Entity Detection and Data Extraction using LayoutLMv3 and Tesseract-OCR

Sahran Khuwaja
40196507
Concordia University

Mohammed Misbah Uddin Shareef
40190808
Concordia University

Nadira Anjum Nipa
40262312
Concordia University

Bhawin Bengani
40229723
Concordia University

Abstract

In today's information-driven world, extracting relevant information from unstructured documents is a critical task that demands on cutting-edge techniques in natural language processing (NLP) and computer vision. To tackle this challenge, researchers have leveraged the power of pre-trained models with fine-tuning to achieve state-of-the-art results in document analysis tasks. In this project, we pushed the boundaries of document analysis further by fine-tuning the LayoutLMv3 model on the FUNSD dataset. Our goal was to detect regions of interest (ROI) in unstructured documents and extract valuable data from them using Tesseract-OCR. To achieve this, we relied on the sophisticated Tessarct OCR technology, which helped us transform our ROI data into highly structured and organized information. Overall, our findings demonstrate the incredible potential of advanced models and tools for document analysis and understanding in various fields. The code is publicly available at ***https://github.com/SahranKhuwaja/COMP6341_CV_Project***

1. Introduction

Document analysis and understanding are critical tasks in various fields, including finance, healthcare, legal, and education. One significant challenge in document understanding is extracting relevant information from unstructured documents, which requires advanced techniques in natural language processing (NLP) and computer vision. Recent advances in NLP and deep learning have led to the development of several models for document analysis, including the LayoutLM and LayoutLMv2 models. These models use pre-training and fine-tuning approaches to achieve state-of-the-art results in document analysis tasks.

In this project, we focus on fine-tuning the LayoutLMv3 model on the FUNSD dataset and extracting data from the anootated input using Tesseract-OCR Engine. The FUNSD dataset is a publicly available dataset for document layout analysis that contains over 1,000 annotated forms from various domains, including finance, insurance, and healthcare. The LayoutLMv3 model is a recently proposed model that outperforms previous models on several document analysis tasks, including named entity recognition (NER) and layout analysis

2. Related Work

Pre-trained models have become a popular approach for document analysis, and several studies have demonstrated their effectiveness in various applications. For instance, Xu *et al*. [5] utilized the LayoutLM model on three different datasets to extract text and layouts from image documents. Similarly, some have applied fine-tuning techniques to the LayoutLM model on the SORIE dataset, achieving similar results. Bunch [4], HIRAY [2] fine-tuned the LayoutLM model on SORIE dataset to extract information from receipts data. In our work, we took the latest version of the LayoutLM model, LayoutLMv3, on the FUNSD dataset to detect question-answers from form images. This allowed us to leverage the latest advancements in pre-trained models and tailor them to our specific needs.

3. Application Architecture & Data Flow

Fig. 1 shows the application architecture and flow of data throughout. Each step of the application pipeline is explained in the following subsections.

