Tablytix

Capstone Project Report

END SEMESTER EVALUATION

Submitted by:

102016040	Ojas Sharma
102016026	Harneet Kaur
102016068	Agamjot Singh
102016012	Mayank Rawat
102016080	Anshul Kanwar

BE Fourth Year, CSE

CPG No: 212

Under the Mentorship of

Dr. Jasmeet Singh(Assistant Professor)

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology, Patiala

December 2023

ABSTRACT

In the present digital age, a substantial volume of data resides in digital documents, presenting both opportunities and challenges for companies. The emphasis on utilizing structured data, primarily organized in tables, for informed decision-making is evident. However, a noteworthy challenge arises from privacy concerns associated with logos embedded in these documents. In order to smoothly share such digital documents with other organizations for collaboration, extensive efforts would be required to eliminate logos. This process is not only time-consuming but also poses a financial burden on the companies involved. Tables are intricate; they might be nested or semi-structured. It takes some effort to comprehend them initially, and extracting valuable insights from the stored information is even more challenging. Efficiently extracting needed information from tables is crucial for time savings. Unveiling insights hidden in these tables can give businesses a competitive advantage.

We decided to address this concern by creating a tool that meets businesses' privacy requirements and enhances their ability to access information effectively. Opting to handle the substantial data generated by companies, it became clear that utilizing state-of-the-art machine learning models was the only viable solution for efficiently managing extensive documents. Hence our capstone project, Tablytix introduces a new framework for thorough document analysis. Our project aims for achieving various goals, including removal of logos, understanding document layouts, and extracting information from tables. Additionally creating an advanced query answering system for .

DECLARATION

We hereby declare that the design principles and working prototype model of the project entitled Tablytix is an authentic record of our own work carried out in the Computer Science and Engineering Department, TIET, Patiala, under the guidance of Dr. Jasmeet Singh during 7th semester (2023).

Date: 19/12/2023

Registration No.	Name	Signature
102016040	Ojas Sharma	
102016026	Harneet Kaur	
102016068	Agamjot Singh	
102016012	Mayank Rawat	
102016080	Anshul Kanwar	

Counter Signed By:

Faculty Mentor: Dr. Jasmeet Singh

Assistant Professor, CSED, TIET, Patiala

ACKNOWLEDGEMENT

We would like to express our heartfelt thanks to our mentor Dr. Jasmeet Singh. He has been of great help in our venture, and an indispensable resource of technical knowledge. He is truly an amazing mentor to have.

We are also thankful to Dr. Shalini Batra, Head of Computer Science and Engineering Department, entire faculty and staff of Computer Science and Engineering Department, and also our friends who devoted their valuable time and helped us in all possible ways towards successful completion of this project. We thank all those who have contributed either directly or indirectly towards this project.

Lastly, we would also like to thank our families for their unyielding love and encouragement. They always wanted the best for us and we admire their determination and sacrifice.

Date: 19/12/2023

Roll No.	Name	Signature
102016040	Ojas Sharma	
102016026	Harneet Kaur	
102016068	Agamjot Singh	
102016012	Mayank Rawat	
102016080	Anshul Kanwar	

TABLE OF CONTENTS

ABSTRACT **2**
DECLARATION **3**
ACKNOWLEDGEMENT **4**
TABLE OF CONTENTS **5**
LIST OF TABLES **8**
LIST OF FIGURES **9**
INTRODUCTION **11**
1.1.1 Technical terminology 11
1.1.2 Problem Statement 12
1.1.3 Goal 13
1.1.4 Solution 13
1.2 Need Analysis 14
1.3 Research Gaps 15
1.4 Problem Definition and Scope 16
1.5 Assumptions and Constraints 16
1.6 Standards 18
1.7 Approved Objectives 18
1.8 Methodology 19
Phase 1 19
1.9 Project Outcomes and Deliverables 20
1.10 Novelty of Work 21
REQUIREMENT ANALYSIS **22**
2.1 Literature Survey 22
2.1.1 Related Work 22
2.1.2 Research Gaps of Existing Literature 24
2.1.3 Detailed Problem Analysis 33
2.1.4 Survey of Tools and Technologies Used 34
2.1.5 Summary 34
2.2 Software Requirement Specification 35
2.2.1 Introduction 36
2.2.1.1 Purpose 36
2.2.1.2 Intended Audience and Reading Suggestions 36

2.2.1.3 Project Scope....37
2.2.2 Overall Description....37
2.2.2.1 Product Perspective....37
2.2.2.2 Product Features....37
2.2.3 External Interface Requirements....38
The various interface requirements are as follows:....38
2.2.3.1 User Interfaces....38
2.2.3.3 Software Interfaces....38
2.2.4 Other Non-functional Requirements....38
2.2.4.1 Performance Requirements....39
2.2.4.2 Safety Requirements....39
2.2.4.3 Security Requirements....40
2.3 Cost Analysis....40
2.4 Risk Analysis....40
METHODOLOGY ADOPTED....42
3.1 Investigative Techniques....42
3.2 Proposed Solution....43
3.3 Work Breakdown Structure....46
3.4 Tools and Technology....47
DESIGN SPECIFICATIONS....48
4.1 System Architecture....48
4.1.1 Block Diagram....48
4.1.2 Component Diagram....49
4.1.3 State Chart Diagram....50
4.2 Design Level Diagrams....52
4.2.1 Data-Flow Diagrams....52
4.2.2 ER Diagram....54
4.2.3 Class Diagram....55
4.3 User Interface Diagrams....56
4.3.1 Use case Diagrams....56
4.3.2 Activity Diagram....66
4.4 Snapshots of Working Prototype....68
Implementation and Experimental Results....70
5.1 Experimental Setup....70
5.2 Experimental Analysis....70
5.2.1 Data....70
5.2.2 Performance Parametres....71
5.3 Working of the project....72
5.3.1 Procedural Workflow....72
5.3.2 Algorithmic Approaches Used....75
5.3.3 Project Development....85
5.3.4 System Screenshots....87

5.4 Testing Process..... 90
5.4.1 Test Plan..... 90
5.4.2 Features to be tested..... 91
5.4.3 Test Strategy..... 91
5.4.4 Test Techniques..... 92
5.4.5 Test Cases..... 93
5.4.6 Test Results..... 94
5.5 Results and Discussions..... 95
5.6 Inferences Drawn..... 95
5.7 Validation of Objectives..... 96
Conclusions and Future Directions..... 96
6.1 Work Accomplished..... 96
6.2 Conclusions..... 97
6.3 Environmental, Economic and Societal Benefits..... 97
6.4 Reflections..... 99
6.5 Future Work Plan..... 100
Project Metrics..... 101
7.1 Challenges Faced..... 101
7.2 Relevant Subjects..... 101
7.3 Interdisciplinary Knowledge Sharing..... 102
7.4 Peer Assessment Matrix..... 103
7.5 Role Playing and Work Schedule..... 103
7.6 Student Outcomes Description and Performance Indicators (A-K Mapping)..... 104
7.7 Brief Analytical Assessment..... 107
APPENDIX A: REFERENCE..... 109
APPENDIX B: Plagiarism Report..... 113

LIST OF TABLES

Table 1: Assumptions of Tablytix..... 17
Table 2: Constraints of Tablytix..... 17
Table 3: Standards of Tablytix..... 18
Table 4: Research Findings..... 25
Table 5: Cost Analysis..... 40
Table 6: Risk Analysis..... 40
Table 7: Use Case #1 Sign Up..... 58
Table 8: Use Case #2 Login..... 58
Table 9: Use Case #3 Forgot Password..... 59
Table 10: Use Case #4 Tablytix Homepage..... 59
Table 11: Use Case #5 Upload Document..... 60
Table 12: Use Case #6 Access Previous Documents..... 60
Table 13: Use Case #7 Edit Profile..... 61
Table 14: Use Case #8 Logo Redaction..... 62
Table 15: Use Case #9 Document Analysis..... 62
Table 16: Use Case #10 Document Download..... 63
Table 17: Use Case #11 Table Extraction..... 64
Table 18: Use Case #12 Query Answering System..... 64
Table 19: Use Case #13 Change Password..... 65
Table 20: Use Case #14 Log-out..... 65
Table 21: Test Cases..... 93
Table 22: Validation of Objectives..... 96
Table 23: Relevant Subjects..... 101

Table 24: Peer Assessment Matrix 103
Table 25: S01-S07 Mapping for course ‘UCS797-Capstone Project’..... 104
Table 26: References for websites..... 112

LIST OF FIGURES

Figure 1: Gantt Chart..... 47
Figure 2: Block Diagram..... 49
Figure 3: Component Diagram..... 50
Figure 4: State Chart Diagram..... 52
Figure 5: Data Flow Diagram (DFD-0)..... 53
Figure 6: Data Flow Diagram (DFD-1)..... 54
Figure 7: Data Flow Diagram (DFD-2)..... 54
Figure 8: ER Diagram..... 55
Figure 9: Class Diagram..... 56
Figure 10: Use Case Diagram..... 57
Figure 11: Activity Diagram..... 67
Figure 12: Screenshot of Tablytix Homepage..... 68
Figure 13: Screenshot of Layout Analysis page..... 68
Figure 14: Screenshot of result of Layout Analysis..... 69
Figure 15: Screenshot of result of Logo Redaction..... 69
Figure 16:Procedural Workflow..... 74
Figure 17: Logo Redaction Model Implementation..... 75
Figure 18: Training of Logo Redaction model..... 75
Figure 19: Training loss graph for trained model..... 76
Figure 20:Validation loss graph for trained model 76
Figure 21: COCO Metric graph for trained model..... 77

Figure 22:Loss and Learning rate graph for trained model 77
Figure 23: Loss and Learning rate graph for trained model..... 78
Figure 24: Layout Analysis diagram.....80
Figure 25:Transformations in Layout Analysis 80
Figure 26: Model Results during Layout Analysis.....81
Figure 27: Component Diagram.....85
Figure 28: Deployment diagram.....86
Figure 29: Web application user interface.....87
Figure 30: Upload file option for Layout Analysis..... 87
Figure 31:Result after Layout Analysis 88
Figure 32: Result after Logo Redaction..... 88
Figure 33: Result after Table Extraction.....89
Figure 34: Results for Query Answering System 1.....89
Figure 35: Results for Query Answering System 2..... 90
Figure 36: Model evaluation results for Table detection..... 94
Figure 37: Gantt Chart..... 104

1.1.1 Technical terminology

Logo Redaction: Logo Redaction is a process used to selectively obscure visual elements such as logos within digital documents. The primary focus is to safeguard sensitive information and maintain privacy when sharing documents. Logo Redaction tools often employ advanced image processing and recognition algorithms to identify and obscure logos efficiently. This not only enhances the privacy of the shared documents but also streamlines the workflow for businesses that regularly exchange information while preserving the integrity of their brand identity.

Document Layout Analysis: Document layout analysis is a process in which computer algorithms examine and interpret the structure and organization of a document's content. The goal is to understand how different elements such as text, images, and other components are arranged within a document. This analysis is particularly useful in the field of document processing, where automation and information extraction from documents are essential.

Table Extraction: Table extraction is a process in document analysis that involves automatically identifying and extracting tabular structures from digital documents. The primary objective is to recognize tables within documents, which may contain structured data, and extract this tabular information for further analysis or use. Table extraction is particularly valuable in scenarios where large volumes of data are stored in tabular form, such as financial reports, payrolls, and spreadsheets by businesses.

Query Answering System: A query answering system is a technology or software that enables users to ask questions in natural language and receive relevant and meaningful responses. These systems are designed to understand user queries,

interpret the intent behind the questions, and retrieve or generate accurate answers from a given dataset or knowledge base.

1.1.2 Problem Statement

In recent times, the digital landscape has witnessed an exponential surge in the availability of information within digital documents. Businesses, in particular, grapple with vast volumes of digital documents containing crucial data. However, a pressing concern emerges when it comes to sharing these documents—privacy issues linked to logos. In the context of businesses, protecting the brand image is indeed crucial. Before sharing any document, businesses often need to undergo the cumbersome process of logo removal to safeguard sensitive information. Given that it is performed manually would be such a waste of time and resources. In order to efficiently perform the required task, we propose the use of state-of-art machine learning models and deep learning models, automating the task. Logo Redaction using machine learning models helps companies maintain the control over the visibility of their logos, preventing unauthorized and unintended exposure. It ensures that the shared information remains confidential, addressing privacy concerns and compliance requirements. This not only enhances the privacy of the shared documents but also streamlines the workflow for businesses that regularly exchange information while preserving the integrity of their brand identity.

Next concern that we are addressing is the vast amounts of information available within these businesses and not so easily accessible, being in the forms of tables. The digital documents being created, and stored in the businesses are filled with tables, containing a wealth of information. However accessing or extracting information from these tables has proven to be a significant challenge, given their complex structures. Recognising this, we are developing a web application designed to streamline this process, making it both easy and efficient. First we aim to perform Document Layout Analysis, which focuses on highlighting the different components contained within the documents. Next we aim to perform Table Extraction which would be converting the information stored in the table into the JSON format, which is easily searchable and indexable. Our web application not only simplifies the

extraction of information from tables but also responds to user queries, providing quick and accurate access to valuable insights stored within these tabular structures.

This innovative solution aims to revolutionize how businesses interact with and extract meaningful information from their extensive digital document repositories. We believe this solution shows potential promise, as with time the machine learning models and deep learning models used in this project are very rapidly gaining better efficiency and accuracy, providing our project with competitive advantage for getting even better results in the future.

1.1.3 Goal

The goal of this project is to develop a reliable, cost effective and efficient web application, which at its core utilizes a combination of state-of-art machine learning models and deep learning models aiming to analyze documents, perform logo redaction and table extraction with less delay and maximum accuracy. Different combinations of datasets and models shall be trained and tested in order to discover the combination with most accuracy. This web application should also contain a query answering system which further exploits the tables extracted from the documents for useful insights. Tablytix provides all essential features combined into one platform for smooth functioning, making Tablytix a one-stop shop for businesses, catering to their digital document needs.

1.1.4 Solution

Tablytix aims to provide logo redaction, document layout analysis, table extraction and query answering system, all combined into one complete platform. All the needs of the businesses related to digital documents have been catered to here. It is not limited to a single functionality of table extraction, but provides a whole variety of functionalities, along with an easy to use user interface.

Datasets: Many different models and datasets would be considered and trained and the best possible model would be considered and further worked upon. Combination of different models and layers would be tested to achieve maximum accuracy.

Models: To achieve optimal accuracy and efficiency, initially a variety of combinations of machine learning and deep learning models shall be experimented

with. Exploring different configurations and integrations of these models to identify the most effective combinations.

Deployment of models: Look into various tools and technologies for seamless integration of machine learning models into applications backend. This involves exploring various frameworks, programming languages, and infrastructure solutions to find the most compatible and efficient setup for our ML models.

1.2 Need Analysis

Digital documents play a pivotal role in the present day, serving as the backbone of modern information exchange and storage. In a world driven by technology, these electronic files have become the primary means of recording, sharing, and archiving information. Businesses also store most of their data, ranging from reports to records, in digital documents.

Given that businesses house an extensive amount of their data in the form of digital documents, efficient management becomes crucial. This includes being cautious, especially when sharing documents that feature their brand logos. Privacy preservation is a top priority, and to achieve this, businesses require tools like logo redaction. These tools ensure that sensitive visual identities, such as logos, are safeguarded during document sharing, maintaining the integrity of the brand and addressing privacy concerns effectively.

To have a good understanding of the importance of redaction, let's look at some examples. Back in 2016, the US Department of Justice conducted an investigation into the interferences within the Presidential Election, and in 2019, all the documents were published. Those documents possessed plenty of information right, from the jury names to the investigative methods. All these things were possible due to redaction. If this method was not used, all the confidential information would be available to the public. [1]

In the realm of digital data, businesses grapple with the challenge of efficiently extracting information from tables. The need for user-friendly search mechanisms within digital documents containing tabular structures is crucial. This task extends beyond mere extraction, requiring the unveiling of valuable insights hidden within

complex table arrangements. Simplifying navigation through these structures is essential for businesses to retrieve meaningful information and gain insights. The era of digitization, driven by technological advancements, has flooded us with unstructured data. Tracking tabular information in this vast pool has become resource-intensive. The solution lies in leveraging computers, despite the drawback of digitized tables losing their original essence. Our focus is on comprehending tables within unstructured data, isolating them, and restoring their original form through computer vision. [2].

1.3 Research Gaps

Research gap 1: Lack of complete integration:

While existing tools offer functionalities like OCR, basic table extraction, search etc., there is a lack of a comprehensive platform that brings together document analysis, logo redaction, advanced table extraction and intelligent querying in an integrated solution. Current tools only address parts of the problem.

Research gap 2: Handling of complex nested tables:

With an increasing number of mobile devices equipped with cameras, an increasing number of customers are uploading documents via these devices, making the need for information extraction from these images more pressing. Currently, these document images are often manually processed resulting in high labor costs and inefficient data processing times. In businesses this cost is high considering the amount of digital documents they produce. [3]

Research gap 3: Contextual querying:

The ability to query extracted tabular data contextually to gain more insights into the tables of the digital documents is lacking. Additionally given the modern tools of NLP provides such an immense powerful tool for handling of user queries, but integration of such is still lacking in order to provide the contextual answers to the user queries.[4]

Research gap 4: Accurate logo redaction:

Existing logo redaction capabilities often fail to completely redact logos, leaving artifacts behind. More robust image analysis and modern deep learning techniques are

needed to precisely identify logos and redact them without affecting document structure. [5]

Research gap 5: Diverse document handling:
Effective handling of documents in diverse formats like PDFs, scanned images, presentations etc. using a unified analysis framework remains difficult. Generalizable solutions that can work across document types require further innovation.\

1.4 Problem Definition and Scope

The process of analyzing financial documents and extracting relevant information can be time-consuming and prone to errors, especially when done manually. This can lead to inefficiencies and delays in decision-making for organizations that rely heavily on financial data. Additionally, the presence of sensitive information, such as company logos, can pose privacy concerns.
To address this problem, there is a need for a tool that can automate the process of analyzing financial documents, redact sensitive information, extract information from financial tables, and store it in an indexable format. The tool should also be able to answer user queries based on the extracted data, providing faster and more accurate insights to aid in decision-making.
The development of such a tool would greatly benefit financial institutions, research firms, and other organizations that rely on financial data. It would improve the efficiency and accuracy of financial analysis while ensuring privacy and security of sensitive information.
Scope:

- Automatic Logo Redaction performed on the documents
- Efficient Document Layout Analysis and Table Extraction
- Accurate conversion of table data into JSON format
- Answering of the queries aimed for information stored in tables

1.5 Assumptions and Constraints

Assumptions:

Based on our knowledge, experience, and information offered by our team, an assumption is something we think to be true. These are occurrences or situations that we expect to occur during the course of our project's life cycle. The risk management method includes assumption analysis. It may have an impact on the project's health if it is not adequately analyzed.

Table 1: Assumptions of Tablytix

S.no.	**Assumptions**
1.	It is assumed that if any new technique with more accuracy arises, it would be considered during the development phase.
2.	It is assumed that a combination of models shall be experimented, in order to achieve the required accuracy and efficiency.
3.	While training our model, it might happen that the inputs are wrongly interpreted and hence the dataset may become misleading.
4.	The structured data entries produced by the machine learning models will be accurate enough to be useful for users.

Constraints:

Anything that restricts or influences the actions of the project team, such as cost, schedule, or resources, is referred to as a project constraint. Projects must be carried out within the confines of the constraints.

Table 2: Constraints of Tablytix

S.No.	Constraints
1.	Our web application supports pdf, ppt, doc, jpg, and png file formats for uploading.
2.	Resource and cost constraints limit access to certain computing resources for training of the models.

3.	Data constraints limit access to a dataset of financial documents
4.	Accuracy constraints may prevent the machine learning models from achieving the desired level of accuracy in extracting structured data from the financial documents
5.	Regulatory compliance with financial documents.

1.6 Standards

Table 3: Standards of Tablytix

S.No.	Standards	Description
1.	IEEE 730	Software Quality Assurance is a method of observation of the process of software engineering and its principles and strategies used to improve quality. The strategies used to achieve this are several and may include standards such as ISO 9000 and many other models can be included.
2.	IEEE 830	A SRS (software requirements specifications) could be a description of a software that has to be deployed. It tells the non-functional and functional requirements of a project, and it embodies use-cases which tell about the user interactions of software and is very useful.
3.	IEEE 1012	An efficient method of finding if a software is following the specification and if it fulfills the purpose, verification and validation. It can also be called quality control.
4.	IEEE 1058	Project management is a very important aspect of software development. These projects are planned, monitored and controlled for perfect execution.

1.7 Approved Objectives

From the need analysis and project overview section, it is clear that this project will be a one-stop shop for businesses digital document needs. We have listed the objectives of the project in this section.

- Logo Redaction from documents to preserve privacy
- Document Layout Analysis on a number of file formats
- Table Extraction and conversion of table data into JSON format
- Query Answering System based on NLP.

1.8 Methodology

Phase 1

1. Data Collection: Collecting a diverse set of documents (in different formats such as PDF, DOC, etc.) containing company logos. Using publicly available datasets such as Logo-Det3K datasets preferably having a balanced representation of logos of different shapes, sizes, colors, and backgrounds.
2. Data Preprocessing: Cleaning the data and removing any unwanted artifacts from the images. We can also use data augmentation techniques such as rotation, scaling, and flipping to increase the size of the dataset.
3. Logo Annotation: Manually annotating the logos in the dataset using bounding boxes or polygonal masks. This will be used as ground truth for training the model.
4. Model Selection: Exploring different convolutional neural network architectures suitable for semantic segmentation. Some popular choices include U-Net, Mask R-CNN, and other CNN based models. This also includes selecting backbones such as resnet50 or 'resnet50_fpn_1x'.
5. Model Training: Training the selected model on the annotated dataset using a suitable loss function such as Focal Loss Function. Using techniques such as transfer learning and fine-tuning to improve the performance of the model.
6. Model Evaluation: Evaluate the performance of the trained model using metrics such as COCOMetric.

Phase 2

1. Data Collection: Collecting various types of document data in terms of pdfs, jpegs and pngs.
2. Data Preprocessing: Converting data into COCO format along with annotations.

3. Model Selection: Selecting and tuning different models like Detectron2.
4. Model Tuning: Tuning the hyperparameters of the selected model.
5. Model Training:Training the selected model on the annotated dataset using a suitable loss function.
6. Model Testing and Optimization: Testing the model on the test data. Developing suitable pipelines to convert the model's output to a format that is suitable for the phase 2 of the process.

Phase 3

1. Identifying a suitable format to extract information regarding table structure. Different approaches include using HuggingFace Table Transformers for table structure recognition, using statistical approaches such as complex
2. Using appropriate metrics to evaluate the performance of such algorithms.
3. Integrate the algorithms in streamlined pipeline of the system

Phase 4

1. Development of a web based interface for handling queries related to extracted structured data and unstructured data.
2. Using suitable LLM, passing the LLM with appropriate context which includes OCR as well as parsed table information, to handle user queries effectively.
3. Deployment: Deploying the model in a production environment and integrating it with the existing system. Monitoring the model's performance and fine-tuning as necessary.

1.9 Project Outcomes and Deliverables

This project aims to help businesses by providing them with one-stop shop web application features, catering to their digital document needs.

1. Documentation Upload: Users can easily upload documents through website interface, providing support for various file types including pdf, docs, png, jpegs, ppts.

2. Logo Redaction: The system identifies and recognizes logos within the uploaded documents, and further performs automatic redaction of logos.
3. Table Extraction: The system extracts tables from the documents and converts the table data into JSON format, which is searchable and indexable.
4. Query based Information Retrieval: Users can input a query requesting certain data from the table. The system analyzes and provides the relevant information using NLP models.
5. User-Friendly Interface: The website provides an intuitive interface for users interactions, such as uploading documents, inputting of queries, and receiving query based answers.
6. Cloud-based Hosting: Hosting application on cloud ensuring scalability and availability.

