	Layout Analysis Models – Assessment report 2024Q1	NTT DATA

Summary

In this document we present the prototype for Layout analysis model generated by NTTDATA GGAO team.

Version	Description	Author	Date Created	Approved by	Date Finished
1.0	Initial Version	NTT DATA	10/06/2024		

1. Introduction

1.1. Motivation

In this document we present the development of a new prototype model for layout analysis. This model's objective is to enhance document processing for consequent downstream tasks such as Question-Answering or document summarization.

1.2. Key Definitions

Large Language Models (LLMs): are advanced AI models trained on vast amounts of text data, capable of understanding and generating human-like text.[1]

Document Layout: the visual design of a document. A layout can be defined as the collective arrangement of information presented on forms, paragraphs, tables, cells, images, logos, etc.[2]

Document Layout Analysis: Task of identifying the physical structure of a document by interpreting content and spatial relationships.

Optical Character Recognition (OCR): is the 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.

2. Document Layout Analysis

2.1. Document AI

Document AI (also known as Document Intelligence) is a field of technology that employs machine learning (ML) techniques, such as natural language processing (NLP) and Computer Vision, to analyze documents in a manner like human review. It is used to extract information from both digital and printed documents, recognizing text, characters, and images in various languages. [3] [4]

Document AI is a powerful technology that can help businesses streamline document processing workflows, improve data accuracy, and enhance decision-making by automating tasks such as data entry, document classification, and form parsing.

Document AI's main tasks include:

1. **Data Extraction:** Extracting relevant information from documents, such as text, numbers, and dates, to create structured data suitable for analysis and consumption. [5]

[1] What Are Large Language Models (LLMs)? | IBM
[2] Introduction to Document Layout - Why OCR Solutions Need it? (docsumo.com)
[3] https://en.wikipedia.org/wiki/Document_AI
[4] https://www.process.st/document-ai/
[5] https://cloud.google.com/document-ai/docs

2. **Document Classification:** Identifying and categorizing different document types, such as tax forms, invoices, and receipts, to facilitate efficient processing and storage.
3. **Form Parsing:** Automatically identifying and extracting relevant information from structured forms, such as tax forms and invoices, using Optical Character Recognition (OCR) and pattern recognition algorithms.
4. **Layout analysis** involves identifying the physical structure of a document by interpreting its content and spatial relationships. This task is crucial for understanding the layout of extracted content and enhancing semantic analysis.

2.2. The task of Document Layout Analysis

The use of Layout analysis models can provide the following advantages and benefits for downstream tasks:

1. **Improved Document Understanding:** Layout analysis enhances semantic analysis by providing a better understanding of the document's structure and content.
2. **Automated Document Processing:** it can help automate tasks like data entry, document classification, and form parsing, streamlining workflows and improving efficiency.
3. **Enhanced Decision-Making:** By providing accurate and structured data, it can support better decision-making in various industries, such as finance, healthcare, and government.

2.2.1. Key models

We have identified two different state-of-the-art approaches based on transformers:

- **Joint text and vision models:** It uses both the text of the document associated with their bounding boxes and image patches extracted from the document page to analyze the layout. That means that it needs some sort of OCR system before analyzing the document. The main model of this approach is **LayoutLMv3.** It is a pre-trained model that jointly models interactions between text and layout information across scanned document images, enhancing document image understanding tasks like information extraction. Additionally, **UDOP model** designed for multimodal generative tasks is composed by an encoder with the same architecture of LayoutLMv3 and a decoder that can generate text, layout or image data.

