TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS

A
PROJECT REPORT
ON

INFORMATION EXTRACTION FROM STRUCTURED DOCUMENT

SUBMITTED BY:

AAYUSH SHAH KANU (PUL075BCT007)
ADITHYA POKHREL(PUL075BCT009)
BISHAL BASHYAL(PUL075BCT026)
JANAK SHARMA (PUL075BCT040)

SUBMITTED TO:

DEPARTMENT OF ELECTRONICS & COMPUTER ENGINEERING

30TH APRIL, 2023

Page of Approval

TRIBHUVAN UNIVERSIY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS
DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

The undersigned certifies that they have read and recommended to the Institute of Engineering for acceptance of a project report entitled **"Information Extraction From Structured Document"** submitted by **Aaysh Shah Kanu**, **Adithya Pokharel**, **Janak Sharma**, **Bishal Bashyal** in partial fulfillment of the requirements for the Bachelor's degree in Electronics & Computer Engineering.

....
Supervisor
Sarad Kumar Ghimire
Associate Professor
Department of Electronics and Computer Engineering,
Pulchowk Campus, IOE, TU.

....
Internal examiner
Person B
Assistant Professor
Department of Electronics and Computer Engineering,
Pulchowk Campus, IOE, TU.

....
External examiner
Person C
Assistant Professor
Department of Electronics and Computer Engineering,
Pulchowk Campus, IOE, TU.

Date of approval:

Copyright

Acknowledgments

We would like to express our deepest appreciation to **Associate Professor. Sharad Kumar Ghimire**, our project supervisor, for his invaluable guidance and unwavering support throughout the entire duration of the project.

We are also grateful to **Docsumo Nepal** for providing us with the opportunity to work on a real-world project and gain valuable experience in the field. We would like to extend our thanks to the **Institute of Engineering, Pulchowk Campus**, for providing us with the platform to pursue our goals and aspirations.

We would like to express our sincere gratitude to the lecturers of our department, whose unwavering support and inspiration have been instrumental in guiding us through every stage of this project. Their expert guidance and mentorship have been invaluable in helping us develop our skills and abilities, and we are deeply grateful for their contributions.

Last but not the least, we extend our heartfelt thanks to everyone who has contributed to this project, directly or indirectly, and helped us achieve our goals.

Abstract

This project proposes the use of the LayoutLMv2 model, a deep learning model, for information extraction from form-like documents. Specifically, the IRS 990 tax form was used as the dataset for testing and optimization. The information extraction process from form-like documents can be challenging due to the complex layout analysis and text recognition required to identify fields and corresponding values. The proposed model, LayoutLMv2, has demonstrated its effectiveness in these tasks, making it a promising solution for information extraction from form-like documents. The project resulted in the development of a web application and annotation tools that provide users with a user-friendly interface to upload documents and extract relevant information accurately and efficiently. The annotation tool enables users to label data and train custom models, while the web application streamlines document processing for businesses and organizations.

Keywords: *Transformers, OCR engine, LayoutLMv2,Information Extraction*

Contents

List of Figures

List of Tables

List of Abbreviations

NLP Natural Language Processing

JASPER Joint Actinide Shock Physics Experimental Research

IE Information Extraction

MUCs Message Understanding Competitions

NOSC Naval Ocean Systems Centre

DARPA Defence Advanced Research Project Agency Centre

OCR Optical Character Recognition

CNN Convolutional Neural Network

RNN Recurrent Neural Network

LSTM Long Short-Term Memory

PDF Portable Document Format

VGG Visual Geometry Group

ResNet Residual Neural Network

JSON JavaScript Object Notation

RPN Region Proposal Network

DLA Document Layout Analysis

BERT Bidirectional Encoder Representations from Transformers

RPN Region Proposal Network

CV Computer Vision

SVM Support Vector Machines

1. Introduction

Manual extraction of information from documents can be a time-consuming and resource-intensive task. However, automated information extraction systems aim to locate and interpret specific pieces of relevant information from a large number of data sources, which can include text or images. By producing a structured representation of this information, it can be efficiently processed and analyzed to generate valuable insights.

Various types of documents, such as forms, receipts, bills, and insurance quotes, play a critical role in diverse business workflows. An automated workflow can drastically reduce the need for costly human resources and minimize errors. Additionally, implementing such a system can ensure a higher level of data security since information is not accessible to unauthorized parties.

1.1 Background

The task of extracting information from form-like documents such as bills, invoices, and forms can be time-consuming and prone to errors. However, it is an essential activity that is performed regularly to extract valuable data and insights. Manual extraction of this information can lead to inefficiencies and errors, ultimately affecting productivity and efficiency.

To address this challenge, we offer a software platform that leverages the power of deep learning to automate the extraction of data from documents and deliver it in a structured format. By automating this process, organizations can save time, reduce errors, and increase productivity, allowing them to focus on more critical activities.

Our software platform is designed to meet the needs of modern organizations. It is highly customizable, easy to integrate, and can be adapted to meet the unique needs of different industries and business processes. With our platform, organizations can gain valuable insights from their data faster and more accurately, enabling them to make better decisions and stay ahead of the competition.

1.2 Problem statements

Manually updating form data is a time-consuming and labor-intensive task for companies, financial institutions, institutions, and banks. The process of digitizing printed structured data forms into a digital database requires a significant investment of resources and capital. Our automated solution aims to optimize this process by reducing the need for physical labor and financial investment.

1.3 Objectives

The objectives of the project are:

1. To be able to locate the important features of the document
2. To extract text from the localized features
3. To map the extracted text to key entities

1.4 Scope

The scope of information extraction from structured data is vast and varied, with applications across numerous industries and domains. In the healthcare sector, it is used for processing medical records, lab reports, and clinical trial data. In the finance and accounting sector, it can be used for automating data population from invoices, receipts, and financial statements. It is also used in the legal industry for document discovery and analysis, as well as in customer service for sentiment analysis and chatbot training.

By automating the process of data extraction, organizations can significantly reduce the time and resources required for manual data entry and processing. This can result in a significant increase in efficiency and productivity, allowing employees to focus on higher-level tasks and decision-making processes. Furthermore, automated data extraction can help organizations to minimize errors and inconsistencies in data processing, leading to more accurate and reliable insights.

2. Literature Review

Information extraction dates back to the late 1970s in the early days of Natural Language Processing (NLP)[4]. An early commercial system from the mid-1980s was JASPER built for Reuters by the Carnegie Group Inc with the aim of providing real-time financial news to financial traders.

The present significance of IE pertains to the growing amount of information available in structured and unstructured form. Tim Berners-Lee, inventor of the World Wide Web, refers to the existing Internet as the web of documents [5] and advocates that more of the content be made available as a web of data. Message Understanding Competitions (MUCs) have played an important role in the development of information extraction as a field of study. This conference was initiated by the Naval Ocean Systems Centre (NOSC) and was sponsored by the Defence Advanced Research Project Agency Centre (DARPA). MUCs took place seven times from 1987 until 1998.

Unlike traditional OCR techniques, CNN, RNN and LSTM are achieving high performance in text recognition in images. Deep learning techniques are showing prevalent results to date. CNN as feature extractor to detect, slice and recognize pipeline [6] and as encoder in attention mechanism outperformed others.[7]

2.1 Related Works

Rule Based algorithms were prominent historically in the Document Understanding. In the paper [8] the author presents the method that is based on bottom up approach to document analysis. It is based on linking of characters together to form blocks which are further segmented, labelled and merged into paragraphs.Simulataneously, graphics are extracted from image.

More recently , the gear has shifted towards using the recent Computer Vision (CV) and NLP methods especially in 2010s.

In [9] the authors review different techniques for document understanding for documents written in English and concludes that Document understanding is a valuable but understudied field due to the lack of publicly available datasets. However, recent advancements in deep neural network modeling have made end-to-end document understanding achievable by integrating layout analysis, optical character recognition, and domain-specific information extraction.

Some of the document understanding tasks that are relevant for our project are explained below:

2.1.1 Document Layout Analysis

Document Layout Analysis involves identifying regions of interest in a document, which can range from page segmentation to semantic classification. While rule-based approaches [8] have been used since the 1990s, recent advances in machine learning and multi-modal transformer architectures such as LayoutLMv3 [10] have enabled more sophisticated DLA.

2.1.2 OCR

Optical Character Recognition (OCR) is the electronic or mechanical conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo (for example the text on signs and billboards in a landscape photo) or from subtitle text superimposed on an image (for example: from a television broadcast). [1]

According to [9] OCR has two primary components: text detection and text transcription.

[1] https://www.wikiwand.com/en/Optical_character_recognition

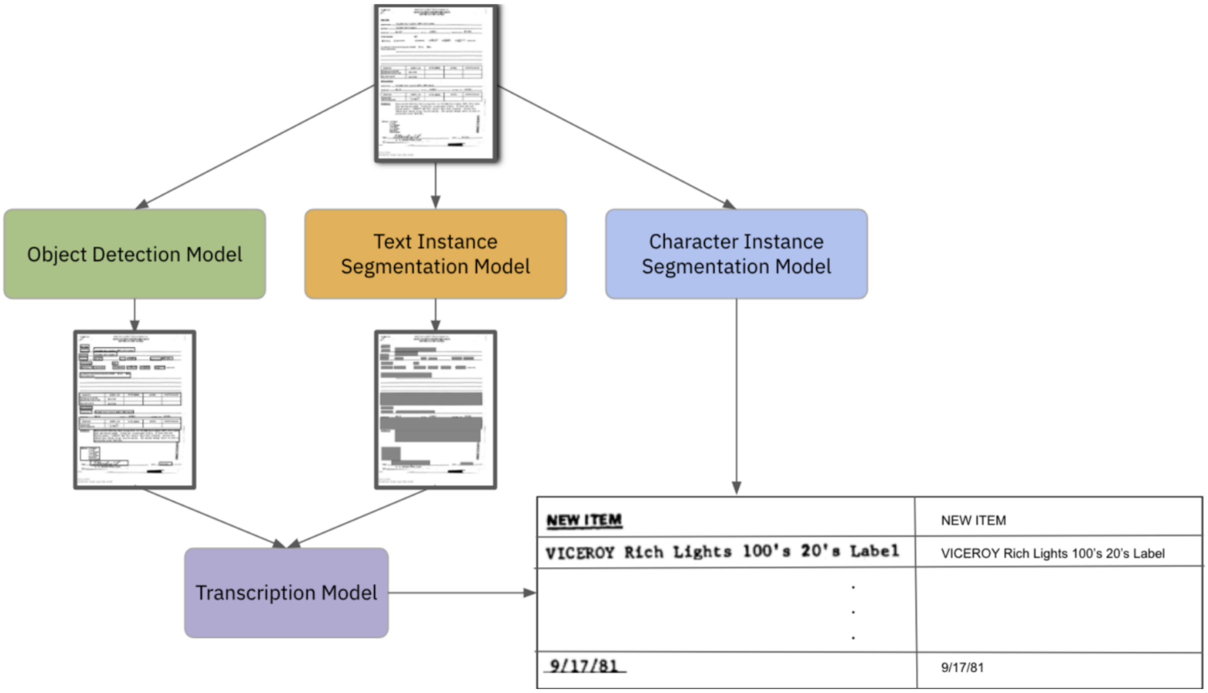

Object Detection Model
Text Instance Segmentation Model
Character Instance Segmentation Model
Transcription Model
NEW ITEM
NEW ITEM
VICEROY Rich Lights 100's 20's Label
VICEROY Rich Lights 100's 20's Label
9/17/81
9/17/81

Figure 2.1: General OCR Process

The general OCR process is shown in Figure 2.1.The document can take any path and all paths produce the same structured output [9]. The document comes from FUNSD [11]

2.1.3 Rule Based Methods

Rule Based Methods could be broadly classified into top-down, bottom-up and hybrid methods.[12] Bottom-up methods are commonly used to detect the basic computational units in document images, which are usually the connected components of black pixels. The goal of the document segmentation process is to combine these components into higher-level structures using various heuristics and label them based on different structural features.

One of the earliest successful bottom-up algorithms that utilizes connected component analysis is the Docstrum algorithm, as described by [13] in 1993. This algorithm groups connected components on a polar structure to derive the final segmentation.

When utilizing a top-down approach, the main goal is to divide the document into blocks in a recursive manner. To achieve this, various methods are employed, one of which is the X-Y cut algorithm, as presented in [14]. In this algorithm, the authors utilize the X-Y cut technique to divide the document into blocks.

A hybrid methods combines both of the methods mentioned previously.[15] uses this approach.

2.1.4 Machine Learning Methods

In the past decade, statistical machine learning approaches have become the mainstream for document segmentation tasks, in conjunction with the development of conventional machine learning. One notable approach, outlined in [16], considers document layout information as a parsing problem and searches for the optimal parsing tree based on a grammar-based loss function. This approach employs a machine learning technique to select features and train all parameters during the parsing process.

Meanwhile,Artificial Neural Networks(ANNs) have also been extensively utilized for document analysis and recognition, with significant success in recognizing isolated handwritten and printed characters [17].

In addition to ANNs, support vector machines (SVMs) and Gaussian mixture models (GMMs) have been utilized in document layout analysis tasks. Machine learning approaches can be time-consuming to design manually crafted features, and it is challenging to obtain highly abstract semantic context. Moreover, these methods often rely on visual cues and overlook textual information.

2.1.5 Deep Learning Methods

In recent times, deep learning techniques have gained widespread popularity and are now widely accepted as the standard approach for solving many machine learning problems. These methods are based on the concept of multi-layer neural networks, which have the ability to approximate arbitrary functions. Numerous studies have demonstrated the efficacy of deep learning across a variety of research fields.

There has been huge advancements in multistage models such as multistage Faster R-CNN [18] and single-stage models such as YOLO [19] Recent studies have attempted to enhance the representation of multi-modal data and address data limitations. For instance, LayoutLM [12] was developed as a pre-trained transformer model that is based on BERT [20] and is specifically designed for document understanding. This model combines textual information, layout information, and image embeddings using Faster R-CNN to improve its performance on various downstream tasks. By integrating these different types of data, LayoutLM provides a more comprehensive representation of documents, which leads to better results in real-world applications.

2.2 Related Theory

2.2.1 Natural Language Processing(NLP)

Natural language processing (NLP) is a field that combines linguistics, computer science, and artificial intelligence to explore how computers and humans can interact through language. Its primary aim is to develop programs that enable computers to process and analyze vast amounts of natural language data in a way that mimics human comprehension. In other words, NLP seeks to create a computer that can understand the meaning of documents and the subtle nuances of language that give them context. By doing so, NLP can help extract valuable insights and information from documents while also categorizing and organizing them more effectively.

2.2.2 Neural Network

A neural network is a type of machine learning model that is designed to learn from data by adjusting its internal parameters in response to the input. The architecture of a neural network consists of multiple layers of interconnected nodes, called neurons, which receive inputs and produce outputs. Each neuron is typically connected to several other neurons in the previous layer, and the outputs from the previous layer are used to calculate the inputs to the next layer. The parameters of the network, including the weights and biases of each neuron, are adjusted using a process called backpropagation, which minimizes the difference between the predicted outputs and the actual outputs for a given set of inputs.

