LayoutQT - Layout Quadrant Tags to embed Visual Features for Document Analysis

Patricia Medyna Lauritzen de Lucena Drumond, Lindeberg Pessoa Leite, Teofilo E. de Campos, Fabricio Braz

Universidade Federal do Piauí (UFPI), Picos, Brazil
Departamento de Ciencia da Computacao (CIC / IE) and Gama Campus (FGA), Universidade de Brasilia, Brazil

Abstract

The relative position of text blocks plays a crucial role in document understanding. However, the task of embedding layout information in the representation of a page instance is not trivial. Computer Vision and Natural Language Processing techniques have been advancing in extracting content from document images considering layout features. We propose a set of Layout Quadrant Tags (LayoutQT) as a new way of encoding layout information in textual embedding. We show that this enables a standard NLP pipeline to be significantly enhanced without requiring expensive mid or high-level multimodal fusion. Given that our focus is on developing a low computational cost solution, we focused our experiments on the AWD-LSTM neural network. We evaluated our method for page stream segmentation and document classification tasks on two datasets, Tobacco800 and RVL-CDIP. In the former, our method improved the F1 score from 97.9% to 99.1% and in the latter the F1 score went from 80.4% to 83.6%. Similar levels of performance improvement were also obtained when we applied LayoutQT with BERT.

Keywords: Document Representation, Information Extraction, Document Classification, Document Image Layout Analysis

1. Introduction

Organizations produce business documents daily that are essential for their transactions. These documents include purchase orders, reports, sales agreements, supplier contracts, letters, invoices, receipts, and resumes. In addition, information in business documents is presented in various ways, from

Preprint submitted to International Scientific Journal Engineering Applications of AI *November 15, 2022*

plain text to multi-column formats and a wide variety of tables. However, the huge amount of digitized documents produced in the last decades requires a significant effort in the development of methods of processing images of documents for information extraction.

Several document image processing systems have been proposed using machine and deep learning models for downstream tasks, considering only visual resources [1, 2, 3], only textual resources [4, 5] or combining both features [6, 7, 8, 9] to extract information from documents. One of the main problems with processing document images is the huge variety of layout formats used in practice [10]. Document layout analysis requires understanding texts in various layouts, and a combination of computer vision and natural language processing techniques is needed. Computer vision techniques used to be burdensome, but methods and hardware have evolved, enabling their ubiquitous applications even for real-time applications, including embedded person detection in cameras and crowd analysis systems [11].

Document image processing deals with understanding document page layout, which includes structural information and some visual and specific models (for example, from the source and geographic position of the text). Furthermore, extracting textual content from documents that have been scanned or photographed is a complex task due to the loss of quality. Document layout analysis (DLA) is an important step in developing an effective and complete document image processing system and enables many important applications, such as document retrieval, digitization, and editing [12]. DLA is a segmentation process that separates the image of a scanned document into its structural elements and classifies them. The segmentation obtained can be combined with the textual information contained in the blocks detected [13]. Layout analysis methods have been actively studied in document analysis and recognition.

The methods can be divided into two main categories: appearance-based analysis and semantic-based analysis. Appearance-based analysis refers to page segmentation to distinguish text regions from non-text regions like figures, tables, symbols and line segments. On the other hand, the semantic-based analysis, often referred to as logical structure analysis, categorizes each region into semantically-relevant classes (e.g., caption, paragraph separation, headings) using a rule-based model [12]. Compositing text and non-text regions in the document image layout adds an extra burden to managing layout analysis methods. This composition can cause a compromised system in terms of accuracy or a high computational cost [14]. A major challenge for researchers is how to efficiently explore textual and visual features of documents for a richer semantic representation to extract information unambiguously. Another challenge is the scalability of the method [15].

In this paper, we present LayoutQT - Layout Quadrant Tags, a lightweight preprocessing method focusing on combinations of texts and their spatial information without relying on visual features or activations from the visual modalities. Specifically, we propose a new set of tokens that encode spatial regions language models and show that they improve results in downstream tasks with low computational cost. We evaluated our method with page stream segmentation and document classification task with Tobacco800 and RVL-CDIP datasets, respectively.

Our main contributions are:

- A novel approach to fuse textual and layout information which exploits a by-product of the text digitalization process, incurring in insignificant additional computational cost.
- The simple yet effective fusion of textual and layout features for extracting information from documents, which consists in injecting spatial tokens related to text block positions.
- The source code of our library, which is available from `https://github.com/fabraz/docSilhouette` and the package on `https://pypi.org/project/docSilhouette`. It can be used immediately in the engineering of other products.

The rest of this paper is organized as follows. In Section 2, we examine previous work on multimodal document classification. In Section 3, we describe our approach, LayoutQT. In Section 4, we detail the datasets used in this paper. Next, we describe the experiments and discuss the results in Sections 5 and 6. Finally, we conclude the paper in section 7.

2. Related Works

In recent research, several studies have approached extracting visual and textual features for the downstream tasks. CharGrid [16] and its extensions [17, 18] assume the layout contents are visually interpreted via computer vision techniques such as OCR and proposed learning frameworks to understand the documents from a 2D aspect semantically. Bakkali et al. (2020)

[19] presented a hybrid cross-modal feature learning approach that combines image features and text embedding to classify document images.

Aggarwal et al. (2020) [20] proposed a hierarchical multi-modal bottom-up approach to detect larger constructs in a form page. Specifically for the task of extracting higher-order constructs from lower-level elements. However, this method shows insufficient capabilities in layout modeling. Li et al. (2021) [21] proposed the VTLayout model to locate and identify different category blocks by merging the document's deep visual, shallow visual, and text features. First, it applies the Cascade Mask R-CNN model to find all the document category blocks. Then, the deep visual, shallow visual, and text features are extracted for fusion to classifier the category blocks of documents.

Furthermore, Natural Language Processing has advanced with representations of contextualized embedding. The emergence of the Transformer architecture [22] has boosted the creation of language models like BERT [4] that are used as pre-training strategies using visual and textual features for downstream tasks. Lu et al. (2019) [23] developed ViLBERT, a model for learning task-agnostic joint representations of image content and natural language. They extended the popular BERT architecture to a multi-modal two-stream model, processing visual and textual inputs in separate streams that interact through co-attentional transformer layers.

