2024 International Conference on Digital Image Computing: Techniques and Applications (DICTA) | 979-8-3503-7903-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/DICTA63115.2024.00087

Enhancing Multimodal Information Extraction from Visually Rich Documents with 2D Positional Embeddings

Aresha Arshad
School of Electrical Engineering and Computer Science (SEECS)
National University of Sciences and Technology (NUST)
Islamabad, Pakistan
aarshad.msds22seecs@seecs.edu.pk

Momina Moetesum
School of Electrical Engineering and Computer Science (SEECS)
National University of Sciences and Technology (NUST)
Islamabad, Pakistan
momina.moetesum@seecs.edu.pk

Adnan ul Hasan
Deep Learning Laboratory, National Center of Artificial Intelligence (NCAI)
Islamabad, Pakistan
adnan.ulhasan@seecs.edu.pk

Faisal Shafait
Deep Learning Laboratory, National Center of Artificial Intelligence (NCAI)
School of Electrical Engineering and Computer Science (SEECS)
National University of Sciences and Technology (NUST)
Islamabad, Pakistan
faisal.shafait@seecs.edu.pk

***Abstract*—Visually rich document understanding involves the interpretation of documents with varied formats and complex layouts, including multi-line entities, presenting a significant challenge. This study addresses these challenges by introducing a document comprehension model based on the LayoutLMv3 architecture, incorporating two-dimensional (2D) positional embeddings to capture both row and column information. Additionally, a multi-stage Transformer network is employed for hierarchical processing of document features. The proposed model is evaluated through extensive experiments on FUNSD dataset, achieving improved performance in both text-centric and image-centric document understanding tasks. Results demonstrate enhanced spatial comprehension and computational efficiency as compared to the state-of-the-art, establishing our approach as a significant contribution to the field of visually rich document understanding.**

***Index Terms*—Visually Rich Documents, Multi-modal Transformer, positional embeddings**

I. Introduction

Visually Rich Documents (VRDs) combine diverse textual and visual elements, such as paragraphs, tables, charts, and images (fig. 1), which are commonly found in domains like finance, medicine, and academia. These documents present challenges for automated systems, as they require understanding of not only the textual content but also the spatial relationships between different visual elements [1]. Pre-trained transformer-based models [3], [12] have recently gained significant attention in Visual Document Understanding (VDU), due to their substantial advancements in document comprehension tasks. Many VDU models have adopted BERT's masked language modeling (MLM) technique [2], designed to learn bidirectional representations for text. However, aligning pre-training objectives for both text and image modalities remains complex. Models such as DocFormer [3] and SelfDoc [6] attempt to

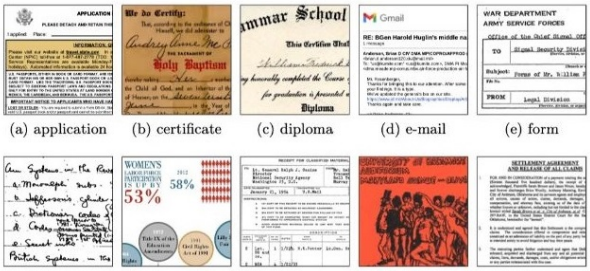


Fig. 1: Samples of Visually Rich Documents [27]

address these challenges through various approaches. For instance, DocFormer emphasizes learning granular features over high-level structures like document layouts by reconstructing image pixels via a CNN decoder [5]. In contrast, SelfDoc addresses more complex tasks by regressing masked region characteristics, which is more challenging than classifying traits from a predefined vocabulary [8]. Nonetheless, issues in accurate comprehension of multimodal entities in VRDs persist.

LayoutLMv3 [23] improves upon these models by integrating multimodal learning, combining masked language and image modeling to better capture the structure and content of VRDs. LayoutLMv3 also uses target image tokens derived from latent codes of a discrete VAE, with each text word corresponding to an image patch, drawing inspiration from models like DALL-E [4] and BEiT [9]. It proposes a Word-Patch Alignment (WPA) objective that predicts whether a text word's associated image patch is masked to achieve cross-modal alignment. Influenced by ViT [10] and ViLT [11],

979-8-3503-7903-7/24/$31.00 ©2024 IEEE
DOI 10.1109/DICTA63115.2024.00087

LayoutLMv3 eliminates the need for complex pre-processing, such as page object detection, by directly utilizing raw image patches from document images. This model is the first multimodal pre-trained VDU model that does not depend on CNNs for image embeddings, saving parameters and eliminating the need for region annotations. Experimental results validate LayoutLMv3's state-of-the-art performance in both text-centric and image-centric VDU tasks [23]. One particular challenge that remains is understanding visually rich multi-line entities. Such entities often present complexities that even LayoutLMv3 struggles to address adequately and therefore requires further exploration.