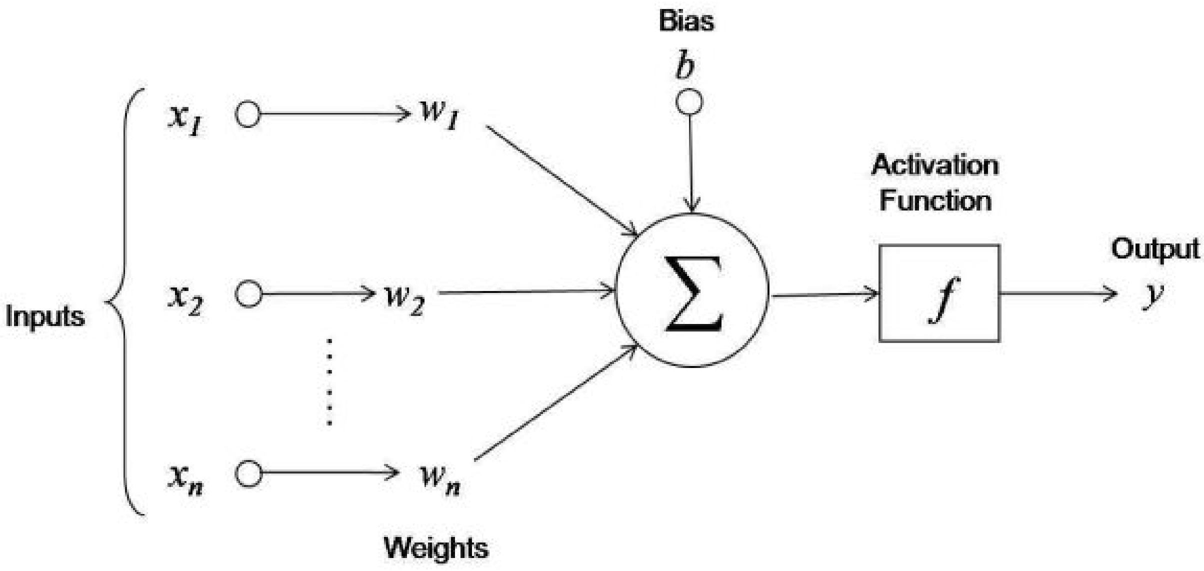


Figure 2.2: Architecture of Neuron

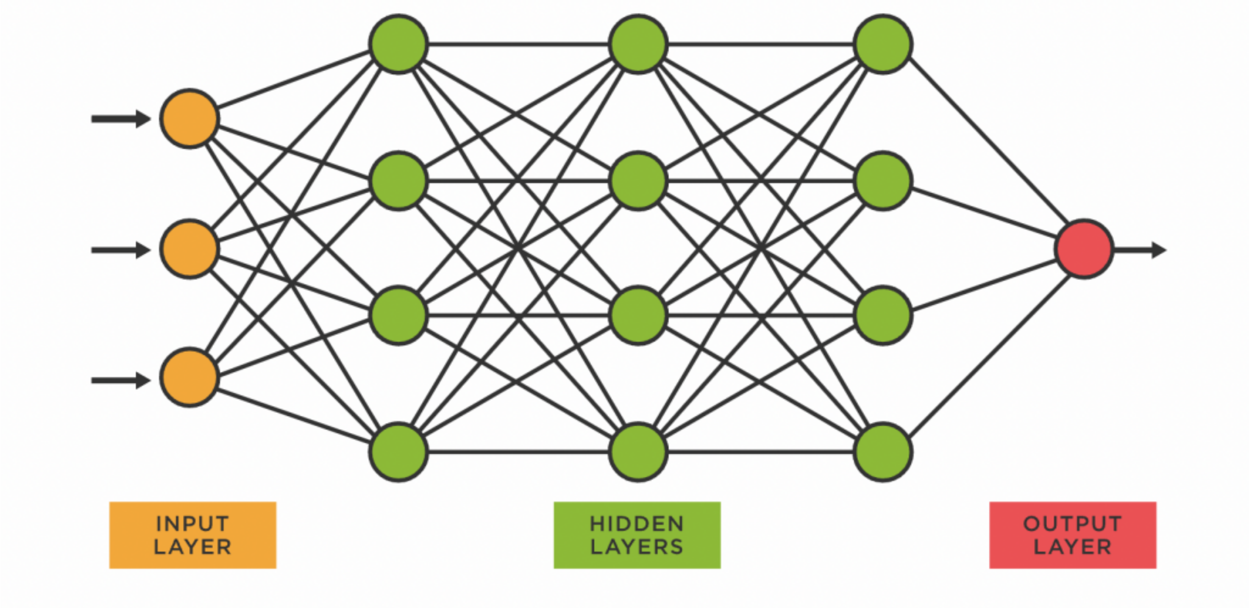


Figure 2.3: Architecture of Neural Network

Neural networks are commonly used for tasks such as image and speech recognition, natural language processing, and predictive analytics. They are highly adaptable and can learn to recognize complex patterns and relationships in the input data, even when these patterns are not immediately apparent to humans. This ability to learn and generalize from examples makes neural networks highly effective for a wide range of applications.

2.2.3 Convolutional Neural Network Architecture

CNN is a neural architecture that applies sliding window over the image to capture the spatial relation and context over that window. The sliding window works as filters for spatial representation to understand the feature of the spatial data. This architecture includes Convolutional layers connected to Pooling and Normalization layers for better feature extraction. For classification, the feature extraction portion of CNN is used with Fully Connected Neural Layer and softmax layer.

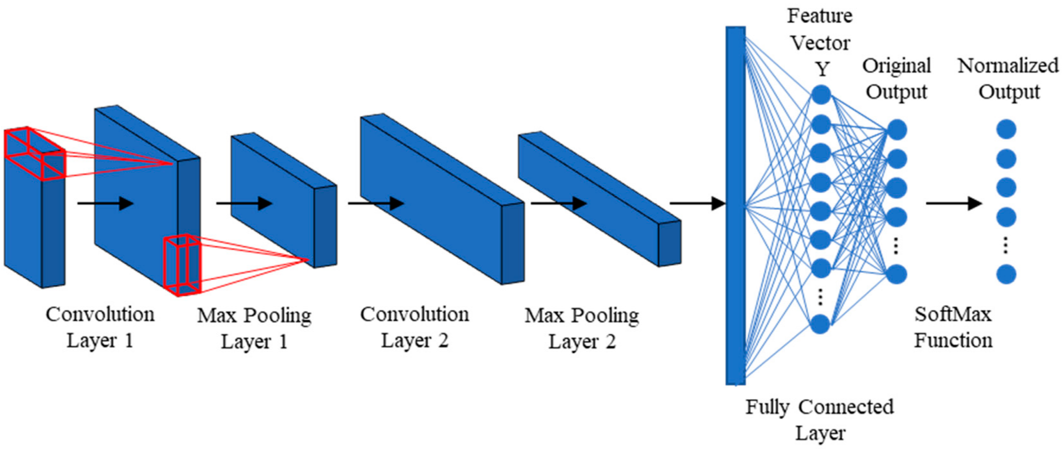


Figure 2.4: One-dimensional convolutional neural network (1D CNN) model [1]

2.2.4 Transformers

A transformer is a deep learning model that adopts the mechanism of self-attention, differentially weighting the significance of each part of the input data. It is used primarily in the fields of NLP and computer vision.[2] Transformers were introduced in 2017 by a team at Google Brain [2] and are increasingly the model of choice for NLP problems, replacing Recurrent Neural Network (RNN) models such as Long Short-Term Memory (LSTM).

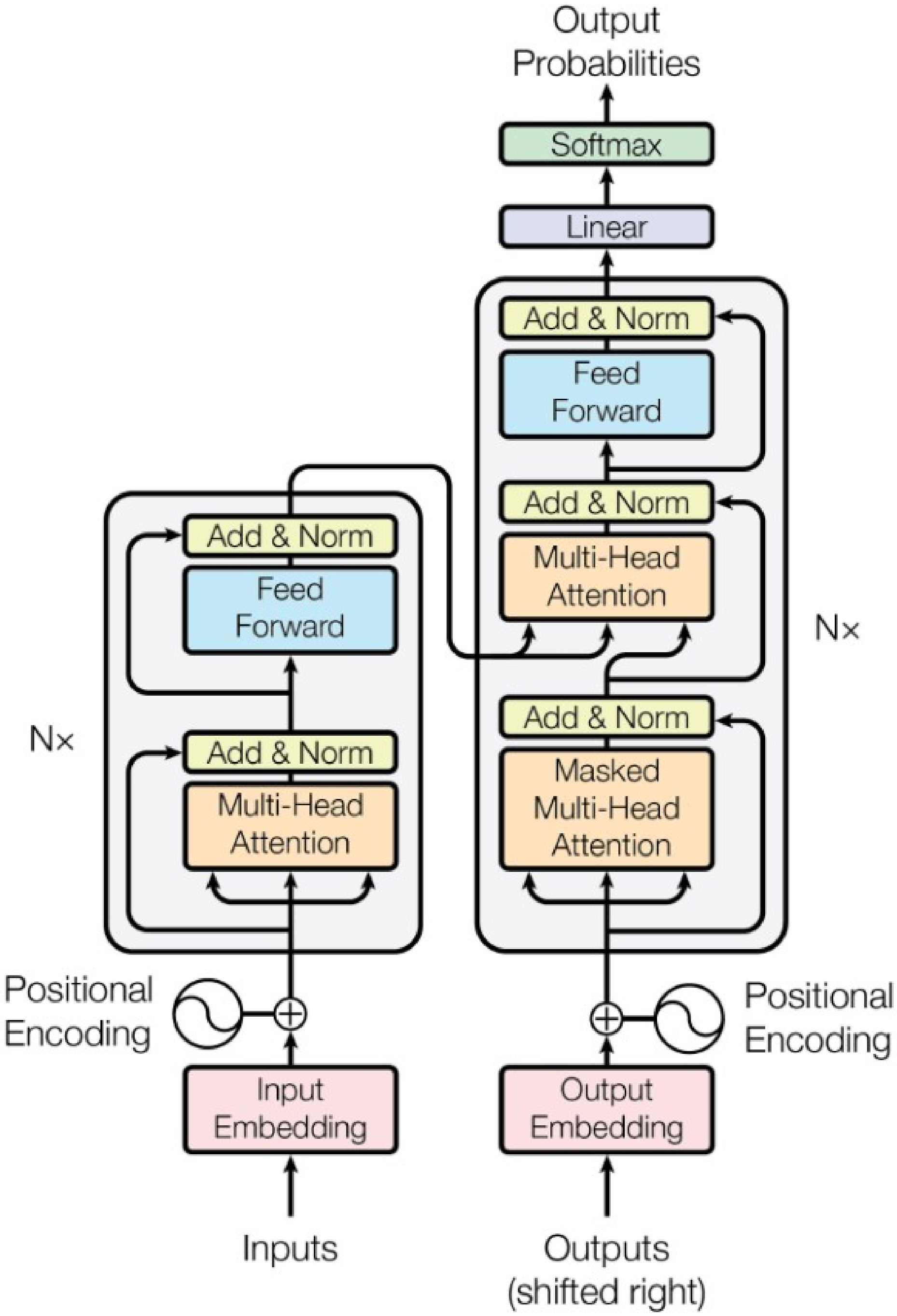


Figure 2.5: Transformer Architecture [2]

Encoder

Each encoder consists of two major components: a self-attention mechanism and a feed-forward neural network. The self-attention mechanism accepts input encodings from the previous encoder and weighs their relevance to each other to generate output encodings.

The feed-forward neural network further processes each output encoding individually. These output encodings are then passed to the next encoder as its input, as well as to the decoders.The first encoder takes positional information and embeddings of the input sequence as its input, rather than encodings. The positional information is necessary for the transformer to make use of the order of the sequence, because no other part of the transformer makes use of this.[2]

Decoder

Each decoder consists of three major components: a self-attention mechanism, an attention mechanism over the encodings, and a feed-forward neural network. The decoder functions in a similar fashion to the encoder, but an additional attention mechanism is inserted which instead draws relevant information from the encodings generated by the encoders.[2]

2.2.5 Embeddings

In natural language processing, embeddings are a way to represent words or phrases as vectors in a high-dimensional space. These vectors capture the semantic and syntactic relationships between the words, allowing for computational models to better understand the meaning of text. Embeddings are created through a process called ”word embedding,” where words are transformed into dense numerical vectors. There are several types of word embeddings, including one-hot encoding, frequency-based embeddings, and context-based embeddings.

One-hot encoding represents words as sparse vectors where each dimension corresponds to a word in the vocabulary, and only one dimension has a value of one while the others are zero. Frequency-based embeddings assign a numerical value to each word based on its frequency in the corpus. Context-based embeddings use neural networks to learn a mapping between words and their surrounding context, which captures the semantic and syntactic relationships between the words.

2.2.6 BERT

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language model developed by Google in 2018 for natural language processing tasks. It is based on the transformer architecture, which was introduced in [2]

BERT is a deep neural network that is trained on large amounts of unannotated text data to generate high-quality embeddings for various natural language processing tasks. Unlike previous models, BERT is bidirectional, which means that it takes into account both the left and right context of each word in a sentence.

BERT is trained using two tasks: masked language modeling and next sentence prediction. In masked language modeling, the model is trained to predict a randomly masked word within a sentence, while in next sentence prediction, the model is trained to predict whether two sentences are consecutive or not. By training on these tasks, BERT is able to capture the semantic relationships between words and understand the context in which they are used.

BERT has achieved state-of-the-art performance on a wide range of natural language processing tasks, including question answering, sentiment analysis, and text classification. It has also paved the way for other pre-trained language models, such as GPT-2 and RoBERTa, which have further improved the state-of-the-art performance on various language tasks.

2.2.7 LayoutLM

LayoutLM is a deep learning model that combines text recognition and layout analysis to improve the performance of natural language processing (NLP) tasks on documents with complex layouts, such as forms, receipts, and invoices. The model is based on the popular BERT architecture, which uses a transformer network to encode input text into a contextualized representation. However, LayoutLM extends BERT by also encoding spatial information about the position of words and their relationships within the layout of the document.

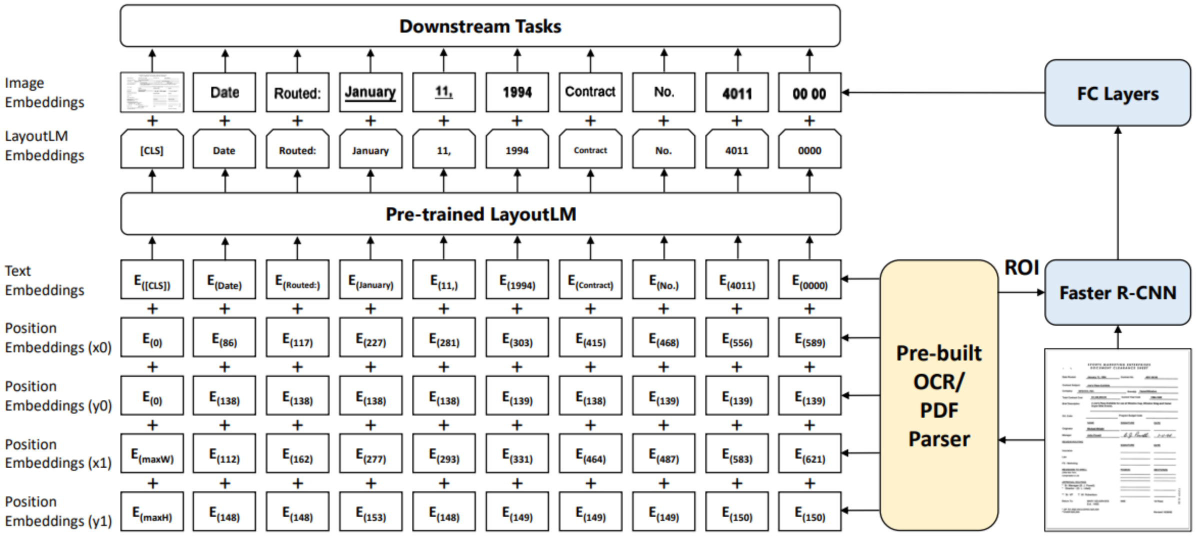


Figure 2.6: Architecture of LayoutLMv1

To achieve this, LayoutLM uses a two-stage approach. In the first stage, the model processes the input document as an image to extract layout features, such as bounding boxes, coordinates, and visual features of text and non-text elements. In the second stage, the model processes the text within each bounding box and combines the layout features with the BERT-encoded text to create a final representation.