LayoutLM [8] model is proposed as the pioneer pre-training method of text and layout for document image understanding tasks, which expands 1D positional encoding of BERT to 2D to avoid the loss of layout information. Image embeddings are combined in the fine-tuning stage, and the image information is integrated into the pre-training stage. Subsequently, several pre-trained language models were developed by combining additional visual features to improve results [9, 24, 25].

Unlike LayoutLM, StructuralLM [26] is a structural pre-training approach that jointly exploits cell and layout information from scanned documents. It uses cell-level 2D-position embeddings with tokens in the cell sharing the same 2D position. LAMPreT was proposed by Wu et al. (2021)[27] to explore both the structure and the content of documents and consider image content to learn a multi-modal document representation. BROS [28] encode relative positions of texts between text blocks in 2D space, focusing on the combinations of texts and their spatial information without relying on visual features.

These deep learning-based algorithms contain large numbers of trainable model parameters, which require a significant amount of training data and lead to an increase in computation time to train the classifier [29]. A major drawback of such pre-trained models based on the Transformer architecture [22] is that they require a high computational cost. Unlike these previous methods, the approach in this paper aims to improve the performance of language models by combining texts and their spatial information with a low computational cost. Specifically, we propose a spatial layout encoding method that combines text blocks' textual and spatial information.

3. Method

This section discusses our approach to creating a document representation that encodes layout features alongside textual tokens. We use a method to detect text blocks on the document page and use quadrants to compose spatial tokens for a joint textual and layout representation.

3.1. Preprocessing LayoutQT

Our algorithm is based on a bottom-up approach, which defines primitive components to start the clustering process. It starts with the bounding box of words as a primitive component of the page. The word grouping process identifies a group of nearest neighbours of each bounding box to form lines and blocks of text until the page end. Furthermore, each document page is divided into rectangular regions with the same *height* and *width* dimensions. Each quadrant has layout location information that is represented by spatial tokens.

Spatial tokens are added at the beginning and end of each line when indicating the quantized coordinates of the bounding box that the line belongs to. The text group beginning tag considers the distances from the top left corner of the bounding box to the image's left edge and top edge. Likewise, the end tag considers the distance between the bottom right corner of the bounding box and the image's bottom edge and right edge. Table 1 presents spatial tokens and their descriptions used in our LayoutQT model. For example, the beginning of a text block is marked with $xxQw_i\_h_j$ *xxbob* to indicate the position (quadrant) of the beginning of the bounding box text. The centralized parts of the text are also marked with spatial tokens *xxeob* and *xxbcet*.

LayoutQT's algorithm (3.1) takes single-page or multi-page documents as input and generates tokenized text t with layout information. The algorithm

Table 1: **Descriptions** of the spatial tokens

Special Token	**Descriptions**
$xxPn_k$	Page_Number
$xxbob$	Begin_Of_Block
$xxeob$	End_Of_Block
$xxbcet$	Begin_Of_Centered_Text
$xxecet$	End_Of_Centered_Text
$xxQw_i\_h_j$	Quadrant-w_iRow-h_jColumn

scans the page from top to bottom and left to right to find the boundaries of text groups and identify the group's top left corner. Initially, it adds a spatial token to the text to indicate the page. It then starts using an OCR engine [30] to generate word bounding boxes. For that, we used the combination of heuristics that is included in the Tesseract package, but more modern techniques can be applied by using an object detection neural network trained to detect the bounding boxes of textual elements. An example of such networks is the series of YOLO networks, which was originally proposed for object detection benchmarks [31] then it has been adapted for all sorts of objects, including human body parts [32] and even tomatoes [33].

Having obtained textual bounding boxes, our algorithm exploits their coordinates by injecting that information through the spatial tokens. It sorts the groups that belong to the same column on the page to check which groups are centralized and adds the tokens. Moreover, it ends by adding the end-of-group spatial token. The text extraction with spatial tags is saved in a text file.

Figure 1 presents a visual illustration from LayoutQT to the document page. The document input image is divided into quadrants and text groups on the left. Each row is numbered from left to right, and each column is numbered from top to bottom, so the tags of the first and last quadrants are, respectively, $xxQ00\_00$ and $xxQn-1\_m-1$. Inspired by the tokenization of Fastai [34], which adds spatial tokens at the beginning and end of the sentence, LayoutQT adds spatial tokens with information about the bounding box position. All spatial tokens start with the character xx, which is not a common English word prefix. They are added using rules for the model to recognize the important parts of a text. The image of the text file tokenized by our model is on the right side.

Algorithm 1 LayoutQT Algorithm

Input: multi page document

Output: tokenized text t

```
for page = 0, ..., N − 1 do
    t+ = add page token (where + = means insert symbol in string t)
    triage each word bounding boxes into line and group
    triage groups into coherent page columns
    for each group do
        t+ = quadrant coordinate of group top left corner
        for each text line in this group do
            check line centralization w.r.t. its page column center position
            if the line is centralized then
                t+ = centre tag
            end if
            t+ = textual contents of the line
            if the line is centralized then
                t+ = centre tag
            end if
        end for
        t+ = quadrant coordinate of group bottom right corner
    end for
end for
```