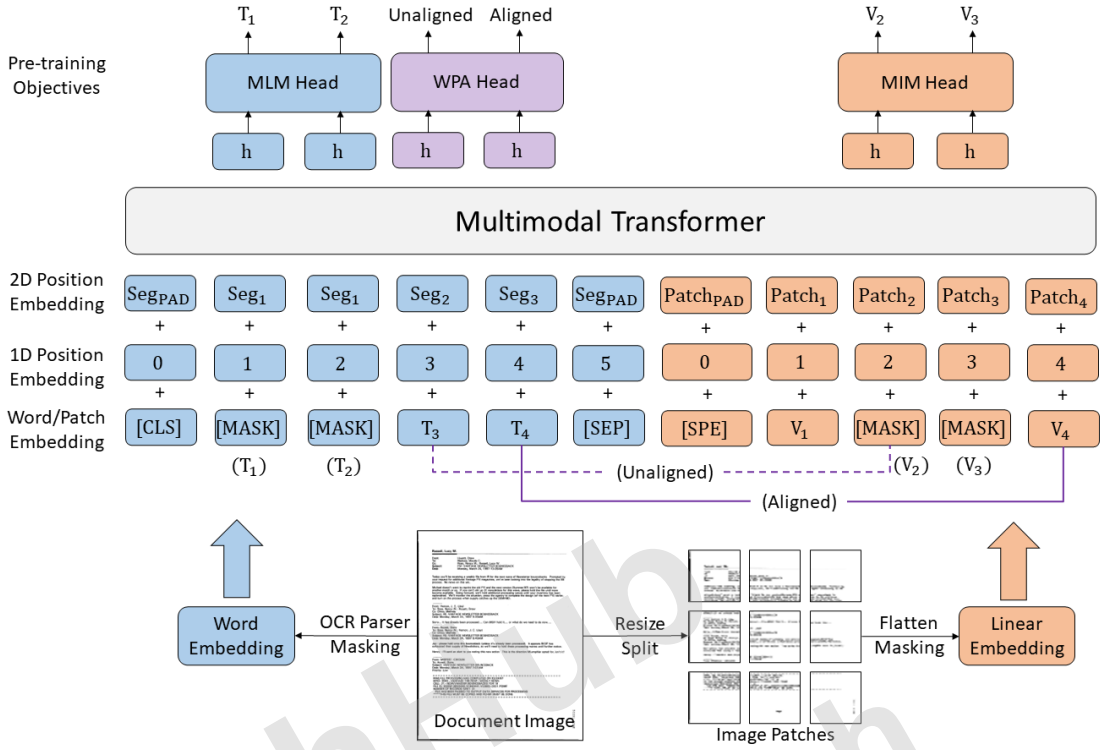


FIGURE 1 LAYOUTLMV3 ARCHITECTURE

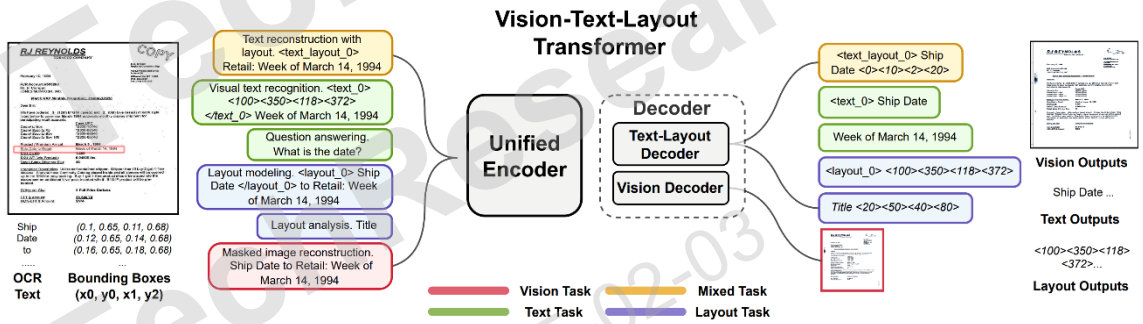


FIGURE 2 UDOP ARCHITECTURE. NOTE THAT THE ANALYSIS IS PERFORMED BY THE ENCODING PART THAT IS SIMILAR TO LAYOUTLMV3.

- **Pure vision models** are models that process each page as an image without explicit text and bounding box information, hence it does not need the use of an OCR step. **DiT** (Document Image Transformer) is one of these pure vision models that uses the Mask R-CNN framework for object detection, achieving high accuracy in tasks like document image classification and table detection. Another example, similar to the case of UDOP in the OCR-based approach, is **DONUT.** This is a generative model based on pure vision encoder and text decoder. In this report, we have not included DONUT in the analysis, but it would be interesting to evaluate it in the future.