LayoutLM can be trained on a variety of tasks, such as named entity recognition, information extraction, and document classification, and has shown to outperform traditional NLP models on tasks that require understanding of both textual and spatial information. Additionally, the model can be fine-tuned on specific document types to further improve performance.

2.2.8 LayoutLMv2

LayoutLMv2[3] is the improved version of LayoutLM which incorporates visual information in the pre-training stage through a spatial-aware self-attention mechanism. The model uses a 2-D relative position representation for token pairs to provide a broader view for contextual spatial modeling. In addition, two new training objectives are used, including a text-image alignment strategy and a text-image matching strategy, to further improve performance. These modifications help LayoutLMv2 achieve better accuracy and correlation between document images and textual content compared to the original LayoutLM model.

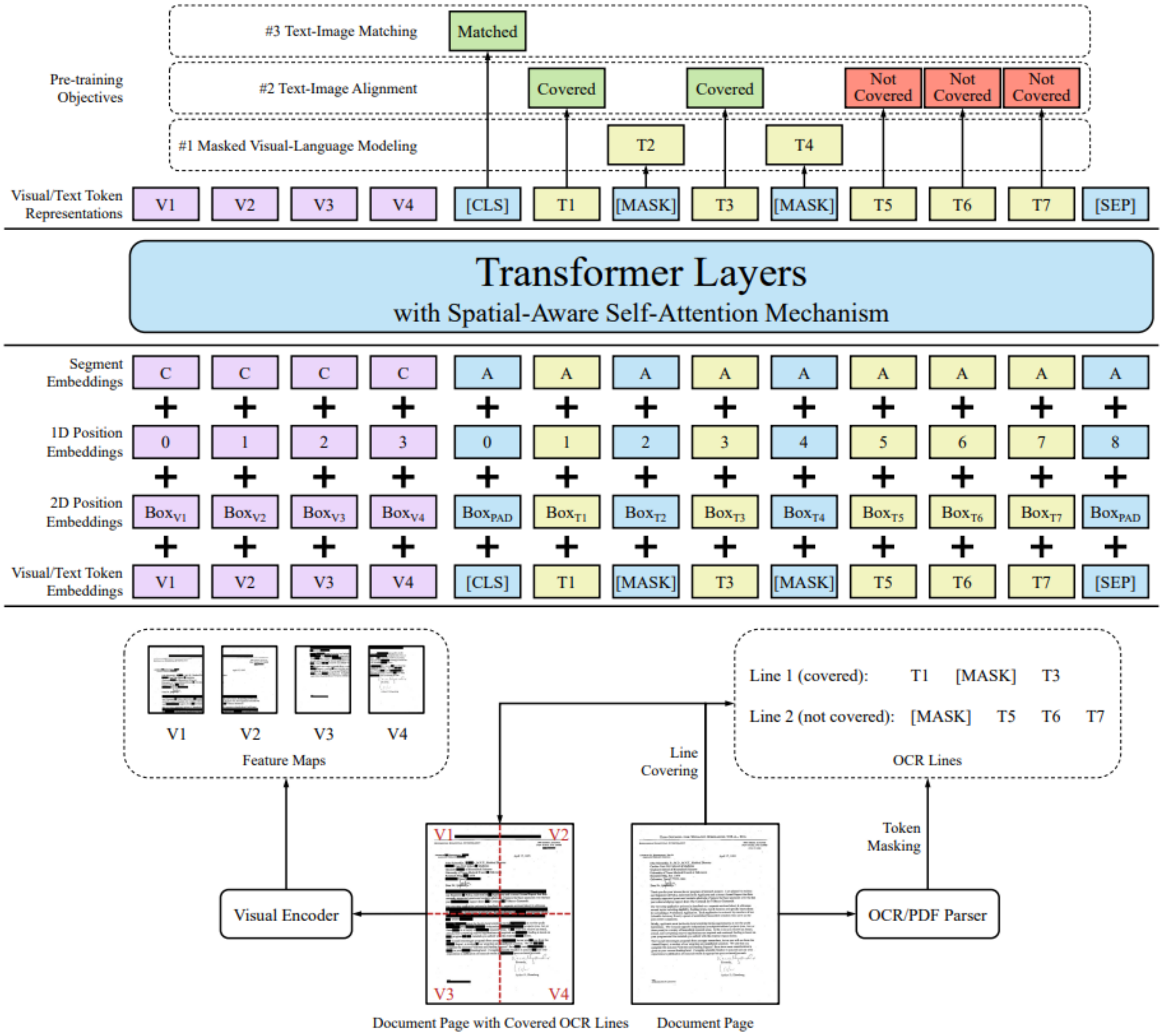


Figure 2.7: An illustration of the model architecture and pre-training stragegies for LayoutLMv2 [3]

The model uses three types of Embeddings which are explained below:

Text Embedding

In LayoutLMv2, the text embedding is made up of three components that are combined using addition. The first component is the token embedding which is the actual text of the token. The second component is the positional embedding which represents the position of the token in the sequence. The third component is the segment embedding which is used to differentiate between text and visual tokens. The model uses the segment embedding to identify the type of token in the sequence. The text sequences are of equal length and shorter sequences are padded with additional tokens. The text embedding for a given word in the sequence is defined by an equation 2.1 that includes the word itself, its segment embedding, and the maximum length of the sequence.

$$t_i = \text{TokEmb}(w_i) + \text{PosEmb1D}(i) + \text{SegEmb}(s_i),\ 0 \leq i < L \tag{2.1}$$

Visual Embedding

In LayoutLMv2, the visual embedding is produced using ResNeXt-FPN [21, 22] as the backbone. For each document page, a feature map is generated with a resolution of 224x224. This assumes that all pages have the same aspect ratio, such as the commonly used A4 format. However, since A4 is not a square, the receptive field is not symmetrical in the x and y directions. To match the dimensionality of the text embeddings, a linear projection layer is applied to the visual embedding sequence. The visual embedding sequence is represented by the visual token embeddings, the positional embedding, and the segment embedding. The visual embedding for each token in the visual embedding sequence is defined by an equation 2.2 that includes the visual token embedding, the index of the token (positional embedding), and the segment embedding. Since the CNN-based backbone does not capture positional information, a positional embedding $PosEmb1D$ is added to the visual embedding of each token. Finally, the segment embedding is set to [C] since all visual embeddings are visual segments.

$$v_i = \text{Proj}(\text{VisTokEmb}(I)_i) + \text{PosEmb1D}(i) + \text{SegEmb}([C]), \quad 0 \leq i < WH \tag{2.2}$$

Layout Embedding

A 2D layout embedding sequence is constructed using existing information from documents. The bounding box for each word is described using six parameters where $x - min$, $x - max$, $y - min$ and $y - max$ are corner coordinates and width and height are sizes. The bouding box coordinates are all normalized to integers in the range $[0, 1000]$. In $PosEmb2D$ the three parameters for each direction are concatenated and two embeddings are obtained, one for the $x - direction$ and one for the $y - direction$. The two embeddings are then concatenated into a resulting layout token embedding shown in Equation 2.3

$$l_i = \text{Concat}(\text{PosEmb2D}_x(x_{min}, x_{max}, width), \text{PosEmb2D}_y(y_{min}, y_{max}, height)) \tag{2.3}$$

Multi-model Encoder with Spatial-Aware Self-Attention Mechanism

The encoder consists of two parts. The first is the input sequence that contains both visual and text embeddings for each sequence

$$\text{InputSequence} = \text{Concat}(\text{VisualEmbedding}, \text{TextEmbedding}) \tag{2.4}$$

As for any Transformer network this sequence is then used in the self-attention layer. Furthermore, after obtaining a self-attention map a, a spatially aware self-attention mechanism is used. In this phase, the layout embeddings are used to update the self-attention map with the layout information. This is achieved by taking the previous attention score and adding the positional 1D embedding bias plus the 2D embeddings biases in $x - direction$ and $y - direction$ respectively as shown in Equation 2.5. In equation 2.5 a'_ij denotes the updated attention score in the self-attention map for query i and key j. x_i and y_i denote the top-left corner coordinates for the i:th bounding box, and x_j and y_j denote the j:th key-query pair in the self-attention map. To avoid adding too many parameters, each embedding is represented as a bias term b.

$$a'ij = aij + \mathbf{b}^{(1D)}_{(j-i)} + \mathbf{b}^{(2D_x)}_{(x_j-x_i)} + \mathbf{b}^{(2D_y)}_{(y_j-y_i)} \tag{2.5}$$

In the end, the output vectors are obtained by taking a weighted average of all the projected value vectors. The weights are determined by normalized spatial-aware attention scores, which reflect the importance of each vector in the context of the input data. Essentially, the algorithm is using these attention scores to focus on the most relevant parts of the input data and combining them to produce the output vectors.

$$h_i = \sum_j \frac{\exp(\alpha'_{ij})}{\sum_k \exp(\alpha'_{ik})} x_j \mathbf{W}^V \tag{2.6}$$

2.2.9 ResNeXt

ResNeXt is a variant of the popular ResNet architecture that extends the basic ResNet block with a split-transform-merge operation. This operation enables ResNeXt to capture more diverse and fine-grained features by increasing the network's representational power.

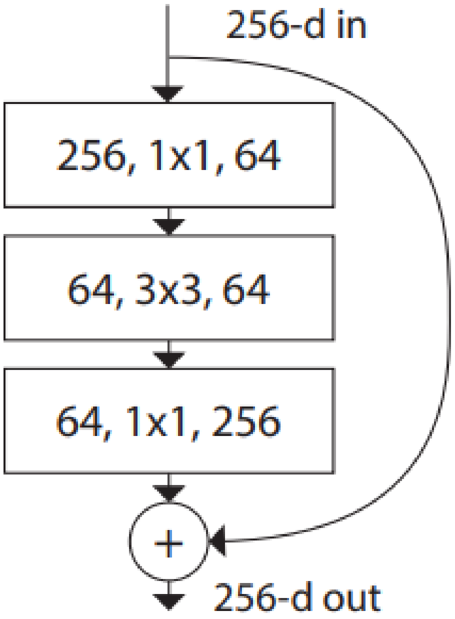


Figure 2.8: A block of ResNet

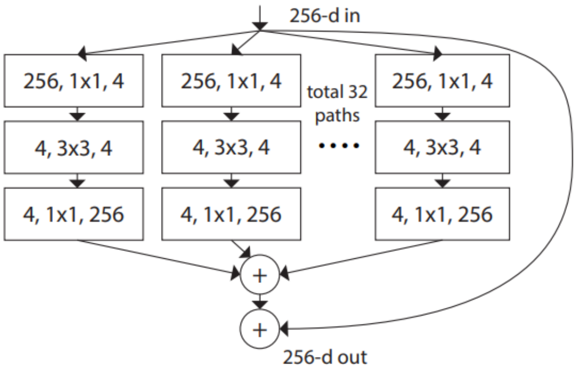


Figure 2.9: A block of ResNeXt with cardinality = 32

In contrast to ResNet, where the focus is on increasing depth to improve performance, ResNeXt achieves better performance by increasing the width of the network. Instead of using a single path to learn features, ResNeXt splits the input into multiple paths or cardinality, each processing a subset of the input. The outputs of these paths are then combined through a summation operation.

This cardinality parameter allows ResNeXt to increase the width of the network without adding too much computational cost.

2.2.10 FPN

Feature Pyramid Network (FPN) [22] is a technique for building deep neural networks that can effectively detect objects at different scales in an image.

The basic idea behind FPN is to combine feature maps from different layers of a network to create a pyramid of features, with each level in the pyramid corresponding to a different scale of the input image. This pyramid structure allows the network to detect objects at different scales in the image, which is essential for many computer vision tasks such as object detection and segmentation.

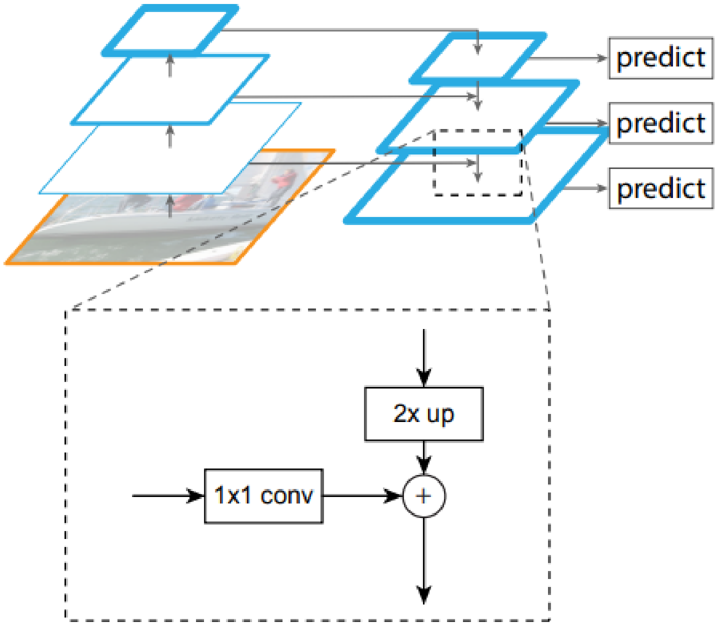


Figure 2.10: Feature Pyramid Network

To create the feature pyramid, FPN first takes the highest resolution feature map from the network and applies a 1x1 convolution to it to reduce the number of channels. This feature map is then upsampled and combined with the feature map from the layer immediately below it, creating a new feature map that contains information from both layers. This process is repeated to create the next level of the pyramid, and so on until the desired number of levels is reached.

The resulting feature pyramid can be used for a variety of tasks. For example, in object detection, the pyramid is used to detect objects of different sizes by matching them to features at different levels of the pyramid. In semantic segmentation, the pyramid is used to generate a dense prediction by upsampling the final feature map to the original input resolution.

3. Methodology

3.1 Research

This project's research has been crucial, accounting for more than half of the total effort expended. Our primary objective was to develop a more efficient method of extracting information from financial documents, specifically Form 990s. To achieve this goal, we began by studying various research papers to better understand the current state of the art in document extraction.

Our initial approach was rule-based since the Form 990s were supposed to follow a fixed format. However, we soon realized that the forms had variations in dimensions and content, which made this approach unreliable.

Our second approach was entity linking, which aimed to identify and link entities such as names, addresses, and financial figures in the forms. However, the extracted information may not be in standard form. For example, some of the words in the key might be missing or some keys may have large blocks of text while the value might be a single word. These discrepancies can greatly affect the entity linking process.

Finally, we settled on a third and current approach using token classification. For this, we used the LayoutLMv2 model, which allowed us to train our system to identify relevant tokens in the financial documents accurately. This approach has shown promising results and has significantly increased the efficiency of extracting information from Form 990s.