Recently, there has been growing interest in explainable techniques, leading to the development of methods to interpret model behaviors, especially for specific data modalities such as text and images. Local Interpretable Model-Agnostic Explanations (LIME) [25] is one such popular technique, as it provides insight into specific predictions without necessarily explaining the entire model. This focus on interpretability is even more critical in the complex multimodal context of VDU. However, due to the complex integration of multiple data modalities into a single model, explainability remains underexplored in the domain of VDU. In this paper, we leverage explainability techniques to analyze the performance of the widely-used LayoutLMv3 model on the FUNSD [24] dataset comprising visually rich documents. As a result, it is observed that despite an overall high F1 score, the model performs poorly on multi-line entities like headings, due to insufficient spatial comprehension. Consequently, we propose to employ 2D positional embeddings to better encode the spatial arrangement of patches to capture the representation of entities spanning across multiple lines. The main contributions of this study are as follows:

1) We employ the LIME technique for the explainability of the LayoutLMv3 model's performance on visually rich documents.
2) We incorporate 2D positional embeddings to better encode the spatial arrangement of patches. These embeddings are combined to provide a more comprehensive representation of each patch's location in the document.
3) We explore a multi-stage transformer approach where the model is divided into several stages, each consisting of multiple transformer layers. This hierarchical processing enhances the overall model's performance without compromising efficiency.

The rest of the paper is organized as follows. Section 2 focuses on the relevant literature review. Section 3 covers methodology, which includes image embeddings with 2D positional embeddings, multistage transformers, and explainable AI. Section 4 provides an overview of experiments, Section 5 discusses results, and Section 6 concludes the paper.

II. Literature Review

Various methods have been explored in the literature to fuse image, spatial, and text features for document understanding, especially for extracting information from structurally rich documents like forms, tables, receipts, and invoices. Despite advancements, the optimal fusion of multimodal features remains an open research challenge.

Earlier models focused on region-based features [19]–[21]. SelfDoc [6] introduced a more challenging pre-training task by regressing masked region features, which are noisier and more difficult to learn compared to classifying discrete features within a limited vocabulary [7], [8]. This complexity adds to the difficulty of cross-modal alignment learning, a critical aspect of multimodal representation learning. LayoutLM [13] extended the BERT architecture by integrating 2D spatial coordinate embeddings with text token embeddings, allowing the model to better process spatial layouts in documents. Visual features for each word token, derived using Faster-RCNN, were incorporated alongside bounding box coordinates. Another model called BROS [12], used a BERT-based encoder with a graph-based classifier derived from SPADE [26] to predict entity relationships within documents. Similar to LayoutLM, BROS combined 2D spatial embeddings with text tokens and is tested on various document types, including receipts and forms. Document images imply a fine-grained, word-level alignment relationship between text and image areas. UNITER [19] proposed an optimal transport-based word-region alignment objective, which ViLT [11] further extended to patch-level image embeddings. UDoc [22] effectively aligned images and text using the mask operation provided by MIM.

Recent advances, such as grid-based features [15], have further improved performance by addressing limitations like predefined object classes and regional supervision. Grid-based approaches [3], [14], for example, have been employed for processing invoice images as collected in FUNSD [24] benchmark dataset, where text pixels are represented through character or word vectors and classified into specific field types using convolutional neural networks (CNNs). DocFormer [3] and LayoutLMv2 [14] are popular examples of such VDU models. DocFormer proposed learning of granular features using a CNN decoder by reconstructing image pixels. LayoutLMv2 [14] further improved the performance of LayoutLM by treating visual features as separate tokens instead of embedding them into text tokens. This modification enabled the model to more effectively use unlabeled document data through new pre-training tasks.

For document processing, models like LayoutLMv3 introduced the Word-Patch Alignment (WPA) objective, which predicts whether a text word's corresponding image patch is masked, facilitating better cross-modal alignment. LayoutLMv3 shows the power of masked image modelling (MIM) for linear patch image embedding to construct aligned/unaligned pairs effectively and uniformly. Other approaches, such as masked grid modeling (MGM) in SOHO, predict the mapping index for masked grid features in a visual dictionary [15]. Visual Parsing [18] uses attention weights in self-attention encoders to mask visual tokens, enabling patch-level image embedding.

Additionally, inspired by the Vision Transformer (ViT) [10],

TABLE I: Summary of Popular Models and their Performance of FUNSD Dataset

Model	Parameters	Modality	Image Embedding	FUNSD F1 Score
BROS-base [12]	110M	T+L	None	83.05
LayoutLM-base [13]	160M	T+L+I (R)	ResNet-101 (fine-tune)	79.27
SelfDoc [6]	-	T+L+I (R)	ResNeXt-101	83.36
UDoc [22]	272M	T+L+I (R)	ResNet-50	87.93
DocFormer-base [3]	183M	T+L+I (G)	ResNet-50	83.34
LayoutLMv2-base [14]	200M	T+L+I (G)	ResNeXt101-FPN	82.76
LayoutLMv3-base [23]	133M	T+L+I (P)	Linear	90.29
BROS-large [12]	340M	T+L	None	84.52
LayoutLM-large [13]	343M	T+L	None	77.89
DocFormer-large [3]	536M	T+L+I (G)	ResNet-50	84.55
LayoutLMv2-large [14]	426M	T+L+I (G)	ResNeXt101-FPN	84.20
LayoutLMv3-large [23]	368M	T+L+I (P)	Linear	92.08