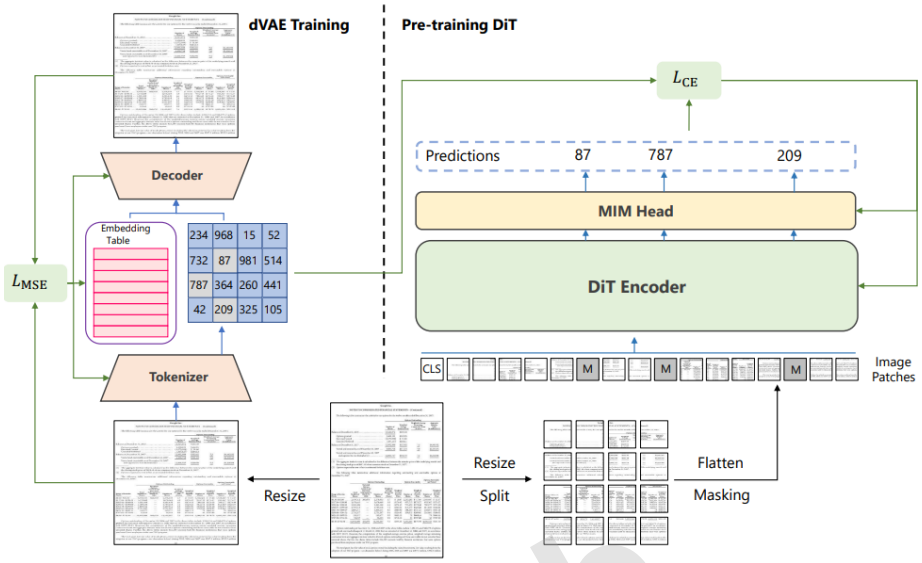


Figure 2: The model architecture of DiT with MIM pre-training.

Recently, there has been presented more models but follow a similar approach and their performance is also comparable.[6]

2.2.2. Evaluation datasets

There are two main training or evaluation datasets for layout analysis:

1. **DocLaynet**[7]: a human-annotated document layout segmentation dataset containing page-by-page layout segmentation ground-truth using bounding-boxes for 11 distinct class labels on 80863 unique pages from 6 document categories.
2. **PubLaynet**[8]: a large dataset of document images for PubMed[9], of which the layout is annotated with both bounding boxes and polygonal segmentations.

3. Comparison of Document AI models for Layout Analysis

3.1. Features of Models

Among the presented models, we focus the following ones for the rest of the report:

- **LayoutLMv3:** Is one of the best performant models according to preliminary test but its Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) license forbids the commercial use of the model.
- **UDOP:** We build an "encoder-only" version of UDOP to try to replicate the behavior of layoutlmv3, as the license of UDOP is less restrictive. We follow the pattern presented in HuggingFace to do so.[10]

[6] PubLayNet val Benchmark (Document Layout Analysis) | Papers With Code
[7] IBM Developer: Doclaynet
[8] ibm-aur-nlp/PubLayNet (github.com)
[9] PubMed Central Open Access Subset (commercial use collection).
[10] Document-AI/UDOP_DocLayNet_Inference.ipynb at main · mit1280/Document-AI (github.com)

- **DiT:** Pure-vision transformer model to check the behavior of a model that does not depends on an OCR, hence can have better performance in figure detection for instance.

In the following table we summarize the main features of each model.

Model Name	UDOP	DiT	LayoutLMv3
Architecture	Vision-Text Layout Transformer (Unified Vision, Text, and Layout Encoder – Vision-Text-Layout Decoder)	Vanilla Vision Transformer Architecture	Text-image multimodal Transformer
Tasks	• Document image classification • Layout analysis • Document visual question answering	• Document image classification • Layout analysis • Table detection	• Document image classification • Layout analysis • Document visual question answering
Pre-training phase	Unlabeled documents and labeled data (pretraining supervised tasks)	Unlabeled text images	Vision language models: pre-trained with MLM (Masked Language Modeling), MIM (Masked Image Modeling) and WPA (Word-Patch Alignment) objectives
Datasets for self-supervised learning	IIT-CDIP, text and token-level bounding boxes extracted by OCR	IIT-CDIP	IIT-CDIP, text and token-level bounding boxes extracted by OCR
Datasets for supervised pretraining tasks	• Classification (RVL-CDIP) • Layout Analysis (PubLayNet) • Information Extraction (DocBank, KLC, PWC and DeepForm) • Question Answering (WebSRC, VisualMRC, DocVQA, InfographicsVQA, WTK) • Document NLI(TabFact)	<No supervised pretraining>	<No supervised pretraining>
License	MIT	MIT[11]	Non-Commercial (CC BY-NC-SA 4.0 DEED)