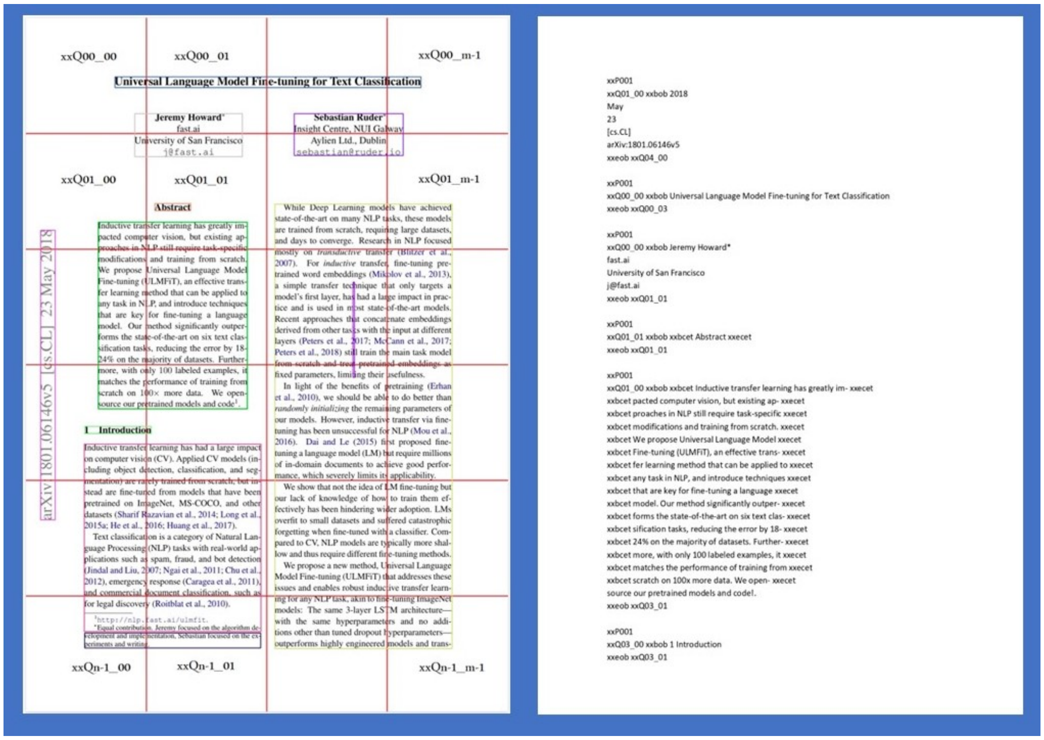

Figure 1: Illustration of Layout Quadrant Tags (LayoutQT). The rectangles represent the bounding boxes of text. On the left side, an input document is divided into quadrants and receives spatial tokens $xxQw_i\_h_j$ according to row i and column j positions. On the right side is the text extracted by the OCR system, with the tags indicating the position (quadrant) of each text block's beginning and end.

3.2. Long Short Term Memory networks (LSTMs)

To prove the efficiency of our encoding method, we chose to perform most of our experiments using the simplest contemporary textual analysis tool, an LSTM network. LSTMs [35] are a special kind of recurrent neural nets (RNNs) capable of learning long-term dependencies. They work tremendously well on many problems and are now widely used in NLP. LSTMs are explicitly designed to handle the long-term dependency problem, as remembering information for long periods is practically their default behaviour.

In addition, to performing experiments with a vanilla LSTM architecture, we evaluated the ASGD Weight-Dropped LSTM [36], a.k.a. AWD-LSTM. It is a recurrent neural network that employs a strategy DropConnect mask on the hidden-to-hidden weight matrices to prevent over-fitting across the recurrent connections. We used that architecture as the backbone of a Universal Language Model Fine-tuning (ULMFiT) [5], a transfer learning method that can be applied to NLP tasks. ULMFiT consists of the following steps: the LM is trained on a general-domain corpus to capture general features of the language in different layers. The full LM is fine-tuned on target task data using discriminative fine-tuning following a slanted triangular learning rate policy to learn task-specific features. Finally, the classifier is fine-tuned on the target task using gradual unfreezing. This strategy preserves low-level representations and adapts high-level ones.

3.3. LayoutQT

First, following the flow of Figure 2, we provided document images as input to our preprocessing step, which virtually maps page space into equally spaced quadrants. After that, we map each text block's start and end position into the related quadrant and inject spatial tokens to mark each text box's start and end position. Then the text of each bounding box is extracted along with the spatial tokens taking into account their position on the document page. In the processing language model phase, we tokenize and applied Universal Language Model Fine-Tuning (ULMFiT) [5] with ASGD Weight-Dropped LSTM (AWD-LSTM) [36].

4. Datasets

The methods described above were evaluated in two quite distinct datasets. The Tobacco800 [37, 38] and the RVL-CDIP [3] datasets, whose properties are described below.

4.1. Tobacco800

The Tobacco800 is a subset of the Truth Tobacco Industry Documents dataset. The original dataset has over 14 million documents of many types, such as letters, fax, memos, etc., the subset has only 1,290 pieces, manually annotated, targeting document signature and logos segmentation. Since the Tobacco800 dataset sample file name comes with the page, like the ones shown in Figure 3, when merged, it mimics a stream of pages from multiple documents proper to split by the PSS model.

The classification problem here involves two classes: whether the transition between consecutive pages indicates the continuity of the same document or the beginning of a new document. Document images are classified

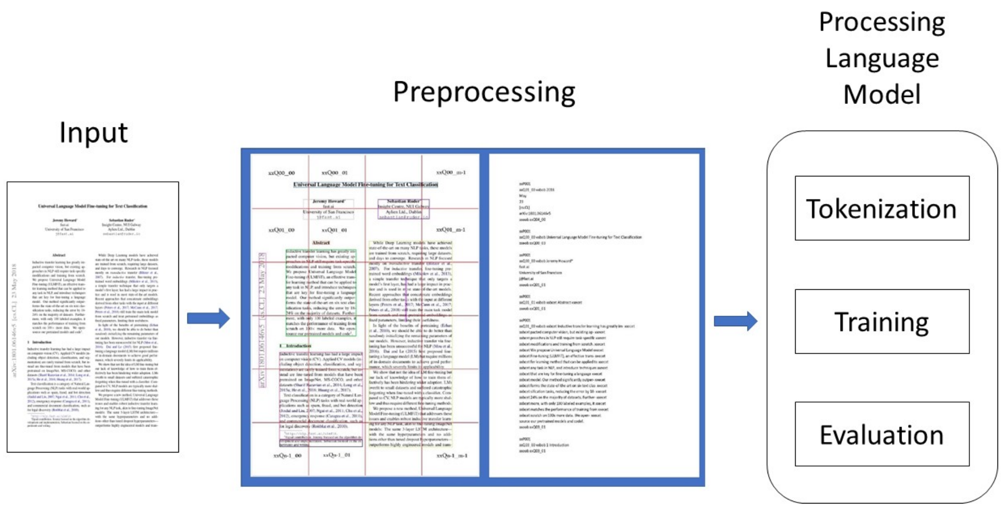

Figure 2: Illustration of the LayoutQT pipeline, going from input textual images to an NLP system. Our method computes layout tags as part of an OCR pipeline which is injected as special tokens in the text.

in FirstPage or NextPage, in which FirstPage represents the first page of a document, and NextPage class is formed by all pages of a document except the first page.

