NOVA IMS Information Management School

MDSAA

Master Degree Program in

Data Science and Advanced Analytics

Fine-tuning a Multimodal Machine Learning Model for Key Information Extraction from Invoices and Receipts

Rodrigo Miguel Vidal da Silva

Master Thesis

presented as partial requirement for obtaining a Master's Degree in Data Science and Advanced Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

Fine-tuning a Multimodal Machine Learning Model for Key Information Extraction from Invoices and Receipts

by

Rodrigo Miguel Vidal da Silva

Master Thesis presented as partial requirement for obtaining the Master's degree in Data Science and Advanced Analytics, with a specialization in Data Science

Supervised by

Bruno Damásio, PhD

Nova Information Management School Instituto Superior de Estatística e Gestão de Informação

June, 2025

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledged the Rules of Conduct and Code of Honor from the NOVA Information Management School.

[June, 2025]

ACKNOWLEDGEMENTS

First and above all, I would like to express my profound gratitude to my parents. The constant support, encouragement, and trust placed in me have served as the foundational elements of my academic journey.

I am very grateful to my supervisor, Bruno Damásio for his valuable guidance and advice during this research.

Finally, a special thank you to my friends, colleagues and teachers who have been with me on this journey. Their support, stimulating discussions, and shared moments of frustration and success have made this experience both valuable and satisfying.

ABSTRACT

The automated extraction of important information from different types of documents, especially invoices, is essential for improving business operations and increasing efficiency in finance. In the past, this was a time-consuming and error-prone manual task. Recently, progress in deep learning and transformer-based learning has renewed interest in automating this work. It offers promising solutions for smart document processing. This thesis tackles this issue by focusing on fine-tuning LayoutLMv3, a transformer-based model, to extract key fields from Portuguese invoices and receipts. The main goal of this research is to adjust LayoutLMv3 for a custom dataset of 813 invoice and receipt images in Portuguese. The model will be trained to clearly identify and extract important details like company name, address, date and total amount. This information is essential for keeping financial records and streamlining workflows. To prepare the training data, we first use Tesseract for an OCR step. This extracts raw text and their corresponding bounding box coordinates from the images. After that, we use a custom algorithm to accurately label text categories that either match or closely resemble the predefined annotations. This process ensures the dataset is properly formatted for LayoutLMv3's multimodal input needs. After the preprocessing and labeling steps, the LayoutLMv3 model is fine-tuned and evaluated. Its effectiveness is measured by comparing its performance to a well-known commercial solution, Google Document AI. This comparison aims to show the practical use and limitations of a custom-trained open-source model in a real-world scenario. The results show that Google Document AI outperforms the fine-tuned LayoutLMv3 model by a large margin. However, the findings offer valuable insights into the strengths and weaknesses of fine-tuned Transformer models for extracting information from documents in a low-resource language context and semi-structured document types. Additionally, this research can help improve automation in financial processes, reduce manual work, and provide a solid framework for similar document understanding tasks in different industries.

KEYWORDS

Optical Character Recognition; Key Information Extraction; Invoices; Multimodal Machine Learning Models

Sustainable Development Goals (SDG):

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ANN	Artificial Neural Networks
BERT	Bidirectional Encoder Representations from Transformers
CLAHE	Contrast Limited Adaptive Histogram Equalization
CNN	Convolutional Neural Networks
DONUT	Document Understanding Transformer
GMM	Gaussian Mixture Models
GPU	Graphical Processing Unit
HMM	Hidden Markov Models
KIE	Key Information Extraction
KNN	K-Nearest Neighbour
MLM	Masked Language Modeling
MIM	Masked Image Modeling
MLP	Multilayer Perceptrons
OCR	Optical Character Recognition
PICK	Processing Key Information from Documents
RNN	Recurrent Neural Networks
SVM	Support Vector Machines
VDU	Vision Document Understanding
WPA	Word-Patch Alignment

1. INTRODUCTION

1.1. RESEARCH BACKGROUND

Optical Character Recognition (OCR) has had a profound impact on the field of document digitization, enabling the extraction of text from scanned or photographed documents. Over the decades, this technology has become a cornerstone in various domains, including data entry automation, archival systems, and digital libraries. While early research primarily focused on character recognition, beginning with machine-printed characters and later expanding to handwritten text, recent advances have pushed the boundaries far beyond basic OCR, (Wang et al., 2021).

As OCR technology advanced, it became imperative to expand beyond rudimentary text extraction from structured formats, such as book pages, to address the complexity inherent in diverse document layouts. This evolution was pivotal for the effective processing of varied document types, including forms, business cards, and invoices, which require layout analysis to extract relevant information. The advent of deep learning has further transformed OCR, paving the way for the development of end-to-end models capable of identifying both structure and text with remarkable consistency. Notable examples of these advancements include LayoutLMv3 (Huang et al., 2022) and DONUT (Kim et al., 2022).