To extract information from Form 990s more efficiently, we utilized the LayoutLMv2 model. Initially, we trained the model on the FUNSD dataset, which contains annotated forms from various domains. The results of the model were analyzed to determine its performance in extracting key fields such as the organization's name, address, and financial information from the FUNSD dataset. Afterward, we trained the same base LayoutLMv2 model on the Form 990s dataset to make it more specific to our use case.

3.2 Dataset Acquisition

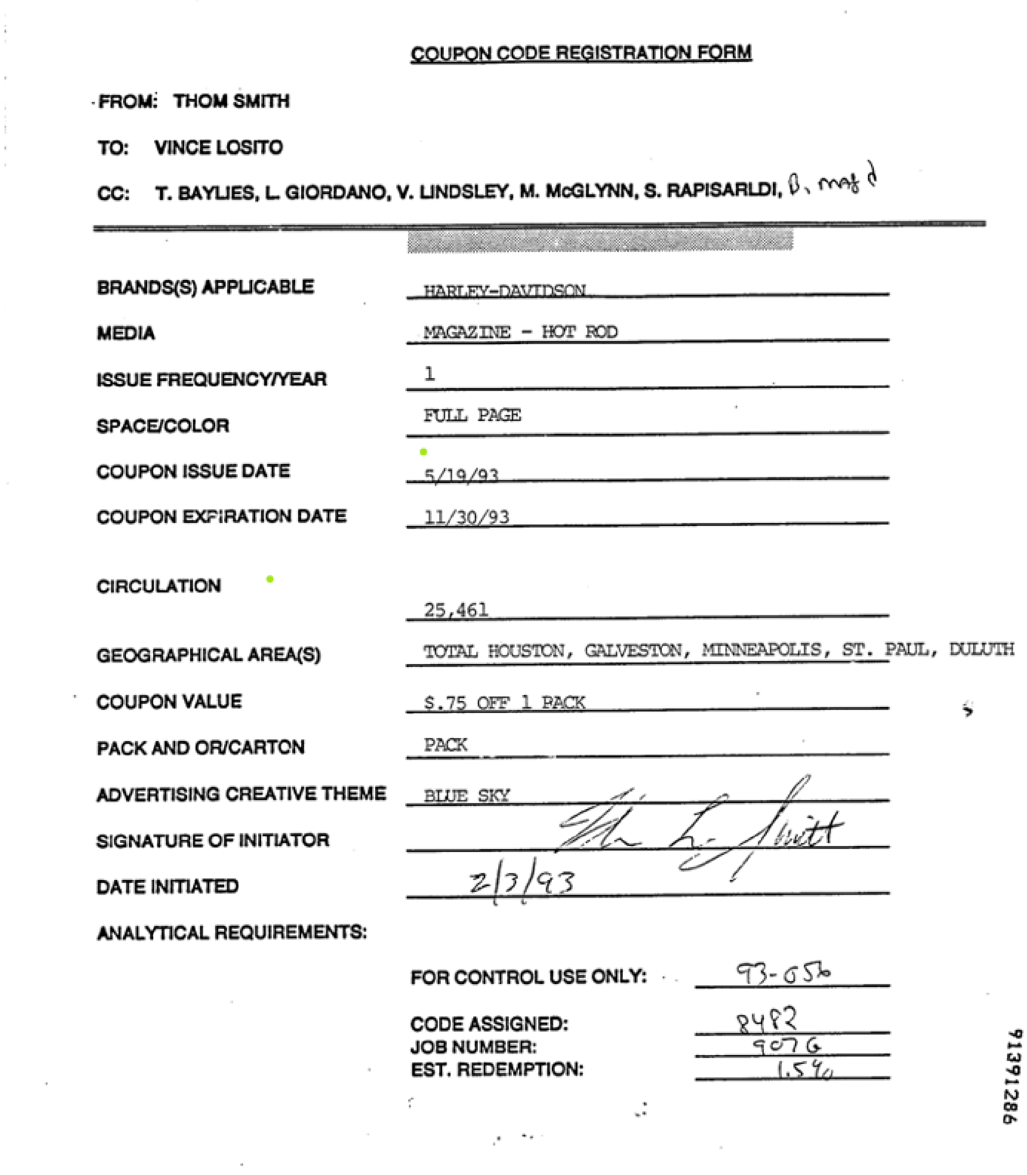
Acquiring high-quality datasets is essential for developing machine learning models that can effectively extract information from financial documents. In this project, we utilized two datasets: the FUNSD dataset and the Form 990 dataset.

3.2.1 FUNSD

The FUNSD [11] dataset is publicly available and a widely-used benchmark dataset for document segmentation and information extraction.

It consists of 199 annotated scanned forms from various domains, including finance, insurance, and medical industries. The dataset is designed to support form understanding tasks, such as layout analysis, information extraction, and classification, and it contains a diverse range of document layouts and formats.

The annotations in the FUNSD dataset include information such as the bounding boxes of text, lines, checkboxes, and other form elements, as well as the corresponding labels or values. The dataset also includes a separate set of testing data to evaluate the performance of form understanding systems. Researchers and practitioners in the fields of document analysis, machine learning, and natural language processing use the FUNSD dataset to develop and evaluate algorithms for form understanding tasks.



COUPON CODE REGISTRATION FORM

FROM: THOM SMITH

TO: VINCE LOSITO

CC: T. BAYLIES, L. GIORDANO, V. LINDSLEY, M. McGLYNN, S. RAPISARLDI, B, mag d

BRANDS(S) APPLICABLE	HARLEY-DAVIDSON
MEDIA	MAGAZINE - HOT ROD
ISSUE FREQUENCY/YEAR	1
SPACE/COLOR	FULL PAGE
COUPON ISSUE DATE	5/19/93
COUPON EXPIRATION DATE	11/30/93
CIRCULATION	25,461
GEOGRAPHICAL AREA(S)	TOTAL HOUSTON, GALVESTON, MINNEAPOLIS, ST. PAUL, DULUTH
COUPON VALUE	$.75 OFF 1 PACK
PACK AND OR/CARTON	PACK
ADVERTISING CREATIVE THEME	BLUE SKY
SIGNATURE OF INITIATOR	
DATE INITIATED	2/3/93
ANALYTICAL REQUIREMENTS:	

FOR CONTROL USE ONLY:	93-656
CODE ASSIGNED:	8482
JOB NUMBER:	9076
EST. REDEMPTION:	1.5%

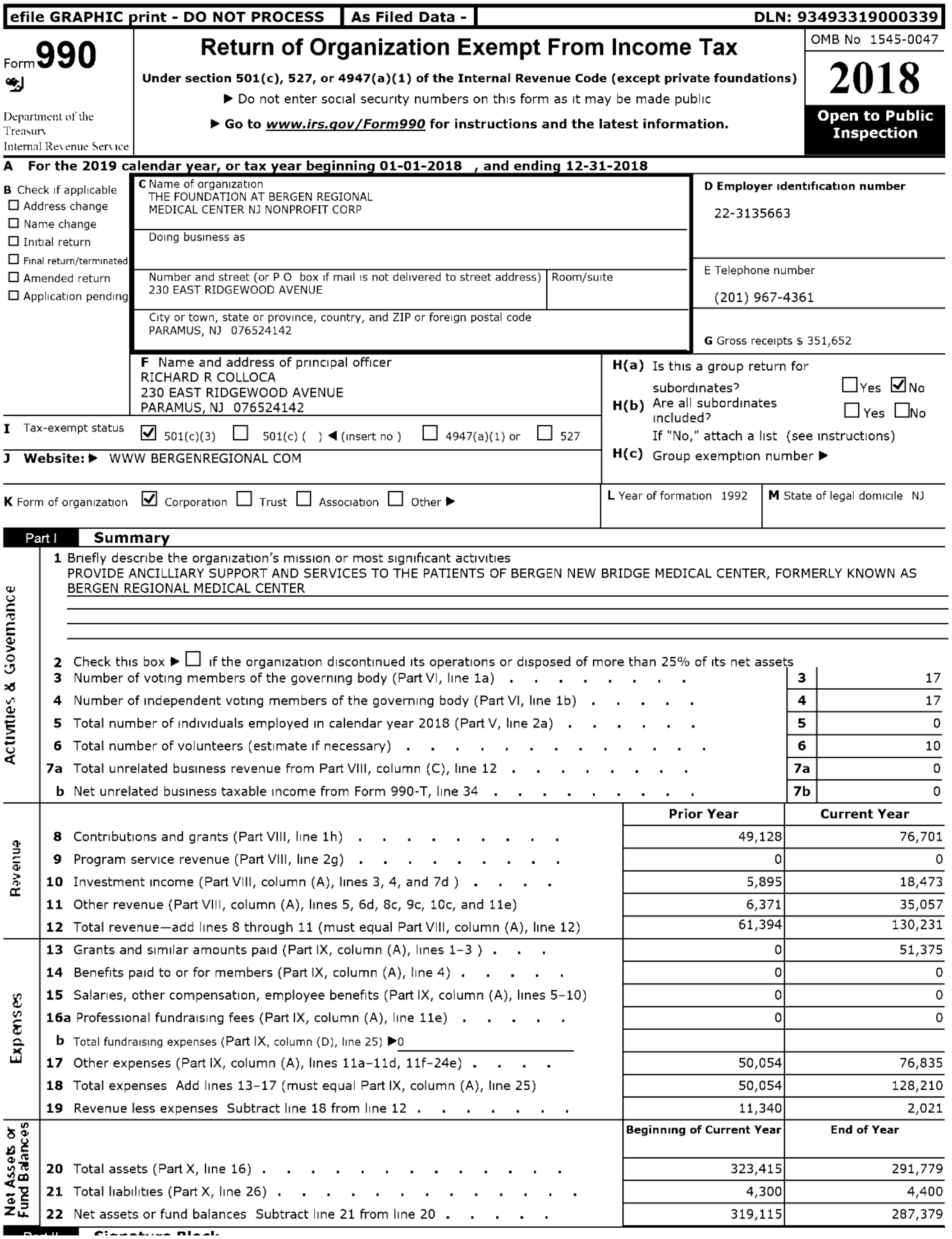
91391286

Figure 3.1: Sample Image from FUNSD Dataset

We chose this dataset because it provides a diverse set of document layouts and allows for training and evaluating our model on a broad range of document types.

3.2.2 IRS FORM 990

From a technical point of view, IRS Form 990 contains a wealth of information that can be extracted and analyzed to gain insights into the financial and operational performance of nonprofit organizations. The data on Form 990 is publicly available and can be accessed from the IRS website, making it a valuable resource for researchers, analysts, and document processing companies.



efile GRAPHIC print - DO NOT PROCESS | As Filed Data - | DLN: 93493319000339

Form 990

Department of the Treasury
Internal Revenue Service

Return of Organization Exempt From Income Tax

Under section 501(c), 527, or 4947(a)(1) of the Internal Revenue Code (except private foundations)

▶ Do not enter social security numbers on this form as it may be made public

▶ **Go to *www.irs.gov/Form990* for instructions and the latest information.**

OMB No 1545-0047

2018

Open to Public Inspection

A For the 2019 calendar year, or tax year beginning 01-01-2018 , and ending 12-31-2018

B Check if applicable
□ Address change
□ Name change
□ Initial return
□ Final return/terminated
□ Amended return
□ Application pending

C Name of organization
THE FOUNDATION AT BERGEN REGIONAL
MEDICAL CENTER NJ NONPROFIT CORP

Doing business as

Number and street (or P O box if mail is not delivered to street address) 230 EAST RIDGEWOOD AVENUE
Room/suite

City or town, state or province, country, and ZIP or foreign postal code
PARAMUS, NJ 076524142

D Employer identification number
22-3135663

E Telephone number
(201) 967-4361

G Gross receipts $ 351,652

F Name and address of principal officer
RICHARD R COLLOCA
230 EAST RIDGEWOOD AVENUE
PARAMUS, NJ 076524142

H(a) Is this a group return for subordinates? □ Yes ☑ No
H(b) Are all subordinates included? □ Yes □ No
If "No," attach a list (see instructions)
H(c) Group exemption number ▶

I Tax-exempt status ☑ 501(c)(3) □ 501(c) ( ) ◀ (insert no) □ 4947(a)(1) or □ 527

J **Website: ▶** WWW BERGENREGIONAL COM

K Form of organization ☑ Corporation □ Trust □ Association □ Other ▶

L Year of formation 1992

M State of legal domicile NJ

Part I **Summary**

Activities & Governance

1 Briefly describe the organization's mission or most significant activities
PROVIDE ANCILLIARY SUPPORT AND SERVICES TO THE PATIENTS OF BERGEN NEW BRIDGE MEDICAL CENTER, FORMERLY KNOWN AS BERGEN REGIONAL MEDICAL CENTER

2 Check this box ▶ □ if the organization discontinued its operations or disposed of more than 25% of its net assets

3	Number of voting members of the governing body (Part VI, line 1a)	3	17
4	Number of independent voting members of the governing body (Part VI, line 1b)	4	17
5	Total number of individuals employed in calendar year 2018 (Part V, line 2a)	5	0
6	Total number of volunteers (estimate if necessary)	6	10
7a	Total unrelated business revenue from Part VIII, column (C), line 12	7a	0
b	Net unrelated business taxable income from Form 990-T, line 34	7b	0

			Prior Year	Current Year
Revenue	8	Contributions and grants (Part VIII, line 1h)	49,128	76,701
	9	Program service revenue (Part VIII, line 2g)	0	0
	10	Investment income (Part VIII, column (A), lines 3, 4, and 7d)	5,895	18,473
	11	Other revenue (Part VIII, column (A), lines 5, 6d, 8c, 9c, 10c, and 11e)	6,371	35,057
	12	Total revenue—add lines 8 through 11 (must equal Part VIII, column (A), line 12)	61,394	130,231
Expenses	13	Grants and similar amounts paid (Part IX, column (A), lines 1–3)	0	51,375
	14	Benefits paid to or for members (Part IX, column (A), line 4)	0	0
	15	Salaries, other compensation, employee benefits (Part IX, column (A), lines 5–10)	0	0
	16a	Professional fundraising fees (Part IX, column (A), line 11e)	0	0
	b	Total fundraising expenses (Part IX, column (D), line 25) ▶0		
	17	Other expenses (Part IX, column (A), lines 11a–11d, 11f–24e)	50,054	76,835
	18	Total expenses Add lines 13–17 (must equal Part IX, column (A), line 25)	50,054	128,210
	19	Revenue less expenses Subtract line 18 from line 12	11,340	2,021
Net Assets or Fund Balances			Beginning of Current Year	End of Year
	20	Total assets (Part X, line 16)	323,415	291,779
	21	Total liabilities (Part X, line 26)	4,300	4,400
	22	Net assets or fund balances Subtract line 21 from line 20	319,115	287,379

Part II Signature Block

Figure 3.2: Sample Image of Form 990 from the dataset

One of the key pieces of information that can be extracted from Form 990 is revenue data. Nonprofits are required to report their revenue from various sources, including donations, grants, and investment income. This data can be used to analyze trends in fundraising and identify opportunities for organizations to improve their revenue generation.

Another important piece of data that can be extracted from Form 990 is expense data. Nonprofits are required to report how they spend their money, including details of program expenses, administrative expenses, and fundraising expenses. This data can be used to analyze the efficiency of an organization's operations and identify areas where cost savings can be made.