Note: "T/L/I" denotes "text/layout/image" modality. "R/G/P" denotes "region/grid/patch" image embedding.

modern latest models have shifted away from CNNs toward using self-attention networks for extracting visual features, improving computational efficiency [16]–[18]. ViLT [11], for instance, learns visual features using a lightweight linear layer, reducing both model size and runtime. Table I summarizes the comparison of state-of-the-art models in this domain, highlighting differences in model size, modalities used, image embeddings, and their performance on the FUNSD dataset [24]. LayoutLMv3, with linear patch embeddings, demonstrates superior performance, achieving the highest F1 score among models that use text, layout, and image modalities.

The current study builds upon these advancements to address the specific challenge of visually complex, multi-line documents. While existing models such as LayoutLMv3 perform well on various tasks, they struggle with multi-line entities, underscoring the need for more advanced techniques in visual document understanding.

III. Methodology

The research proposes a method to extend the capabilities of a document comprehension model, in which the LayoutLMv3 framework has been used. To address the challenges of interpreting visually complex multi-line documents, the model's capacity is enhanced by incorporating 2D positional embeddings of image segments and implementing a multi-stage transformer approach. This development is designed to improve the model's ability to represent spatial configurations within documents while ensuring computational efficiency. Additionally, Explainable AI (XAI) techniques are leveraged to provide transparency and interpretability, offering deeper insights into the model's decision-making processes, particularly when handling visually rich documents. Fig. 2 shows the graphical overview of our proposed methodology. Each step is elaborated in the subsequent sub-sections.

A. Multimodal Transformer

1) Image Embedding with 2D Positional Embeddings: The original LayoutLMv3 uses position embeddings to represent the locations of image patches but struggles with capturing spatial relationships between them. To address this, our proposed method incorporates 2D positional embeddings, using both row and column embeddings to more accurately encode the spatial structure of documents. In this approach, the document image is divided into non-overlapping patches based on a defined patch size. For instance, a 224×224 image segmented with 16×16 pixel patches results in a 14×14 grid. This is done using a 2D convolutional layer with the kernel size and stride set to the patch size, ensuring spatial consistency in patch extraction. A different embedding matrix is generated for the grid, one embedding matrix for each set of row and column positions. The embedding matrix for the row position embeddings has a form of [num_rows, embed_dim], implying that each row in the grid is assigned a different embedding vector of size embed_dim. Similarly, the embedding matrix for column position embeddings takes the form [num_cols, embed_dim], such that each column in the grid is assigned an embedding vector of size embed_dim accordingly. In other words, it means that every coordinate (i, j) in the grid is uniquely defined by a combination of the row position embedding vector, row_pos_embed[i], and the column position embedding vector, col_pos_embed[j]. This results in a process where to get the embedding vector for the patch at position (i, j) in the grid, one either concatenates or performs the sum of these two particular embedding vectors; hence, every coordinate in the grid gains a unique embedding depending on its row and column positions. During the forward pass, the corresponding row and column position embeddings are added to each patch embedding. Specifically, for a patch at position (i, j), its final embedding is computed as shown in eq. 1.

$$\begin{aligned} \text{patch\_embedding}_{i,j} = &\text{patch\_embedding}_{i,j} \\ &+ \text{row\_pos\_embed}[i] \\ &+ \text{col\_pos\_embed}[j] \end{aligned} \tag{1}$$

This addition helps the model to understand the relative positions of patches more effectively, providing richer spatial context.

2) Multi-Stage Transformer Network: The initial LayoutLMv3 model employs a singular extensive Transformer net-