An important extension of OCR is Key Information Extraction (KIE), which focuses on identifying and extracting specific data points from documents and images. This capability is particularly valuable for applications such as efficient archiving, document analytics, and fast indexing, where targeted information such as dates, names or prices, needs to be quickly located. The manual execution of such tasks on a large scale is often time-consuming and costly, underscoring the necessity for ongoing research to develop KIE solutions with superior performance, (Huang et al., 2022; Kim et al., 2022). These advancements hold the potential to significantly enhance automation, reduce costs, and improve efficiency in managing complex document workflows. For that reason, this research focuses on refining a solution to carry out KIE on Portuguese invoices and receipts. These documents are common but come in many different formats, which makes our task very interesting.

1.2. RESEARCH OBJECTIVES

This thesis aims to develop and evaluate a key information extraction system for invoices and receipts, leveraging the capabilities of a multimodal LayoutLMv3 model. The main goal of this research is to determine the effectiveness of a model, incorporating visual, textual, and layout information, in accurately identifying and extracting critical data points, including company, address, date, and total amount, from a set of Portuguese documents with a broad array of layouts.

Given the availability of the dataset, the methodology advances to the pre-processing, entity matching and fine-tuning phase. This process enables the model to comprehensively interpret both the textual content and the spatial arrangement of elements within the invoices.

Subsequent to the training of the model, a robust evaluation framework is designed. This framework utilizes a set of key performance metrics to assess the model efficacy, including precision, recall, and f1-score. Subsequently, a comparative analysis is conducted with commercial model from Google. The anticipated outcome of this research is the development of a trained multimodal system that exhibits extraction accuracy and adaptability across a range of invoice formats. This work promises to offer valuable insights into the practical benefits and challenges of applying advanced multimodal models to real-world document understanding tasks. Ultimately, the knowledge gained from this work can significantly automate and improve invoice processing in business contexts.

1.3. DISSERTATION STRUCTURE

The present thesis is structured across a total of five sections. The first section introduces the context of text extraction from documents and the importance and practical value of this research area. It also explains the main reasons for the study and outlines the specific research goals that guide the entire dissertation.

The second chapter reviews the basic concepts and existing literature related to the study. It starts by detailing the steps typically involved in a traditional pre-processing pipeline for document image analysis. Then, it looks at multimodal models, highlighting their architecture and places the related work within the specific context of our research.

The third section describes the methods used to achieve the research goals. It explains the procedures for preparing and pre-processing images, the training process for the proposed model, and the inference approach used with a commercial model for comparison. This section also outlines the evaluation metrics to assess the performance and effectiveness of the proposed method.

The fourth chapter presents the experimental results from the proposed approach. It offers a detailed analysis and discussion of these results, comparing them to the performance of the commercial model. The discussion also explores potential reasons for the results.

Finally, the fifth section wraps up the dissertation by summarizing the key contributions and insights from the research. It reflects on the limitations faced during the study and suggests directions for future research, proposing ways to expand and improve the work to further advance the field of text extraction from invoices and receipts.

2. LITERATURE REVIEW

This section presents a comprehensive examination of existing OCR techniques and technologies. It investigates the fundamental components and stages of OCR systems, from pre-processing to the post-processing phase. It elucidates the structure of a conventional OCR pipeline and underscores the significance of various OCR frameworks, some of them to explored in Methodology chapter.

2.1. TRADITIONAL OCR PIPELINE

There are different methods and pipelines for recognizing text from images, Figure 2.1 represents a classic step-by-step structure that is used in OCR systems.