1.10 Novelty of Work

The Tablytix project introduces a novel approach by seamlessly bridging the gap between unstructured and complex tables found in businesses digital documents and the implementation of a query-based system for efficient data retrieval. While existing solutions focus on individual components such as document analysis, logo redaction, and even structured data extraction, our innovation lies in providing the integration of these components into a single, powerful platform.

The central novelty comes from our unique methodology to convert unstructured and complex tables into highly structured and query-ready data. Using advanced data extraction algorithms and techniques, our system dynamically recognizes table structures, extracts tabular data, and organizes it into a structured format. This intricate process is aimed at preserving the integrity of intricate tables present in documents, ensuring that users can access the information they need with precision.

Furthermore, the implementation of a query system adds a layer of sophistication to our innovation. Our project integrates Natural Language Processing (NLP) capabilities to decipher user queries written in natural language. This transformative step bridges the gap between user intent and data retrieval, allowing complex queries to be translated into actionable insights. The system's ability to comprehend and

respond to user queries signifies a novel paradigm in document analysis systems.

The culmination of these innovations is a user-centric web application hosted on Google Cloud, where users can upload documents, experience logo redaction, and witness the transformation of unstructured and complex tables into structured, query-ready data. This amalgamation of document analysis, data extraction, NLP, and cloud infrastructure signifies a novel approach that simplifies information access, streamlines decision-making, and enhances the utility of complex documents.

In conclusion, the Tablytix project's novelty stems from its innovative methods of converting unstructured and complex tables into structured data and implementing a sophisticated query system. This integration showcases a pioneering solution that empowers users to seamlessly transform documents into actionable insights, setting new standards in document analysis, data extraction, and user-centric information retrieval systems.

REQUIREMENT ANALYSIS

2.1 Literature Survey

Before proceeding with the development of a project, it is essential to thoroughly survey the current market to analyze the risks and the current competition. The project aims to develop a Web Application to provide the users with redacted documents, from which table data has been efficiently extracted, additionally providing users with answers to their queries concerning table data. Numerous projects and research papers with remarkable contributions to this field were referred to structure our project accordingly, keeping in mind the feasibility and quality standards. With the objectives already defined, the research works with direct or indirect correspondence with our project were selected, and the critical ideas and methodology were used. This shows the analyses and research made in the field of text classification, taking into account the requirements and extent of the project.

2.1.1 Related Work

There do not exist tools or systems that combine all the functionalities i.e. document analysis, logo redaction, table extraction, and query-based information retrieval, into a single integrated platform like our Tablytix project. The integration of these functionalities is a unique aspect of our project.

Veritas Data Privacy: AI-powered text and image redaction solution, tailored for compliance with data protection regulations such as GDPR and HIPAA, ensuring data privacy and security.

Adobe: Adobe has been actively involved in addressing the challenge of logo redaction, particularly in the context of document security and privacy. They have developed innovative solutions that focus on efficiently redacting logos from digital documents while maintaining the integrity of the surrounding content.

Tabula (tabula.technology): Tabula allows extracting data tables from PDF documents through a simple web interface or Python code. It uses optical character recognition and lattice detection to identify tabular structures in PDFs. Users can convert tables to CSV/JSON formats or Excel spreadsheets for analysis.

ABBYY FineReader (www.abbyy.com): ABBYY FineReader is a commercial OCR and document conversion software. It can identify tables in PDFs and export them to Excel spreadsheets while retaining formatting and structure. Its integrated OCR engine provides accurate table extraction from image-based and scanned PDFs. FineReader also offers document comparison, conversion to searchable PDFs, and integration with other systems.

ABBYY FlexiCapture: ABBYY FlexiCapture is a commercial platform that combines deep learning techniques for data extraction, and document analysis capabilities. It can handle structured and semi-structured data extraction, making it suitable for forms and tabular data.

DocParser: DocParser is a document processing platform that can extract data from documents, including tables. While it doesn't explicitly mention logo redaction, it focuses on automating data extraction from various document formats.

Rosoka Text Analytics: As mentioned earlier, Rosoka offers text analytics solutions, including table extraction, which can be part of a broader system that processes

unstructured text data.

Custom Solutions: Some organizations develop custom solutions that integrate deep learning architectures for data extraction, and analysis tools to meet specific document processing needs. These solutions are typically tailored to their requirements.

2.1.2 Research Gaps of Existing Literature

Insufficient Exploration of Document Analysis Challenges: While existing literature extensively covers various document analysis datasets and methodologies, there is a notable gap in addressing the specific challenges posed by scanned document images with low quality, noise, and low resolution. The research lacks a comprehensive exploration of solutions tailored to handle these characteristics effectively.

Limited Focus on Medieval Manuscript Analysis: The literature review reveals a dearth of research dedicated to the detailed analysis of medieval manuscripts. Current datasets like DIVA-HisDB provide annotated pages for layout analysis, text line segmentation, binarization, and writer identification, highlighting the need for more in-depth investigations into methodologies specifically designed for the complexities of historical documents.

Neglect of Logo Detection Challenges: Despite the availability of diverse logo detection datasets such as LogoDet-3K and WebLogo-2M, there is a research gap in addressing the specific challenges related to accurate logo redaction, efficient object detection, and handling a wide variety of logo categories. Further exploration is needed to enhance the robustness and efficiency of logo detection models.

Limited Integration of Deep Learning in Document Layout Analysis: While datasets like DocBank and PubLayNet indicate the integration of weak supervision approaches and multimodal networks, there is a gap in exploring the full potential of deep learning, particularly in tasks like document layout analysis. The literature lacks comprehensive studies that leverage the capabilities of deep neural networks for fine-grained token-level annotations.

Unexplored Synergy between Object Detection and NLP: The literature review highlights the significance of models like BERT in NLP tasks and object detection frameworks like YOLO and Faster R-CNN in computer vision. However, there is a

research gap in exploring the synergy between these domains, specifically in tasks related to document structure recognition and information extraction, where both visual and textual information play crucial roles.

Table 4: Research Findings

S. No.	Roll Number	Name	Paper Title	Tools/ Technology	Findings	Citations
1	102016040	Ojas Sharma	RVL-CDIP	DICOM Libraries; TensorFlow	The RVL-CDIP dataset includes scanned document images with 16 classes like letter, form, email, resume, memo, etc. It comprises 320,000 training, 40,000 validation, and 40,000 test images. The images exhibit low quality, noise, and low resolution, around 100 dpi.	A. W. et. al. [1]
2	102016040	Ojas Sharma	DIVA-HisDB	DICOM Libraries; TensorFlow	DIVA-HisDB is a well-annotated dataset for Document Image Analysis (DIA) tasks like layout analysis, text line segmentation, binarization, and writer identification. It features 150 annotated pages from three challenging medieval manuscripts with diverse layouts.	Simistira, et. al.[2]

3	102016040	Ojas Sharma	Viola-Jones Detector	Haar-like Features; Integral Images; AdaBoost; Cascade Classifier; Machine Learning (SVM);	"Viola-Jones algorithm excels in face detection using Haar-Like Features, capturing edges and textures. Internal Image employs a summed area table for rapid pixel intensity computation. AdaBoost selects effective Haar-like features, and Cascade Classifier efficiently rejects non-object regions through Training and Cascading."	Jones, et. al.[11]
4	102016040	Ojas Sharma	HOG Detector	Gradient Computation,histogram calculation algorithms	HOG detector, a non-deep learning method, is used for object detection, especially in pedestrian images. It employs traditional computer vision techniques, focusing on object structure by counting gradient orientation occurrences and generating histograms for image regions using gradient magnitude and orientations.	Dalal, et. al. [12]
5	102016040	Ojas Sharma	Deformable Part-based Model (DPM)	LIBSVM; Pascal VOC; INRIA Person	The Deformable Part-based Model (DPM) is a machine learning framework for object detection in images, predating recent deep learning methods. It relies on traditional computer vision techniques, combining deformable parts with a root filter to model object structures and variations. DPM employs HOG features and SVMs for part-based object detection.	Felzenszwalb, et. al [13]

6	102016026	Harneet Kaur	DSSE-200	Convolutional Neural Networks	The DSSE-200 is a complex document layout dataset including various dataset styles. The dataset contains 200 images from pictures, PPT, brochure documents, old newspapers and scanned documents. Introduced while learning to Extract Semantic Structure From Documents Using Multimodal Fully Convolutional Neural Networks	Yang, et al. [13]
7	102016026	Harneet Kaur	PubLayNet	DICOM Libraries; TensorFlow	PubLayNet, a dataset for document layout analysis, includes over 360,000 annotated images of titles, paragraphs, figures, and captions. It matches XML representations and content of 1 million PDF articles from PubMed Central, commonly used for tasks like academic paper layout detection.	Zhong, et. al.[4]

8	102016026	Harneet Kaur	Regions with CNN features (R-CNN)	CNN; Classifiers;	R-CNN, designed for object detection and localization, comprises key steps: Region Proposal Generation, using methods like selective search; Feature Extraction, employing a Convolutional Neural Network (CNN) for hierarchical feature learning; Classification, using extracted features to classify object presence and category; Bounding Box Regression, predicting object bounding box coordinates; and Non-Maximum Suppression, eliminating duplicate detections by retaining the most confident one.	Ross Girshick, et. al.[22]
9	102016026	Harneet Kaur	Fast R-CNN	RPN; CNN; RoI	"Fast R-CNN, an improved object detection algorithm, addresses speed and efficiency issues of R-CNN. Key enhancements include integrating Region Proposal Network (RPN) for faster proposal learning, sharing convolutional features for efficiency, introducing RoI pooling for variable-sized regions, and adopting single-stage training for a streamlined and efficient process."	Girshick, et. al.[14]
10	102016068	Agamjot Singh	DocBank	Weak supervision	DocBank, a large-scale dataset with 500K document pages,	Li,et. al.[5]

				ml model	employs weak supervision for integrating textual and layout information. It offers 400K training, 50K validation, and 50K testing pages, providing fine-grained token-level annotations for layout analysis in a unique and effective way compared to traditional human-labeled datasets.	
11	102016068	Agamjot Singh	FlickrLogo-32	COCO annotator; labelbox	A logo dataset for computer vision tasks includes annotated images with details like filename, class name, training subset, and coordinates (x1 y1 x2 y2) indicating logo location.	Wang,et. al.[6]
12	102016068	Agamjot Singh	Faster R-CNN	RPN; CNN; RoI; Regression model	Faster R-CNN is a computer vision object detection framework with a Region Proposal Network (RPN) for generating potential object bounding boxes, a shared convolutional backbone for feature extraction, an RPN head for refining proposals, RoI pooling for fixed-size feature extraction, and separate heads for classification and regression.	Ren, et. al.[15]
13	102016068	Agamjot Singh	Feature Pyramid Networks	CNN; Top-Down pathway	FPN, or Feature Pyramid Network, improves object detection by merging features from different levels of a CNN into a unified pyramid. The process includes using a backbone network (e.g.,	Lin, et. al.[16]

					ResNet) for initial feature extraction, constructing a feature pyramid, and employing both top-down and bottom-up pathways to capture details and context. The resulting multi-scale feature pyramid enhances tasks like detection and segmentation.	
14	102016012	Mayank Rawat	Logo Det 3k	RectLabel; Bounding boxes	LogoDet-3K, the largest logo detection dataset, comprises 3,000 categories, 200,000 annotated logo objects, and 158,652 images. It offers a more challenging benchmark for logo detection with comprehensive coverage and a wider variety of logo categories and annotated objects.	Wang, et. al.[7]
15	102016012	Mayank Rawat	WebLogo-2M	Bounding box	WebLogo-2M is a large-scale, weakly labeled logo detection dataset, sampled from Twitter stream data, featuring 194 logo classes and over 2 million images. Designed for training models with noisy data and computational challenges, it includes 6,569 test images with manually labeled logo bounding boxes for performance evaluation.	Su,et. al.[8]
16	102016012	Mayank Rawat	Mask R-CNN	RPN; RoI; CNN layers	Mask R-CNN extends concepts from R-CNN and Faster R-CNN by introducing a mask generation branch. It utilizes an RPN for region proposals,	He, et. al[17]

					followed by object detection to refine and classify these proposals. Additionally, it predicts binary masks for each detected object using a fully convolutional network.	
17	102016012	Mayank Rawat	Single Shot Detector (SSD)	Single Shot; Loss Function	SSD efficiently detects objects in images in a single neural network pass, combining accuracy from two-stage detectors like Faster R-CNN with real-time processing benefits of one-stage detectors like YOLO. Key features include single-shot processing, multi-box predictions, multi scale feature maps, conv. pred., and non-maximum suppression for accurate detections.	Fu, et. al[18]
18	102016080	Anshul Kanwar	TopLogo-10	Bounding box	The dataset TopLogo-10 contains photos depicting 10 clothing related popular brand logos for the evaluation of multi-class logo detection in real-world images. With 70 images per one logo classes, making a dataset of 700 images. The datasets are annotated manually with strict bounding boxes.	Su, et.al. [9]
19	102016080	Anshul Kanwar	QMUL-OpenL [10]	Combination of datasets	QMUL-OpenLogo contains 27,083 images from 352 logo classes, built by aggregating and refining 7 existing datasets and establishing an open logo detection evaluation protocol.	Su, et.al. [10]

20	102016080	Anshul Kanwar	YOLO V5	YOLO family	YOLO (You Only Look Once) is a family of real-time object detection models used in computer vision tasks. YOLO models are known for their efficiency and speed in detecting objects in images and videos. YOLOv5 is a cutting-edge, state-of-the-art (SOTA) model that builds upon the success of previous YOLO versions and introduces new features and improvements to further boost performance and flexibility.	Redmon, et. al.[19]
21	102016080	Anshul Kanwar	DeepDeSRT	Rule based Heuristics	DeepDeSRT contributes by offering a deep learning solution for table detection in document images and presenting a novel approach for recognizing table structures, including rows, columns, and cell positions. Unlike rule-based methods, this system is data-driven and does not rely on heuristics or additional PDF metadata for both table detection and structure recognition in document images.	Schreiber, et. al. [20]
22	102016080	Anshul Kanwar	Bert	NLP; fine-tuning	BERT, or "Bidirectional Encoder Representations from Transformers," is a	Bao, et. al.[21]

					transformer-based natural language processing (NLP) model designed for sequential data processing, especially text. Key points about BERT include bidirectional contextualization, pretraining on a massive unsupervised text corpus, a transformer architecture with self-attention mechanisms, embedding layers capturing semantic and syntactic information, a Masked Language Model (MLM) for predicting masked words, and transfer learning for achieving state-of-the-art results on various NLP tasks with minimal task-specific training.	

2.1.3 Detailed Problem Analysis

In present times, there has been an unprecedented surge in the abundance of information within digital documents, posing a significant challenge for businesses, particularly in managing vast volumes of crucial data. However, a notable concern arises regarding privacy issues associated with logos when sharing these documents. Preserving brand image is paramount for businesses, necessitating the removal of logos before sharing documents. Manual logo removal is time-consuming and resource-intensive. To address this challenge efficiently, we propose leveraging state-of-the-art machine learning and deep learning models to automate the task of logo redaction.

Logo redaction using machine learning models empowers companies to control the visibility of their logos, preventing unauthorized exposure. This approach ensures that shared information remains confidential, addressing privacy concerns and compliance requirements. Beyond enhancing document privacy, it streamlines workflows for businesses engaged in regular information exchange while preserving brand identity

integrity. Commonly used datasets for logo redaction include BelgaLogs, FoodLogoDet-1500, QMUL-OpenLogo, and Logo At-3k. Widely employed models encompass RCNN, Fast-RCNN, Faster-RCNN, and various versions of the YOLO model.

Addressing another concern related to the accessibility of vast information within businesses, particularly in tabular forms, we are developing a web application to streamline the extraction of information from tables within digital documents. The application focuses on Document Layout Analysis to highlight different document components. Subsequently, Table Extraction converts table information into JSON format, enabling easy search ability and indexability. This web application simplifies information extraction from tables, responding to user queries for quick and accurate access to valuable insights within tabular structures. Datasets relevant to Document Analysis include RVL-CDIP, DIVA-HisDB, and DSSE-20, with commonly used models such as RCNN, OCR, and various YOLO model versions.

This innovative solution seeks to revolutionize how businesses interact with and extract meaningful information from their extensive digital document repositories. The rapidly advancing efficiency and accuracy of machine learning and deep learning models used in this project position it competitively for continued improvement and success in the future.

2.1.4 Survey of Tools and Technologies Used

The Software Tools used by the system are:

1. Python
2. Applied deep-learning techniques for document analysis
3. Word Embeddings
4. Keras
5. PyTorch
6. Scikit-Learn
7. NodeJS
8. TailwindCSS
9. Supabase

10. Deep Learning
11. Next.js
12. FastAPI
13. Google cloud
14. Dockers
15. Icevision framework
16. Google colab

2.1.5 Summary

The final year project, Tablytix, aims to develop a comprehensive web application that addresses the challenges associated with document analysis and table extraction. The project involves a thorough literature survey to identify existing tools and research contributions in the field. The key objectives include document analysis, logo redaction, table extraction, and query-based information retrieval, all integrated into a single platform.

The literature survey reveals a gap in existing tools, making Tablytix unique in its combination of functionalities. Notable tools and solutions in the field include Veritas Data Privacy, Adobe, Tabula, ABBYY FineReader, ABBYY FlexiCapture, DocParser, Rosoka Text Analytics, and custom solutions developed by organizations.

Research gaps in the existing literature are identified, such as insufficient exploration of challenges in document analysis, limited focus on medieval manuscript analysis, neglect of logo detection challenges, and the limited integration of deep learning in document layout analysis. These gaps provide a foundation for the innovation and uniqueness of the Tablytix project.

The detailed problem analysis emphasizes the need for efficient logo redaction and streamlined information extraction from tables within digital documents. The proposed solution involves leveraging machine learning and deep learning models for automated logo redaction and developing a web application for document layout analysis and table extraction.

The tools and technologies used in the project include Python, deep-learning techniques, word embeddings, Keras, PyTorch, Scikit-Learn, NodeJS, TailwindCSS,

Supabase, FastAPI, Google Cloud, Docker, Icevision framework, Google Colab, and more. These tools contribute to the efficiency and accuracy of the Tablytix project.
In conclusion, Tablytix represents an innovative solution to the challenges posed by the abundance of information in digital documents. The project integrates advanced technologies to provide a comprehensive platform for document analysis and table extraction. The use of machine learning and deep learning models, coupled with a diverse set of tools and technologies, positions Tablytix competitively for continued improvement and success in the future.

2.2 Software Requirement Specification

A software requirements specification could be a description of a piece of software that has to be created. A stakeholder requirements specification is another name for it. The SRS is modeled using the business requirement specification.

2.2.1 Introduction

The SRS Document seeks to provide a comprehensive summary of the final deliverable product, its parameters, and objectives. The purpose of this paper is to collect, analyze, and provide an in-depth understanding of the full Tablytix software by thoroughly defining the matter statement. Throughout this document, the Tablytix's needs are laid out in depth.

2.2.1.1 Purpose

This article explains each and every feature of the Tablytix software, as well as the concept behind the project's creation and its practical uses. It gives a comprehensive summary of our product, including its parameters and objectives, as well as the project's target audience, user interface, and software requirements.

2.2.1.2 Intended Audience and Reading Suggestions

This document is intended for:

- The system's designers and implementers; it establishes rules for future

development. Because some defects are easier to find when using a requirements document, project testers might use this document as a starting point for their testing strategy. Testing becomes more carefully planned in this manner.

- Anyone interested in learning more about the project and its activities.
- The goal of the document is to help both developers and stakeholders. It will assist developers in fully comprehending and documenting the needs of stakeholders. The project's functionality and, as a result, the requirements are described in detail in the following sections. It will be read in order to gain a thorough understanding of each and every module of our project.

2.2.1.3 Project Scope

The scope of the project is to develop a reliable, cost effective and efficient technique for real-time table extraction from documents with less delay and maximum accuracy. Tablytix provides a solution with all features combined into one complete platform for smooth functioning. Different combinations of datasets and models will be trained and tested in order to discover the combination with most accuracy. Not only these but it will offer many more features like a logo redaction, document layout analysis, query answering system etc.

2.2.2 Overall Description

2.2.2.1 Product Perspective

Tablytix is an app that is intended for businesses and commercial use. It would be a standalone platform which would be one of its kind since various features of different categories will be accommodated by it. All the user's digital documents needs would be satisfied by Tablytix as it uses various techniques (Logo Redaction, Document Layout Analysis, Table Extraction, Query answering system) to cover every segment.

The app would ask input from the user and would use the necessary technique to answer the query from the user using modern nlp techniques. Each interface would

have different functions and users would be able to easily use & understand the working of this app.

It will also provide the option for using the individual functionalities of the app or to streamline the document through all the functionalities at once. All these various features would be provided in one single app which would become a one-stop shop for businesses for their digital document needs.

2.2.2.2 Product Features

The main features of our application is:

- A fully developed software which offers the features of maximum accuracy and real-time table extraction from the documents.
- The software also has features supporting logo redaction from the documents.
- The software also include features for query answering system, using which the user can get the table information as an answer
- The software supports a variety of the file formats including pdfs, ppts, docs, jpegs, pngs.

2.2.3 External Interface Requirements

The various interface requirements are as follows:

2.2.3.1 User Interfaces

The UI provides a user-friendly environment. The user submits a document which is then returned after logo redaction, document layout analysis and table extracion. The user can additionally provide a query to which the user interface would return relevant data from the tables contained in the documents uploaded. In order to improve the experience and better the privacy to users, users must go through user authentication i.e. login/signup, which is also very intuitive.

2.2.3.2 Hardware Interfaces

The web app will work in the browser. Hence there are no external interface requirements apart from a device (mobile, tablet, laptop, or PC) and an internet connection to access the web application.

2.2.3.3 Software Interfaces

Some of the important software interfaces used in the project:

- Node.js
- Google cloud platform
- Tailwind CSS
- FastAPI
- Docker
- Python

2.2.4 Other Non-functional Requirements

A non-functional requirement is a description of the system's operation capabilities and limitations that improve its functionality. These are maintainability, portability, fast execution speed, security, reliability, robustness, accuracy etc.

2.2.4.1 Performance Requirements

These are the main performance requirements for our proposed system:

Simplicity: The User Interface has been kept relatively simple to use and intuitive to use for all kinds of users. It should not be uselessly sophisticated, complicated, and usable for a layman. The user should only give the input and receive the result.

Availability: The interface will be in use and accessible to the user until the internet works and that particular web page is open.

Maintainability: The development team will follow best practices for clean code and software modularity in order to make the application as maintainable as possible.

Portability: Users will be able to access this application 24/7 any time on their respective devices.

Fast execution speed: Users will be able to change between interfaces with minimum or no delay.

Reliability: The website would be continuously updated making sure its failure free.

Robustness: The website will be able to handle unexpected termination and unexpected actions.

Accuracy: The system will provide best possible accuracy and efficient technique for real-time audio to text conversion alongside summarization and Question/Answer

formation.

Security: Sensitive user information is encrypted to ensure user privacy.

Change Password- The user can change their account password.

2.2.4.2 Safety Requirements

These are the safety requirements for the proposed system:

- The app data should be highly protected
- Necessary protocols and measures should be taken, so that the personal information of the users would not get leaked.
- The system should implement an automated backup and recovery mechanism to safeguard against data loss.

2.2.4.3 Security Requirements

The following are the proposed security requirements:

- The website's communication with the google cloud server must be safe and unchangeable.
- Strong user authentication to be implemented with role of the user clearly defined
- All the security features should be compliant with the data protection and privacy regulations of the region.

2.3 Cost Analysis

This project is entirely software based and does not involve any hardware components. The libraries and the datasets used are either free or open source, thus adding nothing to the project's total cost.