3.2. Preliminary evaluation

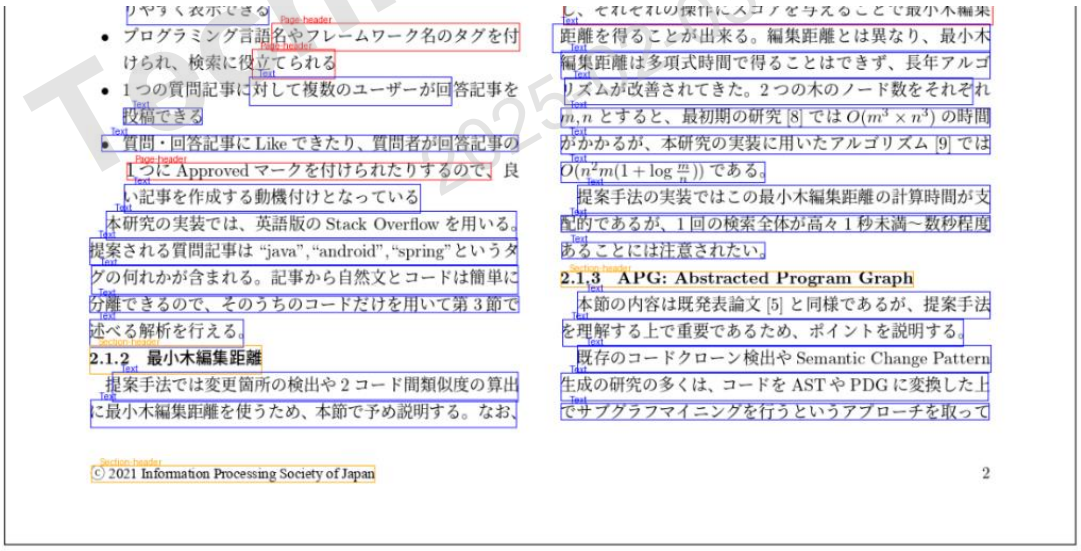
[11] There is not explicit license stated in the folder of this model, hence users assume that MIT license applies (as it is in the root of the repository). Nevertheless, developers have not commented on it when being asked DiT Licence? · Issue #1140 · microsoft/unilm (github.com)

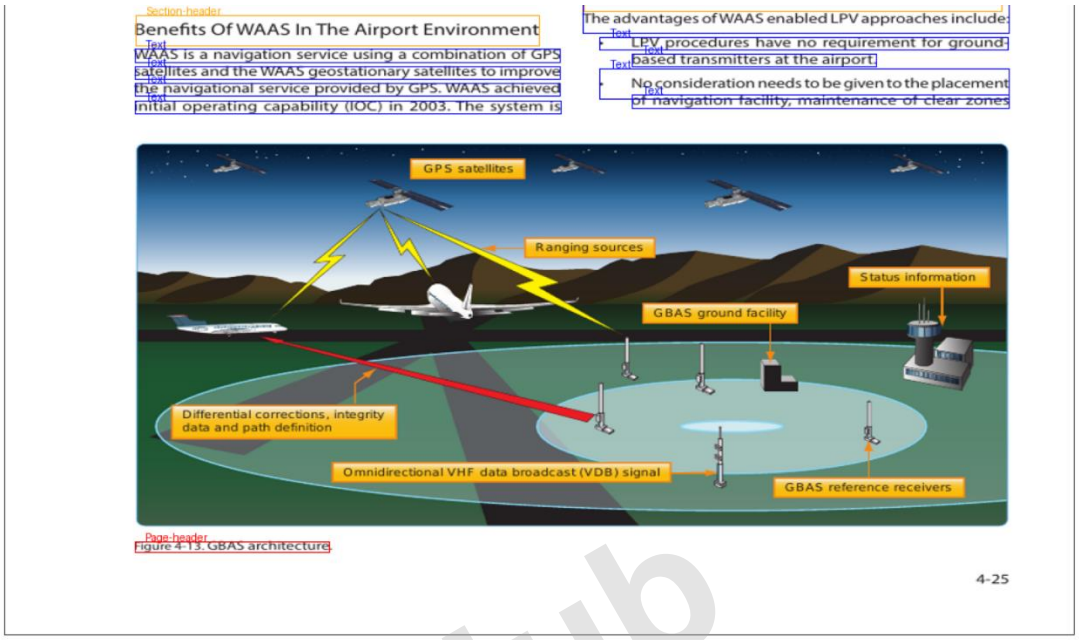
We performed a preliminary evaluation of the available pretrained models. We tested the following configurations:

- **UDOP**
  - UDOP fine-tuned model (HuggingFace) with DocLayNet small dataset (pierreguillou/DocLayNet-small · Datasets at Hugging Face).
  - Bounding box labels: [Caption, Footnote, Formula, List-item, Page-footer, Page-header, Picture, Section-header, Table, Text, Title]
    - Used pytesseract library for OCR with extracted text from each line detected by OCR, then UDOP model is initialized with a bounding box per line and the text detected for each line
- **DiT**
  - Used a fine-tuned model as publaynet using Cascade Mask R-CNN
  - Bounding box labels: [Text, Title, List, Table, Figure].
- **LayoutLMv3**
  - Used LayoutLMv3 fine-tuned model (HuggingFace: Mit1208/layoutlmv3-finetuned-DocLayNet · Hugging Face) with DocLayNet small dataset(pierreguillou/DocLayNet-small · Datasets at Hugging Face).
  - Bounding box labels: [Caption, Footnote, Formula, List-item, Page-footer, Page-header, Picture, Section-header, Table, Text, Title]
  - Used pytesseract library for OCR.

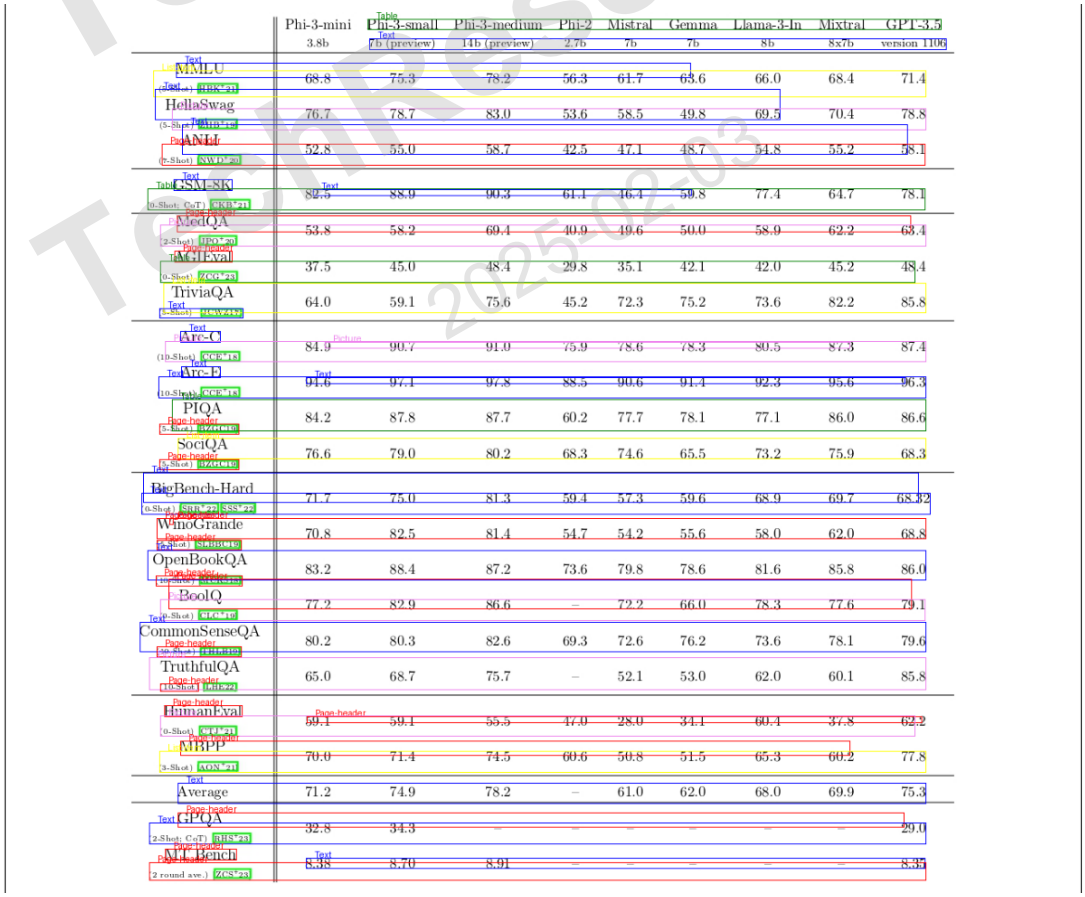
We performed a simple test with 18 documents a got the following preliminary conclusions:

- UDOP and LayoyutLMv3 have text detection in the case of headers, footers and captions.





- DiT was able to detect all the pictures (high recall) but failed in the sense that detected some tables as pictures (low precision). LayoutLMv3 and UDOP were not able to detect pictures, as expected as they are not identified by the OCR.
- Tables are not properly detected in many cases. For the case of UDOP or Layoutlmv3 there are many cells identified as other layout parts: headers, pictures, etc. In the case of DiT, as stated in the previous point, some tables are detected as pictures.



	Phi-3-mini 3.8b	Phi-3-small 7b (preview)	Phi-3-medium 14b (preview)	Phi-2 2.7b	Mistral 7b	Gemma 7b	Llama-3-In 8b	Mixtral 8x7b	GPT-3.5 version 1106
MMLU	68.8	75.3	78.2	56.3	61.7	63.6	66.0	68.4	71.4
HellaSwag	76.7	78.7	83.0	53.6	58.5	49.8	69.5	70.4	78.8
ANLI	52.8	55.0	58.7	42.5	47.1	48.7	54.8	55.2	58.1
GSM-8K	82.5	88.9	90.3	61.1	46.4	59.8	77.4	64.7	78.1
MedQA	53.8	58.2	69.4	40.9	49.6	50.0	58.9	62.2	63.4
AGIEval	37.5	45.0	48.4	29.8	35.1	42.1	42.0	45.2	48.4
TriviaQA	64.0	59.1	75.6	45.2	72.3	75.2	73.6	82.2	85.8
Arc-C	84.9	90.7	91.0	75.9	78.6	78.3	80.5	87.3	87.4
Arc-E	91.6	97.1	97.8	88.5	90.6	91.4	92.3	95.6	96.3
PIQA	84.2	87.8	87.7	60.2	77.7	78.1	77.1	86.0	86.6
SociQA	76.6	79.0	80.2	68.3	74.6	65.5	73.2	75.9	68.3
BigBench-Hard	71.7	75.0	81.3	59.4	57.3	59.6	68.9	69.7	68.32
WinoGrande	70.8	82.5	81.4	54.7	54.2	55.6	58.0	62.0	68.8
OpenBookQA	83.2	88.4	87.2	73.6	79.8	78.6	81.6	85.8	86.0
BoolQ	77.2	82.9	86.6	–	72.2	66.0	78.3	77.6	79.1
CommonSenseQA	80.2	80.3	82.6	69.3	72.6	76.2	73.6	78.1	79.6
TruthfulQA	65.0	68.7	75.7	–	52.1	53.0	62.0	60.1	85.8
HumanEval	59.1	59.1	55.5	47.0	28.0	34.1	60.4	37.8	62.2
MBPP	70.0	71.4	74.5	60.6	50.8	51.5	65.3	60.2	77.8
Average	71.2	74.9	78.2	–	61.0	62.0	68.0	69.9	75.3
GPQA	32.8	34.3	–	–	–	–	–	–	29.0
MT Bench	8.38	8.70	8.91	–	–	–	–	–	8.35

To sum up, DiT provides a way of detecting pictures but the errors with respect to table extraction limit its applicability to general document processing. OCR-based models (LayoutLMv3 and UDOP) fine-tuned to small DocLayNet dataset (the ones that are publicly available) have significant performance issues both in text and table detection. However, previous experience in fine-tuning LayoutLMv3 with the DocLayNet-large dataset showed a much better performance in those tasks.

As LayoutLMv3 is restricted to non-commercial use, we focus on fine-tuning UDOP (whose license is permissive) with DocLayNet-large to generate a marketable model with good text and table detection performance.