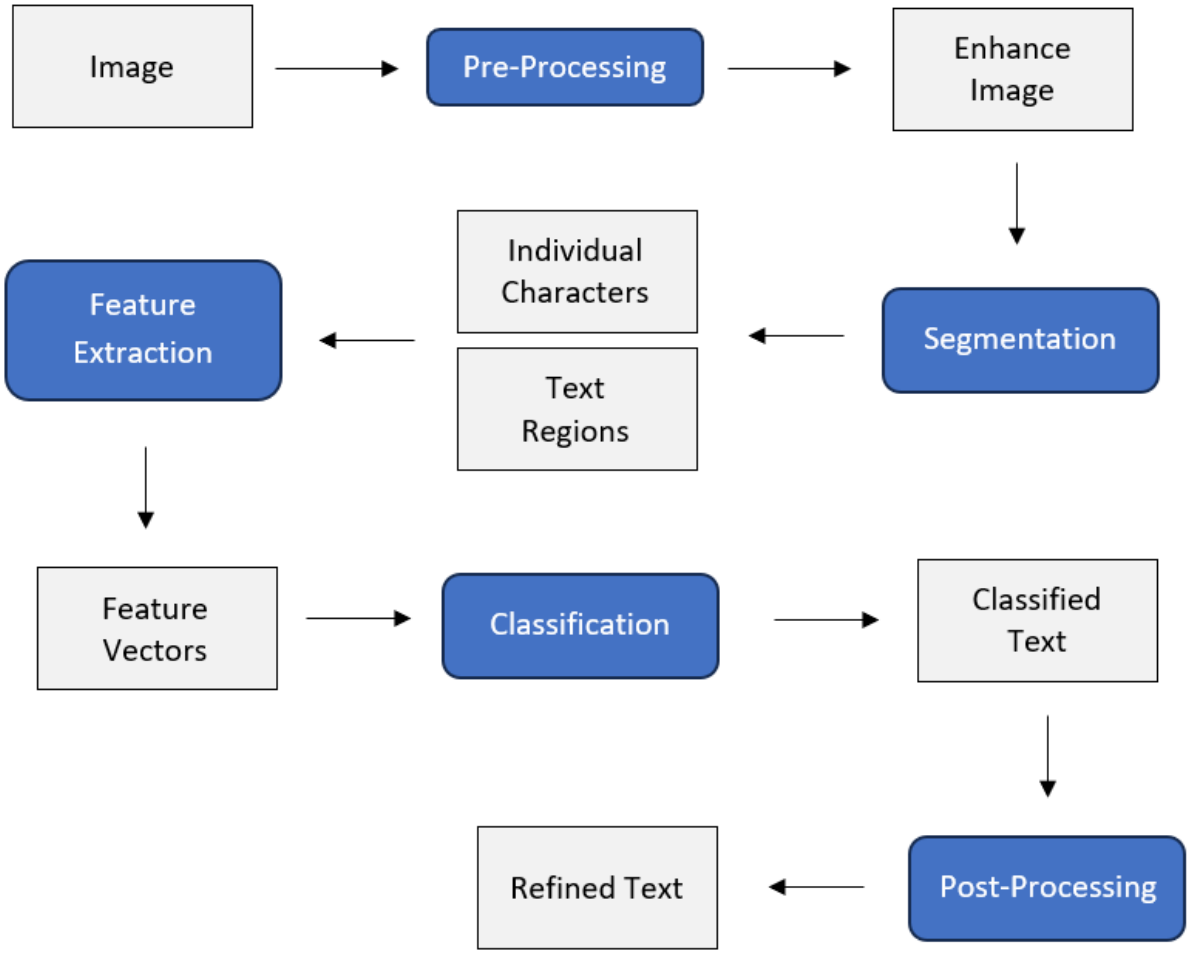

Figure 2.1 - OCR traditional pipeline

2.1.1. Pre-Processing

This is a crucial aspect in the context of this study, given that a common challenge in accounting offices is the presence of poor-quality receipts, often characterised by excessive noise.

There are numerous pre-processing techniques that can be employed to prepare an image for classification. This discussion will focus on three such techniques: binarization, noise reduction, and skew detection.

Binarization can be conceptually understood as a simplified colour image quantization process, in which an image is generated using only two colours (one colour for identifying interesting objects and the other for the background), (Yang et al., 2012). To identify or distinguish pertinent pixels from background pixels, global threshold-based or adaptive threshold-based methods are employed. Global threshold-based use a single delineator for the entire image, (Otsu, 1979; Pavlidis, 1993; Ying Liu & Srihari, 1997), local or adaptive threshold-based is based on the characteristics of the neighbourhood to calculate the threshold, (Sauvola et al., 1997). There are also deep learning-based methodologies that employ trained models to enhance the process of binarization, (De et al., 2020).

In the context of documents, the term "noise" is used to describe any visual distortions that impair readability. One of the most widely recognized forms of noise is salt-and-pepper noise, which occurs when the image contains a significant number of distorted pixels, exhibiting values of either 0 or 255, which represent the minimum and maximum possible values for those pixels. This type of noise may be caused by the presence of dust molecules during the image acquisition process.
Gaussian noise is a specific type of statistical noise that affects images with random intensity variations that follow a Gaussian distribution. It appears as a grainy texture and is typically introduced by electronic sensor noise during the image capture process, especially in low-light conditions or by low-quality scanners. In images that are affected by this noise, pixel values deviate slightly from their actual values, which can obscure fine details and reduce image sharpness (Hussein et al., 2021).