Table 5: Cost Analysis

Category	Description	Price (INR)	Duration	Total Cost (INR.)
GCP	Gcp is used to host Models,as well as website	280 per day	30days	8500
Domain cost	Domain is used	500	1 year	500

	by website			
GPT	Used by QA bot	1000	60 days	1000
Total				10000

2.4 Risk Analysis

Table 6: Risk Analysis

Risk Category	Risk Description	Impact	Likelihood	Mitigation strategy
Technical Risks	Complex Integration challenges	high	moderate	Thorough testing
	Algorithm Performance below expectations	medium	high	Rigorous testing
Resource Risks	Limited expertise among team members	high	moderate	Skill enhancement
	Time constraints due to academic commitments	medium	high	Effective scheduling using Gantt chart
Communication Risks	Misalignment with mentor's guidance	medium	moderate	Regular communication
	Coordination challenges among team member	low	moderate	Team collaboration
Security and Privacy Risks	Data breach due to inadequate security measures	high	low	Robust security
User Acceptance Risks	Application not meeting user expectations	high	moderate	User feedback

	Usability issues impacting user engagement	medium	moderate	Usability testing
External Dependencies Risks	Downtime/disruptions in services of third-party	high	low	Contingency planning
	Cloud platform technical issues	medium	low	Redundancy measures
Financial Risks	Budget constraints leading to resource strain	high	low	Resource planning before any real development starts
Regulatory Compliance Risks	Non-compliance with data protection regulations required	high	moderate	Adhere to regulations by following standards set by IEEE

METHODOLOGY ADOPTED

3.1 Investigative Techniques

Investigative Technique Involved: EXPERIMENTAL.

Experimental : An organized investigation includes a control group and is designed to test the hypothesis, which includes independent and dependent variables.

Our independent variables are the documents, images, tabular contents, table structures and user queries. The dependent variables in this project are logo redaction, document layout analysis, table data extracted and answers to user's queries.

Testing the Feasibility of the project:

1. Technical Feasibility:

Existing Technologies: The project leverages established technologies such as CNN, Next.js, Tailwind CSS, FastAPI, and Docker.

Algorithm Availability: Suitable deep learning algorithms are available, and the required libraries are accessible.

Integration Complexity: Integrating logo redaction and table extraction requires moderate complexity, but resources and tutorials are available.

2. Financial Feasibility:

Budget Constraints: The project budget is allocated for development, tools, cloud hosting, and operational expenses.

Resource Availability: As undergraduate students, the team has access to university resources, and the mentor provides guidance without additional financial burden.

Revenue Generation: Potential revenue streams include user subscriptions for advanced features or offering commercial licenses to enterprises.

3. Operational Feasibility:

Team Expertise: The team possesses programming skills and access to relevant tools and technologies. Limited expertise may require learning curves.

Time Constraints: Balancing academic commitments with project development may lead to extended development timelines.

Resource Allocation: The team can allocate sufficient time for development, testing, and iteration based on academic schedules.

4. Market and User Feasibility:

Market Demand: Document analysis tools are in demand for business, research, and education sectors. User surveys indicate potential interest.

User Acceptance: Users are interested in functionalities such as table extraction, sentiment analysis, and user-friendly visualizations.

Competition: Similar platforms exist, but "Tablytix" aims to differentiate itself through advanced features and user-centric design.

5. Legal and Regulatory Feasibility: (to-check)

Data Protection: The project will adhere to data protection regulations (e.g., GDPR) by implementing encryption, pseudonymization, and user consent mechanisms.

Intellectual Property: Intellectual property concerns are addressed through proper attribution and compliance with licensing terms of third-party libraries.

6. Environmental Feasibility:

Carbon Footprint: As a web-based application, Tablytix minimizes environmental impact compared to physical alternatives.

Resource Efficiency: Cloud hosting and containerization (Docker) optimize resource utilization and reduce waste.

3.2 Proposed Solution

Objective:

The proposed solution outlines the key features, components, and functionalities of the "Tablytix" project, aimed at providing users with a comprehensive document analysis platform.

1. User Authentication

The user will have to sign up for the system. Only then he will be given a credential which he can later use for logging into the system.

2. Document Upload and Preprocessing:

The user will have the option to upload his files for processing. The files could be in formats such as ppt, pdf, jpegs or pngs. The system is using different libraries to convert these all into suitable formats. Then these suitable formats are passed one by one to each model for further processing.

3. Logo Redaction

The parsed images in the form of numpy arrays are passed into the logo redaction model. The model is currently using resnet_50_fpn_1x as the backbone. It is a state of the art feature pyramid network based variation of the classic resnet_50 architecture. We are using RetinaNet[23], Faster RCNN architecture for training this particular model. The dataset used is LogoDet-3K[7]. It contains images of different logos and the model is accurately able to detect logos present in the given image. We then use the coordinates outputted by the model to redact that part from the parent image. The main aim of this sub-system was to redact company’s logos to maintain privacy of the financial documents, as the further sub-systems would only use the processed documents without logos. It also ensures that when the documents are stored or

further processed, it doesn't disclose the company's logos and name information hence preserves privacy.

4. Document Layout Analysis

Our next subtask includes performing document layout analysis. It is important to know what different kinds of data is present in the document. For this task we trained a Faster RCNN based architecture for layout analysis containing classes Text, Title, List, Table, Figure. The model uses resnet_50_fpn_3x architecture and is trained on PublayNet Dataset. The model is highly accurate with an mAP value 0.80. If the system detects that tables are present then the document is forwarded to the table data extraction system in the form of images.

5. Table Data Extraction

The images that contain tables are then sent to the table data extraction model. We are using Anssi Nurminen's master's thesis approach along with inspiration from Tabula.

The approach can be summarized as:

For any given PDF Or Image, find the lines that are

(a) explicitly defined and/or

(b) implied by the alignment of words on the page.

1. Merge overlapping, or nearly-overlapping, lines.

2. Find the intersections of all those lines.

3. Find the most granular set of rectangles (i.e., cells) that use these intersections as their vertices.

4. Group contiguous cells into tables.

6. Question Answering System

We are using large language models such as GPT4 for processing the user query. GPT4 is given with the user's query, context information such as OCR scan of the document and parsed information of the table. After this response is displayed on the screen.

7. User Authentication and Security:

Implement user authentication mechanisms to secure uploaded documents and analysis results.

Ensure user data privacy and comply with relevant data protection regulations (e.g., GDPR).

8. Cross-Platform Availability:

Design the application to be accessible across different operating systems, and devices, including desktops, tablets, and mobile phones (with desktop mode).

Optimize the user interface for responsive and consistent experiences.

9. Cloud Hosting and Scalability:

Host the "Tablytix" application on cloud platforms to ensure scalability and resource efficiency.

Implement containerization (Docker) to facilitate deployment and management.

10. Continuous Improvement and Iteration:

Establish a feedback loop with users to gather insights and suggestions for enhancement.

Regularly update the application with new features, improvements, and bug fixes.

11. Educational and Commercial Licensing:

Offer educational licenses for students and researchers to access the application's basic features. Provide commercial licenses for enterprises seeking advanced functionalities and data analysis capabilities.

Conclusion:

The proposed solution for the "Tablytix" project encompasses a comprehensive document analysis platform that incorporates deep learning techniques for table extraction, and query-based access. By focusing on user-friendliness, and security, the solution aims to address users' needs while providing valuable insights from unstructured documents.

3.3 Work Breakdown Structure

This project spans 12 months, from February 2023 to January 2024. There are 10 major activities planned:

1. Problem Identification (February - March 2023)

2. Study and Analysis of existing techniques (March - April 2023)
3. Logo Redaction (April - May 2023)
4. Document Layout Analysis (May - June 2023)
5. Table Information Extraction (June - July 2023)
6. Question Answering System (July - August 2023)
7. Implementation and Evaluation of model (August - September 2023)
8. Design interface to showcase (September - October 2023)
9. Result Evaluation (October - November 2023)
10. Final report (November 2023 - December 2024)

The chart shows the planned timeline for each activity with expected start and end weeks. It will allow tracking progress by comparing the actual timeline to the planned timeline.

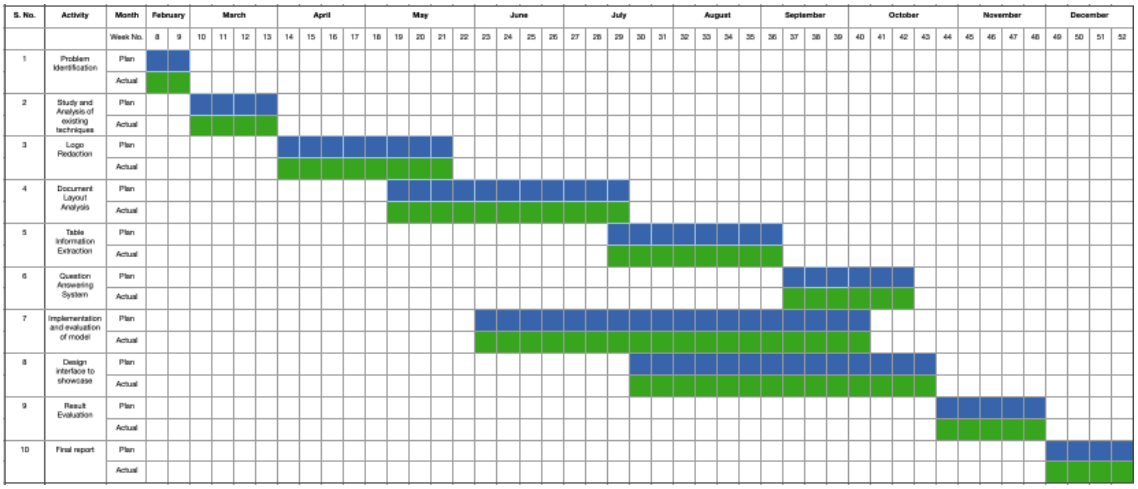


Figure 1: Gantt Chart

3.4 Tools and Technology

Frontend Development:

Next.js: A React framework for building server-rendered React applications with better performance and SEO.

Tailwind CSS: A utility-first CSS framework for rapid UI development with a focus on simplicity and flexibility.zxz

Backend Development:

FastAPI: A modern, fast web framework for building APIs with Python. It offers automatic documentation generation and high performance.

Supabase: Supabase is an open-source Backend-as-a-Service that helps developers build scalable and web and mobile applications with speed and simplicity. It provides developers with a range of backend features and services, such as database management (based on PostgreSQL), real-time updates, authentication, and storage.

Docker: Containerization platform to package applications and dependencies for consistent deployment across different environments.

DevContainers: Use Visual Studio Code Dev ContainersDev Containers for consistent development environments across team members.

Database and Storage:

Database Management System: Choose a database system like PostgreSQL or MySQL to store user data, analysis results, and metadata.

Cloud Storage: Use cloud storage services like Google Cloud Storage or Amazon S3 to store uploaded documents securely.

Document Analysis:

Tesseract OCR: Open-source OCR engine to extract text from images within uploaded documents.

IceVision: Utilize the IceVision library for image analysis tasks such as logo detection and extraction.

DESIGN SPECIFICATIONS

4.1 System Architecture

4.1.1 Block Diagram

Block diagram provides a high-level view of the overall architecture and flow of our document processing system.

It shows the Orchestration Server which acts as the central controller. The Orchestration Server receives the input document and then coordinates routing it through each of the core document processing services in sequence:

- Logo Redaction Service - Detects and redacts any logos in the document.
- Layout Analysis Service - Identifies the structural layout of the document.
- Table Extraction Service - Finds and extracts tabular data.

Each service produces an output that is passed to the next service by the Orchestration Server.

The final output is the fully processed document with redacted logos, analyzed layout, and extracted tables.

This diagram makes clear the modular architecture with the Orchestration Server coordinating the workflow across the distinct document processing services. It conveys the high-level flow and sequencing to transform the input document to the final output.

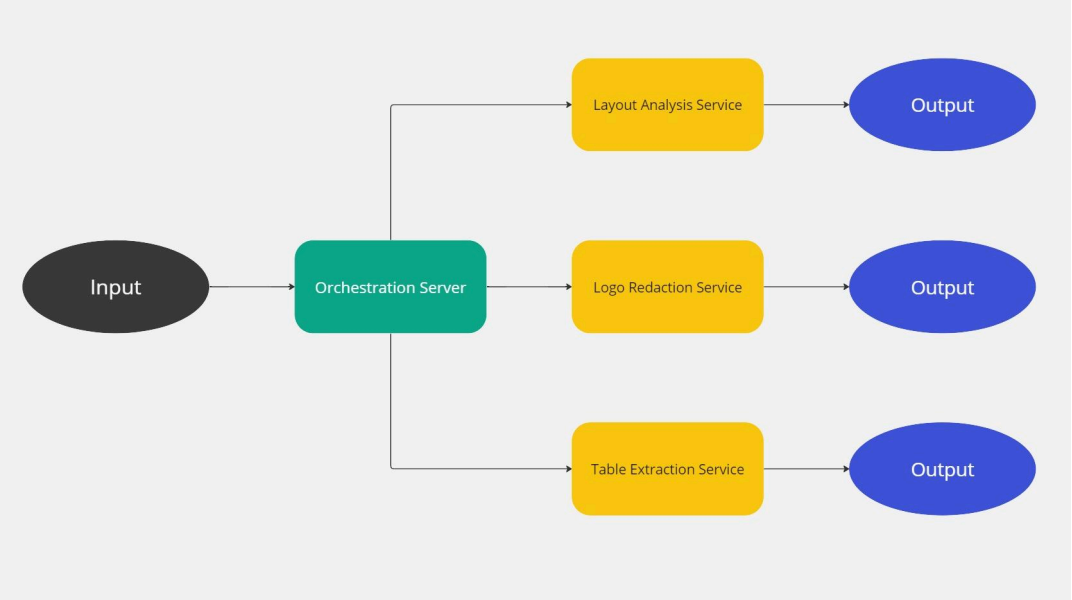


Figure 2: Block Diagram

4.1.2 Component Diagram

This diagram models the key components and interactions in our document processing system using a microservices architecture.

It shows the core subsystem components:

- API Gateway - Single entry point for clients. Routes requests to appropriate microservices.
- Logo Redaction Microservice - Encapsulates logo redaction logic and ML model.
- Layout Analysis Microservice - Encapsulates document layout analysis logic and ML model.
- Table Extraction Microservice - Encapsulates table extraction logic and ML model.
- Storage Volume - Persistent storage for input and output files.

Each microservice contains its own machine learning model wrapped in a common model interface. The microservices read input files from the storage volume, run them through the models, save the outputs, and interact via the API gateway.

This diagram conveys the encapsulated nature of the key document processing functions as independent microservices. It also demonstrates how they interconnect and share data via the storage volume.

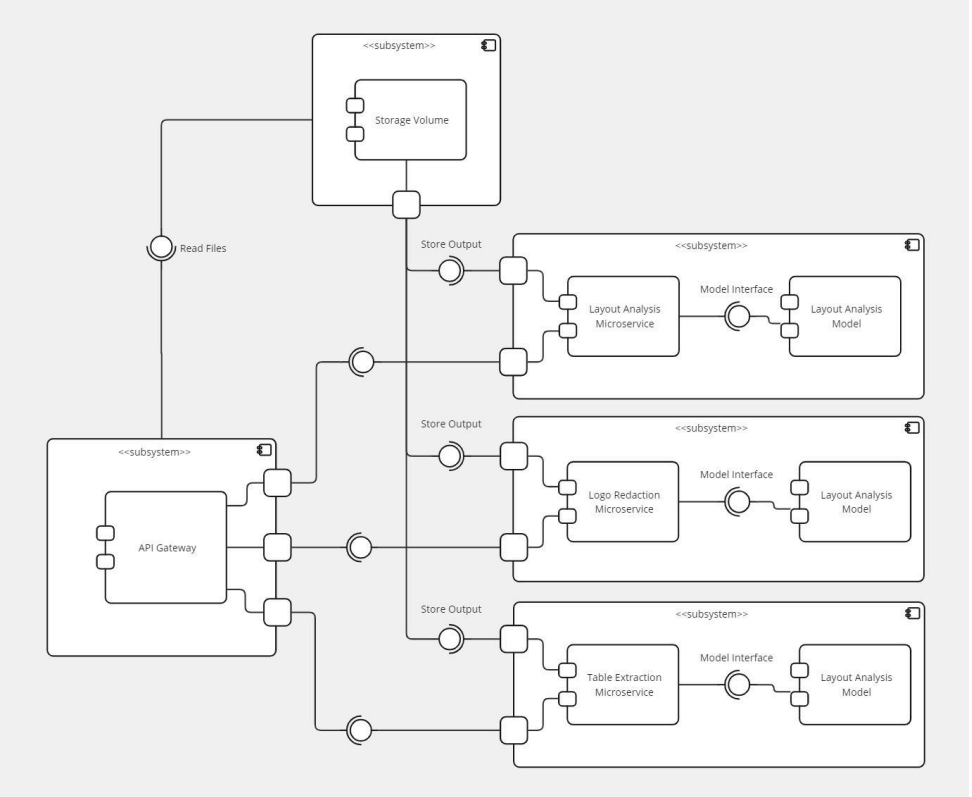


Figure 3: Component Diagram

4.1.3 State Chart Diagram

This state diagram provides an overview of the key states users go through when interacting with our document processing and question answering system.

It begins with an initial state where new users can sign up by entering their information, while existing users can log in with their credentials. If a user forgets their password, they can reset it via a link sent to their registered email.

Upon entering valid login credentials, users reach the Dashboard state. This is the central hub from which they can access the main system functionalities. Key actions here include:

- Uploading a document which kicks off the core document processing workflow. This moves through three main states:
    - Logo Redaction: Any logos detected in the document are blacked out.
    - Document Layout Analysis: The structure and layout of the document is extracted.
    - Table Data Extraction: Tabular data in the document is identified and extracted.

- If a document is in an invalid format, the system returns to the Dashboard state to retry.
- The user can ask questions by entering queries based on the extracted document data. This connects to the Question Answering system state where the system searches the extracted info to return an answer.
- Users can log out to exit the system and return to the initial state.

Walking through the diagram provides visibility into the critical system modes, the logical flow between states, and key actions available to users.

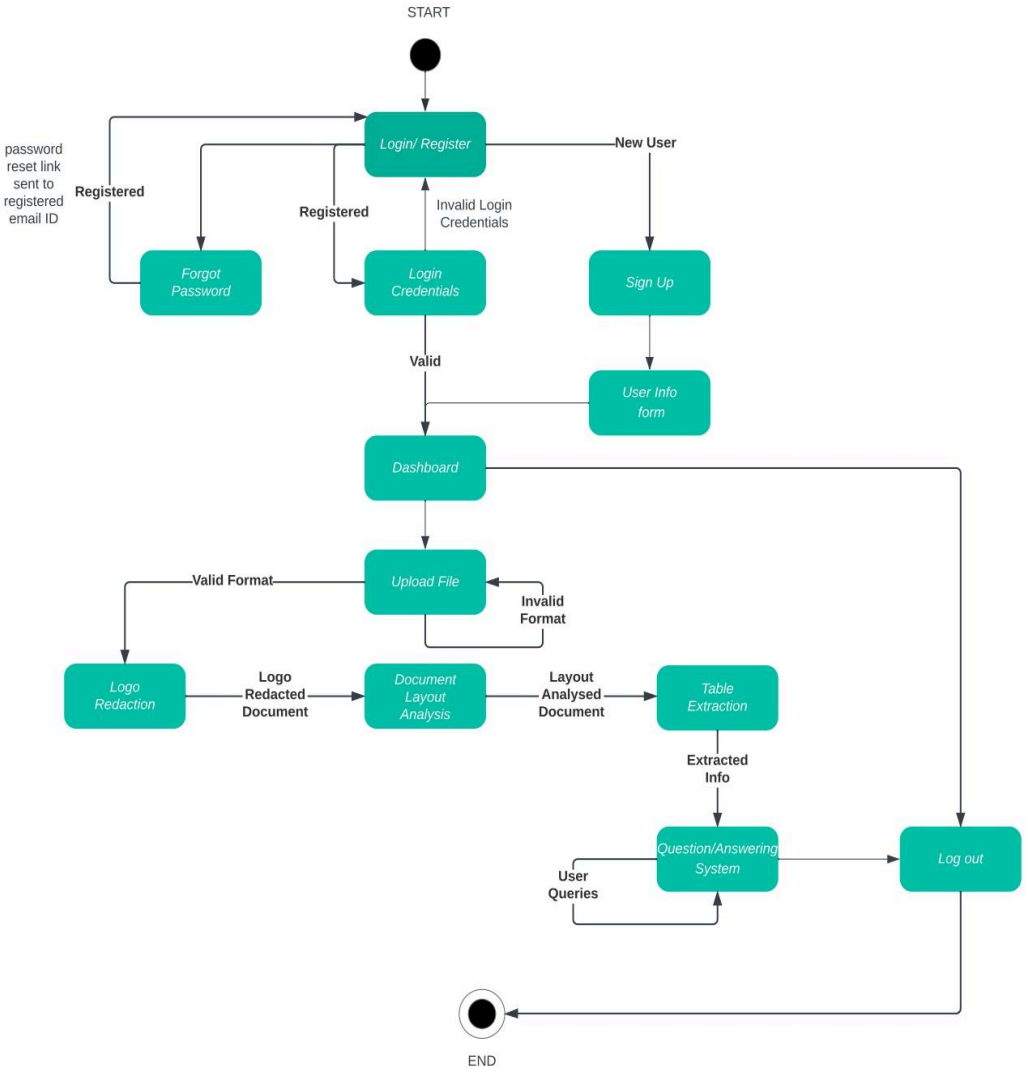


Figure 4: State Chart Diagram

4.2 Design Level Diagrams

4.2.1 Data-Flow Diagrams

The level 0 diagram provides a high-level overview of the overall system architecture and key components. It shows the user interacting with the system and the main functions of user authentication, document processing with machine learning models, and the question answering system.

The level 1 diagram dives into more detail on the core document processing workflow. It illustrates how the user's uploaded documents connect to the different ML models for logo redaction, layout analysis, and table data extraction. The output of each model feeds into the next stage of processing. It also shows how the user's questions connect to the QA system which provides responses by leveraging the extracted data.

Finally, the level 2 diagram decomposes the components further to reveal the sub-processes and data stores involved. This includes the specific databases, model training data, and interim outputs that allow the system to take in documents, process them, and enable the QA functionality. Walking through the levels gives a progressively deeper view into the overall data and process flow of the system.