Form 990 also contains information on the governance and management of nonprofit organizations, including the names and addresses of key individuals such as officers, directors, and trustees. This data can be used to identify potential conflicts of interest and analyze the composition of an organization's leadership.
Various types of 990 Form are explained below:

- **990-N**: For small nonprofits with gross receipts less than \$50,000, a simple and quick electronic notice that includes basic information about the organization's activities and finances.
- **990-EZ**: A shortened version of the full 990 form, for mid-sized organizations with gross receipts between \$50,000 and \$200,000, and total assets less than \$500,000.
- **990**: A comprehensive form for larger nonprofits with gross receipts of \$200,000 or more, or total assets of \$500,000 or more, providing detailed information on the organization's activities and finances.
- **990-PF**: A specialized form for private foundations, providing information about the organization's finances, funding sources, and grants.
- **990-T**: A form filed by exempt organizations to report unrelated business income to the IRS, providing details on any income earned through activities not related to the organization's tax-exempt purpose.

To ensure the quality of the datasets, we performed a thorough data cleaning and preprocessing process. We removed any duplicate or irrelevant forms, corrected any mislabeled or misannotated data, and standardized the formatting and structure of the documents.

Overall, the acquisition and preprocessing of high-quality datasets were critical steps in developing our machine learning model for document extraction. By utilizing the FUNSD and Form 990 datasets, we were able to train and evaluate our model on a diverse set of document layouts and document types, leading to improved performance in extracting information from financial documents.

3.3 Data Preprocessing and Annotation

Data preprocessing and annotation are critical steps in developing a machine learning model. In this section, we will discuss the methods used to preprocess and annotate the dataset used in this study.

To preprocess the dataset, a variety of techniques were employed, including data cleaning, normalization, and feature engineering. Data cleaning involved removing any missing or invalid values from the dataset, while normalization ensured that the data was on a consistent scale. Feature engineering involved selecting the most relevant variables from the dataset and transforming them into a format that could be used by the machine learning algorithm.

Once the dataset was preprocessed, it was annotated using a combination of tools. The first tool used was Docsumo, an AI-powered document processing software that automates data extraction from complex documents like the 990 form. Docsumo helped to extract structured data from the 990 forms, which was then used for annotation. In addition to using Docsumo for data extraction from the 990 forms, we also built our own annotation tool to extract data.The combination of Docsumo and our annotation tool helped us to extract and label the necessary information from the 990 forms more accurately and comprehensively, which in turn facilitated the training and evaluation of our LayoutLMv2 model.

3.4 Nature of Dataset and Choice of Metrices

In this study, the 990 form dataset was used to develop a machine learning model. To evaluate the performance of the model, the F1 score and accuracy metrics were chosen. The F1 score is a measure that combines precision and recall, which are important measures for evaluating classification models. It provides an indication of how well the model is performing overall and takes into account both false positives and false negatives.

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

The accuracy metric measures the percentage of correct predictions made by the model.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

The choice of metrics for evaluating the performance of the model is appropriate for the nature of the dataset and the task at hand. Since the 990 form contains a wide range of financial information about the organization, it is crucial to have a model with high accuracy in its predictions.

3.5 Model Training

The model under consideration was subjected to a training process lasting 8 epochs, with validation conducted after each epoch to ensure effective learning.The split contained 80% on training set and 20% on validation set. During the training, the model was assessed using various metrics, including loss, accuracy, and F1 score. These measures allowed for a comprehensive evaluation of the model's performance, with loss indicating the model's ability to accurately predict outcomes and accuracy measuring the model's precision. The F1 score, which combines precision and recall, was also used to assess classification performance.

Following the training process, a thorough analysis of the model's performance was conducted, with the best checkpoint chosen to be utilized for future predictions. This selected checkpoint represents the optimal performance of the model achieved during the training process and can be used to further develop and improve the model or to apply it to new data.

Overall, this rigorous approach to the development and assessment of the model exemplifies a methodical and comprehensive approach to machine learning, which is critical to ensuring successful outcomes and effective implementation of the model in practical settings.

3.6 Software Development Model

The Iterative Model was selected as the software development model for our information extraction project using Layout LMv2 because the extraction model requires extensive iteration and fine-tuning to produce the desired output. We began by implementing a basic version of the model and continued to enhance and refine it iteratively until a satisfactory system was developed. Since the Layout LMv2 model involves complex neural network architecture, the model's hyperparameters needed to be regularly adjusted and tuned to achieve optimal performance. Additionally, as new challenges and requirements arose during the development process, new methods and procedures were implemented through further iterations. The iterative approach allowed us to efficiently address these challenges and continuously improve the model until it met our project's requirements.