To address the previously mentioned issue, statistical methodologies were initially proposed. These methodologies entail decomposing the components using wavelet, (Portilla et al., 2003), whereby insignificant coefficients (noise) are suppressed while significant ones are preserved. Subsequently, a threshold-based selection is employed, a technique that can also be utilized in other scenarios. For instance, the approach proposed by (Donoho, 1995), can be referenced, wherein a threshold is subtracted from several coefficients exceeding it, and the remainder are set to zero.
Recently, deep learning-based convolutional neural networks (CNN) have been employed extensively for image denoising due to their capacity to learn image feature hierarchies (Divakar & Babu, 2017). Notable methods include DnCNN, proposed by (Zhang et al., 2017), which learns a residual mapping to remove noise, and FFDNe, proposed by (Zhang et al., 2018), which allows the network to adapt to various levels of noise, including the standard deviation of noise as an additional input.

As outlined by (Al-Khatatneh et al., 2015), the skew, or angular misalignment of text or content, can be classified into three distinct categories: Global skew that is defined as a consistent angular misalignment affecting the entire document, frequently resulting from improper alignment during the scanning process; Multiple skew that refers to a document comprising sections or regions exhibiting varying skew angles; Non-uniform text line skew that is characterized by individual lines within the document displaying disparate angles, imparting a wavy appearance to the text.

A variety of strategies have been proposed, including down sampling the image and determining the skew angle through the rotation of the input document image at various angles, with the calculation of a projection profile for each by (Bloomberg & Kopec, n.d.). Furthermore, methods utilising the Hough transform have been employed for the detection of document skew. This approach, initially proposed by (Yu & Jain, 1996) and (Amin & Fischer, 2000), identifies the skew angle by locating lines exhibiting the highest concentration of collinear pixels. These often align with the text baselines within the document. Further methods of this nature, along with a variety of other approaches, can be found in the ICDAR2013 Document Image Skew Estimation Contest, as presented by (Papandreou et al., 2013).

2.1.2. Segmentation

One of the fundamental aspects of OCR is the process of segmentation, which is a critical step in the overall process of optical character recognition. Segmentation is a strategy employed to ascertain the boundaries of a block, sentence, word, or even a single character. Furthermore, segmentation is employed to differentiate between images and text. The field of segmentation can be divided into three main categories: ruled-based approaches, machine learning approaches, and deep learning approaches, (Xu et al., 2020).

Ruled-based approaches can be classified as top-down or bottom-up. Top-down approaches begin with larger blocks and progressively reduce the block size until the character level is reached, (Jaekyu Ha et al., 1995). Bottom-up approaches, on the other hand, start by grouping pixels until a character is formed, and then expand to encompass larger units, (Jaekyu Ha et al., 1995; Lebourgeois et al., 1992).

Machine learning methods, have been the predominant approach to document segmentation tasks. (Jain & Zhong, 1996) treats document presentation as a parsing problem, utilizing grammar-based loss functions to identify the optimal structure while training parameters and selecting features. Artificial neural networks (ANN) have also been widely applied to analyze and recognize handwritten and printed characters with remarkable success. Other models, such as support vector machines (SVM) and Gaussian mixture models (GMM), have also contributed to layout analysis, (Wei et al., 2013). However, these machine learning techniques often require extensive manual design of features and have difficulty capturing high-level semantic context. They rely mainly on visual clues, often ignoring textual information.

In recent years, deep learning methods have emerged as the predominant standard for addressing a range of machine learning challenges. These methods theoretically approximate any function by stacking multiple neural network layers and have been demonstrated to be effective across numerous research domains. For example, one approach treats document semantic structure extraction as a pixel-by-pixel classification task, employing a multimodal neural network that integrates both visual and textual information, (Yang et al., 2017). However, this method is primarily designed to support heuristic algorithms in classifying candidate bounding boxes, rather than providing a comprehensive end-to-end solution.
Further advancements include the incorporation of contextual information into a Fast R-CNN model, which enhances region detection performance by accounting for the localized nature of article content, (Borges Oliveira & Viana, 2017). More recently, (Xu et al., 2020), was proposed a robust method that can leverage multimodal inputs, such as token embeddings, layout embeddings, and image embeddings. This approach enables the model to jointly process textual and layout information, enhancing its ability to understand document structures.

2.1.3. Feature Extraction

As a considerable number of classifiers are unable to process images or raw data effectively, feature extraction represents a crucial step aimed at reducing the data set size while extracting pertinent information. An optimal feature set should effectively represent the specific features of a given class and be as invariant as possible to changes within that class, (Lauer et al., 2007).

Some feature extraction methods work on gray-level subimages of single character, while others work on symbols segmented from the binary raster image, thinned symbols or skeletons, or symbol contours. Here we will present some that work in binary raster images. (Due Trier et al., 1996; Hossain et al., 2012) enumerates some traditional methods that rely on features that are manually designed based on human intuition and mathematical principles. Examples are Template matching that compares the input image to stored binary, with similarity measured by distances or Moments that extract shape, orientation, size, and position of the character.