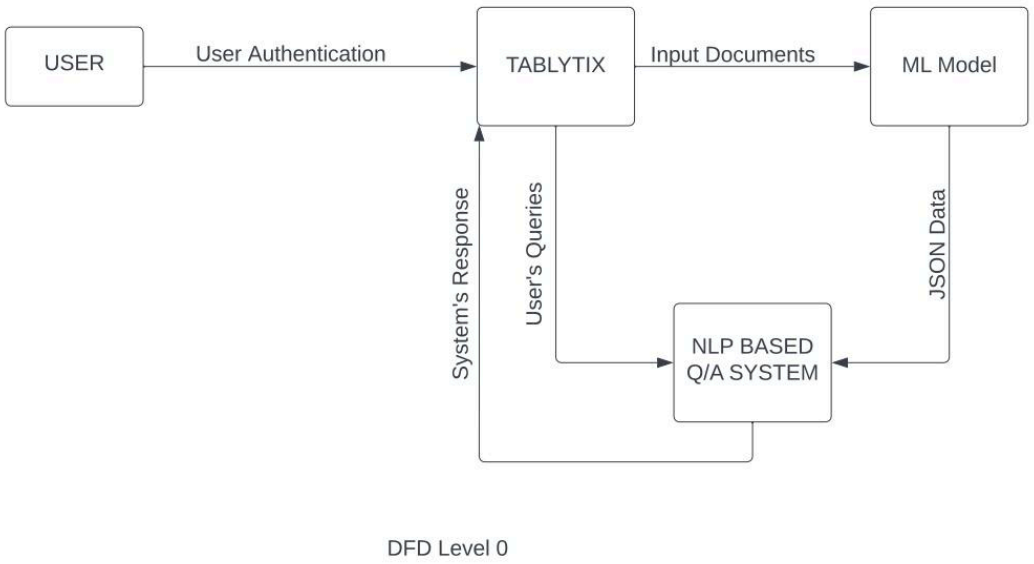


Figure 5: Data Flow Diagram (DFD level 0)

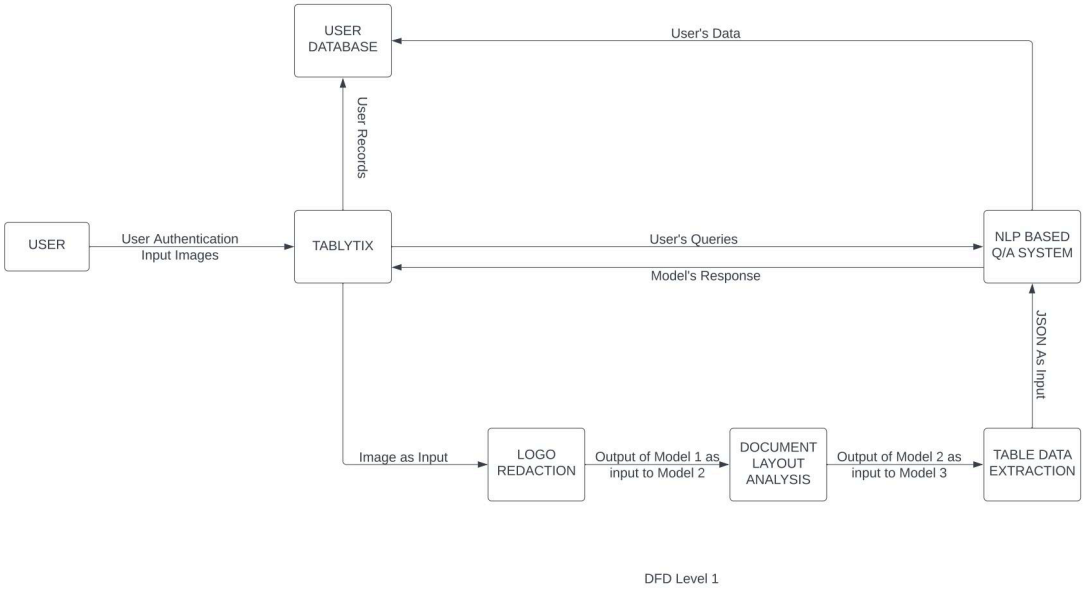


Figure 6: Data Flow Diagram (DFD-1)

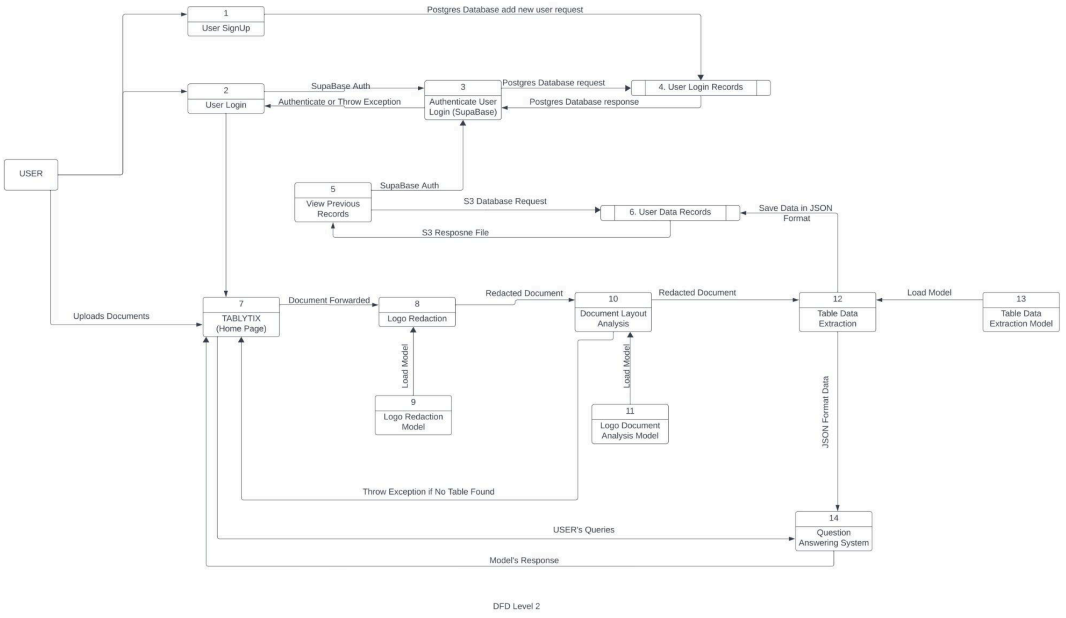


Figure 7: Data Flow Diagram (DFD-2)

4.2.2 ER Diagram

ER diagram models some of the key entities and relationships involved in our system's user authentication and document storage capabilities.

It depicts three main entities:

- Users - Stores registered user information including a unique ID, email, and encrypted password.
- Buckets - Represents storage containers for user documents. Has attributes like a unique ID, name, owner ID linking to a user, and file size limit.
- Objects - Models the actual documents stored in buckets. Has attributes like ID, name, metadata, link to parent bucket, and file size limit.

The diagram shows the one-to-many relationships between entities. A user can own multiple buckets. A bucket can contain multiple object documents.

Modeling these entities and connections conveys the core components and relationships involved in user management and document storage in our system. The database schema can be derived from this diagram.

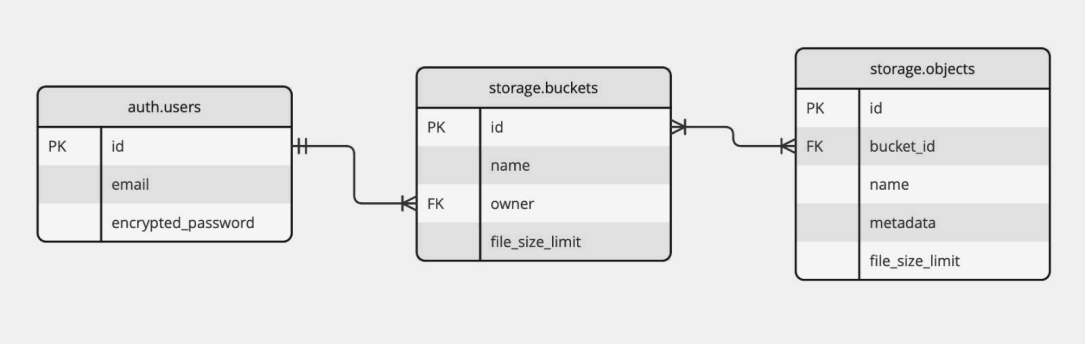


Figure 8: ER Diagram

4.2.3 Class Diagram

The system contains the following main classes:

- Login - Manages user authentication with email and password. Has a method to authenticate users.
- User - Stories user information like ID, email, password. Has methods related to user accounts like password reset and logout.
- Document - Represents documents uploaded to the system. Stores metadata like type, name, size. Has methods to read and write documents.
- Logo - Analyzes logo images in documents. Stores logo properties like location, count, page number. Includes methods to redact logos.
- Layout - Extracts layout information from documents like tables. Stores layout properties. Has methods for layout analysis and table extraction.
- QA_Bot - Provides a question answering bot. Stores requests and responses.

Has a method to query the bot.

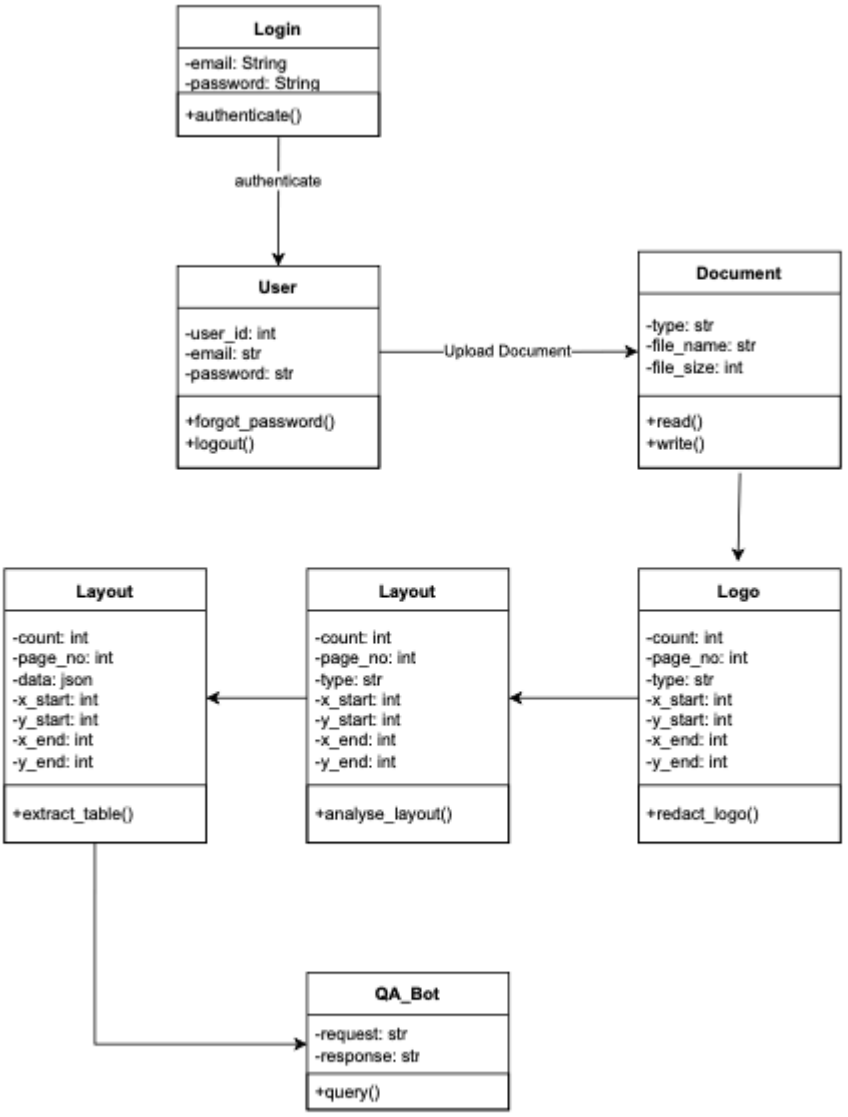


Figure 9: Class Diagram

4.3 User Interface Diagrams

4.3.1 Use case Diagrams

Use case diagram models the key functionalities and workflows available to users in our document processing and question answering system.

It depicts the main Actors - the User and the System. Key Use Cases for the User include:

- Signing up and registering in the system

- Logging in and out
- Uploading documents for processing
- Entering queries and questions
- Accessing previously processed documents

The System handles use cases like:

- Verifying user credentials
- Redacting logos
- Analyzing document layout
- Extracting tables
- Answering user questions
- Storing and allowing download of documents

The diagram shows which actors are involved in each use case and the relationships between them. This provides a high-level view of the key interactions and workflows enabled by the system.

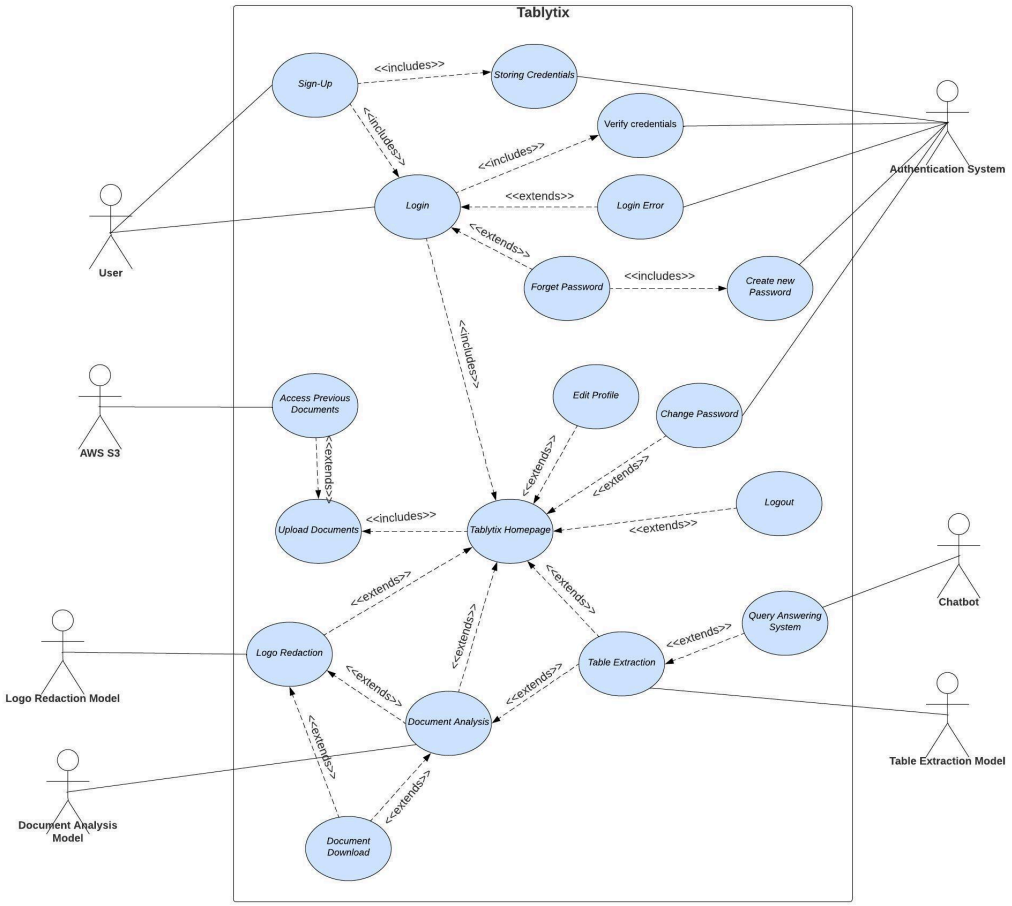


Figure 10: Use Case Diagram

Table 7: Use Case #1 SignUp

Use Case Title	Signup
Abbreviated Title	Signup
Use Case ID	1
Actors	User
Description: With this facility, the user will create an account to take full advantage of this application.	
Pre-Conditions: The user must have the internet.	
Task Sequence: Users can create an account using their email, Google account or Twitter account.	
Post Conditions: User will be redirected to the user information form.	
Date of Modification: 03/05/2023	
Author: Anshul Kanwar	

Table 8: Use Case #2 Login

Use Case Title	Login
Abbreviated Title	Login
Use Case ID	2
Actors	User
Description: With this facility the user can login to their account.	
Pre-Conditions: The user must have an internet connection and an account on our website.	
Task Sequence: Users can login to their account using their email & password, Google account or Twitter account.	
Post Conditions: The user can now access the dashboard.	
Date of Modification: 03/05/2023	
Author: Anshul Kanwar	

Table 9: Use Case #3 Forgot Password

Use Case Title	Forgot Password
Abbreviated Title	Forgot Password
Use Case ID	3
Actors	User
Description: With this facility the user can reset password for their account.	
Pre-Conditions: The user must have an internet connection, an account on our website and forgot the password.	
Task Sequence: User will receive a mail with a new password.	
Post Conditions: The user can now access the dashboard.	
Date of Modification: 04/05/2023	
Author: Anshul Kanwar	

Table 10: Use Case #4 Tablytix Homepage

Use Case Title	Tablytix Homepage
Abbreviated Title	Tablytix Homepage
Use Case ID	4
Actors	User
Description: The user is provided with user-friendly Homepage, which facilities the services such as uploading of documents, document analysis, logo redaction, and so on	
Pre-Conditions: Good connectivity from users end, and successful login into the website	
Task Sequence: This use case follows after the successful login of the user	
Post Conditions: This provides all the facilities as advertised by the project, directly to the user via this webpage	
Date of Modification: 04/05/2023	

Use Case Title	Tablytix Homepage
Abbreviated Title	Tablytix Homepage
Use Case ID	4
Actors	User
Description: The user is provided with user-friendly Homepage, which facilities the services such as uploading of documents, document analysis, logo redaction, and so on	
Author: Agamjot Singh	

Table 11: Use Case #5 Upload Documents

Use Case Title	Upload Documents
Abbreviated Title	Upload Documents
Use Case ID	5
Actors	User
Description: User can upload the documents such as pdfs, docs, from their end to use the services provided on Tablytix platform	
Pre-Conditions: Good connectivity from users end; document type uploaded by users is supported by the Tablytix platform	
Task Sequence: After successful login, the Tablytix platform opens up the Tablytix homepage, which offers various services, amongst is the uploading of documents	
Post Conditions: On successful uploading of document, user can perform document analysis, logo redaction, table extraction tasks on the said documents	
Date of Modification: 04/05/2023	
Author: Agamjot Singh	

Table 12: Use Case #6 Access Previous Documents

Use Case Title	Access Previous Documents
Abbreviated Title	Access Previous Documents

Use Case ID	6
Actors	AWS S3, User
Description: With this feature the user can access previously uploaded and worked on documents	
Pre-Conditions: The AWS S3 service is active and fully functioning on its end, and smooth connectivity between AWS S3 service and user	
Task Sequence: User requests the access to previously accessed documents, while given the option to upload the documents	
Post Conditions: The user can now access the previously uploaded and worked on documents	
Date of Modification: 04/05/2023	
Author: Harneet Kaur	

Table 13: Use Case #7 Edit Profile

Use Case Title	Edit Profile
Abbreviated Title	Edit Profile
Use Case ID	7
Actors	Authentication System, User
Description: With this feature the user can edit their profile	
Pre-Conditions: Good connectivity from users endpoint. Good connectivity with authentication systems. Successful login into the system.	
Task Sequence: Once the user is successfully logged in, can choose to edit profile from Tablytix homepage.	
Post Conditions: Now Onwards user would now login using the edited profile information.	
Date of Modification: 04/05/2023	
Author: Harneet Kaur	

Table 14: Use Case #8 Logo Redaction

Use Case Title	Logo Redaction
Abbreviated Title	Logo Redaction
Use Case ID	8
Actors	User, Logo Redaction Model
Description: With this facility, users are provided with service to redact logos from their documents for privacy purposes.	
Pre-Conditions: The user must have an internet connection and successfully logged in the website. The user must have uploaded some documents or selected from previously accessed documents to perform logo redaction on. The Logo Redaction model must be available online. The User could also perform logo redaction after successful document analysis	
Task Sequence: The successfully logged in user selects a previously accessed document or upload a document, and follows to perform logo redaction on the said documents. The logo redaction use case may also be followed after successful document analysis.	
Post Conditions: The user is now provided with the document without logos, hence catering to privacy needs of users. The logo redacted document is also available for downloading.	
Date of Modification: 04/05/2023	
Author: Ojas Sharma	

Table 15: Use Case #9 Document Analysis

Use Case Title	Document Analysis
Abbreviated Title	Document Analysis
Use Case ID	9
Actors	User, Document Analysis Model
Description: This use case provides the user to use the Document analysis service, which is ML model separately highlights the images, text, tables and so on, for easy information extraction from the documents	

Pre-Conditions: Good connectivity from users end. Successful login of user Successful upload of documents or selection of previously accessed documents
Task Sequence: Document Analysis follows the Tablytix homepage, from where the service can be directly selected, after successful uploading of documents.
Post Conditions: The analyzed document is available for download by the user. The analyzed document can also be used directly for logo redaction services.
Date of Modification: 04/05/2023
Author: Anshul Kanwar

Table 16: Use Case #10 Document Download

Use Case Title	Document Download
Abbreviated Title	Document Download
Use Case ID	10
Actors	User
Description: This use case provides the user with the ability to download the processed documents for further uses.	
Pre-Conditions: The user must have good connectivity The user must be successfully logged in. The user must have successfully uploaded document or selected one from previously accessed documents The logo redaction model successfully provided with a redacted document or the document analysis model successfully did the analysis of the provided document.	
Task Sequence: This use case follows after successful logo redaction or document analysis of provided documents.	
Post Conditions: The user is provided with the downloadable document for further uses.	
Date of Modification: 04/05/2023	
Author: Ojas Sharma	

Table 17: Use Case #11 Table Extraction

Use Case Title	Table Extraction
Abbreviated Title	Table Extraction
Use Case ID	11
Actors	Table Extraction Model
Description: data in tables is converted into key values pairs (JSON format) .	
Pre-Conditions: user uploads a table	
Task Sequence: information in tables is extracted and now users can ask queries from the QA bot.	
Post Conditions: users can get their queries resolved from the bot.	
Date of Modification: 05/05/2023	
Author: Ojas Sharma	

Table 18: Use Case #12 Query Answering System

Use Case Title	Query Answering System
Abbreviated Title	QA bot
Use Case ID	12
Actors	Users , chatbot
Description: User can ask queries regarding the documents they have uploaded	
Pre-Conditions: Information has been extracted from the table	
Task Sequence: user asks questions and gets answers as output	
Post Conditions: users can get their queries resolved	
Date of Modification: 05/05/2023	
Author: Ojas Sharma	

Table 19: Use Case #13 Change Password

Use Case Title	Change Password
Abbreviated Title	Change Password
Use Case ID	13
Actors	User
Description: With this feature, users can change their password.	
Pre-Conditions: The user must have an internet connection and must be logged in to their account.	
Task Sequence: User clicks on the change password button on their dashboard and gets redirected to a new page where they will enter current and new password.	
Post Conditions: User password gets updated.	
Date of Modification: 05/05/2023	
Author: Mayank Rawat	

Table 20: Use Case #14 Log-out

Use Case Title	Log-out
Abbreviated Title	Log-out
Use Case ID	14
Actors	User
Description: With this feature, users can log out of their account.	
Pre-Conditions: The user must have an internet connection and an account on our website.	
Task Sequence: Users can login to their account using their email & password, Google account or Twitter account.	
Post Conditions: The user can now access the dashboard.	
Date of Modification: 05/05/2023	
Author: Mayank Rawat	

4.3.2 Activity Diagram

This activity diagram illustrates the key workflows and actions a user performs when using our document processing and question answering system.

It begins with user registration and login, including resetting forgotten passwords. Upon valid login, it moves to the Dashboard representing the main hub for accessing system capabilities.

Key workflows shown include:

- Uploading a document which invokes the core ML models for logo redaction, layout analysis, and table data extraction.
- Entering queries that get processed by the Question Answering system based on the extracted doc data.
- Viewing the results returned from the QA system.
- Logging out to exit the system.

The diagram depicts the sequential flow of activities from registration to accessing the main system functions of document processing and QA. It highlights the key actions users take along with decision points and different paths based on input or conditions. Walking through the flows demonstrates the connectivity between activities and conveys the overall workflow as users utilize the system's capabilities.