4.2. RVL-CDIP

The RVL-CDIP[3] consists of 400,000 grayscale images in 16 classes, with 25,000 images per class. There are 320,000 training images, 40,000 validation images, and 40,000 test images. The images are resized, so their largest dimension does not exceed 1,000 pixels. The 16 classes include letter, form, email, handwritten, advertisement, scientific report, scientific publication, specification, file folder, news article, budget, invoice, presentation, questionnaire, resume, memo 4. The evaluation metric is the overall classification accuracy and F1-score.

This dataset is a subset of the IIT-CDIP Test Collection 1.0 that is publicly available. The IIT-CDIP dataset itself is a subset of the Legacy Tobacco Document Library. The file structure of this dataset is the same as the IIT collection so that it can query this dataset for OCR and additional metadata.

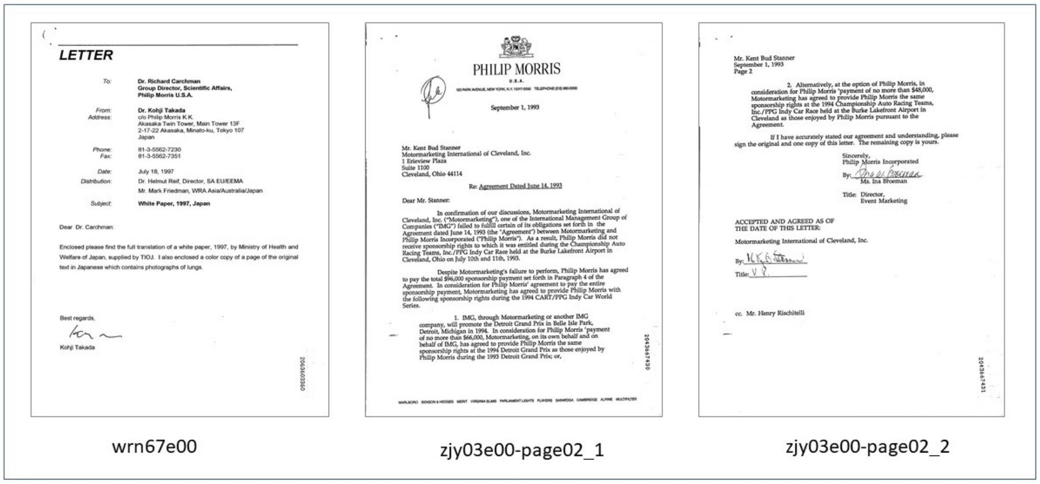

Figure 3: Image documents sample of Tobacco800 dataset. In left-to-right order, the first image is a single-page document, and the next two images are pages of the same document and are in ascending page order.

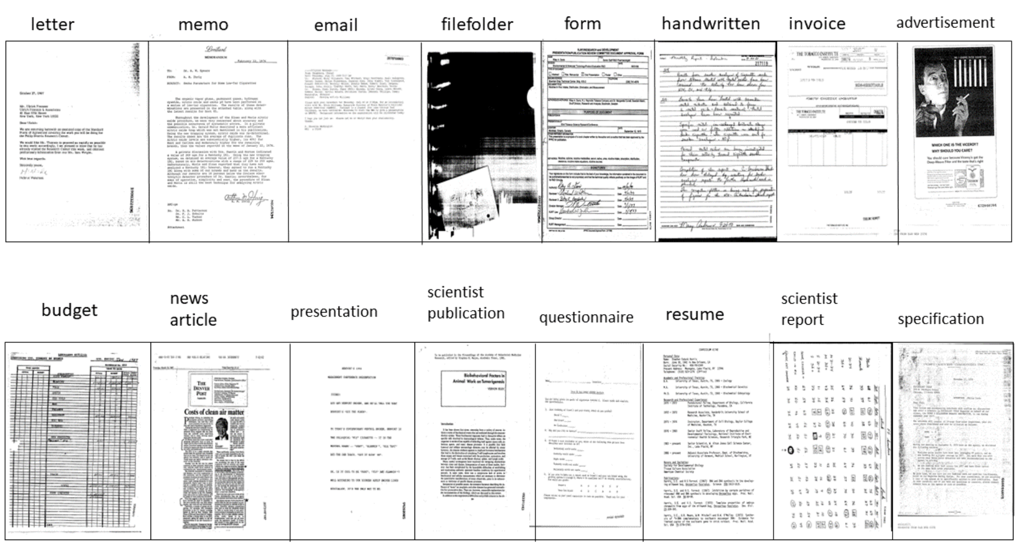

Figure 4: Samples of different document classes in the RVL-CDIP [3] dataset illustrate the low inter-class discrimination and high intraclass variations of document images.

5. Experiments

This section exposes the experiments in detail. We apply our model to two downstream tasks, one for page segmentation and the other for classifying document types. We performed four experiments with the Tobacco-800 dataset for the page stream segmentation task and two with the RVL-CDIP dataset for the document type classification. For Tobacco800, we followed the train, validation and test split defined by [1], whilst RVL-CDIP used the standard split. We performed classification experiments with and without using our model to compare the results. Thus, it identified the location (quadrants) of each bounding box's beginning, middle, and end and added spatial tokens (tags) to the text.

First, following the blue flow of Figure, we provided document images as input to our LayoutQT, which virtually maps page space into equally spaced quadrants. After that, we map each text block start and end position into the related quadrant and inject spatial tokens to mark the start and end position of each text box. Then the text of each bounding box is extracted along with the spatial tokens taking into account their position on the document page. The extracted texts were saved in text files.

5.1. Baseline

As a baseline, the document images fed the Tesseract OCR engine to extract the text without the spatial tokens. Subsequently, the extracted texts were tokenized, trained, tested, and evaluated using the same language model for the document classification task, as shown in Fig. 5 Finally, we compare the results obtained with and without tags.