The most recent approaches are founded upon neural networks that are capable of autonomously discerning pertinent characteristics during the training phase and exhibiting a superior capacity for generalization to previously unobserved data sets, due to their ability to discern more abstract and high-level features, (Ahlawat et al., 2020; Lauer et al., 2007).

2.1.4. Classification

Classification is the part of the process where the extracted features from segmented regions are analyzed and assigned to a specific category. The following describes four types of methods according to (Dongre et al., 2010).

Template matching is probably the most basic classification method and is used to classify a character with a predetermined template. In the case of text on receipts, direct matching can be done with standard character prototypes, according to a similarity measure. In other types of documents, where there may be human handwriting, it is relevant to use Elastic Matching, (Arica & Yarman-Vural, 2001).

Statistical techniques, guided by probability models and statistical rules, are based on assumptions such as the feature distribution for each class follows a specific form, often a uniform distribution. These methods utilize mathematical models to map input features to output classes, enabling systems to make optimal classification decisions. Examples of such methods include K-Nearest Neighbor (KNN), (Cover & Hart, 1967), Bayes classifier, (Friedman et al., 1997), and Hidden Markov Models (HMMs), (Mor et al., 2021).

In contrast to the conventional template matching approach, which relies on direct comparisons with prototypes, SVMs, (Ben-Hur & Weston, 2010), define the classes based on this margin, rather than relying on simple distance measures. Moreover, the decision boundary in SVMs is not constrained to linear separation, it can be nonlinear, achieved through the utilization of kernel functions that map the input data into higher-dimensional spaces. This flexibility enables SVMs to address more intricate classification issues as you can see in (Arora et al., 2010).

Neural networks, including Multilayer Perceptrons (MLP), CNN and Recurrent Neural Networks (RNN), are commonly employed for text classification in OCR applications. MLP networks facilitate simplified and more direct categorization. CNNs and RNNs, conversely, have demonstrated remarkable efficacy in text recognition tasks, as they are adept at discerning patterns, with RNNs particularly suited for leveraging contextual data, (Memon et al., 2020).

2.1.5. Post-Processing

According to (Nguyen et al., 2022), post-processing encompasses the techniques utilized to refine and correct errors in the text generated by OCR systems. It is of paramount importance to implement these processes, as the output of OCR, particularly in the case of documents with high levels of noise, frequently contains inaccuracies that have a detrimental impact on downstream tasks, such as ensuring compliance in accounting statements.

In accordance with the aforementioned source, post-OCR processing methods can be categorized into manual approaches where people correct words manually and semi-automatic methods that consider features of the OCR output words. Examples are the presence of the word in a dictionary, its frequency, its recognition confidence and so on. The latter can be further divided into two subcategories: isolated word processing and context-dependent approaches.

Isolated word processing relies on characteristics of single words, and it can be applied merging OCR outputs, using a distance metric to select candidate errors and implementing post-OCR error models that use complex algorithms. An example of the last approach is a method named OCRSpell (Taghva & Stofsky, 2001) that instead of only splitting text by spaces, it smartly identifies full word boundaries and check the result word against a dictionary. If an error is found, OCRSpell suggests corrections. It generates these suggestions using a confusion matrix and a special rule for words with unrecognizable characters. Finally, a scoring system ranks these suggestions based on how often words appear together and their character patterns, presenting the most likely corrections to the user.

Context-dependent approaches do not handle only a single word, then handle real-word erros by considering the surrounding context of each token. Example of this methods are neural network-based language models, that are trained as probabilistic classifiers to predict the probability distribution over the next word given its context and sequence-to-sequence models that consider OCR post-processing as a machine translation task, which transforms OCR output words in their correct form in the same language.

2.2. MULTIMODAL MACHINE LEARNING MODELS

In contrast with traditional OCR pipeline that relies on a structured workflow, multimodal-models integrate data from multiple modalities, such as text, layout, image or audio to solve complex tasks. This flexibility enables them to adapt more effectively to challenging scenarios.

2.2.1. LayoutLMv3

Designed to understand both the text and layout of documents, the LayoutLMv3 model is the third version of the experiment started by (Xu et al., 2020). First named LayoutLM, the model emerged as the first to interact jointly with text and layout in scanned documents. In 2020, it was improved in its second version (Xu et al., 2021) regarding the interaction between text and layout, accommodating new pre-training objectives that also interacted with images and CNNs to process their graphic elements.