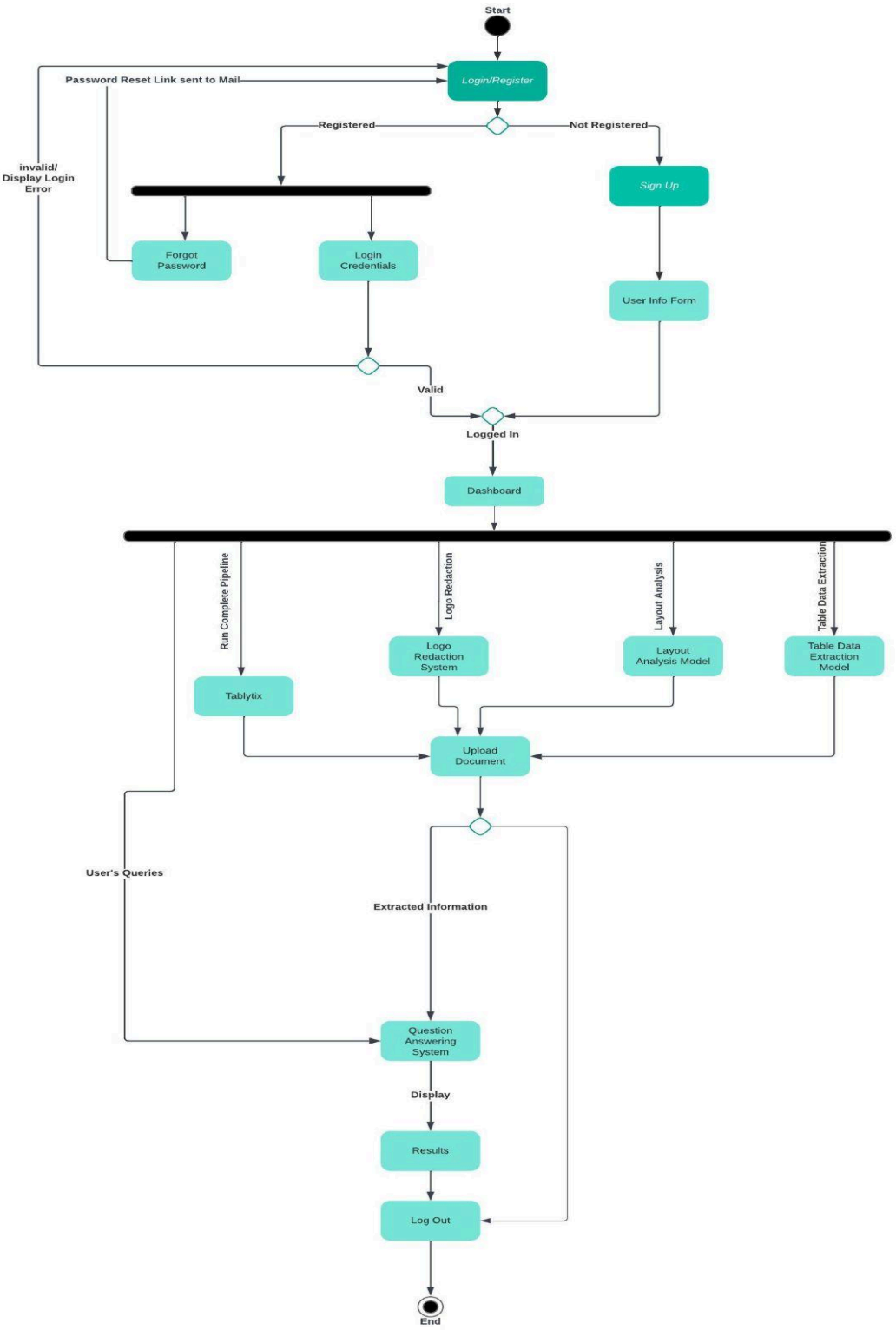


Figure 11: Activity Diagram

4.4 Snapshots of Working Prototype

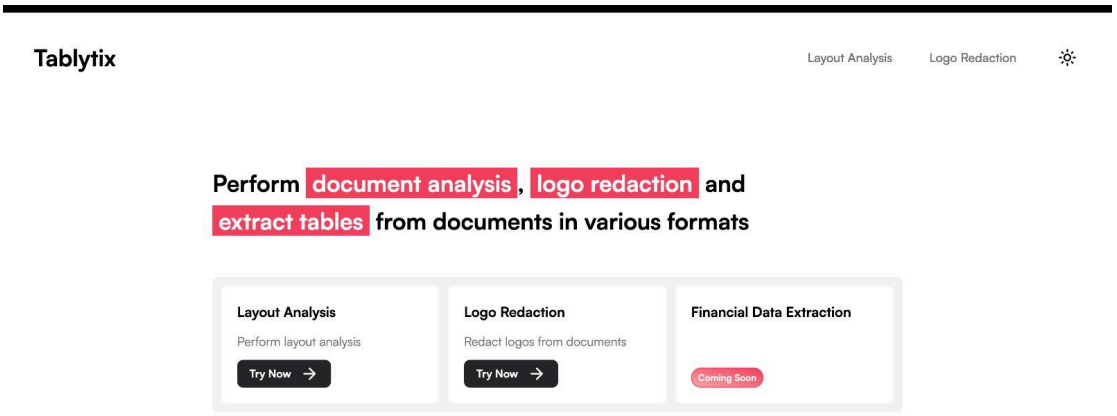


Figure 12: Tablytix Homepage

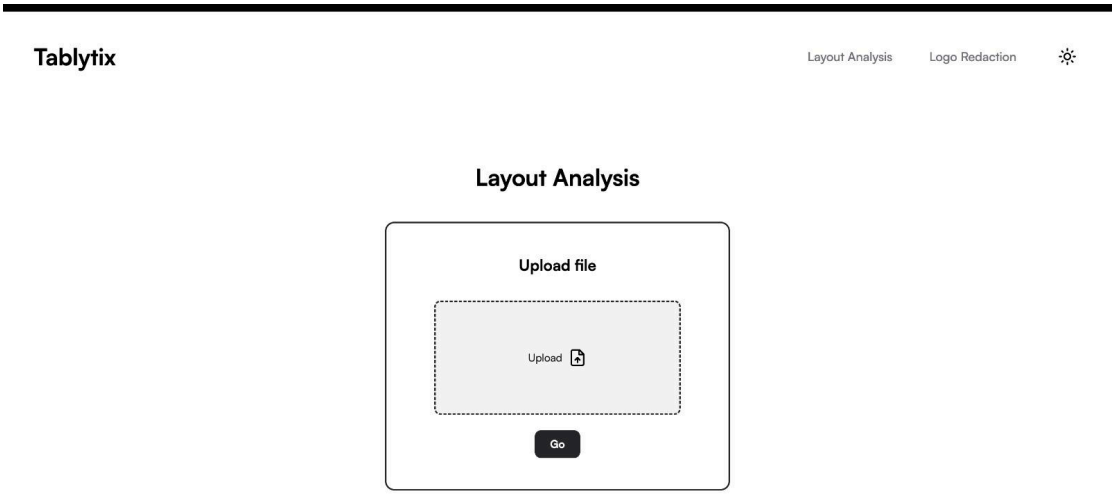


Figure 13: Tablytix Layout Analysis page

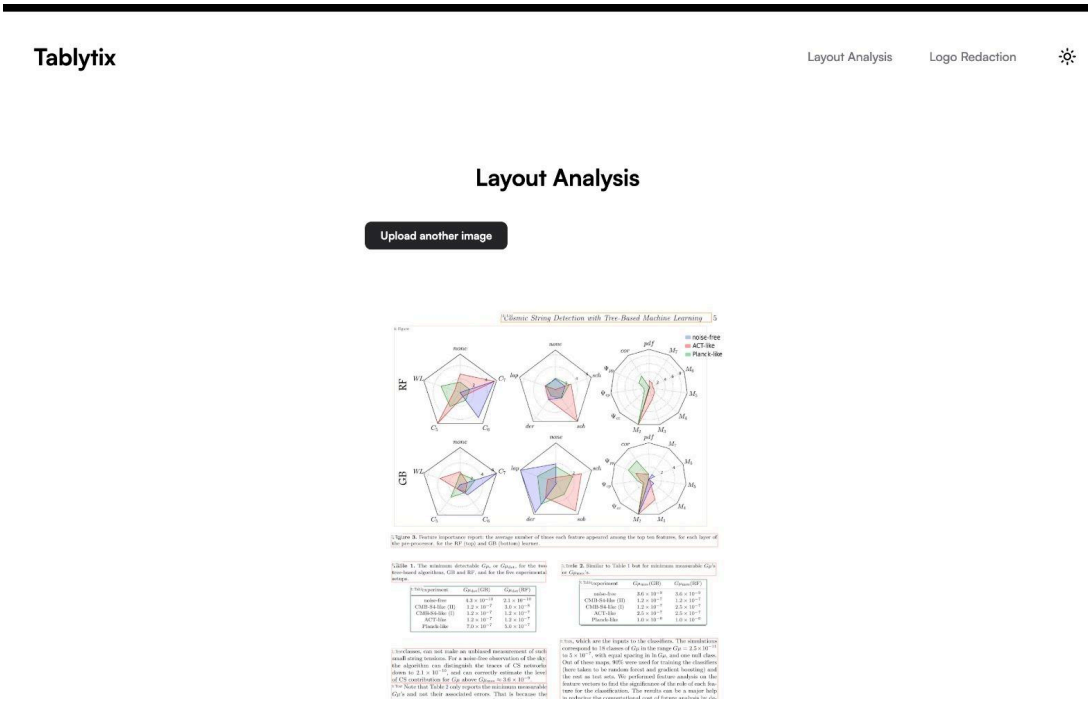


Figure 14: Tablytix result of layout analysis

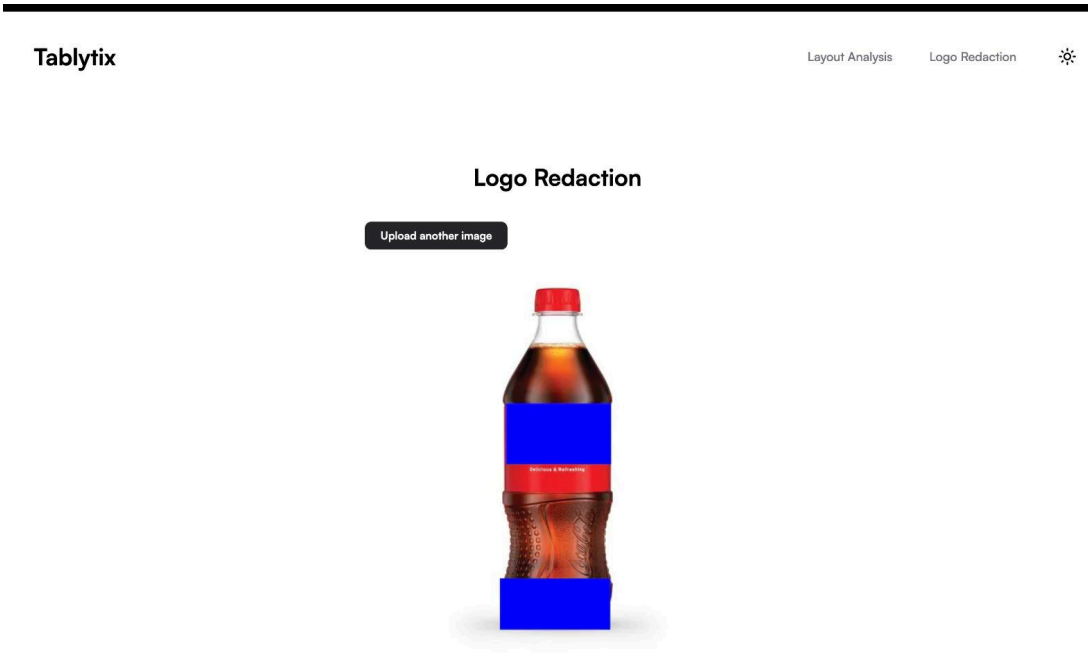


Figure 15: Tablytix result of logo Redaction

Implementation and Experimental Results

5.1 Experimental Setup

The experimental setup involves various modules:

(i) Firstly, the backend technologies employ Supabase for authentication, PostgreSQL for data management, and leverage Supabase's edge functions alongside FastAPI for optimized serverless computing and robust API development.

(ii) AWS S3 serves as the foundation for scalable object storage, complemented by Docker containerization for seamless deployment and development consistency.

(iii) Thirdly, machine learning takes center stage with IceVision models integrated within Docker, addressing logo redaction, document layout analysis, and table extraction. Deployment occurs on the reliable Google Cloud Platform, facilitated by Docker containers for consistent application rollout.

The user needs to upload the document from which the backend first Logo Redaction is performed by Logo-RetinaNet model, which output document is then passed to detectron2 model to perform Document Layout Analysis. The output generated by the Document Layout Analysis is passed to PDFplumber model which for the given image merge overlapping or nearly overlapping lines, finding the intersections of all those lines, hence grouping contiguous cells into tables for Table Extraction Model. Now this table extracted is converted into JSON format which is searchable and indexable. Now question answering bot based on GPT4 takes the parsed table in json format and handle the user given queries regarding the table data

5.2 Experimental Analysis

5.2.1 Data

The dataset used for this project is LogoDet-3k and Visually 29k2. The reason for choosing these datasets is because this logodet-3k creates a more challenging benchmark for logo detection, for its higher comprehensive and wider variety in both logo categories and annotated objects compared with existing datasets. The highlight of Visually 29k is that it is a large data-scale curated infographics dataset. In its

repository metadata, annotations, and processing scripts for tens of thousands of infographics, is provided for computer vision and natural language research.
Cocosplit, requiring python3, is the tool used to split multi-label coco annotation dataset whilst preserving class distribution among train and test sets.

5.2.2 Performance Parametres

The performance parameters of our project are:-

1. Low Latency operation – Real time operation in our project depends on the ability of all our three modules to work together and provide real time results. Unified executable file will need to provide accurate results in real time from the video feed and forward the results to the backend server module. The backend server module then needs to forward this data to the frontend module in real time for the end user to see.
2. Low Cost – Our project uses equipment which are cheap and affordable, and with low power needs. This makes it both affordable and sustainable.
3. Usability – Our project does not need any specialized equipment, making it very convenient and easy to set up and use.
4. Accuracy – Accuracy of our project depends on accuracy of various readings we will be taking of the different text files.
5. Integrity: It gives accurate and reliable results without storing any user information.
6. Reliability: Reliability is a quality indicator for web application services that indicates how well they are maintained. It will be updated on a regular basis to improve the user experience.
7. Regulatory: Webapp service regulation ensures that rules, laws, and earlier negotiated agreements and standards are followed. It is verified that it does not exceed legal limits at any point.
8. Security: The quality feature of web application services is security, which ensures information confidentiality, integrity, authentication of the people involved, and non-repudiation, among other things. The security of the user database is ensured by our web application service.

5.3 Working of the project

5.3.1 Procedural Workflow

In Tablytix, the model consist of 4 different major processes that work consecutively in the background:

1. Logo Redaction

The first process involves analyzing the document and detecting the presence of logos in the images. We are using Logo-Retinanet as offered by keras, after training it on LogoDet-3k and Visually29K or Visuallydata. The model is capable of finding brand independent logos. After the logo is detected, it is redacted from the document with the help of coordinates of bounding boxes and python scripts. This is done to ensure privacy of the documents before parsing it further into the system.

2. Document Layout Analysis

Document layout analysis is performed on the logo redacted document to check for the presence of tables in the document. We are using Detectron2 model, developed by Facebook for this process. The model is able to classify different components of the document. If the table is detected, then it is detected and its contents are sent to the next component for the table extraction process.

3. Table Extraction

Our method is influenced by Anssi Nurminen's master's thesis and takes inspiration from Tabula, a tool used for extracting tables from PDFs. Let's break down the process outlined in the text:

a. Line Detection:

   Identify lines in the document that are explicitly defined or implied by the alignment of words on the page. Explicitly defined lines are likely drawn lines in the document, while implied lines are inferred based on the alignment of text.

b. Line Merging:

   Merge lines that overlap or are nearly overlapping. This step helps in creating a more unified representation of the table structure by combining lines that are very close to each other.

c. Intersection Detection:

Identify the intersections of all the lines detected in the previous steps. These intersections represent the potential vertices of cells in the table.

d. Rectangle Generation:

   Determine the most granular set of rectangles (cells) that can be formed using the identified intersections as their vertices. This step involves defining the boundaries of each cell in the table.

e. Table Grouping:

   Group contiguous cells into tables. This involves organizing the individual rectangles into coherent table structures based on their proximity and connectivity.

In summary, the approach involves detecting lines in a document, merging them for better representation, finding the intersections of these lines, generating rectangles based on these intersections, and finally grouping the rectangles into tables. The thesis involved, by Anssi Nurminen, likely provides more in-depth details about the algorithms, methodologies, and optimizations used in the process of table detection and extraction.

4. Question Answering System

We are using GPT4 to build a question answering system. Firstly the document is scanned for text using OCR techniques. Then the scanned text along with the table information generated in the JSON format in the previous step is passed to the GPT4 model. The user inputs the queries regarding the document, and the GPT4 processes the queries while keeping in context the OCR information and extracted table information. It generates a suitable response and then the response is sent back to the

user.

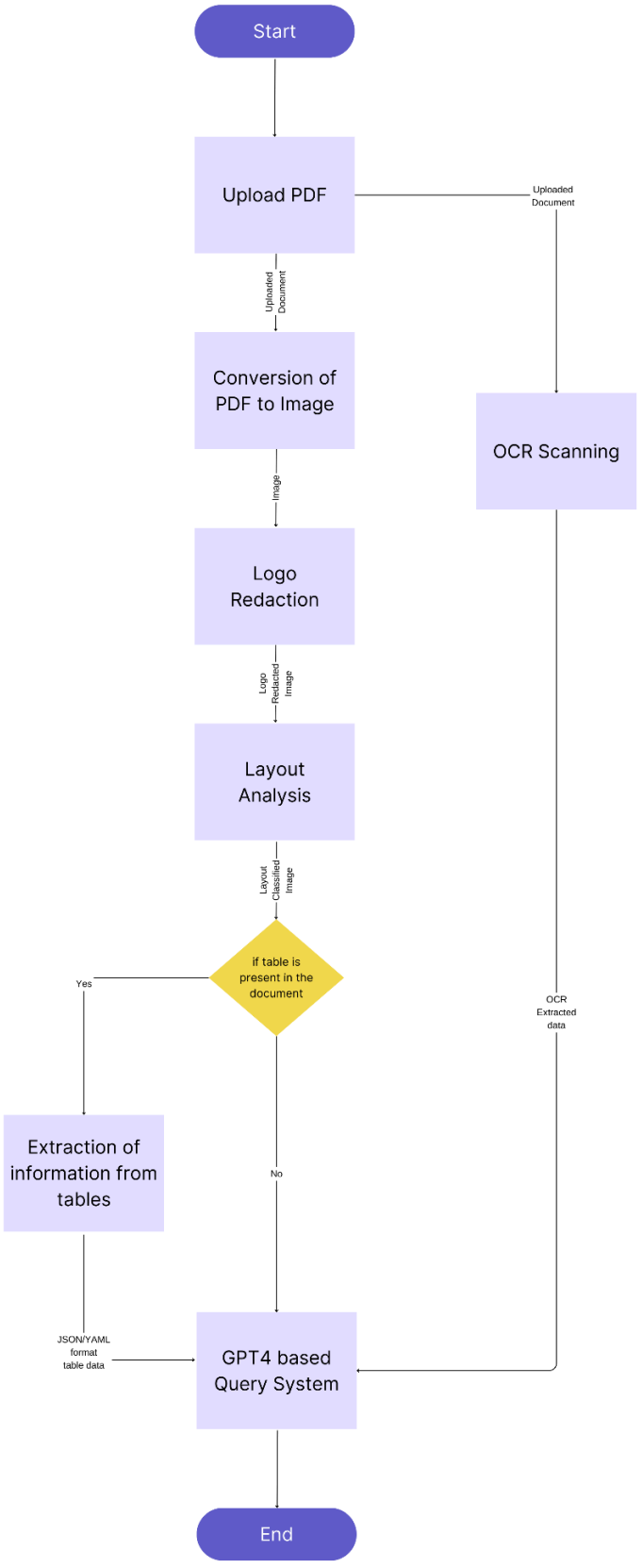

Start
Upload PDF
Uploaded Document
Uploaded Document
Conversion of PDF to Image
OCR Scanning
Image
Logo Redaction
Logo Redacted Image
Layout Analysis
Layout Classified Image
if table is present in the document
Yes
OCR Extracted data
Extraction of information from tables
No
JSON/YAML format table data
GPT4 based Query System
End

Figure 16: Procedural Workflow

5.3.2 Algorithmic Approaches Used

Logo Redaction:

We use the icevision library for our model training and inference purposes. We use mmdetection's retinanet architecture for the object detection model with a pre-trained resnet50_fpn_1x as its backbone. The Metric used was COCOMetric

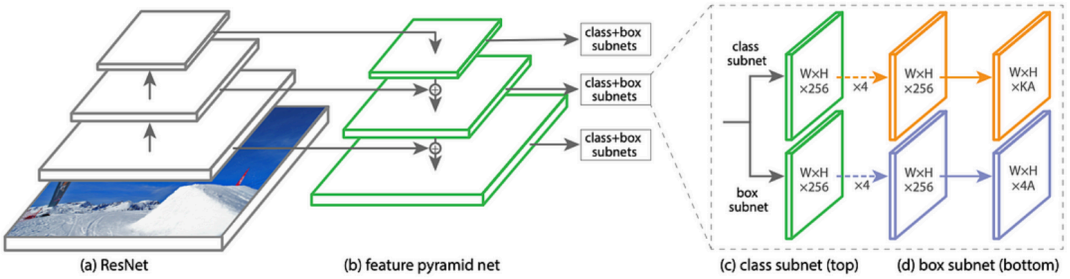


Figure 17: Logo Redaction Model Implementation

Training:

Figure 18: Training of Logo Redaction model

For the training process, first, a custom data parser is used that will parse our data and present it to the model for training and testing purposes. We first train our data on 30,000 images. Later we increased the number of images to 53000. The 30,000 images only include images from LogoDet-3K, and 53,000 contain images from LogoDet-3K and Visually 29K. We train them for 30 epochs and 50 epochs, respectively.

Later it is found that the validation loss or Metric is not increasing significantly, so training is stopped.