5.2. Classification task

We used two classification tasks to evaluate our model. Page segmentation stream (PSS) classifies whether a document page is the first page or a continuity page and the classification of document types. To train and evaluate the document page stream segmentation (PSS), we used the Tobacco800 dataset in three network architectures, a Long Short-Term Memory (LSTM) [39] for text classification. Secondly, we used Universal Language Model Fine-Tuning (ULMFiT) [5] with ASGD Weight-Dropped LSTM (AWD-LSTM) [36] and BERT [4] for ranking the pages as *first_page* or *next_page* class on the same dataset. AWD-LSTM language model which uses DropConnect

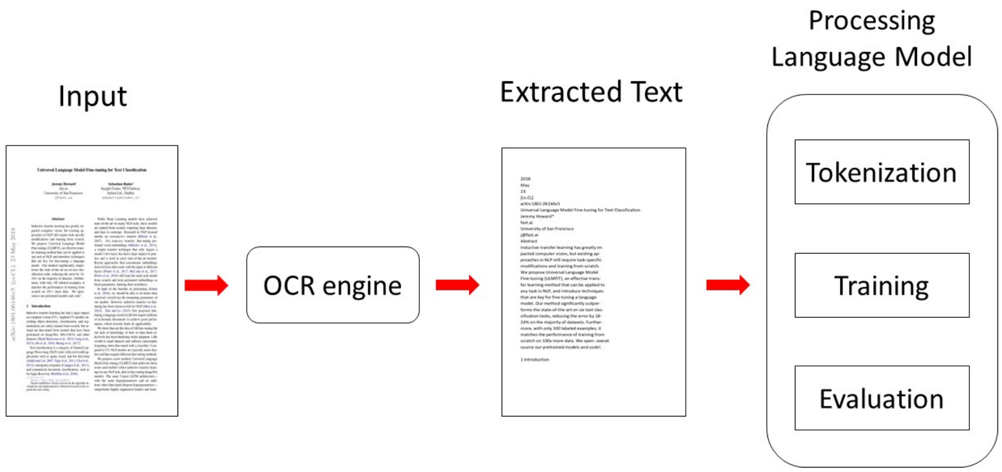

Figure 5: Experiment flow diagram showing the baseline without using the proposed method, to be compared with our pipeline, shown in Figure 2

and the average random gradient descent method, and several other regularization strategies. The weight-dropped LSTM strategy uses a DropConnect mask on the hidden-to-hidden weight matrices, as a means to prevent overfitting across the recurrent connections.

For document classification with the RVL-CDIP dataset, inspired by Howard and Ruder (2018) [5], we used ULMFIT with AWD-LSTM for training, testing and evaluation. Each evaluation dataset was split into training, validation and test subsets in the training phase. We minimized the loss function using the training set and assessed the model from each epoch on the validation set. We saved the model's weights of the lowest loss in the validation set iteration and evaluated the model with these weights in the test set after the whole training. Then we use the BERT [4] model to classify the RVL-CDIP dataset.

To evaluate, we compared the execution of the classifier using the LayoutQT method generating the quadrant tags and without the preprocessing with both Tobacco800 and RVL-CDIP datasets. To compare the results of our approach with the baseline, we used accuracy and F1-score metrics. The loss function used by default is the cross-entropy loss, as we have a classification problem (the different categories are the words in our vocabulary).

5.3. Experiment Setting

This subsection describes the implementation details used for the proposed approach. We used our preprocessing method, which starts with an OCR engine, Tesseract 4.1.1-rc1-7-gb36c, to generate blocks of text (bounding boxes) and delimit textual elements for each image in the document. Then, It drew the horizontal and vertical lines dividing each document page into 24 equivalent quadrants: 4 horizontal x 6 vertical.

Initially, we performed two experiments with the Tobacco800 dataset for binary classification of document pages, one with LayoutQT and one with the baseline. We used an LSTM backbone (composed of 256 nodes fully connected with activation "ReLU" and a dropout of 0.3). Furthermore, we use binary cross-entropy as a loss function with softmax activation and Adam as an optimizer. The model was trained for 100 epochs with a batch size of 128.

Next, we performed the experiments with an AWD-LSTM language model [36] trained with backpropagation through time with a batch size of 128, an embedding size of 400, 3 layers, 1150 hidden activations per layer, using Tobacco800 and RVL-CDIP datasets. The model was trained for one cycle of 100 epochs with a batch size of 128 documents and a sequence length of 72 using the NVIDIA Tesla V100 32GB GPU.

6. Results and Discussion

The document page binary classification, which identifies whether the document is a first page (FirstPage) or a continuation (NextPage), was performed with the Tobacco800 dataset using our LayoutQT method by adding quadrant tags and as a baseline processing without placing tags using only text. Such experiments were processed using the LSTM, ULMFiT with AWD-LSTM and BERT models.

The validation split results in Table 2 brought out that it had a large room for improvement in the baseline by only using text sequence architecture since we have surpassed Braz et al. (2021)[1] and Weidemann (2019) [7] baselines by at least 6 points of F1-score. After applying LayoutQT, we got 1.2 points more out of the 2.1 possible, which turns out to be 57% of possible gain. Furthermore, comparing the results obtained from our model with tags and without tags (baseline) using the LSTM, AWD-LSTM and BERT networks as the backbone, we obtained better results with AWD-LSTM.

Table 2: Accuracy and F1-score of the page stream segmentation on the Tobacco800 dataset obtained with the baseline and LayoutQT.

Model	Modality	Backbone	Accuraccy	F1-score
Braz et al. (2021) [1]	image only	VGG16	92.0%	91.9%
Braz et al. (2021) [1]	image only	EfficientNet-B0	83.7%	81.9%
Wiedemann et al. (2019) [40]	text + image	VGG16	91.1%	90.4%
Baseline	text only	LSTM	84.1%	82.9%
LayoutQT baseline	text + layout	LSTM	85.9%	86.1%
BERT baseline	text only	$BERT_{BASE}$	92.2%	92.0%
BERT with LayoutQT	text + layout	$BERT_{BASE}$	93.0%	93.0%
ULMFiT baseline	text only	AWD-LSTM	97.5%	97.9%
ULMFiT with LayoutQT	text + layout	AWD-LSTM	**99.5**%	**99.1**%

Figure 6 shows the confusion matrix of binary classification to the Tobacco800 dataset without tags (baseline) and with tags of quadrants (LayoutQT) using the ULMFiT (AWD-LSTM) model. It is clear that for the detection of first page images, both the baseline and our model missed only one image, but for detection of the follow-up pages, the model without our tags missed four images, while with our tags, there was only one error.