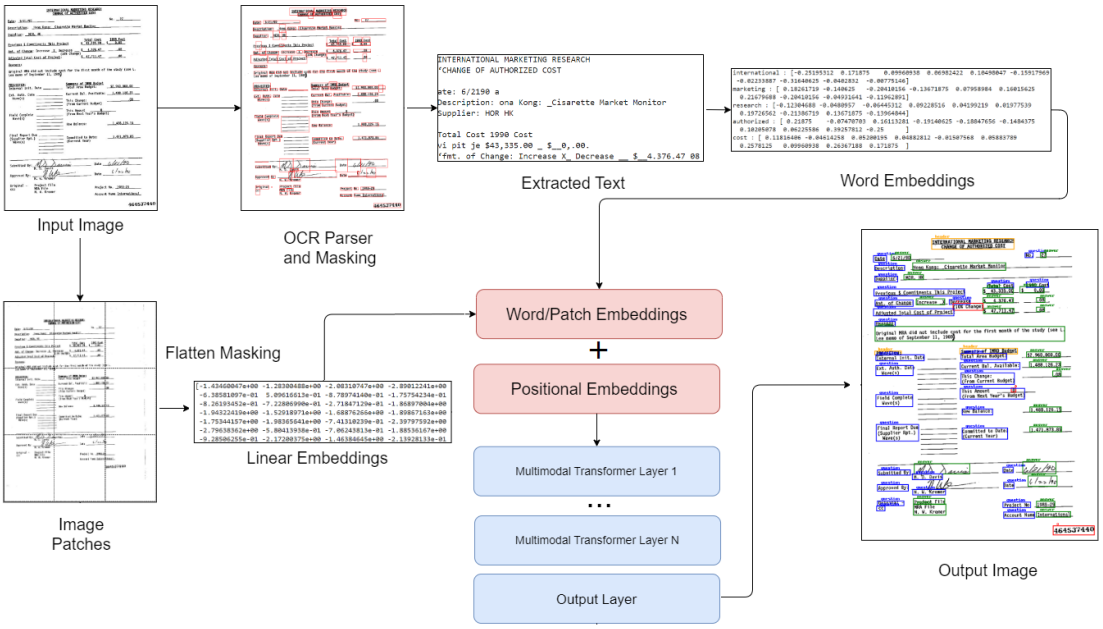


Fig. 2: The proposed model architecture for multimodal document understanding begins by using OCR to extract text and segment the image into patches. Text and image patches are then embedded with 2D positional embeddings to capture spatial relationships. These embeddings are processed through a multi-layer Transformer, designed to identify patterns and enhance information. The final output categorizes various entities present within the document.

work to handle all embeddings, which could pose significant computational demands and potentially overlook hierarchical document patterns efficiently. To address these challenges, we advocate for a multi-stage Transformer methodology. Here, the model is segmented into several stages, each comprising multiple Transformer layers. The early stages concentrate on identifying fundamental patterns, while subsequent stages progressively refine intricate details. This hierarchical processing framework enhances both computational efficiency and overall model performance. The Transformer network is structured into multiple stages, with each stage encompassing a subset of the overall Transformer layers. For instance, in a network composed of 12 layers divided into 3 stages, each stage would consist of 4 layers. The input embeddings are processed sequentially through each stage. The initial stages are responsible for capturing basic patterns and spatial relationships, while the later stages focus on refining these representations. Each stage receives the output from the preceding stage as its input, enabling progressively finer processing of the embeddings. Upon completion of the final stage, the processed embeddings undergo global pooling, employing average pooling, to produce a fixed-size representation. Finally, a classification layer, typically comprising a fully connected layer followed by a softmax activation, is utilized to predict the final output.

B. Explainable AI (XAI)

For explainability, we employ the Local Interpretable Model-agnostic Explanations (LIME). LIME is an interpretability method designed to shed light on the predictions made by complex machine learning models. Its primary objective is to improve transparency and confidence in black-box models by offering human-understandable rationales for specific predictions. LIME achieves this by approximating the decision boundary of a model around a given instance through the construction of interpretable surrogate models that faithfully represent the local behavior of the original model. LIME distinguishes itself as a multimodal explanation tool due to its capability for adaptation into multiple data modalities, which include textual and visual data. This flexibility has high utility in situations where the models handle multiple forms of data, normally images with textual information. A good example is that the LIME technique can explain how changes in parts of images or specific words in the text influence the model's output. This makes LIME highly important for enhancing the interpretability of complex multimodal models, whereby researchers can understand how different elements affect the decisions made by a model across diverse datasets.

IV. Experiments

A series of well-structured experiments were systematically conducted to thoroughly investigate the problem. These experiments were meticulously designed to provide a detailed and comprehensive analysis. The following section offers an overview of the experiments performed.

A. Dataset

The dataset used for the experimentation is known as FUNSD, or Form Understanding of Noisy Scanned Documents focuses on interpreting scanned forms under challenging conditions [24]. FUNSD consists of 199 documents carefully annotated with 9,707 semantic elements, derived from the RVL-CDIP dataset. The main challenge for models using the FUNSD dataset is semantic entity labeling, where each entity is categorized as "question", "answer", "header", and "other". The FUNSD dataset is split into training and test sets consisting of 149 and 50 samples, respectively. Our model aims to classify semantic entities in FUNSD by treating each form as a collection of interconnected semantic entities. In this context, a semantic entity refers to a group of terms that share similar semantic and spatial relevance. Each semantic entity is characterized by several features: a unique identifier, a label indicating whether it is a "question", "answer", "header", or "other", a bounding box, connections to other entities, and a list of constituent words.