Figure 19: Training loss graph for trained model

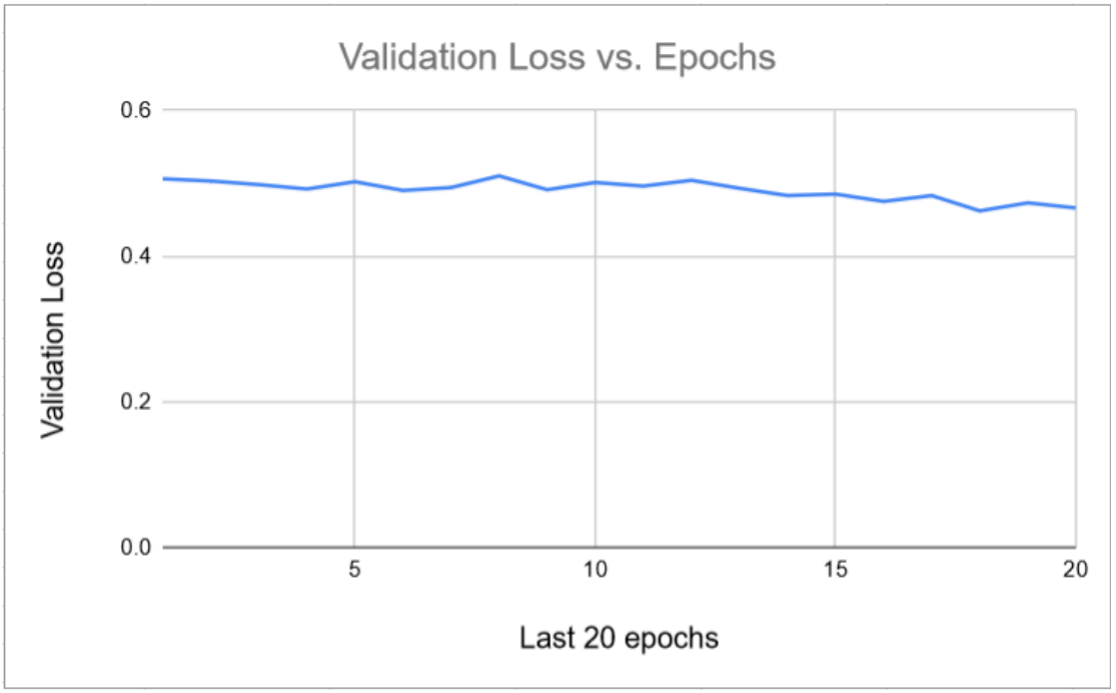


Figure 20: Validation loss graph for trained model

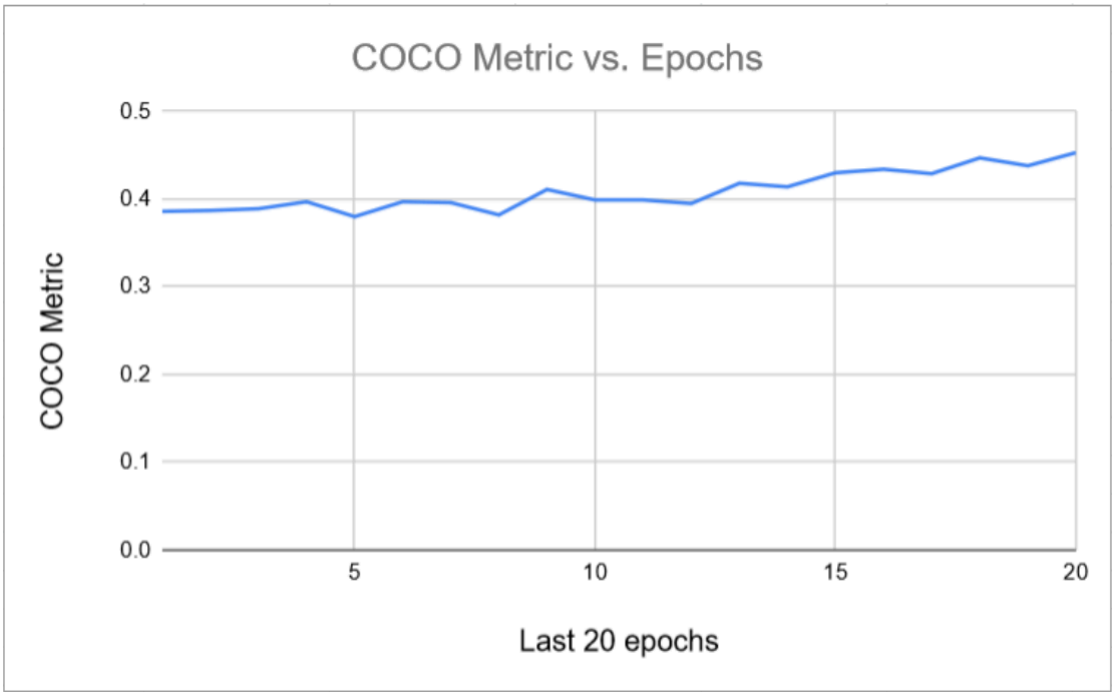


Figure 21: COCO Metric graph for trained model

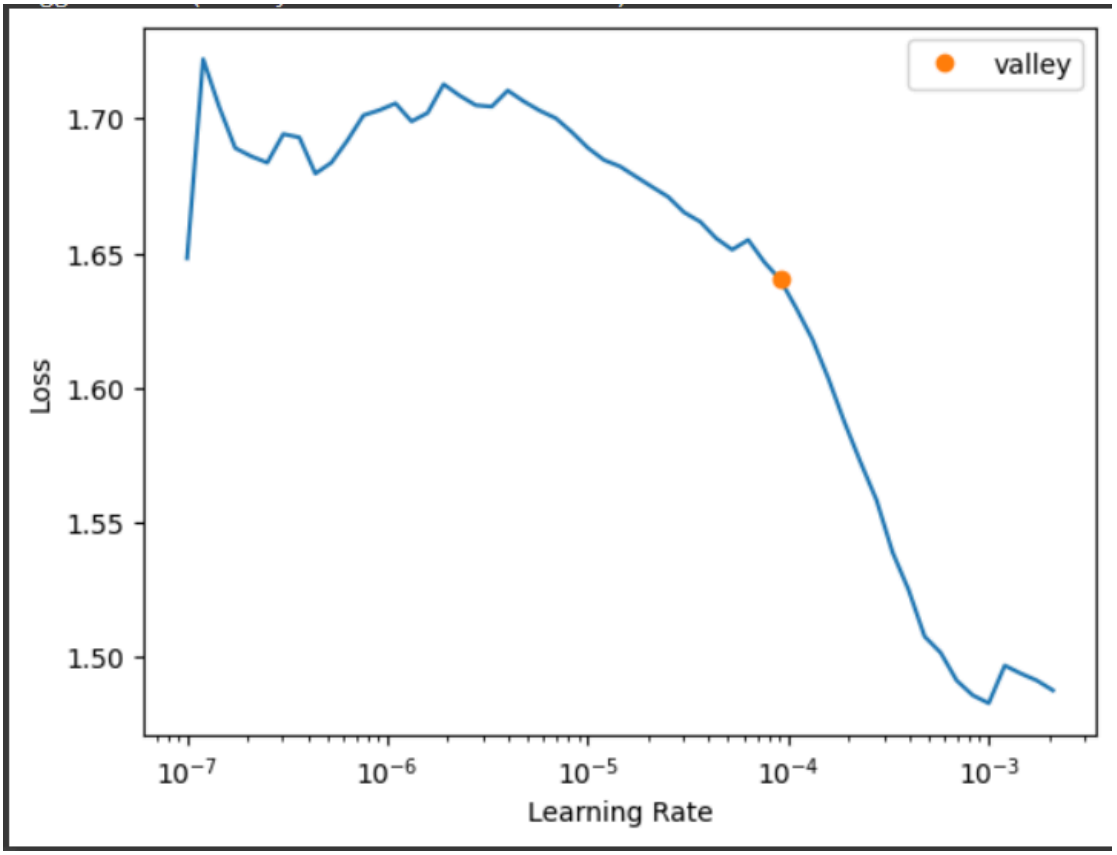


Figure 22: Loss and Learning rate graph for trained model

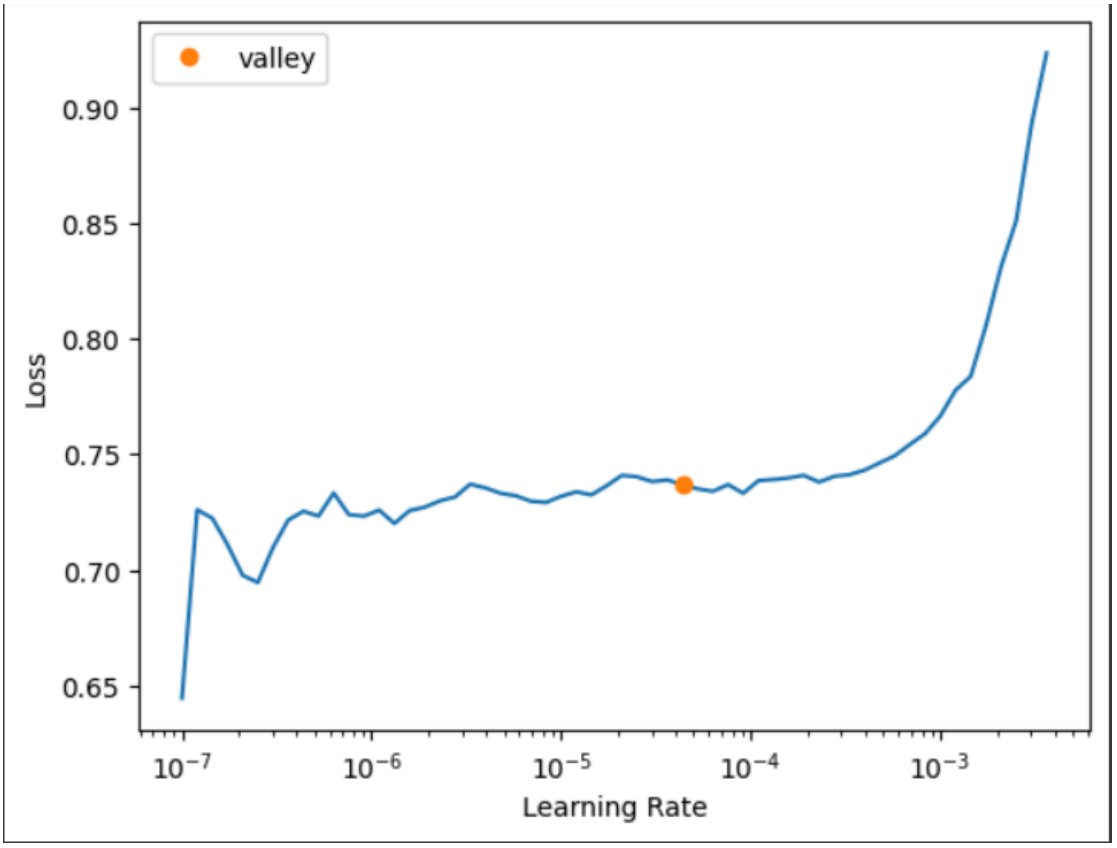


Figure 23: Loss and Learning rate graph for trained model

Pseudo Code / Algorithm:

1. Import Libraries:

Import required libraries such as `fastai`, `icevision`, and `pandas`.

2. Initialize Template Record:

Create an instance of `ObjectDetectionRecord` and generate a template using `Parser.generate_template(template_record)`.

3. Define LogoParser Class:

Create a class `LogoParser` that inherits from `Parser`.

Initialize with a template record, image-to-bbox mapping (`img2bbox`), and directory.

Implement `__iter__`, `__len__`, `record_id`, and `parse_fields` methods to iterate over images and populate records.

4. Set Image Size and Transforms:

Set the image size to 384.

Define training and validation transforms using `tfms.A.Adapter`.

5. Create Datasets:

Create training and validation datasets (`train_ds` and `valid_ds`) using specified records and transforms.

6. Define Model:

Get the model type (`retinanet`).

Set up the backbone (ResNet50 FPN) and create the model with the specified number of classes.

Create data loaders (`train_dl` and `valid_dl`) for training and validation.

7. Initialize Learner:

Define extra arguments and create a fastai learner with the model, data loaders, and specified metrics (COCOMetric).

8. Find Learning Rate:

Use the learning rate finder (`learn.lr_find()`) to determine an appropriate learning rate.

9. Freeze and Train First Layer:

Freeze the model layers except the first one.

Train the model for one cycle using a specified learning rate range and callbacks (CSVLogger, SaveModelCallback, EarlyStoppingCallback, ReduceLROnPlateau).

10. Unfreeze and Find New Learning Rate:

Unfreeze all layers.

Use the learning rate finder again to determine a new learning rate.

11. Train Unfrozen Model:

Train the unfrozen model for additional epochs using the new learning rate, with callbacks for logging and saving models.

Layout Analysis:

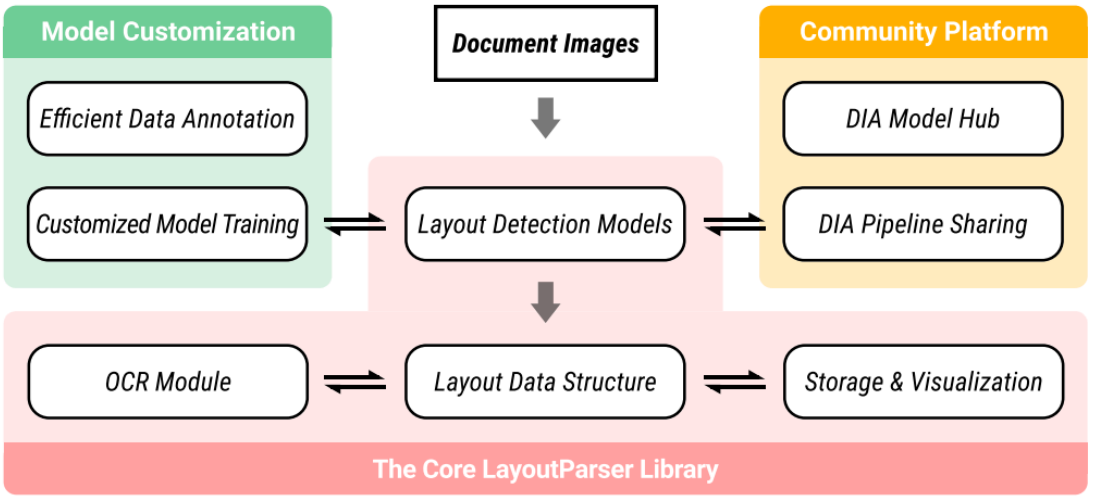


Figure 24: Layout Analysis diagram

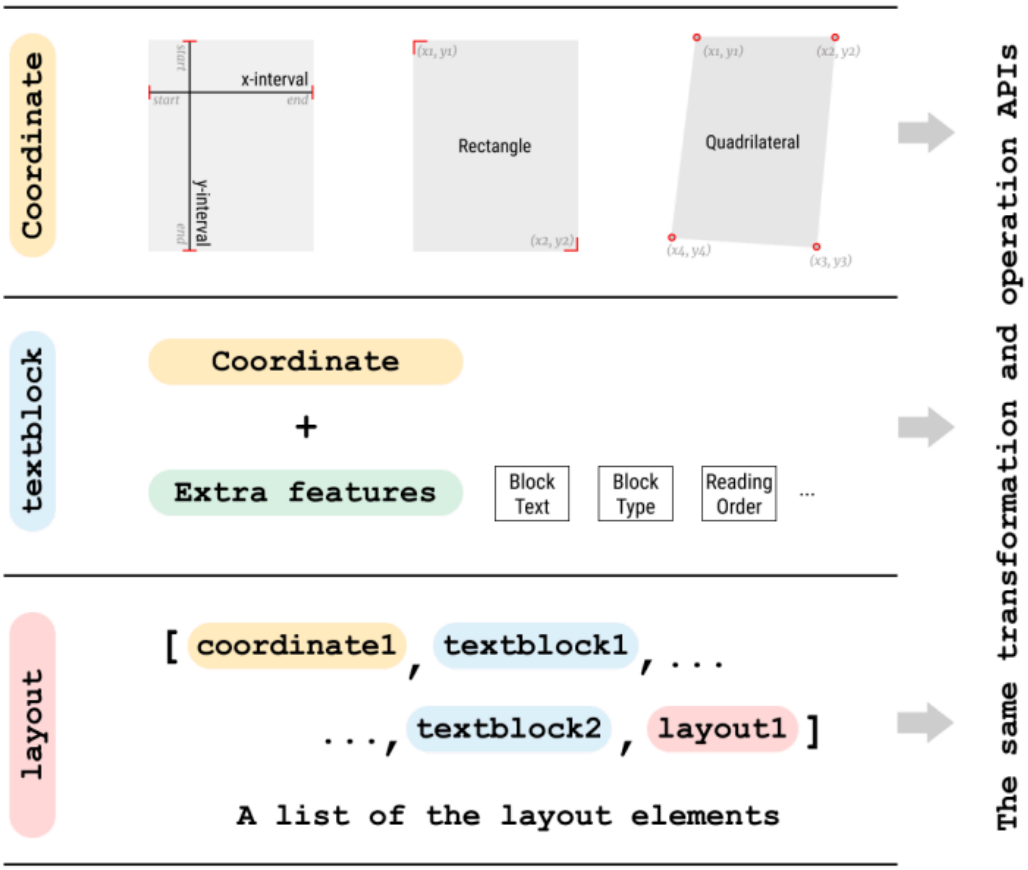


Figure 25: Transformations in Layout Analysis

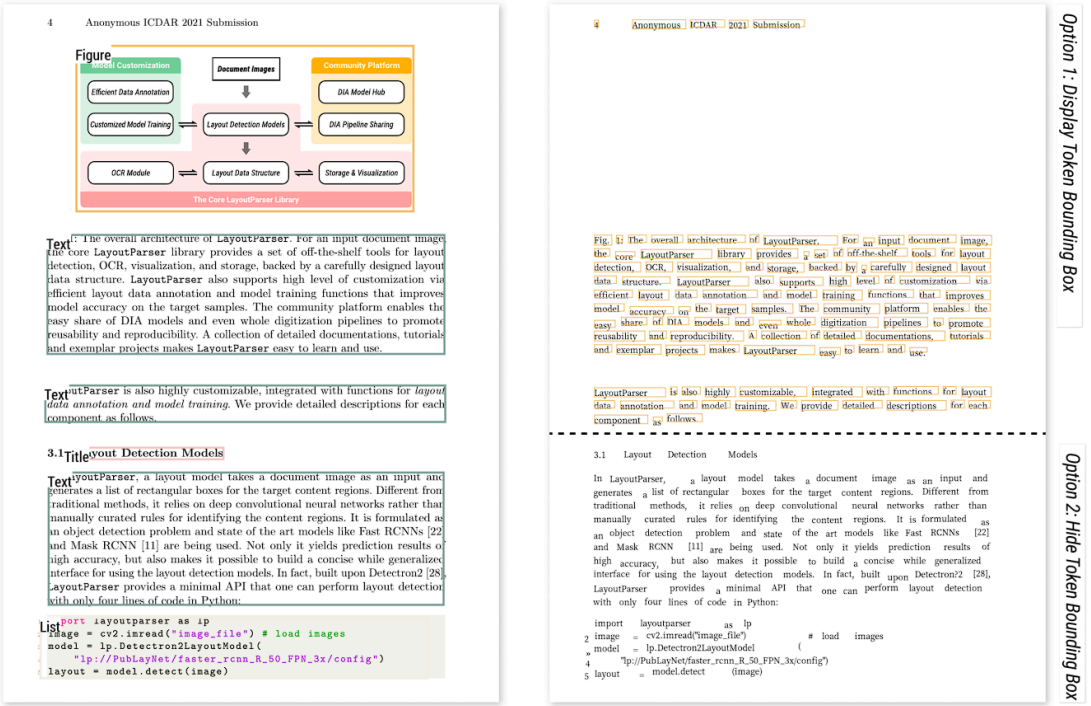


Figure 26: Model Results during Layout Analysis

We trained our detectron2 based layout analysis model on PublayNet and achieved a 0.96 Jaccard score.

Pseudo Code / Algorithm:

1. Download and Load Test PDF:

   Download a PDF from a given URL (https://arxiv.org/pdf/2106.00676.pdf)

   Save the PDF as "test.pdf"

   Load PDF tokens and images using layoutparser

2. Initialize AutoLayoutModel and Detect Layout:

   Initialize AutoLayoutModel with EfficientDet for layout detection

   Detect layout on a specific page (e.g., page 11) of the loaded PDF

   Draw layout boxes on the image

3. Install and Start Label Studio Service:

   Clone the Label Studio repository from GitHub (https://github.com/heartexlabs/label-studio.git)

   Install Label Studio using pip

   Run migrations and start the Label Studio service

4. Download COCO Annotations and Visualize:

Download COCO annotations using a Python script (download_annotation.py)

Load COCO annotations from the downloaded file

Visualize layout with annotated text blocks for two random images

5. Clone Layout Model Training Repository:

Clone the Layout Parser layout-model-training repository from GitHub (git@github.com:Layout-Parser/layout-model-training.git)

6. Split COCO Annotations and Train Layout Model:

Run cocosplit.py script to split COCO annotations into train and test sets

Train a Detectron2LayoutModel using specified configuration and model paths

7. Initialize Trained Layout Model:

Initialize Detectron2LayoutModel with the trained model configuration and weights

Set additional configurations, e.g., "MODEL.ROI_HEADS.SCORE_THRESH_TEST" to 0.8

8. Detect and Draw Layout on Test PDFs:

Download another PDF from a given URL (https://arxiv.org/pdf/2004.07180.pdf)

Save the PDF as "test2.pdf"

Load PDF tokens and images from the new PDF

Detect layout on specific pages (e.g., page 8 and 9) using the trained model

Draw layout boxes on the images

9. Extract References from Reference Section:

For pages in the reference section (e.g., 8, 9, 10), detect bibliographic items using the trained model

Extract text tokens corresponding to the detected bibliographic items

Overlay the bibliographic item blocks onto the original page images

10. Display Extracted Bibliographic Text:

Display the layout with annotated bibliographic items for each page in the reference section

Extract and display the text content of each bibliographic item

11. Train Layout Parser (Training Part):

Define paths to training and testing datasets (e.g., train.json, test.json)

Load training data using layoutparser functions

Define model configuration parameters (e.g., model architecture, input size, output classes)

Initialize a layout model using the specified configuration

Set hyperparameters such as learning rate, batch size, and number of epochs.

Create a DataLoader for the training data with the specified batch size

Initialize an optimizer (e.g., Adam) with the specified learning rate

Define a loss function appropriate for layout detection (e.g., Binary Cross Entropy)

For each epoch in the specified range:

- Iterate over batches from the DataLoader
- Forward pass: Input data through the model
- Calculate the loss based on predictions and ground truth
- Backward pass: Compute gradients and update model parameters

After training, evaluate the model on the testing dataset

Compute and display evaluation metrics (e.g., accuracy, precision, recall)

Save the final trained model weights and configuration to a specified path

Table Data Extraction:

Our method is influenced by Anssi Nurminen's master's thesis and takes inspiration from Tabula, a tool used for extracting tables from PDFs. Let's break down the process outlined in the text:

a. Line Detection:

   Identify lines in the document that are explicitly defined or implied by the alignment of words on the page. Explicitly defined lines are likely drawn lines in the document, while implied lines are inferred based on the alignment of text

b. Line Merging:

   Merge lines that overlap or are nearly overlapping. This step helps in creating a more unified representation of the table structure by combining lines that are very close to each other

c. Intersection Detection:

   Identify the intersections of all the lines detected in the previous steps. These intersections represent the potential vertices of cells in the table.

d. Rectangle Generation:

Determine the most granular set of rectangles (cells) that can be formed using the identified intersections as their vertices. This step involves defining the boundaries of each cell in the table.

e. Table Grouping:

Group contiguous cells into tables. This involves organizing the individual rectangles into coherent table structures based on their proximity and connectivity.

Pseudo Code:

1. Text Element Rotation Analysis:

For each text element on the page:

- Skip elements with less than 3 characters
- If element height is greater than width:
  - Calculate distances from top and bottom
  - Increase rotation count for 90 or 270 degrees based on distances
- Else:
  - Calculate distances from left and right
  - Increase rotation count for 0 or 180 degrees based on distances

Determine page rotation as the maximum count among rotations (0, 90, 180, 270)

2. Access PDF Page and Render as Image:

Access the page of the PDF file using Poppler

Generate a QImage of the rendered page with specified parameters (xres, yres, x, y, width, height)

3. Rectangle Detection:

Initialize an array for found rectangles

Loop for each crossing-point:

- Fetch all points on the same vertical and horizontal line with the current crossing point
- For each pair of points (x_point, y_point) in the respective lists:
  - If edges exist between the points and a hypothetical bottom-right point:
    - Confirm the rectangle with four sides and add it to the found rectangles array

4. Table Extraction Class:

Define a class named TablerInstance with public methods:

- ExtractTables(aFilePath, aPageRange): Extract tables from a PDF document

- SortTables(aPopDoc, aTableRects): Extract tables from specified regions in a Poppler Document.

Question Answering System

We are using GPT4 to build a question answering system. Firstly the document is scanned for text using OCR techniques. Then the scanned text along with the table information generated in the JSON format in the previous step is passed to the GPT4 model. The user inputs the queries regarding the document, and the GPT4 processes the queries while keeping in context the OCR information and extracted table information. It generates a suitable response and then the response is sent back to the user. It required prompt engineering to handle all the user testcases

5.3.3 Project Development

By breaking down a complex object-oriented system into smaller components, a component diagram makes the system more manageable. It represents the physical view of a system that resides within the node, such as executables, files, libraries, and so on. It aids in the creation of an executable system and the visualization of the linkages and arrangement of the system's components.

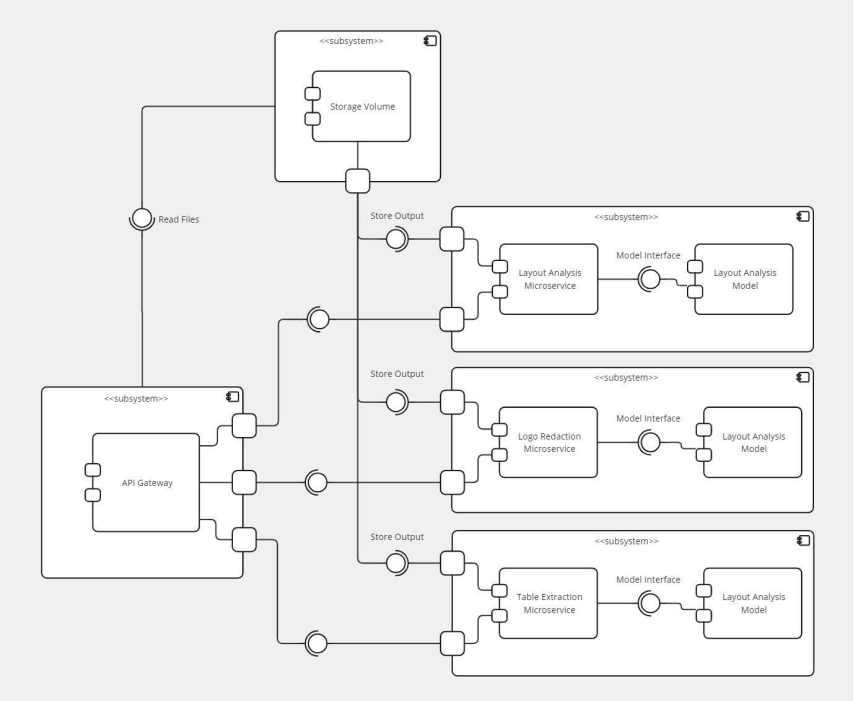


Figure 27: Component

The deployment diagram depicts the physical hardware that will be used to run the software. It depicts a system's static deployment view. It entails the nodes and their connections. The deployment diagram's main purpose is to show how software is deployed on the hardware component. It shows how a piece of software interacts with hardware to carry out its tasks..