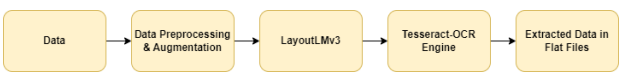

Figure 1. Application Architecture

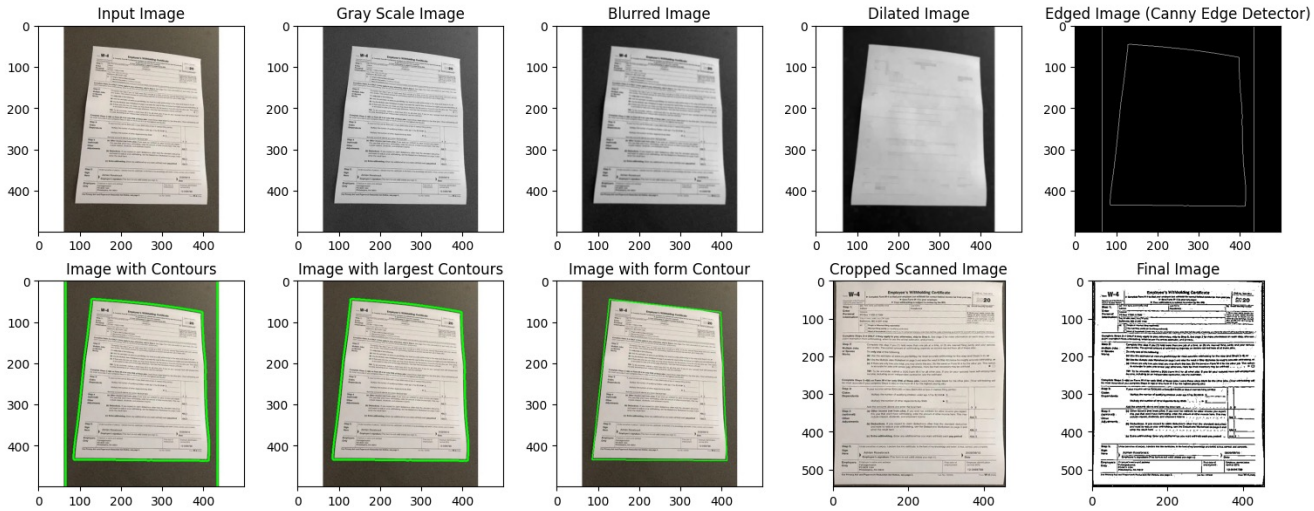

Figure 2. Augmentations & preprocessing applied on input image

3.1. Dataset, Augmentations & Preprocessing

The FUNSD (Forms Understanding and Natural Language Processing in Document Image Analysis) dataset is a publicly available dataset for document layout analysis that contains over 1,000 annotated forms from various domains, including finance, insurance, and healthcare. The dataset includes images of forms and the corresponding annotations, including bounding boxes for each text and image element in the form. We performed the following data augmentation and preprocessing steps:

1. Resizing the image to a suitable size for analysis. This helps to reduce the computation time required for analysis and ensures that the image fits within the memory constraints of the computer.

2. We used contour detection to identify the boundaries of objects in the image. It helps to identify regions of interest and segment the image into smaller regions for analysis.

3. Converting image to greyscale helps to simplify and reduce the number of channels required for analysis. It also helps to reduce the effects of color on the analysis.

4. Applying a low pass filter like gaussian filter to remove noise from the image. It is used to smooth the image and reduce the effects of noise on the analysis.

5. Dilating the image to morphologically expand the boundaries of objects in the image. It is used to detect regions of interest that are connected and help to segment the image into smaller regions for analysis.

6. We used canny edge detector to detect edges in the image. It helps to identify the boundaries of objects in the image and can be used to segment the image into smaller regions for analysis.

7. Cropping the image to extract regions of interest and reduces the computation time required for analysis.

8. Applying perspective restoration on the image to correct the distortion caused by the angle of view of the camera. It helps to improve the accuracy of the analysis.

3.2. Data Annotation using LayoutLMv3

LayoutLMv3 is a state-of-the-art pre-trained model for document understanding tasks such as text recognition, layout analysis, and entity recognition. It is built upon the transformer architecture and uses a combination of image and text embeddings to jointly model text and layout information in documents. The model is pre-trained on a large corpus of documents using a multi-task learning approach, which allows it to learn to perform multiple document understanding tasks simultaneously. It is then fine-tuned on specific downstream tasks such as document classification, form understanding, and receipt recognition.

The applications of LayoutLMv3 involves fine-tuning the pre-trained model on specific document understanding tasks. This can be done by adapting the model to the specific requirements of the downstream task, such as adjusting the output layer, modifying the loss function, and fine-tuning the model on a task-specific dataset. The benefits of using LayoutLMv3 include improved performance on document understanding tasks, reduced need for manual annotation of data, and increased efficiency in document processing tasks.

The LayoutLMv3 architecture consists of a multi-modal encoder for combining text and image embeddings, a task-

specific output layer, and a multi-task loss function that enables the model to learn to perform multiple document understanding tasks.

3.2.1 Model Configuration & Training

We are using the pre-trained base model size of LayoutLMv3 that has a 12-layer Transformer encoder with 12-head self-attention, hidden size of 768, and 3,072 intermediate size of feed-forward networks. For preprocessing the textual input, the model uses Byte-Pair Encoding (BPE) with a maximum sequence length of 512 to tokenize the text sequence. The model adds a [CLS] token at the beginning and a [SEP] token at end of each text sequence. For image embedding, the model parameters are $C * H * W = 3 * 224 * 224$, $P = 16$, $L = 196$. We have modified the model to include an additional classification layer for predicting the label of the text region. The modified model is then frozen, except for the new classification layer, to prevent overfitting.