4. System Design and Implementation

4.1 Diagrams

4.1.1 System Block Diagram

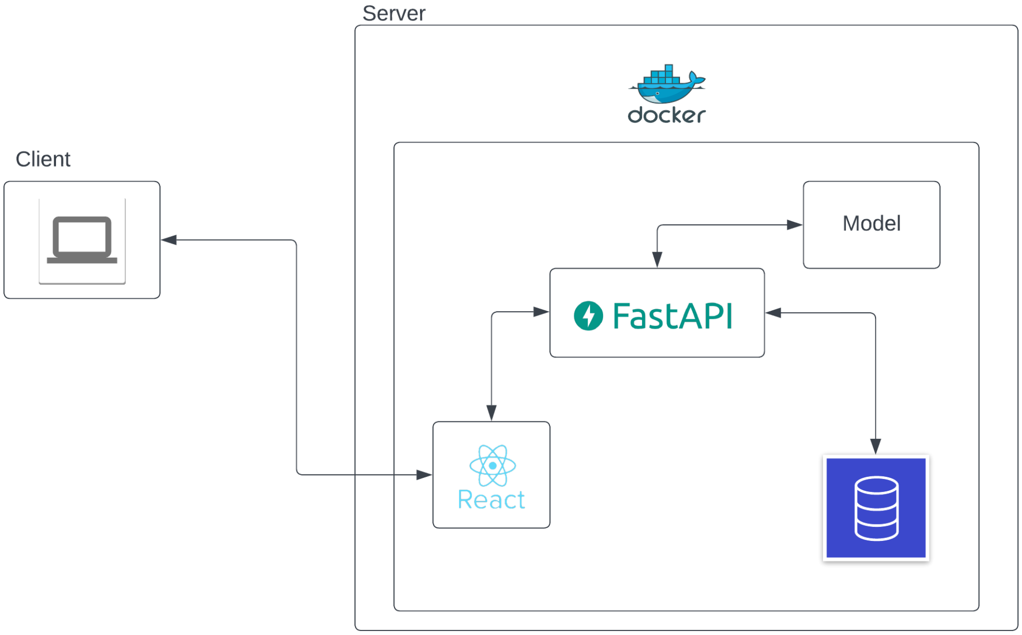


Figure 4.1: General Block Diagram

4.1.2 Use Case Diagram

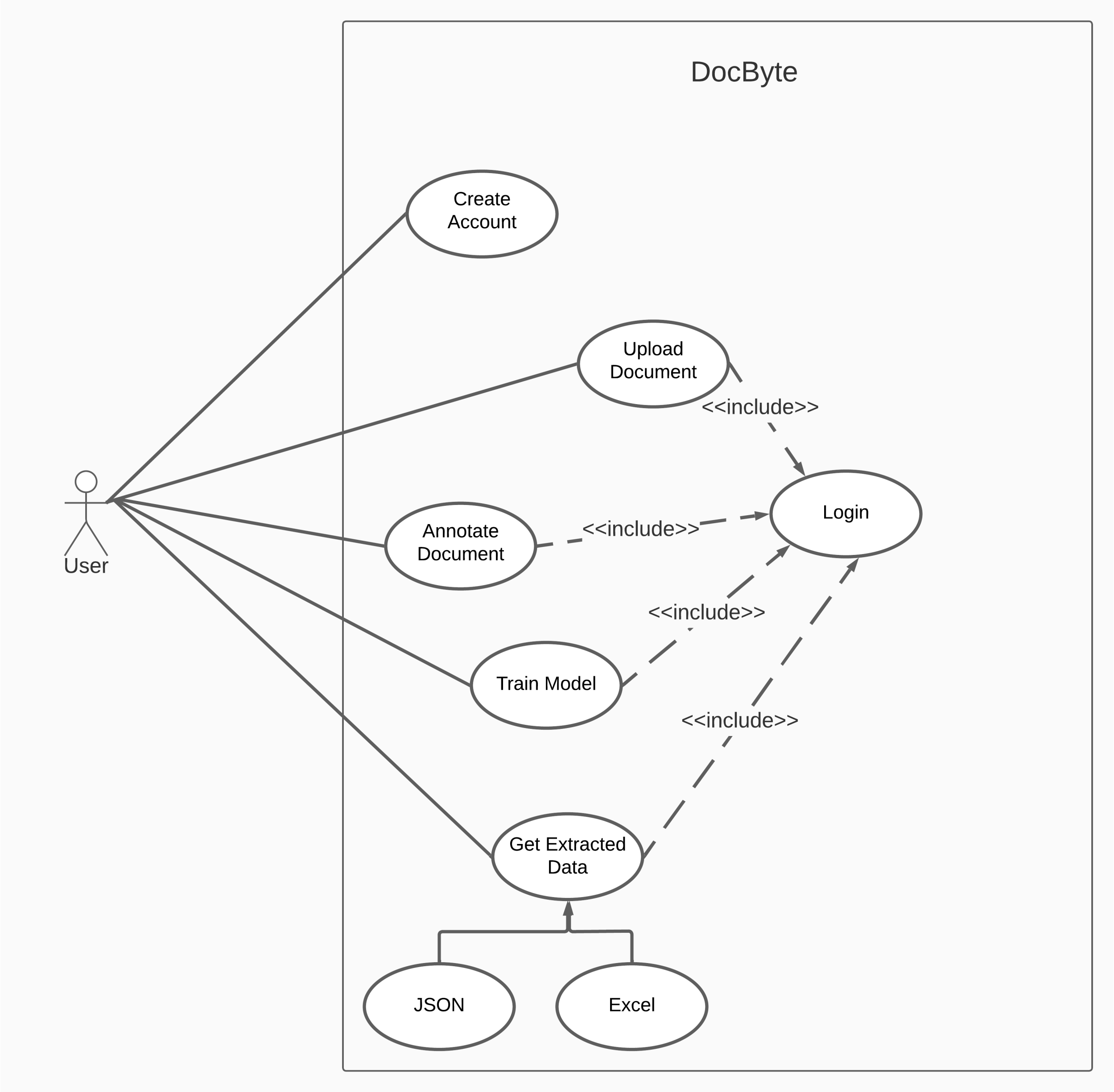


Figure 4.2: Use Case Diagram

4.1.3 Deployment Diagram

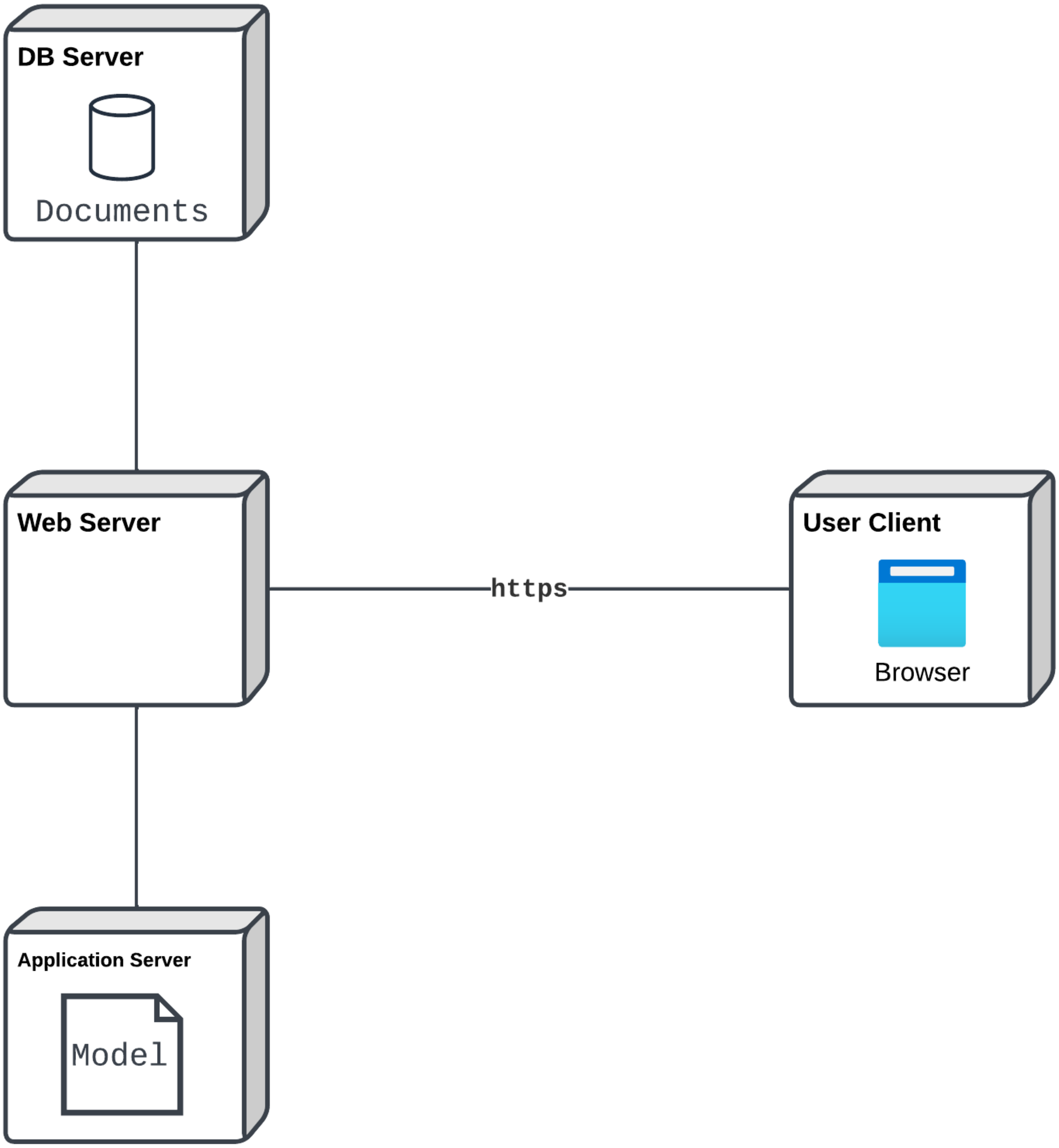


Figure 4.3: Deployment Diagram

4.2 Implementation

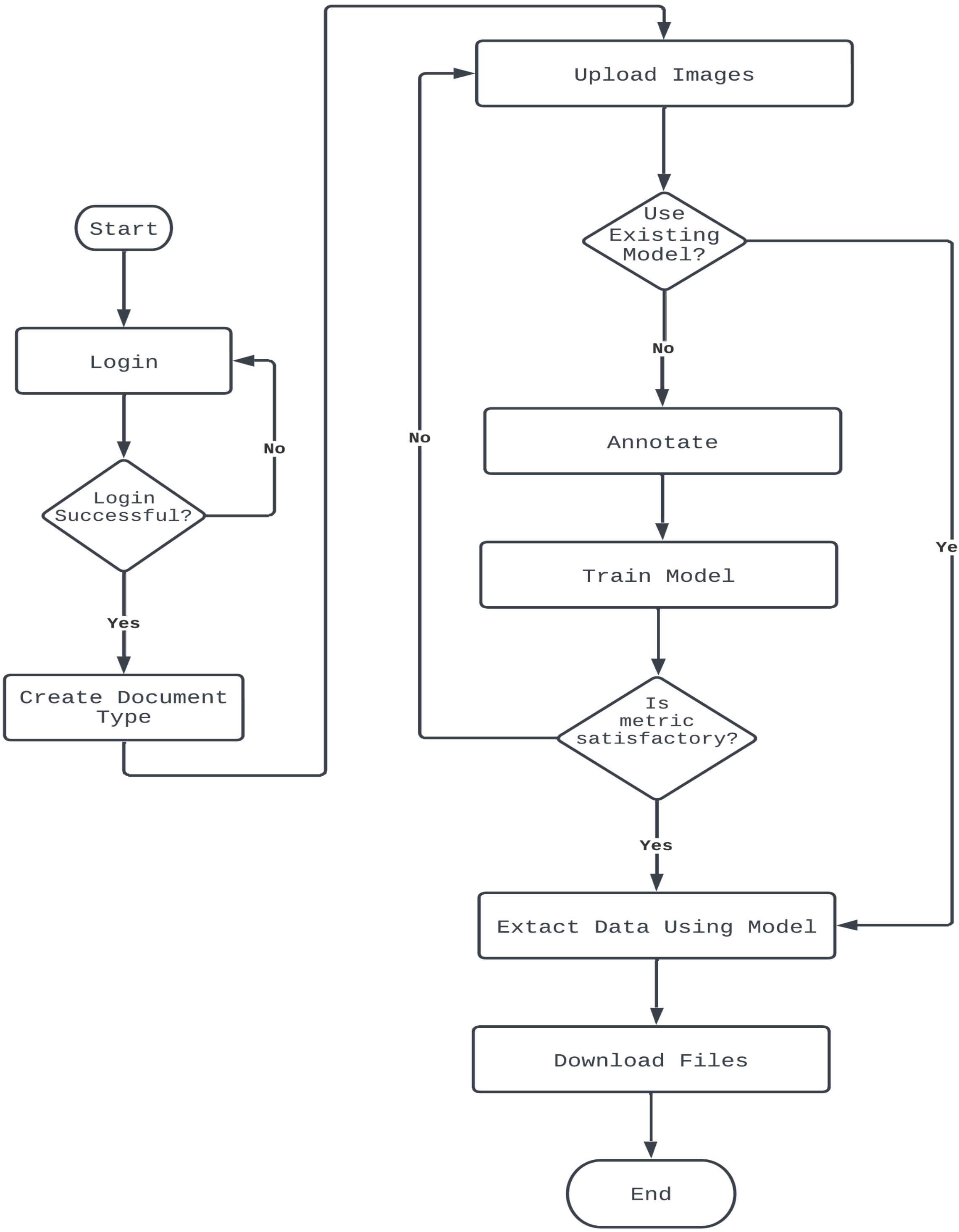


Figure 4.4: Flow chart

4.2.1 API

User Authentication

Cookies based authentication is used for authenticating users. Authentication helps to verify that the user is who they claim themselves to be. Users can login by entering their username or email address and password. When a user logs in, the website creates a unique session ID and stores it in a cookie on the user's browse and the database. This session ID is then used to identify the user's session on subsequent requests to the server. The server checks the session ID stored in the cookie and compares it to the session ID stored on the server to authenticate the user. If the IDs match, the user is considered authenticated and can access protected resources on the website.

REST API

The FastAPI-implemented REST API is a powerful tool for managing user authentication, extracting information, and annotating data. It contains numerous endpoints that allow users to interact with the API either directly or through a frontend web application.

The API's user authentication endpoints enable users to securely authenticate and manage their account information. Users can create accounts and manage their sessions when they are logged in. The doc_type feature allows users to create custom datasets, upload documents, and annotate them. Additionally, the API's information extraction endpoints enable users to extract and process data, while the annotation endpoints facilitate the annotation of various data types.

The frontend web application interacts with the REST API using these endpoints, enabling users to seamlessly utilize the API's functionality. With its ease of use, speed, and flexibility, the FastAPI-based REST API is an ideal solution for any project that requires a robust, secure, and scalable API.

Information Extraction

Users can extract information by uploading their desired document through either the email or the web app. The document is then queued for processing on the application server. Once complete, the extracted information can be downloaded in either CSV or JSON format, or can be directly saved to Google Drive if preferred.

Training

Users have the ability to create custom datasets by uploading their own documents and annotating them. This annotated data can then be utilized to train a model specific to the dataset. So trained model can be used to extract information from newly uploaded documents.

4.2.2 Annotation Tool

The web application includes a annotation tool used for the process of labeling and annotating data/ The tool allows users to identify and highlight specific fields and assign relevant tags or labels to them. This process of annotation is essential in building machine learning datasets, as it helps the model to recognize and learn patterns in the data more accurately.

4.3 Technology Stack

4.3.1 Frontend

React

React [1] or React.js is a popular open-source JavaScript library that is used to create user interfaces for web applications. It was developed by Facebook and is now maintained by Facebook and a community of developers. React allows developers to build reusable UI components and render them in a declarative way. It makes use of a virtual DOM, which is a lightweight representation of the actual DOM. This allows React to efficiently update only the parts of the UI that have changed, resulting in faster and more responsive web applications. React follows a component-based architecture, where each component is a self-contained unit of functionality that can be easily composed with other components to build complex UIs. React components can be created using either JavaScript classes or functions, and can be styled using CSS or inline styles.

Javascript

JavaScript [2] is a dynamic, high-level programming language that is widely used in web development. It enables the creation of interactive, client-side applications and can also be used on the server-side. It is known for its versatility, as it can be used for a variety of purposes, from adding basic functionality to web pages to creating complex web applications.

[1] https://reactjs.org/

[2] https://developer.mozilla.org/en-US/docs/Web/JavaScript

Tailwind CSS

Tailwind CSS [3] is a popular open-source utility-first CSS framework that allows developers to quickly and efficiently create custom user interfaces. Unlike traditional CSS frameworks that rely on pre-built components, Tailwind provides a set of pre-defined classes that can be used to style any HTML element in a consistent and intuitive way.

Tailwind includes a comprehensive set of utility classes that cover a wide range of CSS properties, such as margins, padding, typography, colors, and more. This allows developers to rapidly prototype and build custom designs without having to write CSS from scratch or rely on a design system. One of the key benefits of Tailwind is its flexibility and customization options. Developers can easily extend or override the default classes to meet their specific needs and design requirements. Tailwind also provides a range of configuration options, such as customizing the color palette and typography, that allow developers to create a unique and consistent look and feel for their web applications.

Figma

Figma [4] is a web-based design tool used for creating user interfaces, wireframes, and prototypes. It allows designers to collaborate in real-time, share designs with stakeholders, and create reusable design components. Figma is known for its intuitive interface and ease of use, making it a popular choice for design teams of all sizes.

4.3.2 Backend

FastAPI

FastAPI [5] is a modern, high-performance web framework for building APIs with Python. It is designed to be easy to use and scalable, making it a great choice for developers who want to quickly build high-performance APIs. With its built-in support for asynchronous programming and automatic data validation, FastAPI can help developers to build APIs quickly and with less code. It also integrates well with other Python libraries, making it a popular choice for developers.

[3] `https://tailwindcss.com/`

[4] `https://www.figma.com/`

[5] `https://fastapi.tiangolo.com/`

PyTorch

PyTorch [6] is an open-source machine learning library that is widely used for deep learning applications. It provides a flexible and dynamic computational graph system that allows developers to create complex neural networks and models. The library includes a variety of optimization algorithms and pre-built modules that make it easy to create and train models for tasks like image and speech recognition, natural language processing, and more. PyTorch is built to work seamlessly with other Python libraries, and its efficient implementation of tensor operations allows for fast and scalable computation.

Matplotlib

Matplotlib [7] is a data visualization library for Python that allows developers to create a variety of static, animated, and interactive visualizations in their Python scripts or applications. It provides a high-level interface for creating and manipulating graphs, charts, and plots, and includes support for a wide range of visualization styles and customization options. Matplotlib uses a state-based interface, allowing developers to modify existing plots or create new ones from scratch, and provides support for multiple backends to allow for visualization across multiple platforms and formats. Overall, Matplotlib is a versatile and powerful tool for data visualization in Python.

MongoDB

MongoDB [8] is a document-oriented NoSQL database that is used for storing and retrieving large amounts of data in a flexible and scalable manner. It stores data in documents, which are JSON-like data structures that can be nested and indexed for faster access. MongoDB uses a flexible schema, which means that data can be added or changed without having to modify the database schema. It also supports horizontal scaling, which allows for high availability and faster performance by distributing the data across multiple servers. MongoDB is often used in web applications and big data analytics, where fast and scalable data storage and retrieval is critical.

[6] https://pytorch.org/

[7] https://matplotlib.org/

[8] https://www.mongodb.com/

4.3.3 Platform

Google Colab

Google Colab is a cloud-based development environment that provides a Jupyter notebook interface for working with Python code. It allows users to run and execute Python code on remote servers using a web browser, with access to a variety of pre-installed libraries, including TensorFlow and PyTorch. Google Colab also provides free access to GPU and TPU resources for faster computation, making it a popular choice for machine learning and data analysis tasks. It integrates with Google Drive for data storage and offers collaboration features for sharing notebooks with others.

Azure VM

Azure [9] Virtual Machines (VMs) is a cloud-based service provided by Microsoft Azure that allows users to create and manage virtual machines in the cloud. It offers on-demand computing resources for running Windows and Linux-based applications in a scalable and secure manner. Azure VMs provide a wide range of options for configuring virtual machines, including selecting virtual machine sizes, operating systems, and networking configurations. It also offers options for backup and disaster recovery, making it a popular choice for businesses that require flexibility and scalability in their computing infrastructure.

[9] https://azure.microsoft.com/

5. Results

In this section, we present the results of our analysis using LayoutLMv2 as the deep learning model for document layout analysis and information extraction. The model was trained on two datasets: FUNSD and IRS 990, FUNSD containing 199 annotated images and IRS 990 containing 202 annotated images with an 80/20 split for training and validating. We trained the LayoutLMv2 model on the training set of each dataset and evaluated its performance on the validation set. The results of our analysis are presented in this section, including loss, F1 score and detailed error analysis.

5.1 Loss and F1 Score

The graph for Loss and F1 Score is generated for each dataset that can be seen below.Upon examining the graphs for the FUNSD dataset, it is evident that the validation and training loss curves are decreasing over time, indicating that the model is learning and becoming more efficient. Additionally, the f1 score curve is increasing, which further reinforces the fact that the model is performing well and producing accurate results.

Similarly, the results for the 990 dataset are also very promising. The loss curve for both the validation and training data is decreasing, which indicates that the model is learning and becoming more accurate with time. Furthermore, the f1 score curve is increasing, indicating that the model is producing more accurate and reliable results.