Documents as inputs given by the user to the website.

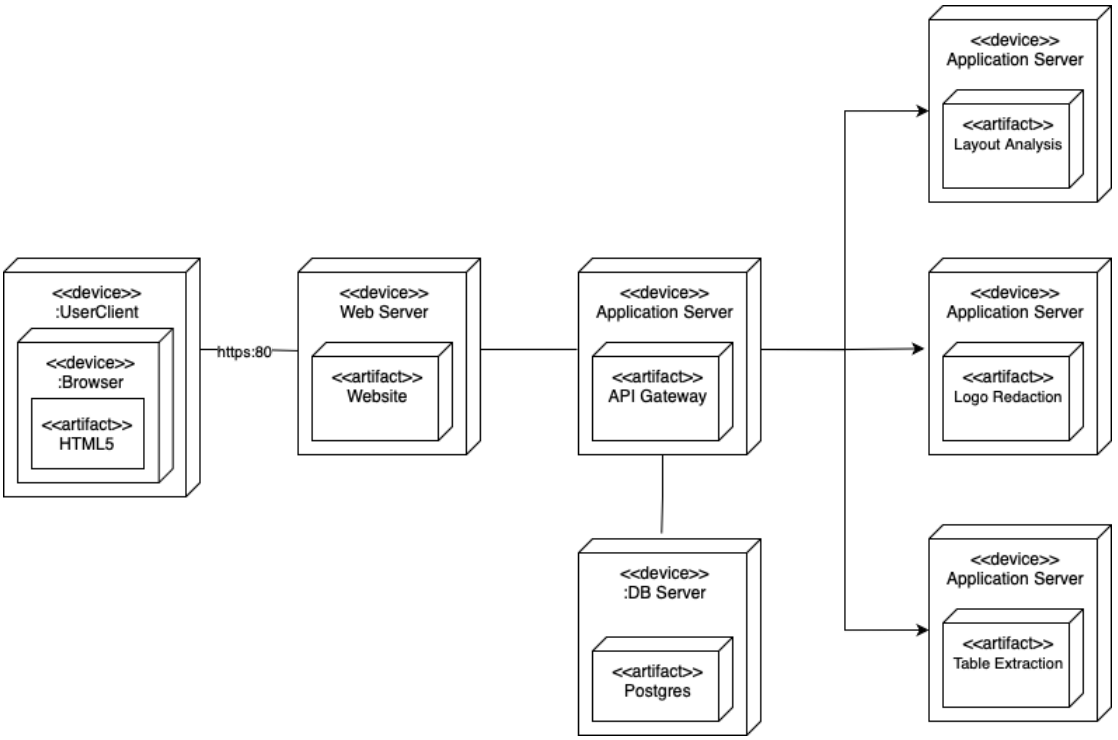


Figure 28: Deployment Diagram

5.3.4 System Screenshots

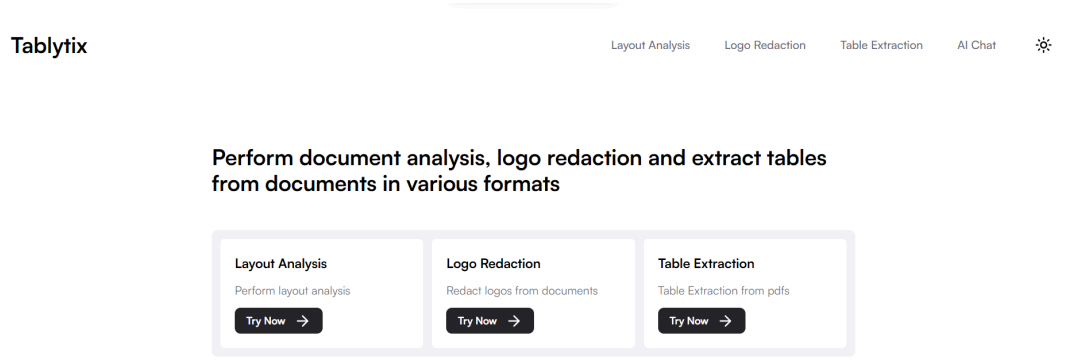


Figure 29: Web application user interface

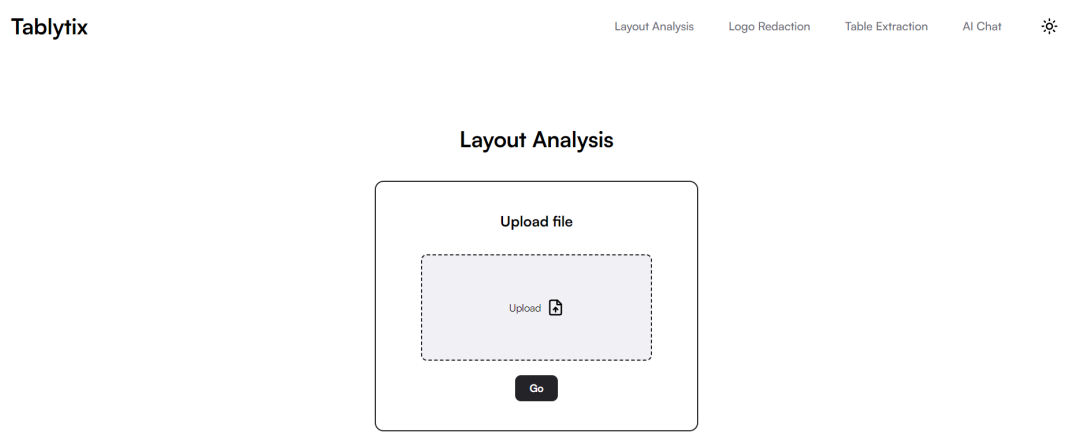


Figure 30: Upload file option for Layout Analysis

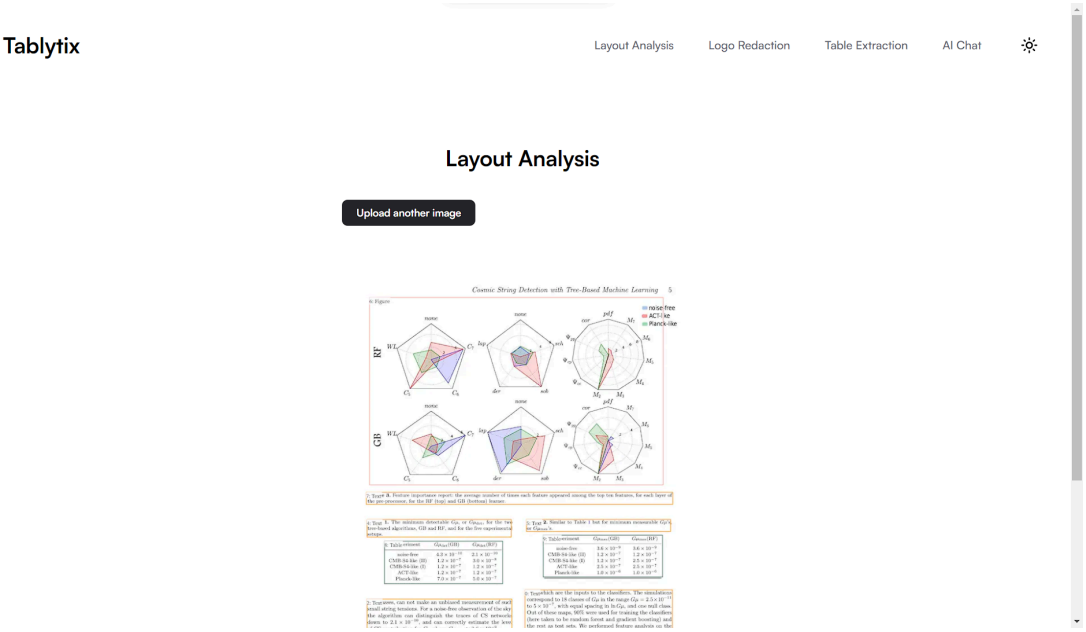


Figure 31: Result after Layout Analysis

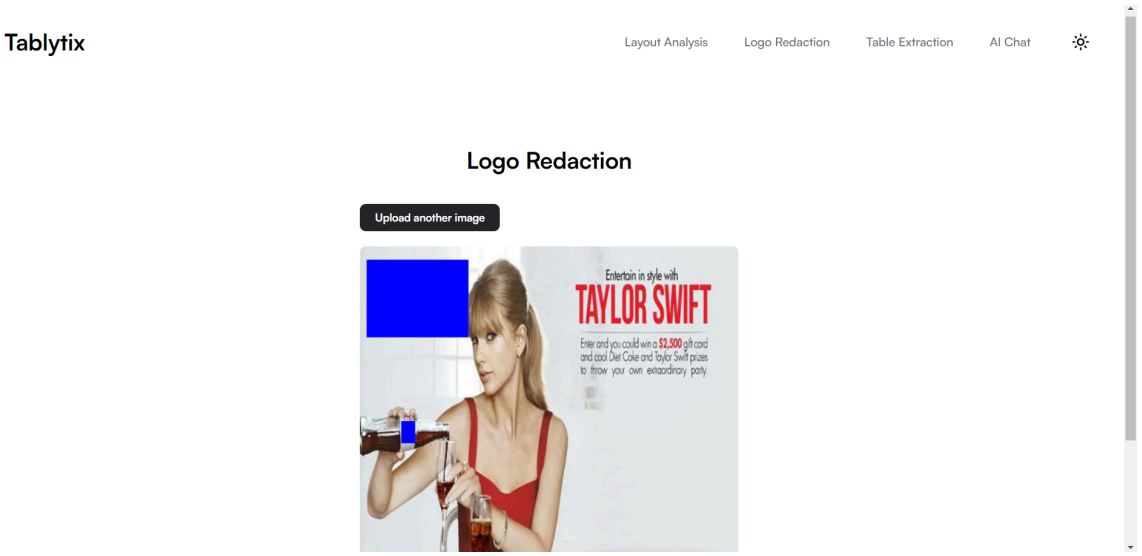


Figure 32: Result after Logo Redaction

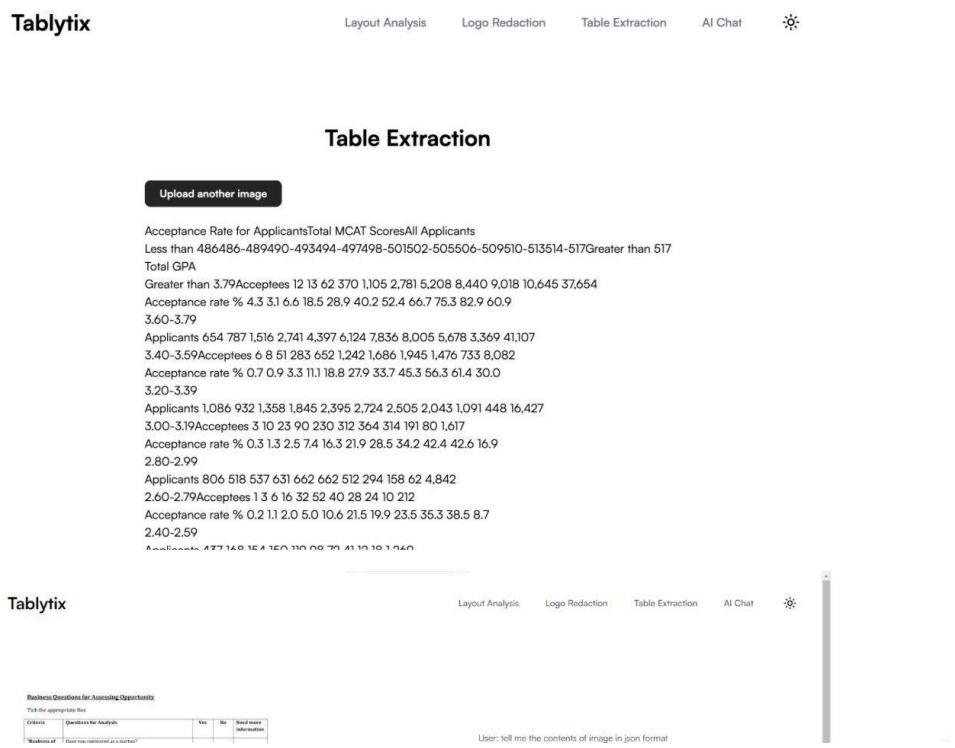


Figure 33: Result after Table Extraction

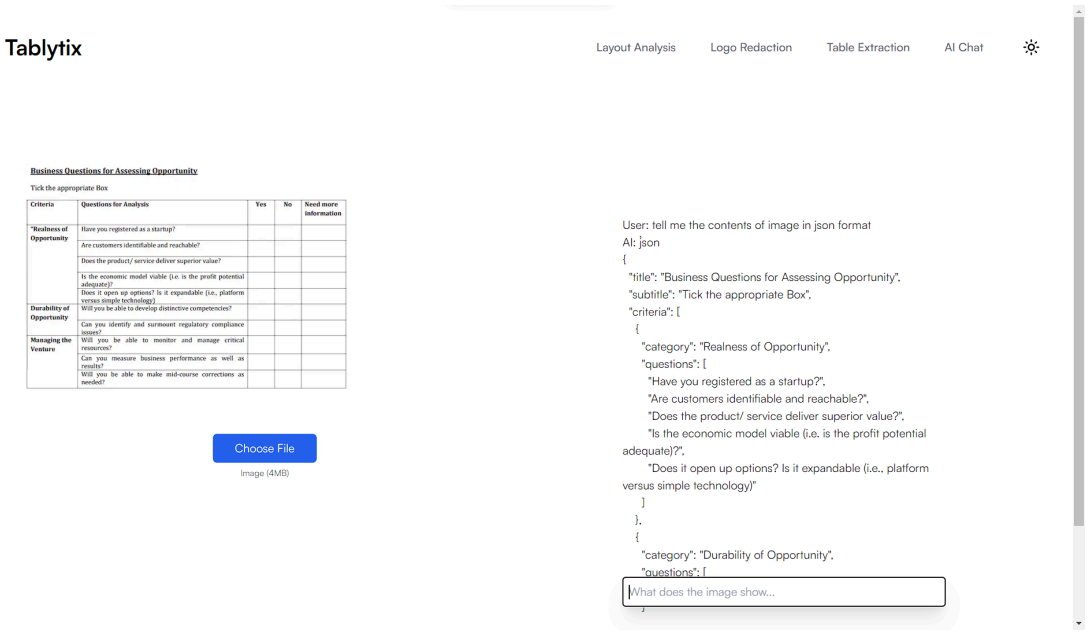


Figure 34: Results for Query Answering System 1

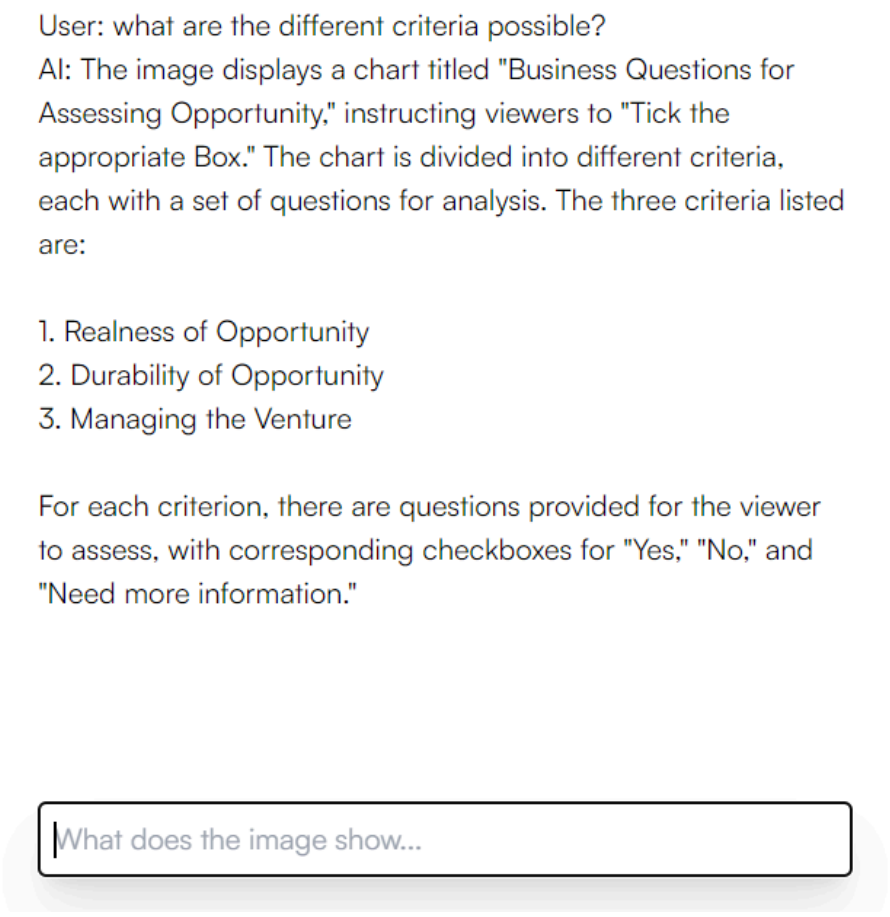


Figure 35: Results for Query Answering System 2

5.4 Testing Process

This section describes the complete testing process including the techniques employed for testing and fixing bugs, test cases and results after testing.

5.4.1 Test Plan

The testing plan majorly involved testing of all the models implemented in our Tablytix project and seeing which one performed best in the terms of accuracy and delay. On comparing the results the best performing models are chosen and integrated into the backend. After the integration all the functionalities are tested individually and later after the integration. The APIs are tested to work for maximum sections of cases.

Apart from this, the user interface was also rigorously tested to make sure all the features being presented to the user are working to their full functionality. It was also

ensured that the interface connections between all the components of the prediction model are communicating properly without any issues.

5.4.2 Features to be tested

There are certain components which need to be tested. Testing points out the bugs/errors that need to be solved/overcome before any advancement in the app release. For models, different performance measures would be used to check their accuracy and correctness.

A list of various components that need to be tested are provided below:

- Log In/Sign up to our platform.
- Uploading of documents for performing Table Extraction
- Checking the accuracy of Logo Redaction, Document Layout Analysis , Table Extraction models
- Query processing performed by the NLP models
- Performance, Usability, and Error-handling of the application
- Security of the sensitive information is preserved throughout the application processing

5.4.3 Test Strategy

The testing strategy should be able to test all the parts individually as well as collectively. The best way to do this is by using different types of testing techniques. We started with Unit Testing performed alongside the development of individual components.

Unit Testing:

- Module1: Document passed to Logo Redacted Document provided
  Input: Document contains logos is passed
  Output: Document provided by web application has all the Logos removed

- Module2: Document passed to Layout Analyzed document provided
  Input: Document passed
  Output: Document with all the various components labeled

- Module3: Document passed to Table Extracted document provided
  Input: Document passed
  Output: All types of tables from documents extracted

In the later stages of developments we also included Black box testing for testing each component separately.

Black box Testing:

- Module1: Query passed to Answer to query is returned
  Input: Query is sent by user
  Output: Answer to the query is returned containing the relevant table data

Lastly Integration testing was performed to ensure testing of all included modules to ensure that they are all useful when combined.

Integration Testing:

- Module1: Complete from Document upload to Query answered
  Input: Document is uploaded along with a query
  Output: Answer to the query contains the relevant table data, which is extracted from the tables contained in the document uploaded

5.4.4 Test Techniques

Software testing is the process of comparing a piece of software to user and system standards. It enables us to assess the functioning of a software application with the goal of determining whether the generated software meets the given requirements and identifying faults in order to ensure that the product is defect-free and thus produce a high-quality product.

Testing is conducted at the phase level in software development life cycle or at module level in program code. Integration testing has also been performed to test the interfaces between the different modules, and interfaces of the GUI, to ensure correct and fast flow of data without any issues.

Super Test library has been used to make sure all the connection interfaces are working at full functionality. For testing of the system as a whole, the prediction

system has been run on various unlabelled test data as well as dummy data to ensure there are no issues, and the system is working properly and with a good accuracy result.

5.4.5 Test Cases

Table 21: Test Cases

S.No.	Test Case	Input	Expected Output	Actual Output	Results
T001	Login	User email id	1.1 Account Logged In	1.1 Account Logged in	successful
			1.2 Error message if account does not exist	1.2 Error message if account does not exist	successful
T002	Sign Up	New user email id	2.1 Account created	2.1 Account created	successful
			2.2 Error message as account exist with those user credentials	2.2 Error message as account exist with those user credentials	successful
T003	Logo Redaction	Document passed	3 Document with logos removed returned	3 Document with logos removed returned	successful
T004	Document Layout Analysis	Document passed	4 Document returned with labels	4 Document returned with labels	successful
T005	Table Extraction	Document passed	5 All the different types of tables extracted	5 All the different types of tables extracted	successful
T006	Uploading of Document	Document passed	6 Document available for the models to process	6 Document available for the models to process	successful
T007	Query asked	String passed	7 Answer to the	7 Answer to the	successful

	by user		query is provided	query is provided	

5.4.6 Test Results

In order to process digital documents, Tablytix employs a variety of deep learning models. Pre-processing converts all types of file formats uploaded by the user into images. This image is further passed to Resnet50 backbone and further into the detectron2 model, and then it is processed for extracting of tables. After the successful extraction of tables, the user uploads a query, which is processed by the GPT4 model and accurately responds back with the relevant table data.

The dataset used in this project is LogoDet-3k and Visually29k and our models are trained after careful pre-processing which includes using cocosplit.

Tablytix uses LogoDet-3k and Visually29k datasets and our models are trained after careful pre-processing which includes using cocosplit.

The models give variable accuracy during document processing. For logo redaction using the Resnet50 backbone model we have achieved around 96.2% accuracy. For Document Layout Analysis using detectron2 model we have achieved 0.96 accuracy.

While performing table extraction, we achieved 68.62% accuracy. Additionally achieved a 0.98 Jaccard score16 and a 0.17 average Levinstein distances17 for token prediction on the test set.

Data set	Documents	Perfect scores	Purity	Completeness	CPF
EU	34	9	77.82%	49.12%	**60.22%**
US	25	11	73.96%	64.00%	**68.62%**

Figure 36: Model evaluation results for Table Detection

5.5 Results and Discussions

Various tests were performed as mentioned above. The majority of test results were positive and in favor. As expected, the overall model proved to be successful. Hence,

the model is efficient in performance and other performance parameters also as mentioned earlier resulted in positive outcome on evaluation. Each and every module in the application is working perfectly.

Tablytix uses a variety of deep learning models to recognise speech. Pre-processing converts all the file types of a document uploaded into the jpeg images format. The image is then passed on to Resnet50 backbone which performs logo redaction. Further on this redacted image is passed on to detectron2 model which performs the document layout analysis. This image is passed further for table extraction, and table data is returned in JSON format which is searchable and indexable. We are further using GPT4 architecture for answering queries from users regarding the table data. All these functionalities are performed in a pipeline.

Tablytix uses LogoDet-3k and Visually29k datasets and our models are trained after careful pre-processing which includes using cocosplit.

The models give variable accuracy during document processing. For logo redaction using the Resnet50 backbone model we have achieved around 96.2% accuracy. For Document Layout Analysis using detectron2 model we have achieved 0.96 accuracy.