B. Hyperparameter Tuning

In our experimentation, we utilized the "microsoft/layoutlmv3-base" checkpoint. We systematically explored the effects of varying hyperparameters, with a particular emphasis on batch size and learning rate. The outcomes provided valuable insights into model performance, summarized in Table II. Notably, the highest F1 scores of 90.64% and 90.71% were achieved with batch sizes of 10 and 12, respectively, both using a learning rate of 1×10^{-5} and training steps of 1000. Conversely, the lowest F1 score of 89.59% was observed with a batch size of 14. These findings highlight the intricate relationship between hyperparameters and model efficacy, suggesting avenues for further refinement and optimization. Table II delineates the performance metrics for four distinct labels: *Question*, *Answer*, *Header*, and *Other*. Notably, as the batch size increases, the F1 scores exhibit a fluctuating trend across the different labels. For instance, the *Question* label attains its highest score of 96.2 when the batch size is set to 10, whereas the *Header* label yields its maximum score of 60 at batch sizes of 8, 10, and 12. Conversely, the *Answer* and *Other* labels demonstrate more varied behavior, with fluctuations observed across the range of batch sizes.

TABLE II: Impact of Batch size on F1 scores of each Label

Batch Size	Question	Answer	Header	Other	Overall
2	93.5	**99.1**	42.8	**90.5**	89.76
8	90.69	93.96	**60**	78.26	90.55
10	**96.2**	92.1	**60**	75.45	90.64
12	91.7	93.6	**60**	76.59	**90.71**
14	90.69	93.96	54.5	75.56	89.59

C. Image Explainer

As discussed earlier, we used LIME to help understand the predictions made by image classification models. Initially, we used `lime_image` for LIME explanations. LIME explanations were computed and visually represented through iteration over each label, aiding in the interpretation of the model's predictions. During the visualization phase, the function generated subplots for each label: the LIME explanations accentuated pertinent areas for the current label's prediction, and marked boundaries delineating regions contributing to the explanation. This systematic approach enabled us to understand the rationale behind the model's predictions across various labels within an image classification task.

D. Upsampling the Data

Experimental analysis revealed that the label "Header" has significantly fewer instances compared to other labels. This imbalance can skew model learning, resulting in suboptimal performance, especially in tasks where accurate representation of all label categories is essential. To address this issue, an upsampling technique is employed. This technique involves synthetically increasing the number of instances in the minority classes to match the count of the majority class. The objective is to create a more balanced dataset where each label category has a similar number of instances, ensuring fair and effective model training. The model iterates through the categorized examples, identifying classes with fewer instances than the desired count. For such classes, random instances are duplicated until they reach the predetermined count. This iterative process ensures that each label category achieves equal representation within the dataset. After upsampling, the examples are shuffled to mitigate any potential biases introduced during the sampling process.

TABLE III: F1 score of Labels after Upsampling

Label	Answer	Question	Header	Other	Overall
F1 Scores	93.4	91.85	63.5	90.5	90.71

Table III presents F1 scores across various label categories after applying an upsampling technique. Each category along with an overall score, showcases the model's ability to classify instances accurately. F1 scores range from 63.5 for "Header" to 93.4 for "Answer", demonstrating the effectiveness of the model in differentiating between label categories. Specifically, the application of upsampling notably improves the "Header" category, increasing its F1 score from 60 to 63.5. The overall F1 score of 90.71 provides a comprehensive view of the model's performance across the entire dataset.

V. Results

A. Analysis of Hyperparameter Tuning

The graph displayed in fig. 3 provides a comprehensive view of how hyperparameter tuning affects the overall F1 score. The model's performance is significantly influenced by the batch size. As depicted in the graph, the model performs optimally with a batch size of 12, achieving an F1 score of 90.72. Experiments were conducted to evaluate the model's effectiveness for different labels: *Header*, *Question*, *Answer*, and *Other*. Extensive tests were performed using consistent

parameters, including a learning rate of 1×10^{-5}, 1000 training steps, and varying batch sizes of 2, 8, 10, 12, and 14. The F1 score was calculated for each label, and the results were analyzed to determine the optimal batch size. The label *Answer* showed the highest F1 score of 99.1 at a batch size of 2, indicating that the model performed exceptionally well in identifying answers. The *Question* label, on the other hand, had a maximum F1 score of 96.2 at a batch size of 10, implying that the model was able to identify questions with a high degree of accuracy. However, it is worth noting that the best overall performance was achieved at a batch size of 12, with optimal label-wise performance. An in-depth analysis of the label-wise performance of the model is provided in Figure 3, which shows that the *Header* label needs improvement. Therefore, the model's performance can be enhanced by improving its ability to identify headers.

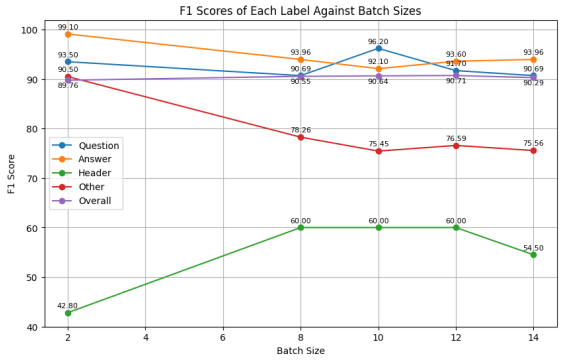


Fig. 3: Impact of Hyperparameter Tuning on F1-Scores.