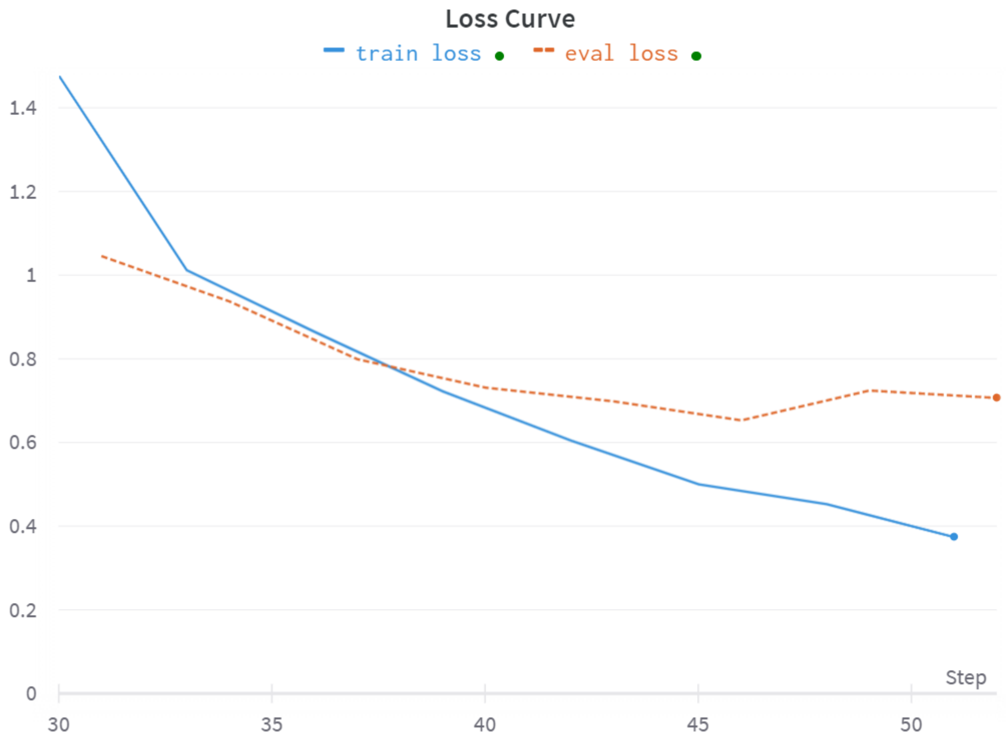


Figure 5.1: Train Vs Validation Loss Plot on FUNSD

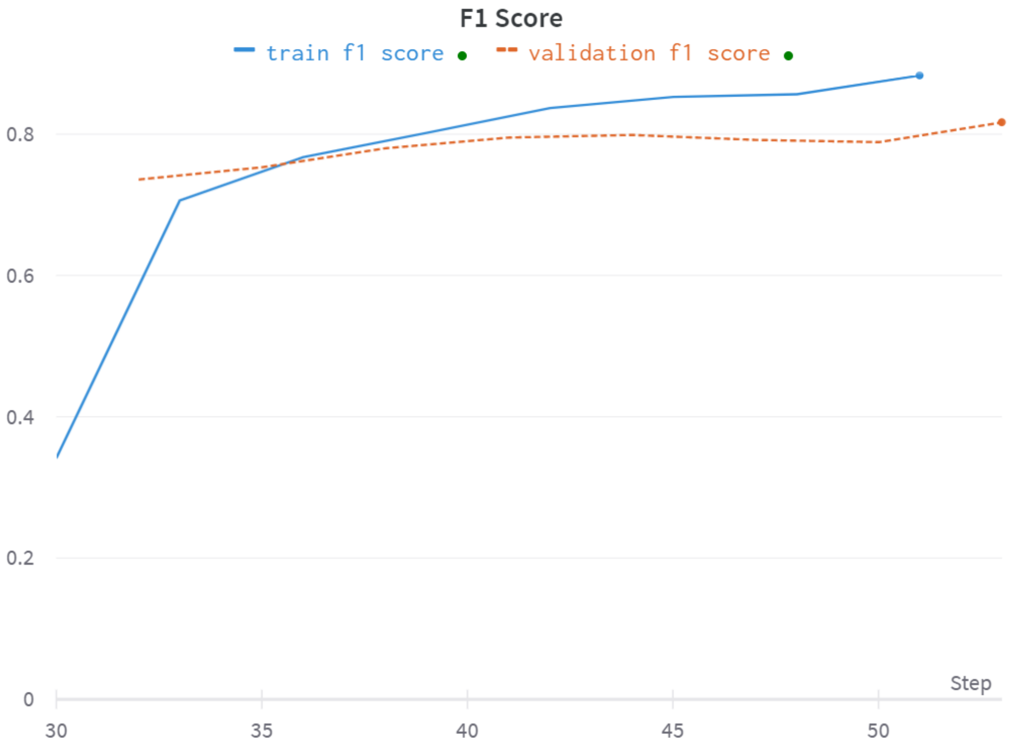


Figure 5.2: Train Vs Validation F1 Score Plot on FUNSD

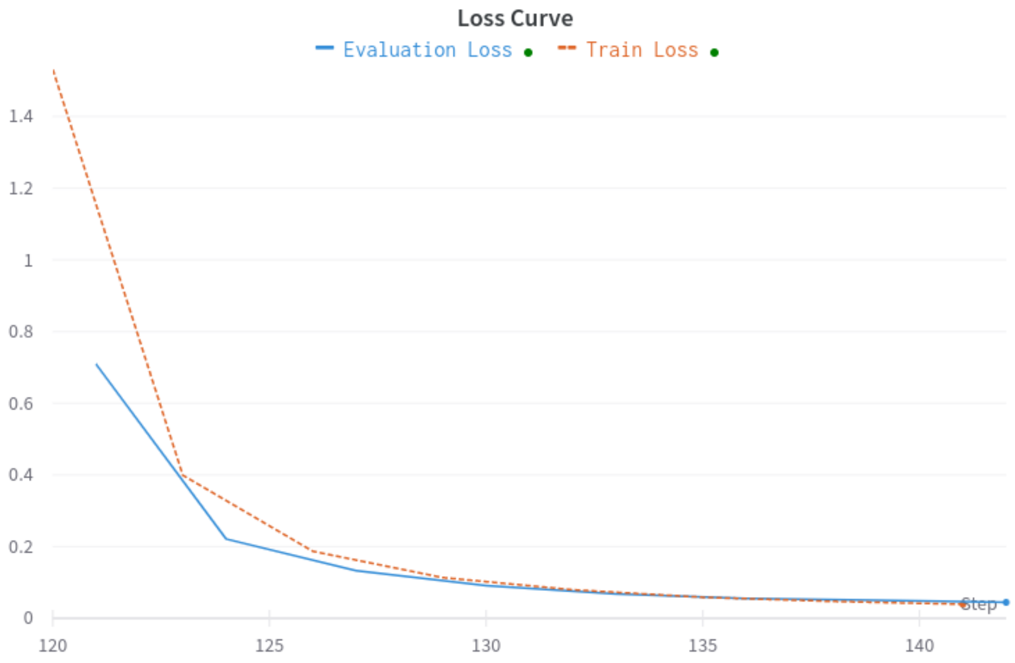


Figure 5.3: Train Vs Validation Loss Plot on 990 Form

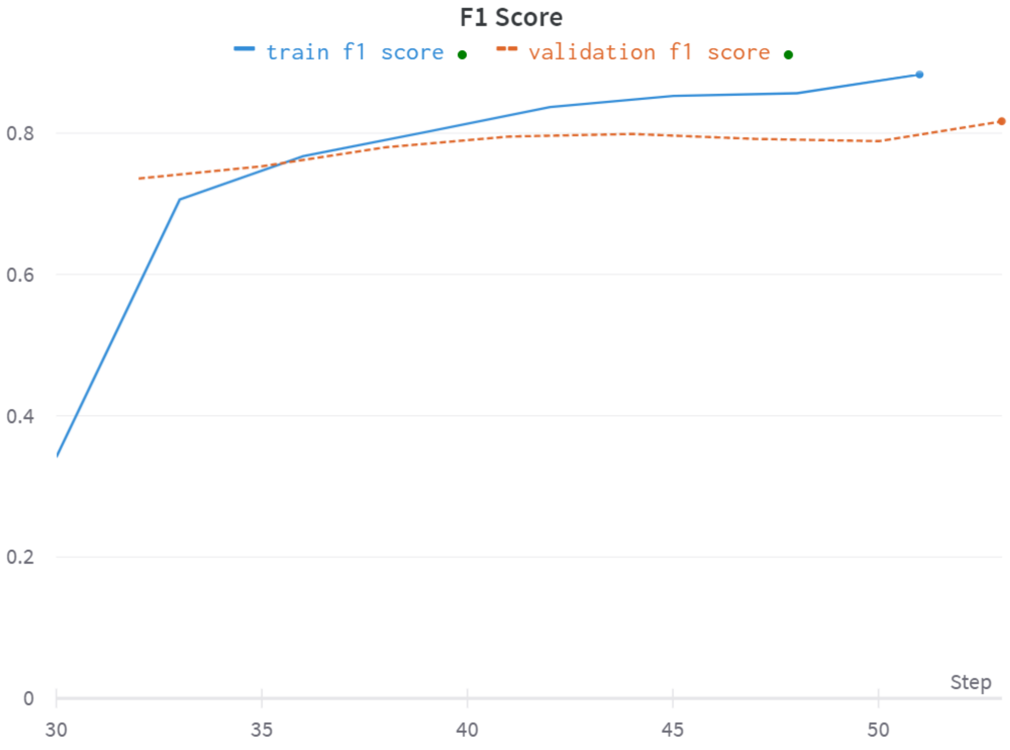


Figure 5.4: Train Vs Validation F1 Score Plot on 990 Form

Overall, the results of the reports suggest that the models used for both the FUNSD and 990 datasets are performing well and producing accurate results. The decreasing loss curves and increasing f1 score curves for both datasets are clear indicators of the model's learning and improving over time. These results are very encouraging and indicate that the models can be used effectively to produce reliable results for a wide range of applications.

5.2 Evaluation Metrices

The table 5.1 shows the performance of LayoutLMv2 on FUNSD and 990 Form dataset.It can be seen that Training LayoutLMv2 performed better on the 990 Form dataset with an F1-score of 0.92 compared to the FUNSD dataset with an F1-score of 0.81. The higher F1-score on the 990 Form dataset indicates a better balance between precision and recall for this dataset compared to the FUNSD dataset.

One possible reason for the lower performance of Training LayoutLMv2 on the FUNSD dataset is the variability in document layouts and formats within the dataset. The FUNSD dataset contains various types of forms with different layouts and formats, which could make it more challenging for the model to learn the underlying patterns and structures in the documents compared to the 990 Form dataset, which contains fewer variants.

Table 5.1: Evaluation Metrices using LayoutLMv2

Datasets	**Total Documents**	**Precision**	**Recall**	**F1-score**
990 Form	202	0.89	0.97	0.92
FUNSD	199	0.81	0.82	0.81

In summary, while Training LayoutLMv2 achieved good performance on the 990 Form dataset, it had a moderate performance on the FUNSD dataset, which could be attributed to the complexity and variability of the dataset. The F1-score is a useful metric to measure the overall performance of a model in identifying the relevant documents, taking into account both precision and recall.

5.3 Label-Wise Evaluation Metrices

The table 5.2 shows the result of validation of LayoutLMv2 on 990 Form data.The results indicate that the model is performing well overall, but there are some underperformed labels, such as **NAME OF ORGANIZATION**, **GROSS RECEIPTS** and **NUMBER AND STREET**.

The underperformance of **NAME OF ORGANIZATION**, **GROSS RECEIPTS**, and **NUMBER AND STREET** can be attributed to several factors. For **NAME OF ORGANIZATION**, the label is complex and involves identifying multiple pieces of information in a specific order. The model may not have learned the correct relationship between them due to label ambiguity, or label complexity.

Similarly, **GROSS RECEIPTS** can vary significantly in format and location within the form, making it difficult for the model to consistently identify it. This variability and ambiguity may also be due to limited training examples or label complexity.

NUMBER AND STREET may have similar challenges as **GROSS RECEIPTS** in terms of its variability and ambiguity. The location and format of the information can vary significantly across forms, and the model may not have learned to consistently identify it due to limited training examples or label ambiguity.

Table 5.2: Label Wise Metrices for IRS 990 Form

Label	**Precision**	**Recall**	**F1-score**
ADDRESS LINE	0.97	0.97	0.97
CAT NO	0.99	1.00	0.99
DLN	0.83	1.00	0.91
EMPLOYER IDENTIFICATION NUMBER	1.00	1.00	1.00
FIRM'S ADDRESS	0.72	0.89	0.80
FIRM'S EIN	0.91	1.00	0.95
FIRM'S NAME	0.80	0.80	0.80
FORM OF ORGANIZATION	0.76	0.84	0.80
GROSS RECEIPTS	0.64	0.92	0.75
NAME AND ADDRESS OF PRINCIPAL OFFICER	0.95	0.86	0.90
NAME OF ORGANIZATION	0.70	0.64	0.67
NUMBER AND STREET	0.72	0.78	0.75
OMB NO	0.81	1.00	0.89
STATE OF LEGAL DOMICILE	0.87	0.93	0.90
TAX YEAR BEGINNING	1.00	1.00	1.00
TAX YEAR ENDING	1.00	1.00	1.00
TELEPHONE NUMBER	0.98	1.00	0.99
WEBSITE	1.00	1.00	1.00
YEAR OF FORMATION	0.89	1.00	0.94
Micro avg	0.88	0.97	0.92
Macro avg	0.87	0.93	0.90
Weighted avg	0.89	0.97	0.92

The table 5.3 shows results of training FUNSD on LayoutLMv2. The results indicates that model has good performance in identifying **ANSWER**and **QUESTION** classes, with weighted average F1-scores of 0.81 and 0.85, respectively. However, the model has relatively low performance in identifying **HEADER** instances, with a weighted average F1-score of 0.50.

For **HEADER** class, the precision is 0.59, indicating that out of all the instances predicted as **HEADER**, only 59% are actually **HEADER** instances. The recall for **HEADER** class is 0.43, which means that out of all the true **HEADER** instances in the dataset, the model correctly identifies only 43%. This low recall suggests that the model is missing many

HEADER instances, which might be due to the variability in the layout and formatting of headers in the document.

Table 5.3: Label Wise Metrices for FUNSD

Label	Precision	Recall	F1-score
ANSWER	0.80	0.82	0.81
HEADER	0.59	0.43	0.50
QUESTION	0.84	0.87	0.85
Micro avg	0.81	0.82	0.82
Macro avg	0.74	0.70	0.72
Weighted avg	0.81	0.82	0.81

We conducted an evaluation of the LayoutLMv2 model on the 990 Forms dataset, specifically looking at its performance with varying amounts of training data and epochs. The results, presented in 5.4, demonstrate that the overall f1 score of the model improves monotonically with more training epochs and larger sample sizes.

The 990 Forms dataset we used for fine-tuning only contains 200 images, making it a low-resource setting. However, our findings confirm that pre-training the model on text and layout is an effective approach for scanned document understanding. Even with the limited sample size, we observed significant improvements in accuracy as more training data and epochs were used. To be more specific, we experimented with sample sizes of 40, 80, 120, and 160, and trained the model with up to 8 epochs which can be seen in 5.4.

Table 5.4: Experimenting with dataset size

Dataset size	No of epochs	Precision	Recall	F1 score
40	7	0.02	0.0	0.0
	8	0.31	0.14	0.19
80	7	0.49	0.30	0.37
	8	0.69	0.45	0.54
120	6	0.81	0.75	78
	7	0.83	0.82	0.83
	8	0.83	0.93	0.88
160	6	0.83	0.89	0.86
	7	0.85	0.88	0.87
	8	0.85	0.95	0.90

In conclusion, Training LayoutLMv2 performed well on both the FUNSD and 990 Form datasets, achieving an F1-score of 0.81 and 0.92, respectively. The results demonstrate the potential of LayoutLMv2 in accurately identifying and extracting relevant information from structured documents. However, further improvements could be made to the model's performance on more complex and varied datasets such as FUNSD. Nonetheless, the model's strong performance on the 990 Form dataset suggests that it could be a valuable tool for automating the extraction of structured data from forms and other similar documents.

6. Conclusion

In conclusion, the project this project utilized the datasets FUNSD and 990 Form to develop a model using LayoutLMv2. The end product is a web application that enables users to train the model and generate inferences. Additionally, the application features an annotation tool that allows for the efficient annotation of data. With this project, users can easily extract information from structured documents, making data analysis and interpretation more accessible and streamlined. Overall, this project is a valuable contribution to the field of information extraction and has the potential to make a significant impact on various industries that rely manual extraction.

7. Limitations and Future Works

There are a few potential limitations and areas for future work that could be considered for this project.

Firstly, the model's performance may be limited by the quality and size of the training dataset. While the FUNSD and 990 Form datasets are comprehensive, they may not represent all possible variations in structured document layouts. Therefore, future work could involve expanding the dataset to include more diverse layouts, which could enhance the model's accuracy and generalizability.

Secondly, the annotation tool could be improved to increase efficiency and accuracy. For instance, implementing an active learning algorithm to select the most informative data for annotation could improve the tool's performance and reduce the time required for manual annotation.

Integration of more export formats can be done to enhance the usability and accessibility of software by allowing users to work with their preferred file types and tools.

Other potential limitations of the current model is its response time during inference, which could be improved by using the ONNX runtime. As part of future work, efforts could be made to reduce the model's response time and enhance its overall performance by exploring the use of ONNX runtime.

Lastly, while the web application is a useful tool for users, its scalability and security may be limited, depending on the size and sensitivity of the data being processed. Future work could involve improving the application's architecture and incorporating additional security measures to support larger-scale data processing and handling.

References

[1] Tongwei Liu, Hao Xu, Minvydas Ragulskis, Maosen Cao, and Wiesław Ostachowicz. A data-driven damage identification framework based on transmissibility function datasets and one-dimensional convolutional neural networks: Verification on a structural health monitoring benchmark structure. *Sensors*, 20(4):1059, 2020.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[3] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. *arXiv preprint arXiv:2012.14740*, 2020.

[4] Peggy M. Andersen, Philip J. Hayes, Steven P. Weinstein, Alison K. Huettner, Linda M. Schmandt, and Irene B. Nirenburg. Automatic extraction of facts from press releases to generate news stories. In *Third Conference on Applied Natural Language Processing*, pages 170–177, Trento, Italy, March 1992. Association for Computational Linguistics.

[5] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data: The story so far. In *Semantic services, interoperability and web applications: emerging concepts*, pages 205–227. IGI global, 2011.

[6] Yi Zheng, Qitong Wang, and Margrit Betke. Deep neural network for semantic-based text recognition in images, 2019.

[7] Hongtao Xie, Shancheng Fang, Zheng-Jun Zha, Yating Yang, Yan Li, and Yongdong Zhang. Convolutional attention networks for scene text recognition. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 15(1s):1–17, 2019.

[8] Frank Lebourgeois, Zbigniew Bublinski, and Hubert Emptoz. A fast and efficient method for extracting text paragraphs and graphics from unconstrained documents. In *11th IAPR International Conference on Pattern Recognition. Vol. II. Conference B: Pattern Recognition Methodology and Systems*, volume 1, pages 272–273. IEEE Computer Society, 1992.

[9] Nishant Subramani, Alexandre Matton, Malcolm Greaves, and Adrian Lam. A survey of deep learning approaches for ocr and document understanding. *arXiv preprint arXiv:2011.13534*, 2020.

[10] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4083–4091, 2022.

[11] Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. Funsd: A dataset for form understanding in noisy scanned documents. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, pages 1–6. IEEE, 2019.

[12] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200, 2020.

[13] Lawrence O'Gorman. The document spectrum for page layout analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 15(11):1162–1173, 1993.

[14] Pınar Duygulu and Volkan Atalay. A hierarchical representation of form documents for identification and retrieval. *International Journal on Document Analysis and Recognition*, 5:17–27, 2002.

[15] Raymond W Smith. Hybrid page layout analysis via tab-stop detection. In *2009 10th International Conference on Document Analysis and Recognition*, pages 241–245. IEEE, 2009.

[16] Michael Shilman, Percy Liang, and Paul Viola. Learning nongenerative grammatical models for document analysis. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 962–969. IEEE, 2005.