While performing table extraction, we achieved 68.62% accuracy. Additionally achieved a 0.98 Jaccard score16 and a 0.17 average Levinstein distances17 for token prediction on the test set.

5.6 Inferences Drawn

Our project satisfies all the basic functionality as we want as per project objectives but still there is much to be improved and various new features must be added to make the user experience more pleasant and exceed expectations. The inferences drawn from the test results and by evaluating the performance parameters are:

- The backend of the website and the neural network models were successfully integrated, and the accepting of documents and their further functionalities performed.
- The majority of the project operations are performed on the google cloud linux instance, hence there is no special requirement on the user’s device, ensuring the faster and higher the performance of the system without

- There should be optimum and noise free surrounding conditions for more efficient results.
- The testing phase led to the inference that with the help of proper data processing algorithms and neural network models, performance of the system can be improved. Also, the accuracy of prediction can be further improved by using better equipment and hardware with stronger computational power.

5.7 Validation of Objectives

Table 22: Validation of Objectives

S.No.	Objectives	Status
1.	Redacting Company's Logos from financial document to preserve privacy	successful
2.	Analyzing Layout of Documents using deep learning models	successful
3.	Extraction of data from tables and storing data in a JSON format, making data indexable and searchable	successful
4.	Creation of Question Answering System using NLP techniques, for efficient access of table data	successful

Conclusions and Future Directions

6.1 Work Accomplished

The following objectives have been accomplished which are listed below:

- Implementation and evaluations of Logo Redaction, Document Layout Analysis, and Table Extraction models
- A NLP powered query answering system for information extraction from tables
- Integration of all the above models into backend
- Our model is compatible with all devices, and operating systems.

- It allows uploading of documents of various file formats
- It outputs a logo redacted document, queries are answered by NLP powered model
- The UI is user-friendly, intuitive and responsive.

All these objectives have been achieved successfully with all the testings performed on individual components and finally integrated product of all the components.

6.2 Conclusions

All the objectives listed above have been successfully achieved. We have implemented a variety of machine learning and deep learning models in order to get the best accuracy within the limited time constraints. Here we have used cloud computing technologies in order to make our project more resource-efficient as well as for any future scalings to be done. Our models are all separately contained in the docker containers, allowing easy management of the versions dependencies of our models. We have implemented the Resnet50, detectro2, and PDFplumber which provided us with the best accuracies in this application. Our team paid great attention to learning and experimenting with new technologies to find the best fit by utilizing rapid prototyping strategies of software engineering. With all this we conclude, the successful completion of our Tablytix project, a cost-effective one-stop shop for the businesses, catering to all their digital document needs, such as Logo Redaction, Document Layout Analysis, Table Extraction and query system providing insights into the vast wealth stored in the tables of digital documents.

6.3 Environmental, Economic and Societal Benefits

Environmental Benefits:

Our project Tablytix provides users with a meaningful digital platform for document analysis, allowsing reduction of paper consumption, helping in the conservation of trees and minimizing waste. This project also enables a lower carbon footprint, since eliminating the need for physical document transportation and carbon emissions. This project provides a solution for feasibly and efficiently sharing of digital documents

within the organizations, by providing an easy and efficient logo redaction method, following the compliances set forth by the companies. This solution is also energy efficient as it utilizes cloud hosting and containerization technologies and promotes resource-efficient energy consumption compared to traditional physical infrastructure.

Economic Benefits:

Tablytix provides an easy and efficient Logo Redaction tool, which makes the document sharing between the organizations efficient and hence also brings in economically better results for both the collaborators. This also helps companies remove the unnecessary financial burden for manual logo removal. Tablytix also provides easy access to information stored in tables within the digital document. This makes the accessing of such insights more fast and efficient, reducing the financial expenses used on this task, and also provides with newer and better insights, makes up for the better decisions for the benefit of the company. Automation of document analysis and its related features helps in operational efficiency and saves time for users and organizations. The businesses can use this technology to their competitive advantage by streamlining document analysis and efficient data extraction from tables.

Social Benefits:

Tablytix is not only limited to large businesses, but also caters to the smaller and newer businesses needs, making it accessible to a wider range of users. Even non businesses can also use this website to their advantage, for example students can also utilize the functionalities offered and get efficient insights from a wide range of data. Tablytix takes special care of the privacy concerns of the businesses and allows them an easy and efficient way for ensuring compliance with data protection regulations. Additionally the table insights features empowers companies to divert their focus and resources for better outcomes.

Conclusion:

Tablytix website’s environmental benefits include easy and efficient digital document related functionalities such as Logo Redaction, Document Layout Analysis and Table Extracion, thereby reducing the physical paper usage and its wastage and hence also energy efficient. While economic benefits encompass cost savings, efficiency gains,

and potential increase in revenue by informed decisions based on the insights from the tables. Concerning the social benefits, the project offers accessibility, privacy protection, educational value, and empowerment to make informed decisions by tapping into the wealth of information stored in the tables in the digital documents. Hence to conclude Tablytix tries to create a positive impact on the environment, economy, as well as society.

6.4 Reflections

On the successful completion of our Tablytix project, our team conclude to the following reflections:

- Throughout the development and implementation of the project, the whole team has learnt a lot of things, gained insightful experience and most importantly learnt the value of Teamwork. The idea of Tablytix started with only an app that would provide efficient access to information hidden in the tables contained in the digital documents, but after realizing the potential of this app, we decided to add a number of more features which would make it more effective and commercial.
- This idea proceeded with us formulating the idea and with the help of our mentor, who helped in adding more clarity, depth to this idea and guiding us what can be done and how. We read various research papers and articles giving us more insight on what we can do and we started implementing these ideas. Throughout the project, we experienced and learnt a lot of new things as we proceeded further with any module.
- We gained a wide wealth of information about writing SRS i.e. Software Requirement Specification on a realistic project. We also learnt application development. This capstone project also taught us about team coordination and research ethics as per industry standards.
- A lot was learnt about data pre-processing and filtering and how much of an impact it has on the accuracy results of a machine learning model. It gave us the opportunity to learn about a lot of techniques to help process raw data in order to extract the useful features from it, as required by the field or

application we need to use the data for.

- Knowledge gained about Machine Learning & Deep Learning could be applied during the development of this project. A lot of difficulties and challenges were encountered during the training of these models, but experience was gained on coping with these problems and overcoming them through systemic testing processes.
- Overall the project enabled us in learning to be effective and efficient, carefully following the set standards, while experimenting with the new technologies in order to bring out the best result possible and giving cost-effective and efficient solutions for the business uses.

6.5 Future Work Plan

The future work plan outlines the additional features and functionalities which can be scaled and implemented in our Tablytix project.

- We would like to store the documents uploaded in repositories, for easy access by users, and eliminating the need for repetition of uploading document in order to perform the functionalities provided
- We would also like to provide users with history of the actions performed and queries made and their results.
- UI enhancements can be done in order to make website more user friendly
- We would like to experiment further with the up-coming machine learning and deep learning models, aiming to get better accuracy and efficiency.

Project Metrics

7.1 Challenges Faced

During the development of the project, the following challenges were faced:

Selection of datasets: Selection of datasets hugely impacts the accuracy and efficiency achieved by the models. Hence finding the right datasets plays a pivotal

role in the ml training models. Hence choosing of the right dataset amongst huge options available was a challenging portion.

Data Pre-processing: Since a publicly available dataset was used in this project, the data needed to be pre-processed thoroughly to make it suitable for our requirements. Data from different sources were combined to make a dataset large enough for the models to learn from and data augmentation techniques were also used.

User Interface Integration: Since none of the team members had worked on integration of deep learning models with web applications, there were difficulties in integration of the model with the interface. The team members had to study online and take the help of various guides to enable completion of the task.

ML Model Training: Due to lack of better machines with higher computational speeds such as GPUs, the training part took significant time and fine tuning the model was also a significant challenge.

Balance between cost and performance: As we aimed to build a cheap and affordable solution, it was necessary to keep the cost of the components down. To tackle this, we made sure that we do not take overly paid software to achieve this.

7.2 Relevant Subjects

Table 23: Relevant Subjects

Subject Code	Subject Name	Description
UML501	Machine Learning	This course provided us with the core technology for our project, covering the basics, evaluation metrics, model selection and Hyperparameter timings.
UCS531	Cloud Computing	This course provided an introduction to very prominent technology of cloud computing, covering ranging topics from cloud storage to hands-on lab practice, which was henceforth used in our project.
UCS503	Software Engineering	This course knowledge helped with the overall efficient functioning of the project, covering requirements to SRS, creating UML diagrams

		while providing the basis of our further work. This also provided us with testing techniques for our project's testing.
UCS761	Deep Learning	Knowledge of necessary deep learning models proved to be very helpful during the entire project. Provided us with much needed basics of neural networks, such as CNNs, RNNs and their architecture

7.3 Interdisciplinary Knowledge Sharing

Our project Tablytix provides users with redacted documents, table extraction and a query answering system based on the documents uploaded. Our project involved a variety of discipline knowledge from various subjects, and their combined knowledge played an essential role in our project development. Understanding different topics in the computer engineering discipline, which has multiple domains, was beneficial for the project's timely completion. The knowledge from Deep Learning was used to select the appropriate model for performing logo redaction and table extraction. Machine Learning further provided us with a basis for evaluation metrics, model selection and hyperparameter timings. Our solid foundation built studying cloud computing was utilized while deploying our whole backend on the cloud infrastructure. The Software Requirements Survey (SRS) was completed, and crucial diagrams (such as UML diagrams) were created using various concepts studied in Software Engineering. Knowledge of object-oriented programming again helped us write reusable code.

7.4 Peer Assessment Matrix

Table 24: Peer Assessment Matrix

	Evaluation of :					
Evaluation by:	Team Members	Ojas Sharma	Anshul Kanwar	Harneet Kaur	Agamjot Singh	Mayank Rawat

	Evaluation of :					
	Ojas Sharma	NA	5	5	5	5
	Anshul Kanwar	5	NA	5	5	5
	Harneet Kaur	5	5	NA	5	5
	Agamjot Singh	5	5	5	NA	5
	Mayank Rawat	5	5	5	5	NA

7.5 Role Playing and Work Schedule

Ojas Sharma: Analysis and trainings of deep learning models, Implementation of Query Answering System utilizing gpt4 NLP model, Analysis of various models for Table Extraction, Implementation of Table Extraction model, Unit Testings

Anshul Kanwar: Backend Designing, Model Integration, Unit Testings, Managing of Cloud services, Interface designing,

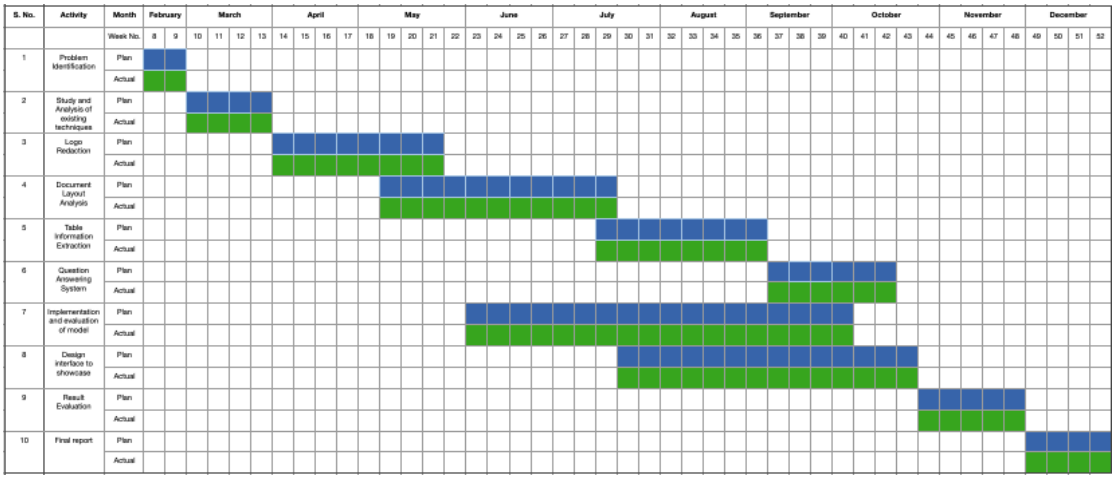
Harneet Kaur: Interface designing, Analysis of various models, Training and Implementation of detectron2 model for Document Layout Analysis, Documentation

Agamjot Singh: Handling User Authentications, Research findings, Pre-processing dataset of Logodet-3k, Model Integration, Documentation

Mayank Rawat: Pre-processing of dataset Visually 29k, Analysis of various models for Document Layout Analysis and Table Extraction, Training & Implementation of Resnet50 backbone for Logo Redaction, Black box Testings

Work Schedule of our Tablytix Project:

Figure 37: Gantt Chart



7.6 Student Outcomes Description and Performance Indicators (A-K Mapping)

Table 25: SO1-SO7 Mapping for the course 'UCS797-Capstone Project'

SO	SO Description	Outcome
1.1	Ability to identify and formulate problems related to the computational domain.	Identified and formulated different datasets and machine learning models used for Logo Redaction, Document Layout Analysis, Table Extraction and tested upon performance parameters like accuracy and latency.
2.1	Design computing system(s) to address needs in different problem domains and build prototypes, simulations, proof of concepts, wherever necessary, that meet design and implementation specifications.	Many models were tested for different parts of the project. The model which was finalized was the one which gave maximum accuracy and minimum time delay and hence for Logo Redaction we use the resnet50 backbone model. For Document Layout Analysis we used detectron model, and for table extraction we used a pdfplumber model. Finally for the query answering system we used the gpt4 nlp model.
2.2	Ability to analyze the economic trade-offs in computing systems.	Many platforms available are quite expensive and not so flexible with their implementation. QAGL provides a cost-effective method which users can use without any difficulties.
3.1		Documentation for the project was completed using

	Prepare and present a variety of/ documents such as project or laboratory reports according to computing standards and protocols.	specified formatting guidelines, and contained all the factual information as well as results inferred during the development of the project. Learned the ability to design, prepare and present documents like Software Requirements Specification and Project reports
3.3	Able to communicate effectively with peers in a well organized and logical manner using adequate technical knowledge to solve computational domain problems and issues.	Development of the project was done under complete guidance and approval of mentors and panel evaluators assigned to us for the duration of the capstone project. Our team put much attention to weekly meetings being held, as well as keeping in touch with our mentor on a monthly basis.
4.1	Aware of ethical and professional responsibilities while designing and implementing computing solutions and innovations.	We have considered privacy as the first priority. We imposed one to one unique connection with each user with the use of sessions. The User Authentication methods used in our project are up to date technologies, covering modern day security implementations.
4.3	Evaluate computational engineering solutions considering environmental, societal, and economic contexts.	With the help of Tablytix, businesses now can tap into the immense knowledge stored away in their digital documents. The logo Redaction allows the easy sharing of the documents between organizations, following the compliances set forth. This is economical as it allows for efficient collaborations. Environmental as the ML models used in this project are providing fast results, reducing unnecessary energy consumption. Given the useful insights hidden away in the tables, accessible only with immense effort, are now all so easily and efficiently available to businesses, also helping them gain those insights for the better implementations of their resources, bringing good to the whole society.
5.1	Participate in the development and selection of ideas to meet	This idea proceeded with us formulating the idea and with the help of our mentor, who helped in

	established objectives and goals.	adding more clarity, depth to this idea and guiding us what can be done and how. We read various research papers and articles giving us more insight on what we can do and we started implementing these ideas.
5.2	Able to plan, share and execute task responsibilities to function effectively by creating collaborative and/ inclusive environment in a team.	Throughout the development and implementation of the project, the whole team has learnt a lot of things, gained insightful experience and most importantly learnt the value of Teamwork. A lot of experience was gained on tackling challenges faced during the development of a project, which is surely to benefit us in our future.
6.1	Ability to perform experimentations and further/ analyze the obtained/ results.	Performed multiple testings and experimentations on the resources available in coherence to our project, and further analyzed and marked the performance based on the results obtained in the experimentations.
6.2	Ability to analyze and interpret data, make necessary judgment(s)/ and draw conclusion(s).	Observed, analyzed, and interpreted different Machine Learning models, and shortlisted them on the basis of performance parameters, and drew conclusions.
7.1	Able to explore and utilize resources to enhance self-learning.	This project helped us to apply our theoretical knowledge for practical purposes, and taught us about overcoming challenges in an organized manner. Gained the ability to make the effective use of resources available.

7.7 Brief Analytical Assessment

Q1. What sources of information did your team look into to come up with a list of potential project issues?

Ans: The entire team decided to research various project problems that could be worked on, and at the end of the week, meetings were held to discuss all the ideas. The decision was made to develop a project which caters to the Digital Documents

needs of the present time businesses, covering Logo Redaction for their privacy needs, Document Layout Analysis, and Table Extraction. We also wished to provide them with the facility of a query answering system aiming for efficient extracting of insights from the tables stored in the documents. Every member was asked to read various research papers and articles related to the field of Speech synthesis, including identification of various problems that would be encountered during the development of the system, in order to come up with solutions to these problems. After realizing the potential of the project we came up with many additional features for the same.

Q2. What analytical, computational, and/or experimental methodologies did your project team employ to find answers to the project's problems?

Ans: In Logo Redaction, we apply a variety of deep learning models. Firstly we deployed a variety of pre-processing methods for further processing of the documents of varying file types including pdfs, docs, ppts, pngs, and jpegs, to jpegs format. During these preprocessing we found converting all the document types into images worked best, and then we used a combination of ml models such as RCNN, Fast-RCNN, Faster-RCNN, detectron2 for logo redaction. Detectron2 was the fastest amongst them but losing on the accuracy part, hence after all the evaluations and comparisons of accuracy metrics of various models and their combinations. Our chosen model metrics included Intersection over Union (IoU) and mean average precision (mAP). Using these metrics we found the best results from the Resnet50 backbone implemented in the icevision framework. Next we had Document Layout Analysis, which is now a logo redacted image, and started with the analysis of the models, and settled on the Detectron model. Next we had its output passed for table extraction. This needed a lot of accuracy, as the outputs of this would be converted into JSON format and the NLP model would be using that to for the query answering system. After a long analysis and training of various models, we choose the one with the best accuracy PDFplumber to extract the tables and convert the table data into JSON format which is easily searchable and indexable. Now we had to choose one of the NLP models, and after careful training and analysis, we found gpt4 to be the best fit for table data.

Q3. Did the project demand demonstration of knowledge of fundamentals, scientific and/or engineering principles? If yes, how did you apply?

Ans: Yes, the project required us to apply a lot of theoretical knowledge that was studied during various courses attended in college. The project required a deep understanding of the Natural Language Processing models and neural networks. A vast understanding of deep learning models was required in order to properly train a neural network that would give accurate results during implementation. Each aspect of the project was thoroughly researched and evaluated before development began, to identify all the problems that would be faced during development, in order to come up with solutions to overcome them in an organized manner.

Q4. How did your team share responsibility and communicate the information of schedule with others in a team to coordinate design and manufacturing dependencies?

Ans: There was constant communication between all the team members throughout the development of the project. All the members of the team were regularly updated about new developments in the project, and we conducted weekly online meetings in order to assess our work and resolve issues faced by any team member, as well as decide on future developments. Constant communication was also maintained with our mentors, including conduction of monthly meetings with them to keep them updated on the project developments.

Q5. What resources did you use to learn new materials not taught in class for the course of the project?

Ans: There were many aspects of the project that had not been taught in college courses, and required the use of online resources and articles in order to implement. It required studying various articles and research papers in order to fully understand the functioning of the newer deep learning models, as well as the working and implementation of these machine learning models used in projects. Online research

was required on various advanced neural networks to implement them properly. The development of the user interface also required the use of online courses as well as several programming language documentations.

Q6. Does the project make you appreciate the need to solve problems in real life using engineering and could the project development make you proficient with software development tools and environments?

Ans: The aim of the project was to provide more accurate and least latency digital document web application catering to the needs of businesses around the globe, especially the smaller businesses which can't afford the heavy charges of for each individual functionality. Even non business users can also use our web application for their digital documents needs, to fulfill the compliances. The development of this project allowed us to apply various engineering principles to solve problems at every step. It allowed us to gain experience in developing projects on a large scale, and provided us the confidence to apply engineering techniques to solve real world problems in the future.

APPENDIX A: REFERENCE

[1] A. W. Harley, A. Ufkes and K. G. Derpanis, "Evaluation of deep convolutional nets for document image classification and retrieval", *Proceedings of the International Conference on Document Analysis and Recognition ICDAR*, vol. 2015-Novem, pp. 991-995, 2015.

[2] Simistira, Foteini, et al. "Diva-hisdb: A precisely annotated large dataset of challenging medieval manuscripts." *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2016.

[3] Yang, Xiao, et al. "Learning to extract semantic structure from documents using multimodal fully convolutional neural networks." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

[4] Zhong, Xu, Jianbin Tang, and Antonio Jimeno Yepes. "Publaynet: largest dataset ever for document layout analysis." 2019 International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2019.

[5] Li, Minghao, et al. "DocBank: A benchmark dataset for document layout analysis." arXiv preprint arXiv:2006.01038 (2020).

[6] Wang, Jing, et al. "Logo-2K+: A large-scale logo dataset for scalable logo classification." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. No. 04. 2020.

[7] Wang, Jing, et al. "Logodet-3k: A large-scale image dataset for logo detection." *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 18.1 (2022): 1-19.

[8] Su, H., Gong, S., & Zhu, X. (2017). Weblogo-2m: Scalable logo detection by deep learning from the web. Proceedings of the IEEE international conference on computer vision workshops, 270–279.

[9] Su, Hang, Xiatian Zhu, and Shaogang Gong. "Deep learning logo detection with data expansion by synthesizing context." *2017 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2017.

[10] Su, Hang, Xiatian Zhu, and Shaogang Gong. "Open logo detection challenge." *arXiv preprint arXiv:1807.01964* (2018).

[11] Jones, Michael, and Paul Viola. "Fast multi-view face detection." *Mitsubishi Electric Research Lab TR-20003-96* 3.14 (2003): 2.

[12] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR 05)*. Vol. 1. Ieee, 2005.

[13] Felzenszwalb, Pedro F., Ross B. Girshick, and David McAllester. "Cascade object detection with deformable part models." *2010 IEEE Computer society conference on computer vision and pattern recognition*. Ieee, 2010.

[14] Girshick, R. (2015). Fast r-cnn. Proceedings of the IEEE international conference on computer vision, 1440–1448.

[15] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.

[16] Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, 2117–2125.

[17] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. Proceedings of the IEEE international conference on computer vision, 2961–2969.

[18] Fu, C.-Y., Liu, W., Ranga, A., Tyagi, A., & Berg, A. C. (2017). Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659.

[19] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." *arXiv preprint arXiv:1804.02767* (2018).

[20] Schreiber, Sebastian, et al. "Deepdesrt: Deep learning for detection and structure recognition of tables in document images." 2017 14th IAPR international conference on document analysis and recognition (ICDAR). Vol. 1. IEEE, 2017.

[21] Bao, Hangbo, et al. "Beit: Bert pre-training of image transformers." *arXiv preprint arXiv:2106.08254* (2021).

[22] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, et al. “Rich feature hierarchies for accurate object detection and semantic segmentation.” *arXiv preprint arXiv:1311.2524 (2013)*

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollár, et al. “Focal Loss for Dense Object Detection.” *arXiv preprint arXiv:1708:02002*

Table 26: References for websites

S. No.	WEBSITE
[1]	Website : Redactable , Internet : https://www.redactable.com/blog/what-is-a-redaction
[2]	Website : Medium , Internet : https://medium.com/analytics-vidhya/table-extraction-using-deep-learning-3c91790aa200
[3]	Website : Medium , Internet : https://medium.com/analytics-vidhya/tablenet-deep-learning-model-for-end-to-end-table-detection-and-tabular-data-extraction-from-1961fb2f97e1
[4]	Website : Nanonet , Internet : https://nanonets.com/blog/table-extraction-deep-learning/
[5]	Website : ResearchGate , Internet : https://www.researchgate.net/publication/221253882_Information_leakage_through_document_redaction_Attacks_and_countermeasures

APPENDIX B: Plagiarism Report