We evaluate our modified pre-trained base model of LayoutLMv3 performance on FUNSD dataset using cross-entropy loss by training it over 10 epochs and optimizing the AdamW with learning rate of 0.01. Below are tables shows the key metrics we monitored during the modeling process:

Accuracy	83.45%
Precision	84.21%
Recall	90.66%
F1 Score	0.873

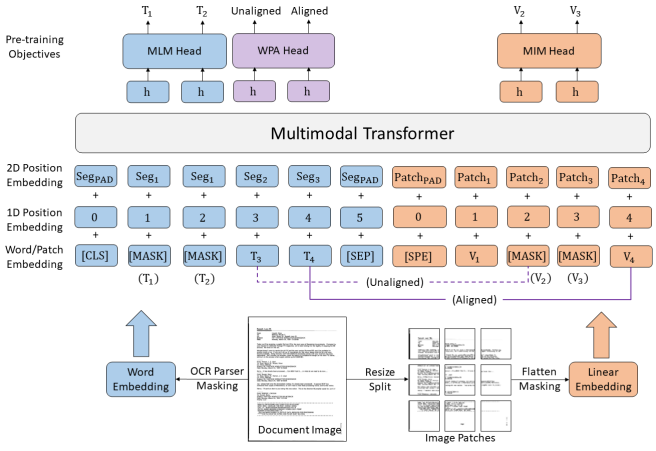

Figure 3. LayoutLMv3 Architecture [1]

3.3. Data Extraction using Tesseract-OCR Engine

Tesseract OCR is a free and open-source optical character recognition (OCR) engine developed by Google [3]. It is designed to recognize text in digital images and convert it into machine-readable text. The architecture of Tesseract-OCR can be divided into two main components: the image processing pipeline and the recognition engine.

The image processing pipeline consists of a series of steps to prepare the input image for recognition. This includes image preprocessing such as binarization, deskewing, and noise reduction. Tesseract OCR uses the Leptonica library for image processing.

The recognition engine uses a combination of traditional OCR algorithms and neural networks to perform text recognition. The engine first performs character segmentation to identify individual characters in the image. It then applies various techniques to recognize each character, including feature extraction, classification, and language modeling. The neural networks in Tesseract OCR are based on the Long Short-Term Memory (LSTM) architecture, which is a type of recurrent neural network (RNN) that is well-suited for sequence modeling.

In our pipeline, we are focuing only on "Question" & "Answer" annotations that are predicted by LayoutLMv3. These annotated images are then passed into Tesseract-OCR which extracts the text sequences from the input. The extracted data is then stored into flat files like CSV.

3.4. Results

In our application, we have used image forms as input, as shown in Fig. 4(a). However, before feeding the images to the model, we performed preprocessing and augmentation to prepare the data for the LayoutLMv3 model, as shown in Fig. 4(b). Next, we used the LayoutLMv3 model to predict the regions of interest (ROI) in the images, which are the bounding boxes for different annotations. This step is shown in Fig. 4(c). The model uses advanced deep learning techniques to accurately identify the different regions in the image that contain the relevant information. Finally, we used Tesseract OCR to extract the data of our interest from different annotations of the images. This step is crucial as it enables us to convert the information contained in the images into a format that can be easily analyzed and processed. The final output of our application is shown in Fig. 4(d), which displays the extracted intended data from the different annotations of the image.

4. Future Work

At present, our application is fine-tuned to extract data of selective annotations like "Question" and "Answer" from forms. In future, we can expand the scope of our application to able extract data of other annotations as well as other

(a) (b)

(c) (d)

	A	B
1	tag	text
2	ANSWER	Asbestos
3	HEADER	CASE FORM
4	QUESTION	DATE FILED:
5	QUESTION	DATE SERVED:
6	QUESTION	CASE TYPE:
7	QUESTION	COUNSEL:
8	QUESTION	TRIAL DATE:
9	ANSWER	Wartnick, Chaber, Harowitz, Smith& TigermanMadelyn J. Chaber4101 California Street, Suite 2200San Franeisco, California 94111415/986-
10	ANSWER	03-Aug-98
11	ANSWER	23-Jul-98
12	ANSWER	Lorillard Tobacco Company
13	ANSWER	San Francisco Superior Court - No. 996378
14	ANSWER	Wanda G. Robinson and Carroll Robinson vy. Raybestos-Manhattan, et al.

Figure 4. (a) Raw Input Image; (b) Augmented & Preprocess Input for LayoutLMv3; (c) LayoutLMv3 Predicted Annotations; (d) Extracted Data

documents like receipts. Additionally, we could try further improving the LayoutLMv3 performance through hyperparameter tuning.

References

[1] 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. 3

[2] Rushikeshhiray. Layoutlm, Apr 2022. 1

[3] Ray Smith. An overview of the tesseract ocr engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE, 2007. 3

[4] Wandb. Wandb/layoutlm$_s roie_d emo$: $Finetunelayoutlmonsroiedatasetusingwamp; btools.1$

[5] 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. 1