4. UDOP Fine-tuning for Layout Analysis

As we concluded before, we opted to fine-tuning UDOP using a large dataset to try to improve the capabilities of available models. We perform the following training process:

- **Model:** UDOP encoding stage used as Token Classification.[12] [13]
- **Training framework:** Huggingface transformers and Pytorch.

Training Arguments:[14]
max_steps=100000,
warmup_ratio=0.1,
per_device_train_batch_size=1,
per_device_eval_batch_size=1,
learning_rate=1e-5,
evaluation_strategy="steps",
eval_steps=1000,
load_best_model_at_end=True,
metric_for_best_model="f1",
greater_is_better = True,
save_total_limit=5,
save_steps=1000

- **Evaluation Metric:** Precision, Recall, F1 and Accuracy over token classification.
- **Dataset:** DocLaynet large.[15] [16]
    - >80k Document images: (69.103 train, 6.480 val, 4.994 test).
    - Vast majority of documents (close to 95%) are published in English language. However, DocLayNet also contains several documents in other languages such as German (2.5%), French (1.0%) and Japanese (1.0%).
    - The pages in DocLayNet can be grouped into six distinct categories, namely Financial Reports, Manuals, Scientific Articles, Laws & Regulations, Patents and Government Tenders.
- Infrastructure:
    - AWS Sagemaker JupyterLab environment.
    - Virtual Machine: ml.g4dn.12xlarge (4 vCPU, 192 GB RAM, 4 NVIDIA T4).[17]
    - Storage 100 GB SSD.

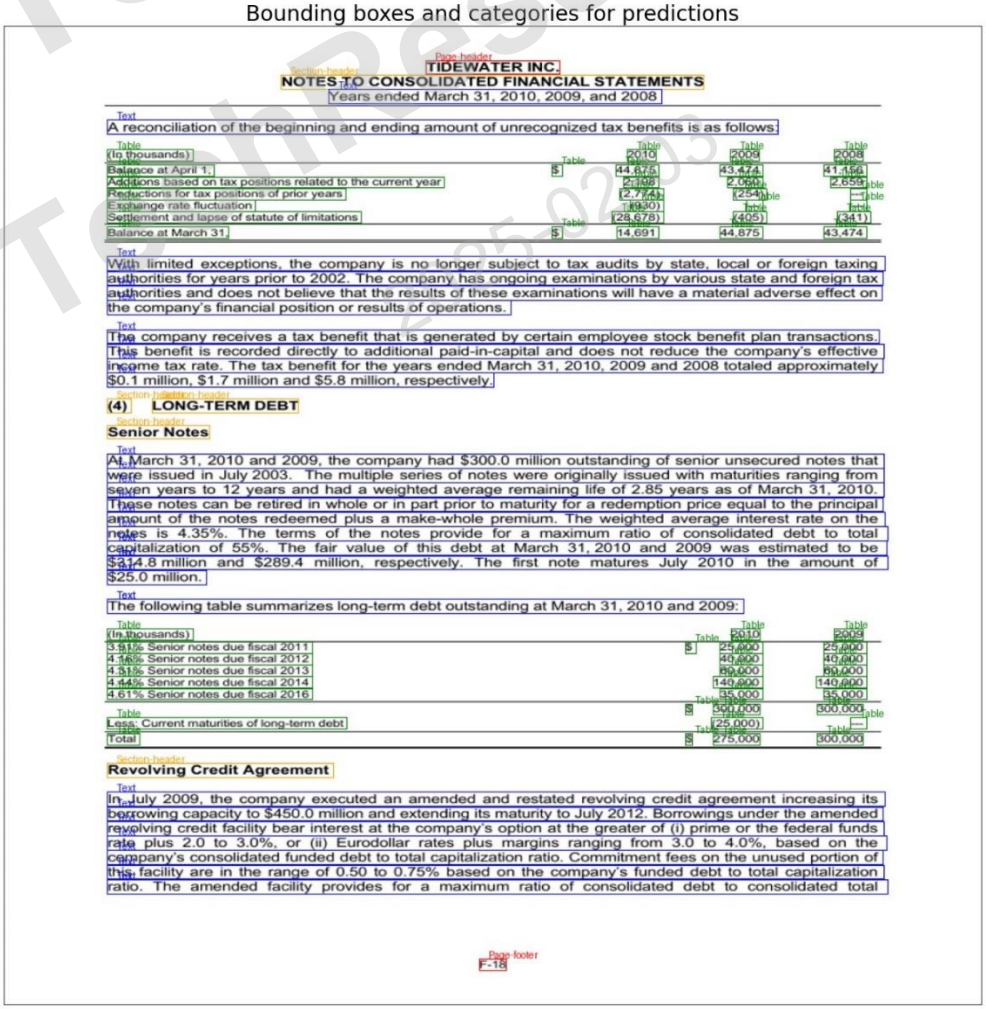
[12] ZinengTang/Udop · Hugging Face
[13] Document-AI/UDOP_DocLayNet_Inference.ipynb at main · mit1280/Document-AI (github.com)
[14] Trainer (huggingface.co)
[15] pierreguillou/DocLayNet-large · Datasets at Hugging Face
[16] [2206.01062] DocLayNet: A Large Human-Annotated Dataset for Document-Layout Analysis (arxiv.org)
[17] Servicio de Machine Learning - Precios de Amazon SageMaker - AWS