LayoutLMv3 (Huang et al., 2022) is the first multimodal transformation model that does not rely on pre-trained CNN to extract visual features, instead it applies a multi-head self-attention to better understand the importance of each word and a position-wise connected feed-

forward networks. The transformer's input is the concatenation of text embedding and image embedding sequences and its architecture is shown in Figure 2.2.

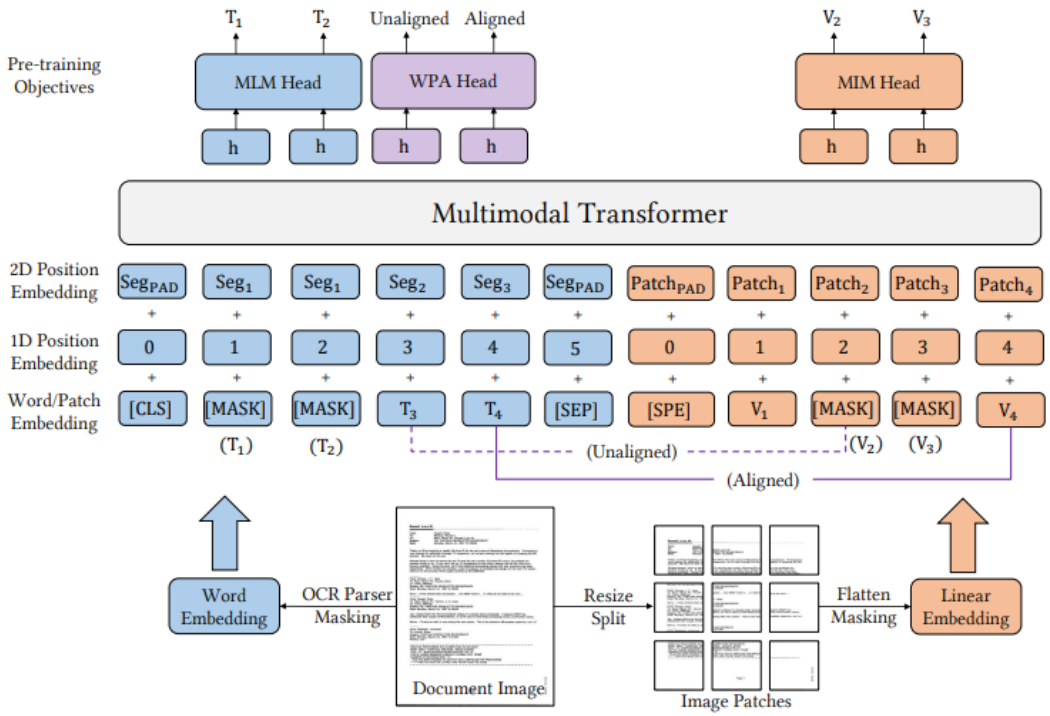

Figure 2.2 - LayoutLMv3 architecture (source: LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking, (Huang et al., 2022))

2.2.1.1. Embeddings

Embeddings are low-dimensional representation of objects such as words, sentences, or images like a numerical vector. These embeddings are designed to encode the relevant features of these objects in a way that can be efficiently processed by machine learning models. (Chen et al., 2018)

This model incorporates text embeddings, which in this case are a combination of word and position embeddings. The word embeddings are initialized with a word embedding matrix from a pre-trained RoBERTA model and the position embeddings include 1D position and 2D layout position embeddings, where the 1D position refers to the index of the tokens in the sentence string, and the 2D layout position refers to the bounding boxes coordinates of the sentence string.

Building upon the methodology established by LayoutLM, a comprehensive normalization of all coordinates is implemented, with this normalization occurring in relation to the image's dimensional parameters. The implementation of embedding layers facilitates the independent encoding of the x-axis, y-axis, width, and height characteristics. In contrast to the word-level layout positions employed by LayoutLM and LayoutLMv2, where each word possesses an independent position, this method utilizes segment-level layout positions. In

this, all words within a segment share the same 2D position, as they generally represent the same semantic meaning.

To build the image embeddings LayoutLMv3 fed into the multimodal transformer linear projection features of image patches. A document image is resized into $\mathrm{H} \times \mathrm{W}$ and denoted with $\mathrm{I} \in R^{\mathrm{C}\times\mathrm{H}\times\mathrm{W}}$, where C, H, and W are the channel size, width, and height of the image, respectively.

The image is then split into a sequence of uniform $P \times P$ patches that are linearly projected to D dimensions and flattened into a sequence of vectors, with a length of $M = \frac{HW}{P^2}$, where *P* and *D* are patch size and the dimension of the embedding space, respectively.

2.2.1.2. Pre-training Objectives

The initial pre-training objective entails the implementation of Masked Language Modeling (MLM), a self-supervised learning method, within LayoutLMv3. The objective of MLM is to empower the model with the capacity to decipher words on the receipt, considering both the preceding and subsequent context of the text.