[17] Simone Marinai, Marco Gori, and Giovanni Soda. Artificial neural networks for document analysis and recognition. *IEEE Transactions on pattern analysis and machine intelligence*, 27(1):23–35, 2005.

[18] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[19] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

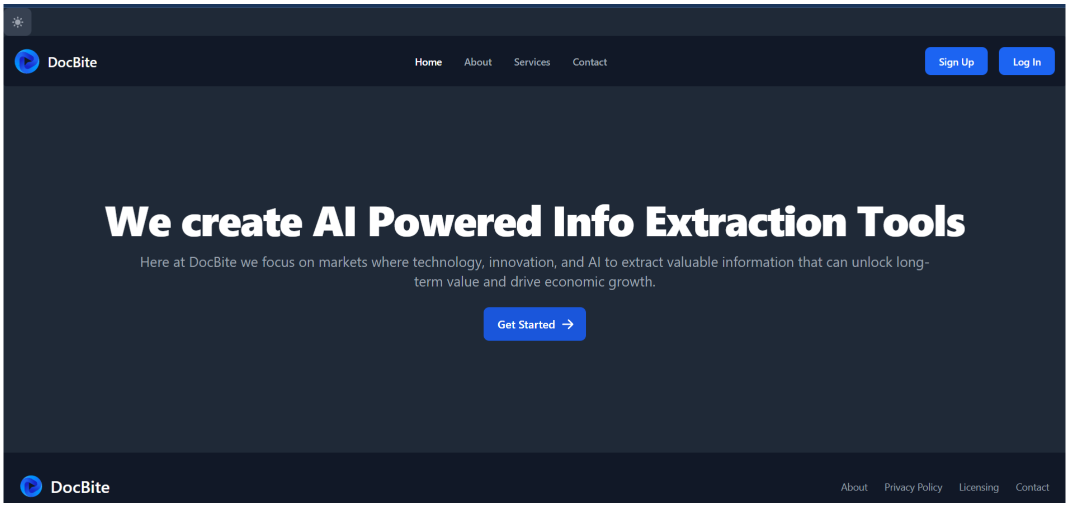
[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[21] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[22] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

8. Appendices

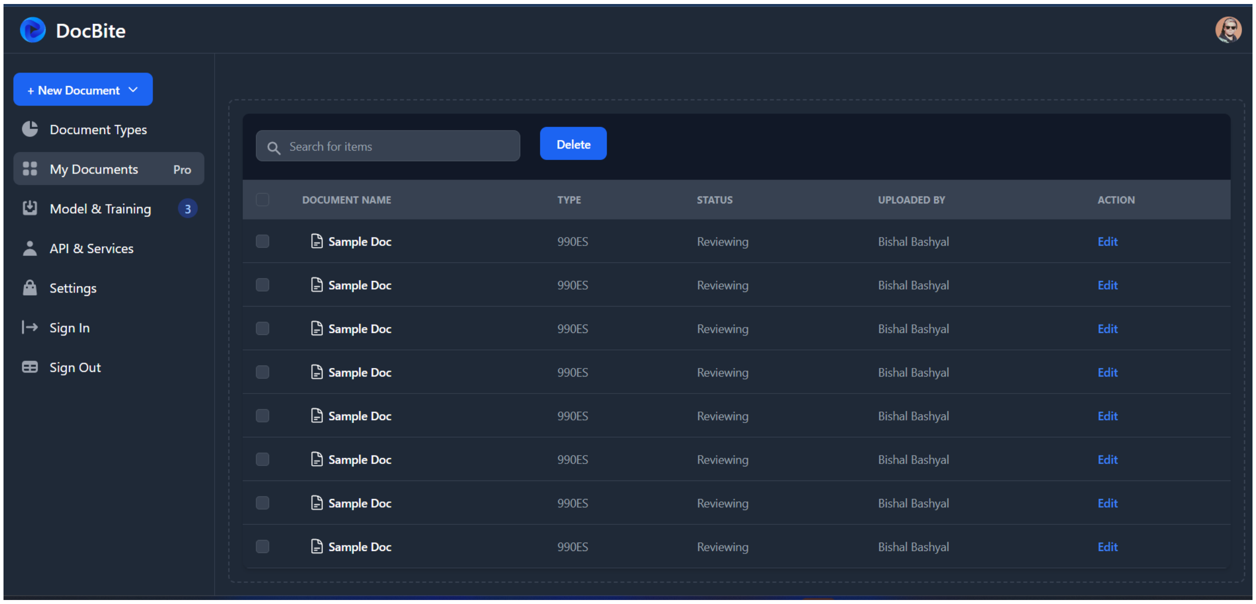
8.1 Screenshots of End Product



We create AI Powered Info Extraction Tools

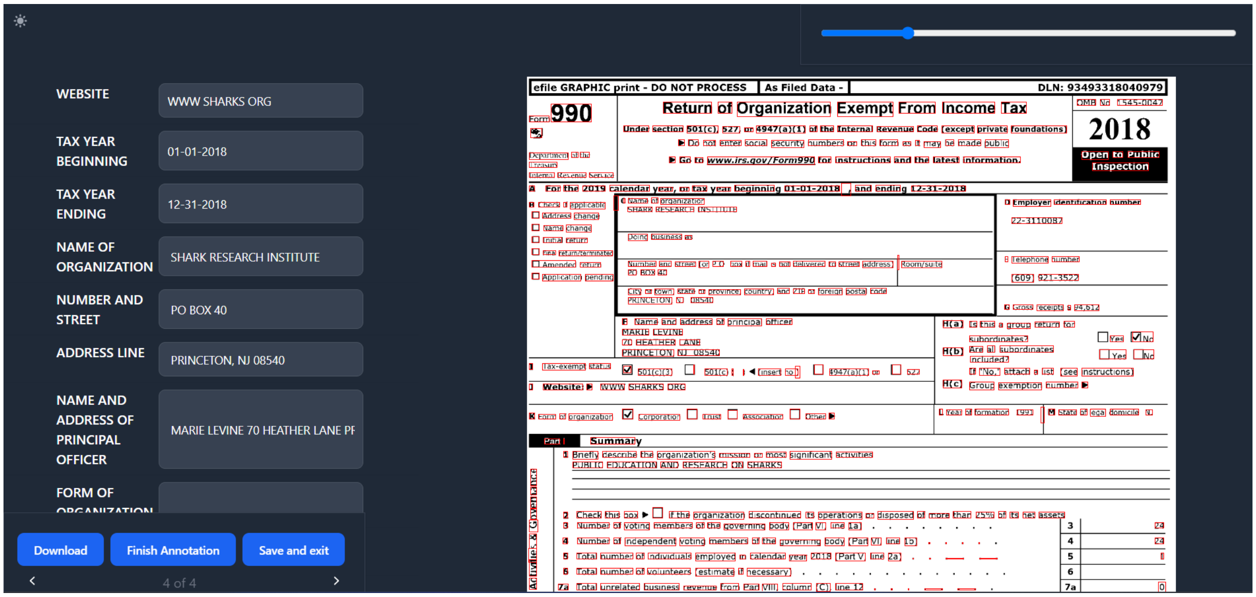
Here at DocBite we focus on markets where technology, innovation, and AI to extract valuable information that can unlock long-term value and drive economic growth.

Figure 8.1: HomePage of WebApp



	DOCUMENT NAME	TYPE	STATUS	UPLOADED BY	ACTION
□	Sample Doc	990ES	Reviewing	Bishal Bashyal	Edit
□	Sample Doc	990ES	Reviewing	Bishal Bashyal	Edit
□	Sample Doc	990ES	Reviewing	Bishal Bashyal	Edit
□	Sample Doc	990ES	Reviewing	Bishal Bashyal	Edit
□	Sample Doc	990ES	Reviewing	Bishal Bashyal	Edit
□	Sample Doc	990ES	Reviewing	Bishal Bashyal	Edit
□	Sample Doc	990ES	Reviewing	Bishal Bashyal	Edit
□	Sample Doc	990ES	Reviewing	Bishal Bashyal	Edit

Figure 8.2: Dashboard Page



WEBSITE	WWW SHARKS ORG
TAX YEAR BEGINNING	01-01-2018
TAX YEAR ENDING	12-31-2018
NAME OF ORGANIZATION	SHARK RESEARCH INSTITUTE
NUMBER AND STREET	PO BOX 40
ADDRESS LINE	PRINCETON, NJ 08540
NAME AND ADDRESS OF PRINCIPAL OFFICER	MARIE LEVINE 70 HEATHER LANE PF
FORM OF ORGANIZATION	

efile GRAPHIC print - DO NOT PROCESS | As Filed Data - | DLN: 93493318040979

Form 990

Department of the Treasury Internal Revenue Service

Return of Organization Exempt From Income Tax

Under section 501(c), 527, or 4947(a)(1) of the Internal Revenue Code (except private foundations)

▶ Do not enter social security numbers on this form as it may be made public

▶ Go to www.irs.gov/Form990 for instructions and the latest information.

OMB No 1545-0047

2018

Open to Public Inspection

A For the 2019 calendar year, or tax year beginning 01-01-2018 , and ending 12-31-2018

B Check if applicable
□ Address change
□ Name change
□ Initial return
□ Final return/terminated
□ Amended return
□ Application pending

C Name of organization
SHARK RESEARCH INSTITUTE

Doing business as

Number and street (or P.O. box if mail is not delivered to street address) | Room/suite
PO BOX 40

City or town, state or province, country, and ZIP or foreign postal code
PRINCETON, NJ 08540

D Employer identification number
22-3110087

E Telephone number
(609) 921-3522

G Gross receipts $ 94,612

F Name and address of principal officer
MARIE LEVINE
70 HEATHER LANE
PRINCETON, NJ 08540

H(a) Is this a group return for subordinates? □ Yes ☑ No

H(b) Are all subordinates included? □ Yes □ No

If "No," attach a list (see instructions)

H(c) Group exemption number ▶

I Tax-exempt status ☑ 501(c)(3) □ 501(c) ( ) ◀ (insert no) □ 4947(a)(1) or □ 527

J Website: ▶ WWW SHARKS ORG

K Form of organization ☑ Corporation □ Trust □ Association □ Other ▶

L Year of formation 1991 | M State of legal domicile NJ

Part I Summary

Activities & Governance

1 Briefly describe the organization's mission or most significant activities
PUBLIC EDUCATION AND RESEARCH ON SHARKS

2 Check this box ▶ □ if the organization discontinued its operations or disposed of more than 25% of its net assets

3	Number of voting members of the governing body (Part VI, line 1a)	3	24
4	Number of independent voting members of the governing body (Part VI, line 1b)	4	24
5	Total number of individuals employed in calendar year 2018 (Part V, line 2a)	5	1
6	Total number of volunteers (estimate if necessary)	6	
7a	Total unrelated business revenue from Part VIII, column (C), line 12	7a	0

Figure 8.3: Annotation Tool

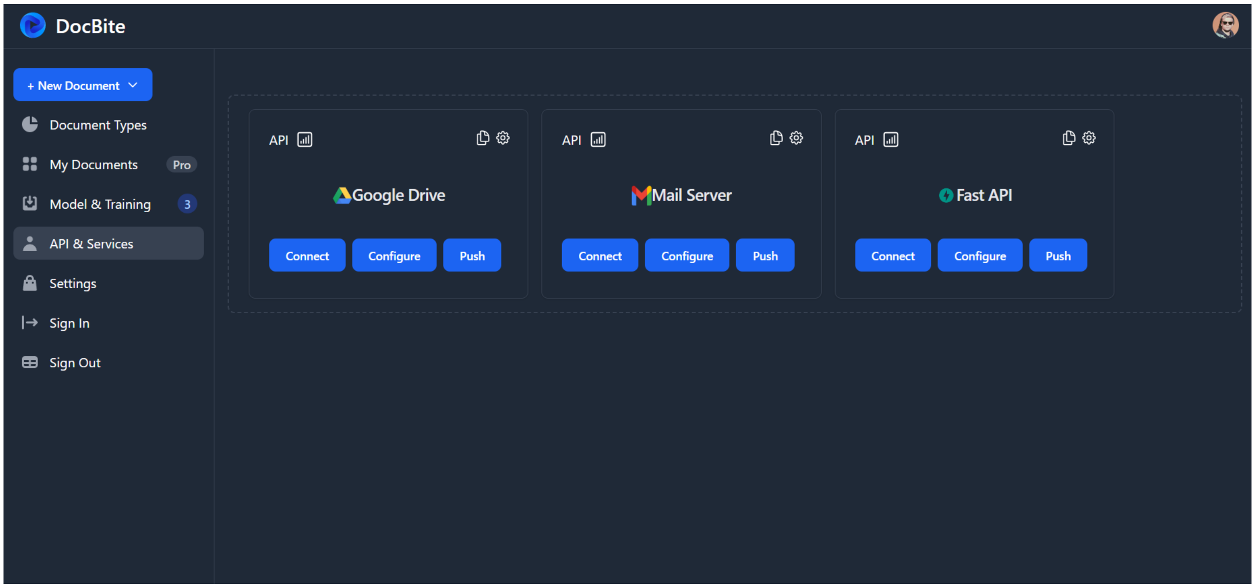


Figure 8.4: API Page

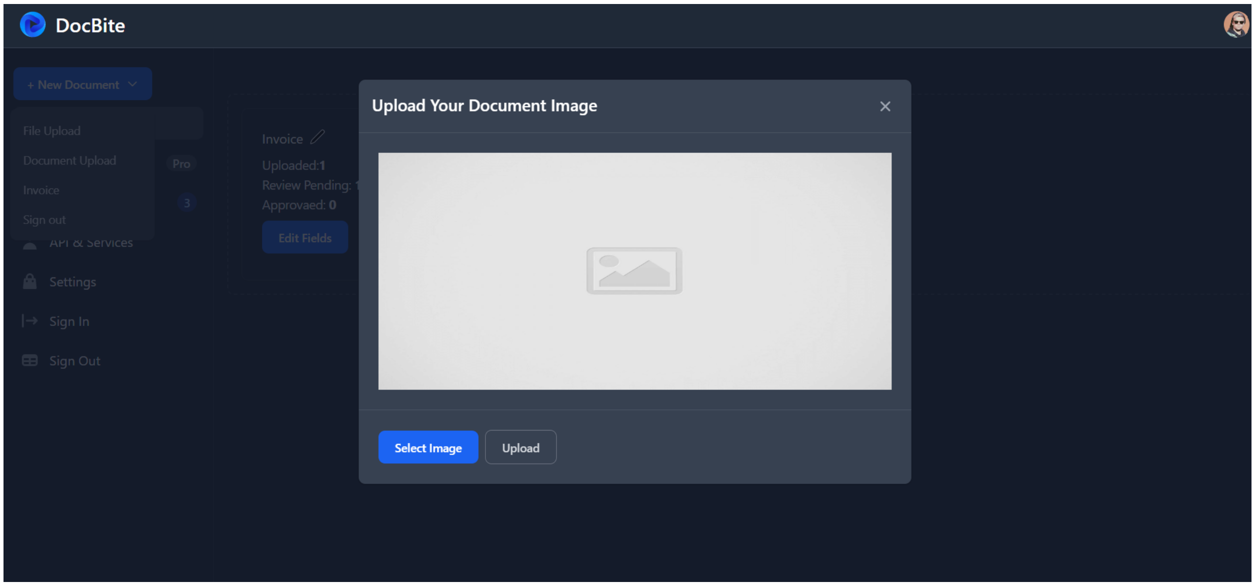


Figure 8.5: Document Upload Page