B. Analyzing the Model's Ability to Process Image

We employed the LIME explainer to visually highlight the areas of an image that contribute to the model's *Header* label predictions, focusing on text objects with specific fonts. These results are presented through marked boundaries on the image, as illustrated in fig. 4. However, the results show that the *Header* label consistently underperforms, with F1 scores ranging from 42.8 to 60.0 across all batch sizes, indicating a significant limitation in the model's prediction accuracy. Our model, enhanced with 2D positional embeddings, demonstrated clear improvements in addressing this issue, particularly in handling spatial information, leading to more accurate *Header* predictions.

C. Results of the Comparison with Other Models

Before testing our proposed model's performance with SOTA, we want to highlight the progressive performance improvements across the LayoutLM, LayoutLMv2, and LayoutLMv3 models on FUNSD samples. The text, layout, and image modalities with linear patch features are integrated with LayoutLMv3, where the CNN backbones are replaced with simple linear embedding to encode image patches. The form understanding task, which involves extracting and structuring

Fig. 4: LIME Visualization on *Header* label

textual content from forms, addresses a sequence labeling problem to tag each word with a label. The focus lies on the semantic entity labeling task within the FUNSD dataset, where each semantic entity is assigned a label such as *question*, *answer*, *header*, or *other*. All experiments were perfomed using the training and test splits comprising 149 and 50 samples, respectively, with officially provided images and OCR annotations used. In the experiment, LayoutLMv3 is fine-tuned for 1,000 steps with a learning rate of 1×10^{-5} and a batch size of 12 for FUNSD. With the base model size, LayoutLMv3 achieves an F1 score of 90.71 on the FUNSD dataset. Notably, LayoutLMv3 employs segment-level layout positions, distinguishing it from other approaches that use word-level layout positions. The results demonstrate that LayoutLMv3 significantly improves text-centric form understanding tasks. Fig. 5 visually represents the *header* label depicted in orange, the *answer* label in green, the *question* label in blue, and the *other* label in lilac. The sample image is a visually rich scanned document from the FUNSD dataset, containing elements such as logos, headers, and computer-written text entities.

D. Results for the Upsampling of Data

The analysis of Table III validates a significant enhancement in the performance of the *Header* label, which exhibited a substantial increase in its F1 score from a baseline of 60.0 to 63.5 after implementing the upsampling technique. This improvement serves as a compelling testament to the efficacy of addressing label imbalance within the dataset. Through the application of upsampling, the representation of minority classes, such as *Header*, is augmented, affording the model a more comprehensive understanding of the intricacies associated with instances labeled as such. Consequently, the model demonstrates an enhanced capability to accurately classify instances from underrepresented categories, ultimately contributing to an overall improvement in performance. This improvement provides compelling evidence for the effectiveness of addressing class imbalance within the dataset. By applying upsampling techniques, the representation of minority classes, such as the *Header* label, is increased, allowing the

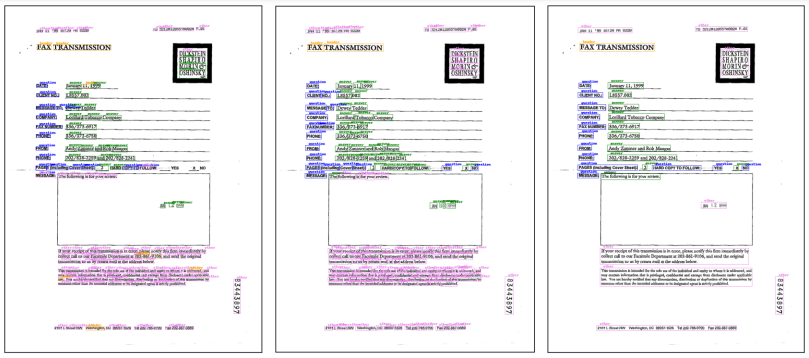


Fig. 5: Visualization of progressive improvements in managing complex document structures and layouts across the models LayoutLM, LayoutLMv2, and LayoutLMv3.

model to better capture the complexities associated with these underrepresented instances. As a result, the model exhibits improved accuracy in classifying instances from imbalanced categories, leading to an overall enhancement in performance. The substantial increase in the F1 score for the *Header* label underscores the importance of mitigating label imbalance. Focusing on undersampled labels through upsampling enhances the model's ability to learn finer details of minority classes, thereby improving classification accuracy across all labels. The observed improvement in test accuracy confirms the efficacy of the upsampling strategy in boosting the model's performance on datasets with skewed label distributions.