Our proposed approach also demonstrated superior performance for document classification on the RVL-CDIP dataset. When our location tokens are not used, the resulting F1 score is 80.4%, and when we use them, the F1 score goes to 83.6%. The confusion matrices for this task are shown in Figure 7, where the reduction can clearly improve off-diagonal values.

Table 3 compares the performance of the two document classification proposals, baseline, and LayoutQT, from the RVL-CDIP dataset for each document class. The results show that our approach to adding positional tags performed better. Of the 16 classes of documents, the F1 metric of our approach was inferior in only five classes (handwritten, scientific report, news article, presentation, and questionnaire). As these document types do not have a default layout or layout information, for example, the handwritten class has only plain text files, so the proposed tags cannot add any useful information. The main limitation of our approach is that it was designed to enrich textual representation by using layout information. However, the overall ranking result with LayoutQT showed an advantage of 3.2% in the F1 metric compared to the baseline. Furthermore, our approach to email documents obtained the highest accuracy at 97.6%.

Despite being a state-of-the-art technique, the use of BERT corresponds

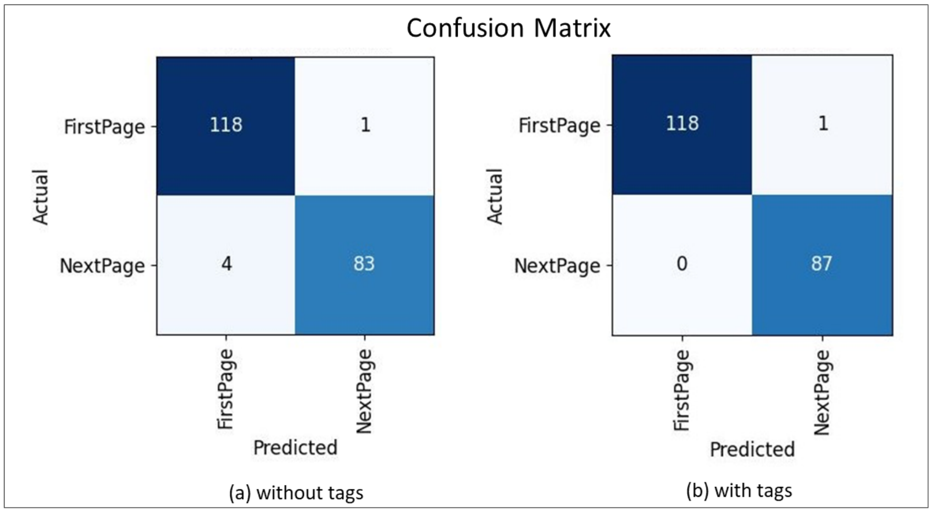

Figure 6: Confusion matrix of Tobacco800 binary classification using AWD-LSTM.(a) results found from the experiment without the tags, that is, with the baseline. (b) results obtained with the tags (LayoutQT).

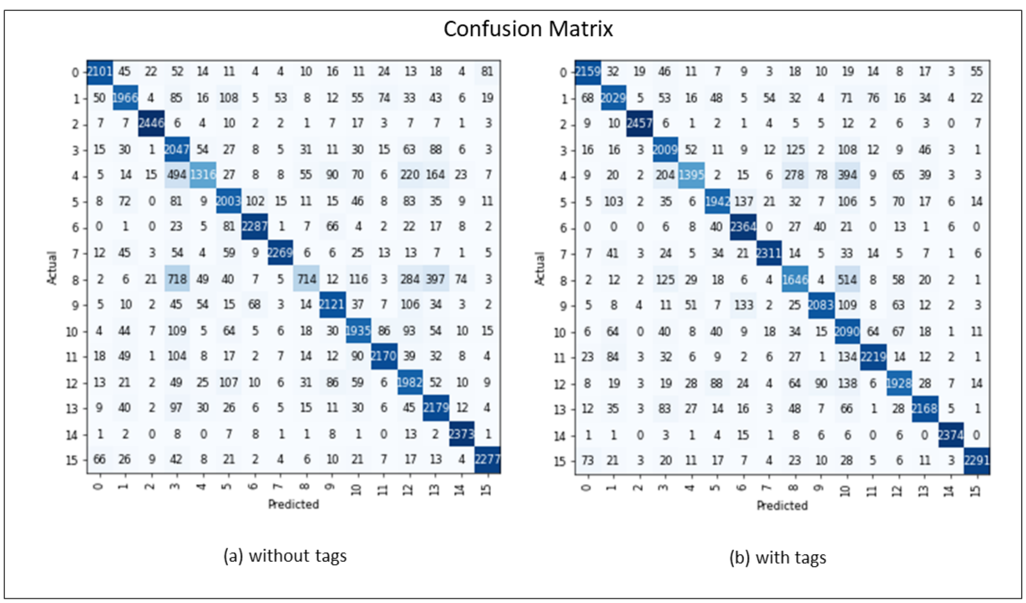

Figure 7: Confusion matrices for document classification on the RVL-CDIP data set using AWD-LSTM. Panel (a) shows the results of processing without our tags, and panel (b) shows the results obtained with our layout tags.