This is facilitated by the model's input of text and its masking of 30% of the tokens with a special token, so that it can identify the token it needs to learn in the text according to the context used by the unmasked ones. In the instance of KIE from receipts, this step assumes paramount importance, as it facilitates the comprehension of key words that may change their nomenclature and positioning within the document, in accordance with the legal requirements of each country. An example could be the final price that in some invoices is named “Total” and in others “Amount Due” or even de currency that in the USA is dollar ($) and in the UE is (€).

The second pre-training objective is Masked Image Modeling (MIM), which is analogous to MLM but applied to images. This step is crucial for enabling the model to develop a comprehensive understanding of the diverse layout configurations, font dimensions, table structures, and text alignment characteristics present in receipts.

The method is tasked with dividing images into patches and transforming them into discrete visual tokens, assigning meaning to each image patch so that the model can interpret the information better than if it were given raw pixels as input. Subsequently, 40% of these patches are masked, and those are then fed to a transformer to unmask and learn the information present in the visual tokens of the original image.

The application of these MLM and MIM separately poses a problem for the architecture of the model, since it is unable to correlate which, textual tokens correspond to image patches. For that reason, LayoutLMv3 model proposes the Word-Patch Alignment (WPA) method to address this issue, with the objective of predicting when patches and tokens are correlated.

The WPA identifies an unmasked text token as aligned when its corresponding image token is also unmasked and as unaligned otherwise. The WPA is trained using a two-layer multi-layer perceptron (MLP), with masked text tokens being excluded through the calculation of loss to prevent the model from learning the relationship between masked text tokens and image patches.

2.2.2. DONUT

DONUT is an end-to-end Visual Document Understanding (VDU) model designed for general document image understanding, which does not rely on OCR (Kim et al., 2022). Rather than employing OCR, it utilizes a transformer-based visual encoder to extract the document's features and a decoder that transforms the features into a sequence of tokens to construct the output. Figure 2.3 provides an overview of the Donut process.

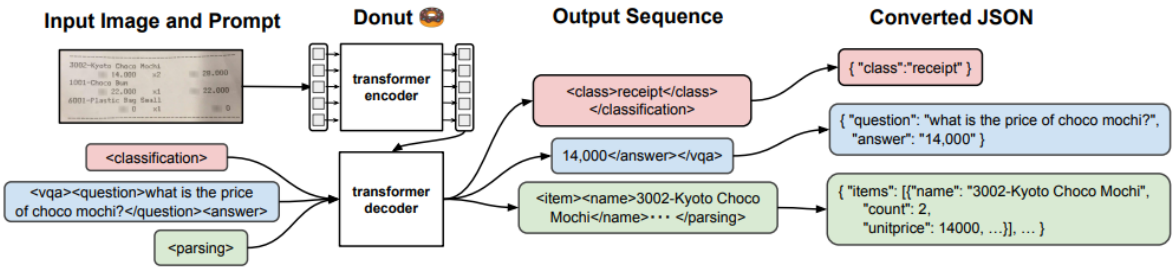

Figure 2.3 – DONUT architecture (source: OCR-free Document Understanding Transformer. (Kim et al., 2022))

2.2.2.1. Encoder

Although CNN-based models or Transformer-based models can be used as the encoder, DONUT uses a Swin Transformer to convert the input document image into a set of embeddings. The Swin Transformer first splits the image into patches which are then embedded in a shifted window-based multi-head self-attention module and a two-layer MLP. Then, the nearby patch tokens are merged into larger tokens in the patch merging layers. The output of the Swin Transformer is then made available to the textual decoder.

2.2.2.2. Decoder

Given the latent representation, the textual decoder produces a sequence of tokens, where each token is represented as a one-hot vector. The decoder is based on the BART architecture, with its weights initialized using the publicly available pre-trained multilingual BART model for a better context and grammar understanding across many languages.

2.2.2.3. Pre-training Objectives

The model processes images in a top-left to bottom-right reading order, like a pseudo-OCR task, generating tokens sequentially. At each step, the model predicts the subsequent token by leveraging the image and text previously generated tokens as inputs. The training objective

is to minimize the cross-entropy loss associated with these predictions. In essence, the model functions as a visual language model trained on a dataset of document images.

DONUT uses ITT-CDIP, a set of eleven million document images with a CLOVA OCR API to obtain the pseudo-text labels. In order for the model to be able to generalize to other languages, the Synthetic Document Generator was used with the help of wikipedia to generate five hundred thousand samples for the Chinese, Japanese, Korean and English languages.