VI. Comparison of Base with Improved Model

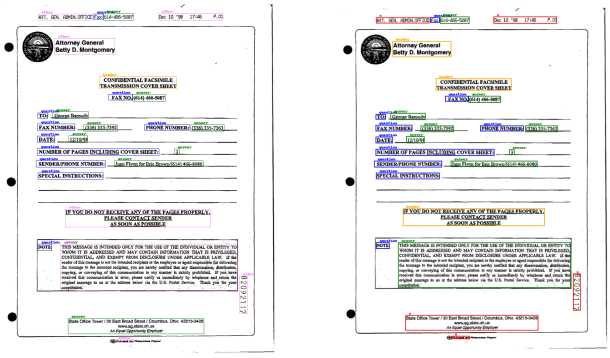
This section compares the performance of our proposed model against various versions of the LayoutLM model in entity recognition tasks, highlighting the specific enhancements in LayoutLMv3. The LayoutLM models are designed to understand visually rich documents by incorporating image region features and spatial layout information. LayoutLMv2 introduced a multi-modal Transformer that integrates text, visual, and layout data using CNN-based visual encoders and token-level positional embeddings. LayoutLMv3 further improves upon this with a spatially-aware self-attention mechanism, enhancing its ability to model relationships between input tokens and their relative positions, thus making it highly effective for document understanding tasks.

TABLE IV: Entity-wise Performance Comparison of LayoutLM Models and Proposed Model.

Model	Header	Answer	Question	Overall
LayoutLM	44.5	74.99	77.76	79.27
LayoutLMv2	50.0	91.60	90.88	82.76
LayoutLMv3 (Base Model)	60.0	93.6	90.69	90.29
Proposed Model	**63.5**	**93.4**	**91.8**	**90.71**

Note: The table represents the F1 scores for each entity.

While models like LayoutLMv3 and ViT rely on 1D position embeddings for patches that offer only linear position information, this enhanced model uses 2D position embeddings



(a) Base Model Result (b) Improved Model Result

Fig. 6: Qualitative performance comparison of our proposed model with LayoutLMv3 showing improved recognition of multi-line headers in documents.

to capture both row and column information, offering richer spatial context. This enhancement enables the model to better represent document layouts and structures, which translates into more accurate information extraction and layout analysis.

In contrast, the model focuses on efficiency by using a multi-stage Transformer network that hierarchically processes embeddings. This is supposed to be computationally intensive for single large transformer networks, which have limited hierarchical pattern recognition. Secondly, this model integrates 2D position embeddings, differentiating between the position of each row and column to capture the accurate location of each patch within a document. It segments the document image into non-overlapping patches using a 2D convolutional layer, while separately initializing two different embedding matrices for the row and column positions. The multi-step process makes it more computationally efficient; hence, the model can learn more complicated representations, accelerate processing for long documents, and optimize resource utilization, which is suitable for real-time applications.

TABLE V: Comparison of the Proposed Model with SOTA

Model	Modality	Image Embedding	F1 Scores
LayoutLMbase	T+L+I (R)	ResNet-101	79.27
SelfDoc	T+L+I (R)	ResNeXt-101	83.36
LaoutLMv2base	T+L+I (G)	ResNeXt101-FPN	82.76
DocFormerbase	T+L+I (G)	ResNet-50	83.34
LayoutLMv3base	T+L+I (P)	Linear	90.29
Proposed Model	**T+L+I (P)**	**Linear + Positional**	**90.71**

Fig. 6 and Table IV illustrate the enhanced results of the proposed model as compared to base models of LayoutLM, especially highlighting a significant difference in detecting the "Header" label more accurately in our case. Comparison of F1 scores among several state-of-the-art models as shown in the Table V also shows that the proposed model has the highest score compared to other SOTA models such as DocFormer and SelfDoc, in addition to LayoutLM versions.

VII. Conclusion

Our proposed enhancements to the LayoutLMv3 architecture significantly improve both spatial understanding and computational efficiency for visually rich document understanding. By using 2D positional embeddings for image patches, the model captures spatial information from both rows and columns, enhancing its ability to handle complex document layouts. The multi-stage Transformer architecture further strengthens this capability by hierarchically encoding sophisticated patterns, improving both accuracy and efficiency in processing long documents. These advancements make the model particularly effective for tasks like form and receipt recognition. The results demonstrate its potential for future applications in data extraction, document accessibility, and usability across various sectors.

References

[1] Ding, Y., Lee, J., & Han, S. C. (2024). Deep Learning based Visually Rich Document Content Understanding: A Survey. arXiv preprint (arXiv:2408.01287).

[2] Devlin, J. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint (arXiv:1810.04805).

[3] Appalaraju, S., Jasani, B., Kota, B. U., Xie, Y., & Manmatha, R. (2021). Docformer: End-to-end transformer for document understanding. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 993-1003).

[4] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M.,& Sutskever, I. (2021, July). Zero-shot text-to-image generation. In International conference on machine learning (pp. 8821-8831). Pmlr.