Table 3: F1-score of the document types classification on RVL-CDIP dataset obtained with the baseline and LayoutQT. The results in absolute numbers of hits and misses by classes are shown in Figure 7

Class	**Document Type**	**Baseline AWD-LSTM**	**LayoutQT AWD-LSTM**	**Baseline** $BERT_{BASE}$	**LayoutQT** $BERT_{BASE}$
0	letter	85.5%	**87.8%**	83.7%	86.0%
1	form	78.8%	**81.3%**	77.8%	77.3%
2	email	97.2%	**97.6%**	93.0%	96.0%
3	handwritten	**84.9%**	83.3%	63.6%	80.0%
4	advertisement	55.2%	**58.5%**	66.0%	70.0%
5	scientific report	**80.6%**	78.2%	74.8%	80.3%
6	scientific publication	89.1%	**92.1%**	87.4%	89.0%
7	specification	91.9%	**93.6%**	90.7%	91.0%
8	file folder	31.9%	**73.5%**	64.0%	73.8%
9	news article	**86.4%**	84.9%	78.8%	82.6%
10	budget	77.8%	**84.0%**	78.1%	82.3%
11	invoice	87.9%	**89.9%**	81.4%	85.9%
12	presentation	**79.9%**	77.8%	70.3%	81.1%
13	questionnaire	**90.0%**	89.5%	83.7%	87.9%
14	resume	93.6%	**93.7%**	98.6%	98.3%
15	memo	91.9%	**92.5%**	85.4%	90.0%
Average		80.4%	**83.6%**	80.1%	84.5%

to a small increase in classification F1 metric on the RVL-CDIP dataset compared to the AWD-LSTM model (84.5% vs 83.6%). In the Tobacco800 dataset, the AWD-LSTM model outperforms the BERT model in the classification F1 metric by a large margin (99.1% vs 93.0%). Considering the fewer parameters of the AWD-LSTM model - while the $BERT_{BASE}$ model has 110M parameters, the AWD-LSTM model has only 24M parameters - we decided to use the AWD-LSTM model in order to reduce the complexity of the LayoutQT architecture. However, the LayoutQT method can be easily adapted to other architecture, including BERT.

7. Conclusion

In this paper, we propose a novel preprocessing approach, LayoutQT - Layout Quadrant Tags, to overcome the challenges of document layout analysis for content extraction. LayoutQT divides the document into quadrants and uses the quadrant's location to add spatial tokens (tags) to mark each text box's start and end position. We compared the performance of our preprocessing method of adding spatial tokens to datasets with a simple baseline without the method to perform a document layout analysis and identified an improvement in the results obtained. For document page binary classification, the LayoutQT method combining text and layout features obtained the best results, obtaining accuracy using an LSTM and AWD-LSTM model, respectively of 85.9% (F1 86.1%) and 99.5% (F1 86.1%). In contrast, the result of baseline obtained an accuracy of 84.1% (F1 82.9%) with LSTM and 97.5% (F1 97.9%) using AWD-LSTM in the Tobacco-800 dataset. Finally, our method will greatly benefit several real-world document understanding tasks, such as document image processing.

Our method is simple and can be applied with any backbone. Even though the main advantage of our method is its potential to improve the performance of the representation with a low computational footprint, we believe that even more sophisticated architectures, like BERT, will benefit from LayoutQT. We therefore suggest that future experiments be performed with pretraining and fine-tuning using architectures with attention mechanisms, like BERT. Furthermore, we suggest applying LayoutQT for other downstream tasks, such as named entity recognition and machine translation.

For future research, we will refine the number of quadrants and tags. Meanwhile, we will also investigate the language expansion to make the multi-lingual LayoutQT model available for different languages, especially Portuguese. Finally, we intend to apply our model to classify legal documents.

References

[1] F. A. Braz, N. C. da Silva, J. A. S. Lima, Leveraging effectiveness and efficiency in page stream deep segmentation, Engineering Applications of Artificial Intelligence 105 (2021) 104394. doi:https://doi.org/10.1016/j.engappai.2021.104394.

[2] J. Lee, H. Hayashi, W. Ohyama, S. Uchida, Page segmentation using a convolutional neural network with trainable co-occurrence features, in: 2019 International Conference on Document Analysis and Recognition (ICDAR), 2019, pp. 1023–1028. doi:10.1109/ICDAR.2019.00167.

[3] A. W. Harley, A. Ufkes, K. G. Derpanis, Evaluation of deep convolutional nets for document image classification and retrieval, in: Inter-

national Conference on Document Analysis and Recognition (ICDAR), 2015, pp. 991–995. doi:10.1109/ICDAR.2015.7333910.

[4] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), volume 1, 2019, p. 4171–4186.

[5] J. Howard, S. Ruder, Universal language model fine-tuning for text classification, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 328–339. URL: https://aclanthology.org/P18-1031. doi:10.18653/v1/P18-1031.

[6] K. Mohsenzadegan, V. Tavakkoli, K. Kyamakya, A deep-learning based visual sensing concept for a robust classification of document images under real-world hard conditions, Sensors 21 (2021). URL: https://www.mdpi.com/1424-8220/21/20/6763. doi:10.3390/s21206763.

[7] G. Wiedemann, G. Heyer, Page stream segmentation with convolutional neural nets combining textual and visual features, in: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), European Language Resources Association (ELRA), Miyazaki, Japan, 2018, pp. –. URL: https://aclanthology.org/L18-1581.

[8] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, M. Zhou, Layoutlm: Pre-training of text and layout for document image understanding, Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2020). URL: http://dx.doi.org/10.1145/3394486.3403172. doi:10.1145/3394486.3403172.

[9] Y. Xu, Y. Xu, T. Lv, L. Cui, F. Wei, G. Wang, Y. Lu, D. Florencio, C. Zhang, W. Che, M. Zhang, L. Zhou, Layoutlmv2: Multi-modal pre-training for visually-rich document understanding, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference

on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 2579–2591. URL: https://aclanthology.org/2021.acl-long.201. doi:10.18653/v1/2021.acl-long.201.

[10] P. Zhang, C. Li, L. Qiao, Z. Cheng, S. Pu, Y. Niu, F. Wu, Vsr: A unified framework for document layout analysis combining vision, semantics and relations, 2021.

[11] F. Matkovic, M. Ivasic-Kos, S. Ribaric, A new approach to dominant motion pattern recognition at the macroscopic crowd level, Engineering Applications of Artificial Intelligence 116 (2022) 105387. URL: https://www.sciencedirect.com/science/article/pii/S0952197622003918. doi:https://doi.org/10.1016/j.engappai.2022.105387.

[12] X. Wu, Y. Zheng, T. Ma, H. Ye, L. He, Document image layout analysis via explicit edge embedding network, Inf. Sci. 577 (2021) 436–448.

[13] S. C. Kosaraju, M. Masum, N. Z. Tsaku, P. Patel, T. Bayramoglu, G. Modgil, M. Kang, Dot-net: Document layout classification using texture-based cnn, in: International Conference on Document Analysis and Recognition (ICDAR), 2019, pp. 1029–1034.