In the following table we shown the training process for 64000 iterations.

Step	Training Loss	Validation Loss	Precision	Recall	F1	Accuracy
1000	2.16	1.78	0.00	0.00	0.00	0.67
10000	0.3762	0.400963	0.00	0.00	0.00	0.920733
15000	0.273	0.306035	0.015957	0.006012	0.008734	0.940164
19000	0.2657	0.272541	0.247253	0.270541	0.258373	0.946025
20000	0.2367	0.257646	0.229642	0.282565	0.253369	0.949142
25000	0.1958	0.240489	0.325464	0.386774	0.35348	0.95201
30000	0.1806	0.243452	0.456044	0.498998	0.476555	0.954449
40000	0.174500	0.202635	0.471667	0.567134	0.515014	0.958333
45000	0.160400	**0.220162**	0.514334	0.611222	0.558608	0.960686
50000	0.136600	0.213572	**0.632911**	0.601202	**0.616650**	0.961484
55000	0.153400	0.210427	0.569343	0.625251	0.595989	0.963180
60000	0.119400	0.213144	0.509121	0.615230	0.557169	0.962224
64000	0.140600	0.205972	0.536627	**0.631263**	0.580110	**0.963764**

We performed an initial result inspection and concluded that there is a significant improvement, especially in detected problems: header, footer identification, table extraction, etc.

Bounding boxes and categories for predictions



5. Evaluation of trained model

We performed the following evaluation: The trained models are being tested against Azure Document AI outputs using 93 extracted pages from 21 evaluation documents of different categories. Those documents are private and have not been part of any training dataset. Then, we evaluate the performance of the extraction of the following label categories:

- Caption.
- Footnote.
- List-item.
- Page-header.
- Section-header.
- Title.
- Text.
- Picture.
- Table.

For text labels (Caption, footnote, list-item, page-header, section-header, title and text) we evaluate the F1 metric in terms of all the tokens labelled as such in each page. For the case of pictures, we measure precision at IoU (intersection over union) 75%, i.e., a picture is correctly extracted if the Bounding boxes of the ground truth and the extracted picture overlaps in more than 75% of their union area).

TABLE 1 TEXT METRICS (F1)

	Caption	Footnote	List-item	Page/footer	Page-header	Section-header	Title	Text
LayoutLMv3	0	0.270667	0	0.32634921	0.355209302	0.527561798	0.123	0.806
UDOP	0	0.100818	0	0.26801639	0.201935484	0.500549451	0.03563	0.812086

TABLE 2 FIGURE METRICS

	Average IOU	Precision @ IOU > 0.75
LayoutLMv3	0.206258065	0.121212121
UDOP	0.148145161	0.078125

TABLE 3 TABLE METRICS

	Average IOU	Precision @ IOU>0.75
LayoutLMv3	0.24773684	0.166666667
UDOP	0.42331579	0.265306122

6. Conclusions

In this document we have presented the trained UDOP model for layout analysis. This model is aimed at text block detection and table extraction. Initial review of **obtained results showed a significant improvement** in the results with respect to publicly available ones. Future work will include a more **thorough evaluation and comparison** with respect to other models.