[5] Salimans, T., Karpathy, A., Chen, X., & Kingma, D. P. (2017). Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. arXiv preprint (arXiv:1701.05517).

[6] Li, P., Gu, J., Kuen, J., Morariu, V. I., Zhao, H., Jain, R., Manjunatha, V., & Liu, H. (2021). Selfdoc: Self-supervised document representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5652-5660).

[7] Cho, J., Lu, J., Schwenk, D., Hajishirzi, H., & Kembhavi, A. (2020). X-lxmert: Paint, caption and answer questions with multi-modal transformers. arXiv preprint (arXiv:2009.11278).

[8] Huang, Y., Xue, H., Liu, B., & Lu, Y. (2021, October). Unifying multimodal transformer for bi-directional image and text generation. In Proceedings of the 29th ACM International Conference on Multimedia (pp. 1138-1147).

[9] Bao, H., Dong, L., Piao, S., & Wei, F. (2021). Beit: Bert pre-training of image transformers. arXiv preprint (arXiv:2106.08254).

[10] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv preprint (arXiv:2010.11929).

[11] Kim, W., Son, B., & Kim, I. (2021, July). Vilt: Vision-and-language transformer without convolution or region supervision. In International conference on machine learning (pp. 5583-5594). PMLR.

[12] Hong, T., Kim, D., Ji, M., Hwang, W., Nam, D., & Park, S. (2022, June). Bros: A pre-trained language model focusing on text and layout for better key information extraction from documents. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 36, No. 10, pp. 10767-10775).

[13] Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., & Zhou, M. (2020, August). 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 (pp. 1192-1200).

[14] Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., Lu, Y., Florencio, D., Zhang, C., Che, W., Zhang, M., & Zhou, L. (2020). LayoutLMv2: Multi-modal Pre-training for Visually-Rich Document Understanding. arXiv preprint (arXiv:2012.14740).

[15] Huang, Z., Zeng, Z., Huang, Y., Liu, B., Fu, D., & Fu, J. (2021). Seeing out of the box: End-to-end pre-training for vision-language representation learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 12976-12985).

[16] Dou, Z., Xu, Y., Gan, Z., Wang, J., Wang, S., Wang, L., Zhu, C., Zhang, P., Yuan, L., Peng, N., Liu, Z., & Zeng, M. (2022). An empirical study of training end-to-end vision-and-language transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 18166-18176).

[17] Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., & Hoi, S. C. H. (2021). Align before fuse: Vision and language representation learning with momentum distillation. Advances in neural information processing systems, 34, 9694-9705.

[18] Xue, H., Huang, Y., Liu, B., Peng, H., Fu, J., Li, H., & Luo, J. (2021). Probing inter-modality: Visual parsing with self-attention for vision-and-language pre-training. Advances in Neural Information Processing Systems, 34, 4514-4528.

[19] Chen, Y., Li, L., Yu, L., Kholy, A. E., Ahmed, F., Gan, Z., Cheng, Y., & Liu, J. (2019). Uniter: Universal image-text representation learning. In European conference on computer vision (pp. 104-120). Cham: Springer International Publishing.

[20] Su, W., Zhu, X., Cao, Y., Li, B., Lu, L., Wei, F., & Dai, J. (2019). Vl-bert: Pre-training of generic visual-linguistic representations. arXiv preprint (arXiv:1908.08530).

[21] Tan, H., & Bansal, M. (2019). Lxmert: Learning cross-modality encoder representations from transformers. arXiv preprint (arXiv:1908.07490).

[22] Gu, J., Kuen, J., Morariu, V. I., Zhao, H., Barmpalios, N., Jain, R., Nenkova, A., & Sun, T. (2023). Unified pretraining framework for document understanding. U.S. Patent Application No. 17/528,061.

[23] Huang, Y., Lv, T., Cui, L., Lu, Y., & Wei, F. (2022, October). Layoutlmv3: Pre-training for document ai with unified text and image masking. In Proceedings of the 30th ACM International Conference on Multimedia (pp. 4083-4091).

[24] Jaume, G., Ekenel, H. K., & Thiran, J. P. (2019, September). Funsd: A dataset for form understanding in noisy scanned documents. In 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW) (Vol. 2, pp. 1-6). IEEE.

[25] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). Why should i trust you? Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135-1144).

[26] Park, T., Liu, M. Y., Wang, T. C., & Zhu, J. Y. (2019). Semantic image synthesis with spatially-adaptive normalization. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 2337-2346).

[27] Van Landeghem, J., Tito, R., Borchmann, Ł., Pietruszka, M., Jurkiewicz, D., Powalski, R., Józiak, P., Biswas, S., Coustaty, M., & Stanisławek, T., (2023). ICDAR 2023 competition on document understanding of everything (DUDE). In International Conference on Document Analysis and Recognition (pp. 420-434). Cham: Springer Nature Switzerland.