[14] S. Umer, R. Mondal, H. M. Pandey, R. K. Rout, Deep features based convolutional neural network model for text and non-text region segmentation from document images, Appl. Soft Comput. 113 (2021). URL: https://doi.org/10.1016/j.asoc.2021.107917. doi:10.1016/j.asoc.2021.107917.

[15] W. Yu, N. Lu, X. Qi, P. Gong, R. X. 0003, Pick: Processing key information extraction from documents using improved graph learning-convolutional networks, in: 25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021, IEEE, 2020, pp. 4363–4370. URL: https://doi.org/10.1109/ICPR48806.2021.9412927. doi:10.1109/ICPR48806.2021.9412927.

[16] A. R. Katti, C. Reisswig, C. Guder, S. Brarda, S. Bickel, J. Höhne, J. B. Faddoul, Chargrid: Towards understanding 2D documents, in: Proceedings of the 2018 Conference on Empirical Methods in Natural

Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 4459–4469. URL: https://aclanthology.org/D18-1476. doi:10.18653/v1/D18-1476.

[17] T. I. Denk, C. Reisswig, Bertgrid: Contextualized embedding for 2d document representation and understanding, in: Workshop on Document Intelligence at NeurIPS 2019, 2019, pp. –. URL: https://openreview.net/forum?id=H1gsGaq9US.

[18] M. Kerroumi, O. Sayem, A. Shabou, Visualwordgrid: Information extraction from scanned documents using a multimodal approach, in: E. H. Barney Smith, U. Pal (Eds.), Document Analysis and Recognition – ICDAR 2021 Workshops, Springer International Publishing, Cham, 2021, pp. 389–402.

[19] S. Bakkali, Z. Ming, M. Coustaty, M. Rusiñol, Visual and textual deep feature fusion for document image classification, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 2394–2403.

[20] M. Aggarwal, M. Sarkar, H. Gupta, B. Krishnamurthy, Multi-modal association based grouping for form structure extraction, 2020 IEEE Winter Conference on Applications of Computer Vision (WACV) (2020) 2064–2073.

[21] S. Li, X. Ma, S. Pan, J. Hu, L. Shi, Q. Wang, Vtlayout: Fusion of visual and text features for document layout analysis, in: Pacific Rim International Conference on Artificial Intelligence, Springer, 2021, pp. 308–322.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017, pp. –. URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[23] J. Lu, D. Batra, D. Parikh, S. Lee, Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks, in: H. M.

Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, 2019, pp. 13–23. URL: https://proceedings.neurips.cc/paper/2019/hash/c74d97b01eae257e44aa9d5bade97baf-Abstract.html.

[24] R. Powalski, Ł. Borchmann, D. Jurkiewicz, T. Dwojak, M. Pietruszka, G. Pałka, Going full-tilt boogie on document understanding with text-image-layout transformer, in: J. Lladós, D. Lopresti, S. Uchida (Eds.), Document Analysis and Recognition – ICDAR 2021, Springer International Publishing, Cham, 2021, pp. 732–747.

[25] Y. Li, Y. Qian, Y. Yu, X. Qin, C. Zhang, Y. Liu, K. Yao, J. Han, J. Liu, E. Ding, Structext: Structured text understanding with multi-modal transformers, Proceedings of the 29th ACM International Conference on Multimedia (2021).

[26] C. Li, B. Bi, M. Yan, W. Wang, S. Huang, F. Huang, L. Si, StructuralLM: Structural pre-training for form understanding, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 6309–6318. URL: https://aclanthology.org/2021.acl-long.493. doi:10.18653/v1/2021.acl-long.493.

[27] T.-L. Wu, C. Li, M. Zhang, T. Chen, S. A. Hombaiah, M. Bendersky, Lampret: Layout-aware multimodal pretraining for document understanding, ArXiv abs/2104.08405 (2021).

[28] T. Hong, D. Kim, M. Ji, W. Hwang, D. Nam, S. Park, Bros: A pre-trained language model focusing on text and layout for better key information extraction from documents, arXiv preprint arXiv:2108.04539 (2021).

[29] A. M. Roy, Adaptive transfer learning-based multiscale feature fused deep convolutional neural network for eeg mi multiclassification in brain–computer interface, Engineering Applications of Artificial Intelligence 116 (2022) 105347. URL: https://www.

sciencedirect.com/science/article/pii/S0952197622003712. doi:https://doi.org/10.1016/j.engappai.2022.105347.

[30] R. Smith, et al., Tesseract ocr engine, Lecture. Google Code. Google Inc (2007).

[31] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[32] W. McNally, K. Vats, A. Wong, J. McPhee, Rethinking keypoint representations: Modeling keypoints and poses as objects for multi-person human pose estimation, in: S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, T. Hassner (Eds.), Computer Vision – ECCV 2022, Springer Nature Switzerland, Cham, 2022, pp. 37–54.

[33] O. Lawal, Tomato detection based on modified YOLOv3 framework, Scientific Reports 11 (2021). doi:10.1038/s41598-021-81216-5.

[34] J. Howard, S. Gugger, Deep Learning for Coders with fastai and PyTorch, O'Reilly Media, 2020.

[35] M. Sundermeyer, R. Schlüter, H. Ney, Lstm neural networks for language modeling, in: Thirteenth annual conference of the international speech communication association, 2012, pp. –.

[36] S. Merity, N. Keskar, R. Socher, Regularizing and optimizing lstm language models, in: International Conference on Learning Representations, 2018, pp. 1–13. URL: https://openreview.net/forum?id=SyyGPP0TZ.

[37] G. Zhu, Y. Zheng, D. Doermann, S. Jaeger, Multi-scale structural saliency for signature detection, in: In Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR 2007), 2007, pp. 1–8.

[38] G. Zhu, D. Doermann, Automatic document logo detection, in: In Proc. 9th International Conf. Document Analysis and Recognition (ICDAR 2007), 2007, pp. 864–868.

[39] M. Sundermeyer, R. Schlüter, H. Ney, Lstm neural networks for language modeling, in: Thirteenth annual conference of the international speech communication association, 2012, pp. –.

[40] G. Wiedemann, G. Heyer, Multi-modal page stream segmentation with convolutional neural networks, Language Resources and Evaluation (2019) 1–24.