2 output
Data Extraction
OCR Engine Tesseract/Google Vision
Data Extraction
Data Parsing for Non-Table Items
Manage Labels
Data Parsing for Table Items
Saving the Results and showing in UI

Fig. 3: Workflow of Software During Inference

Most modules are written in python, manage labels and some UI functionality is written in JavaScript. All the scripts and methodologies are highly optimized to process the document without spending much time. If there are any new templates in the pipeline which are totally different from the old trained templates, then both the models need to be retrained again by labelling the documents as done in the initial training process so that the models will understand the new templates. Our data parsing scripts are based on NLP as well as computer vision. Parsing scripts for table items are based on the corresponding position of table rows, developed after a lot of research on how to parse a table without mixing any column or row data.

The software is so light that it can be deployed on any low-configuration machine. We are also creating a training module so that any non-programmer can easily label and train the models from the UI.

## V. CONCLUSIONS

By combining all the above modules, we have developed software which can process all kinds of documents, our software surpasses most of the current models and software in the market. LayoutLM is the most successful model in document understanding, it is able to understand all the textual data in the document, but fails when it comes to understanding table data. While the software we developed uses a combination of two different AI models for different purposes to handle various kinds of cases, it understands all kinds of structural data and converts the data into the required output with high accuracy. There are many paid software in the market handling tabular and other unstructured data using hard-coded logic, but we are using AI models to make it more generic and scalable. We have also added two non-AI methods to the software. One is the manual mode in which documents can be processed in a manual model where the user needs to mark the coordinates manually instead of getting them from AI models and another one is the semi-automatic mode in which the user needs to first define the template. Then after saving the template for every document, the correct template will get selected automatically, and the coordinate will be applied accordingly. The output of the software can be also integrated with Organization's ERP, Accounting or any 3rd party system.

The software is already available in the market with the name vDigiDocr by vInnovate technologies and is used by 4 of our clients who are processing thousands of invoices, purchase orders and other documents reliably every single day.

## REFERENCES

[1] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou. LayoutLM: Pre-training of Text and Layout for Document Image Understanding by Microsoft arXiv:1912.13318

[2] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91

[3] R. Girshick, "Fast R-CNN," 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440-1448, doi: 10.1109/ICCV.2015.169.

[4] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding arXiv:1810.04805

[6] M. Minouei, M. R. Soheili and D. Stricker, "Document Layout Analysis with an Enhanced Object Detector," 2021 5th International Conference on Pattern Recognition and Image Analysis (IPRIA), 2021, pp. 1-5, doi: 10.1109/IPRIA53572.2021.9483509.

[7] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 29802988, doi: 10.1109/ICCV.2017.322

[8] S. W. Lam and S. N. Srihari, "Frame-based knowledge representation for multi-domain document layout analysis," Conference Proceedings 1991 IEEE International Conference on Systems, Man, and Cybernetics, 1991, pp. 1859-1864 vol.3, doi: 10.1109/ICSMC.1991.169955.

[9] S. Shi, C. Cui and Y. Xiao, "An Invoice Recognition System Using Deep Learning," 2020 International Conference on Intelligent Computing, Automation and Systems (ICICAS), 2020, pp. 416-423, doi: 10.1109/ICICAS51530.2020.00093.

[10] U. S. Unal, E. Unver, T. Karakaya and Y. S. Akgul, "Invoice Content table analysis with feature fusion," 2015 23rd Signal Processing and Communications Applications Conference (SIU), 2015, pp. 2298-2301, doi: 10.1109/SIU.2015.7130337.

[11] A. Conway, "Page grammars and page parsing. A syntactic approach to document layout recognition," Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR '93), 1993, pp. 761-764, doi: 10.1109/ICDAR.1993.395626.

[12] J. Li, C. -M. Lin and S. -x. Hu, "Intelligent Document Processing Method Based on Robot Process Automation," 2021 Global Reliability and Prognostics and Health Management (PHM-Nanjing), 2021, pp. 1-6, doi: 10.1109/PHM-Nanjing52125.2021.9613052

[13] W. Zhang, "Online Invoicing System Based on QR Code Recognition and Cloud Storage," 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2018, pp. 2576-2579, doi: 10.1109/IMCEC.2018.8469461

[14] Official GitHub repository for layoutlm and other layout models https://github.com/microsoft/unilm.

[15] Facebook detectron2 https://ai.facebook.com/tools/detectron/, https://github.com/facebookresearch/Detectro

[16] data labelling tool https://www.makesense.ai

[17] google colab notebook documentation https://research.google.com/colaboratory/faq.htm

[18] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid and S. Savarese, "Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 658-666, doi: 10.1109/CVPR.2019.00075.

[19] tesseract official documentation https://tesseract-ocr.github.io/tessdoc/

[20] google vision documentation and API tutorials https://cloud.google.com/vision/docs.