2.3. RELATED WORK

As stated in the literature review, conventional OCR pipelines have historically served as the basis for extracting textual information from documents. These pipelines frequently employ linear, sequential processing, which transforms document content into a simple, flat text format. This simplification disregards the layout of structured documents, such as invoices, where spatial relationships carry critical semantic information. Typically, these conventional pipelines entail the extraction of text through OCR techniques, which is then followed by the implementation of rule-based methodologies for the recognition of named entities and the classification of fields.

One widely used tool for text extraction within these traditional OCR pipelines is Tesseract (Smith, 2007), a widely adopted open-source OCR engine that was originally developed at Hewlett-Packard (HP) and it was subsequently made open-source by Google in 2005. The architecture of the system is predicated on a recognition pipeline that incorporates connected component analysis, text line formation, word segmentation, and a sophisticated two-pass recognition mechanism. A notable early innovation was its ability to handle both black-on-white and white-on-black text through an outline-based representation of image components. After the segmentation of the image into text lines and words, Tesseract differentiates between fixed-pitch and proportional text, recalibrating its segmentation and recognition strategies as needed.

However, such approaches encounter challenges when confronted with variations in layout and formatting, which limits their adaptability and generalizability across a range of invoice templates.

In order to address this limitations, advancements have been made in the form of layout-aware models that directly incorporate the two-dimensional structure of documents into the learning process. A noteworthy deep learning approach is Chargrid (Katti et al., 2018), which represents scanned documents as a grid of characters, thereby effectively preserving spatial context. The model employs a convolutional encoder-decoder architecture to perform semantic segmentation, labelling different regions of the document (e.g., invoice date, total amount).

The field of information extraction experienced a transformative shift with the emergence of Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018) introduced a new capability for capturing complex dependencies between textual elements in a bidirectional manner. This was particularly impactful in information extraction tasks, where understanding the surrounding context of each word is essential for identifying key fields. BERT's self-attention mechanism enabled models to dynamically weigh the importance of all tokens in a sequence, allowing for more accurate interpretation of semantic roles, even in the presence of ambiguous or domain-specific vocabulary.

Building upon the success of BERT, researchers sought to incorporate spatial and visual features into the Transformer framework to enhance its ability to process document images. The development of LayoutLM and its successor, LayoutLMv2, was prompted by the necessity to address the challenges posed by structured documents. These models, as discussed in the previous section, represent two influential transformer-based approaches that extend the capabilities of BERT to better handle the challenges of structured documents. The development of both models was driven by the recognition of the limitations inherent to flat, sequence-based processing methodologies. To address these limitations, the models incorporated layout information directly into the representation of documents. While they share a common foundation, LayoutLM and LayoutLMv2 differ significantly in their modelling of documents and the types of information they encode.

LayoutLM is an extension of the standard BERT architecture, which augments each textual token with additional positional embeddings. These embeddings encode the token's 2D coordinates on the document page, also known as the bounding box information. This affords the model the capacity to discern relationships not solely predicated on word order but also on spatial proximity and positioning.

LayoutLMv2, which was introduced shortly thereafter, significantly expanded the architectural design by integrating visual features from the original scanned document image, as well as text and layout embeddings. The extraction of these visual features is facilitated by a CNN, thereby empowering the model to discern font size, style, graphical elements, and even handwritten components. The model also introduces new pre-training objectives, including multimodal learning tasks that involve text-image alignment and spatial-aware masked language modelling. A notable aspect of LayoutLMv2 is its integration of self-attention across all three modalities: text, layout, and image. This capability enables the model to reason collectively over these modalities during both the pre-training and fine-tuning phases.

Simultaneously, Processing Key Information from Documents (PICK) emerged as a complementary model focused specifically on extracting predefined fields from documents (Yu et al., 2021). PICK constructs a relational graph from the OCR-extracted text, where each node represents a token and edges encode spatial and semantic relationships. PICK employs a combination of attention-based graph learning and contextual encoding to model both the local layout structure and the surrounding textual context. This approach has proven to be

particularly effective in scenarios such as invoice processing, where only a subset of tokens is relevant and where the spatial arrangement plays a critical role in determining meaning. As with LayoutLM and LayoutLMv2, PICK utilizes the capabilities of transformer-based architectures to model inter-token dependencies. However, it does so with a more task-specific structure that is designed to extract key-value pairs.

Despite significant progress by models such as LayoutLM and PICK, most document understanding pipelines persistently depend on external OCR systems to extract textual content prior to processing, which introduces several challenges. This reliance introduces several challenges, including cascading errors from OCR inaccuracies, loss of fine-grained visual context, and difficulty handling handwritten or degraded text. DONUT and LayoutLMv3 stand out among these models. To address this issue, DONUT was published and mentioned in detail in the past section to explain that it operates directly on document images, eliminating the need for OCR entirely. DONUT treats document understanding as a sequence-to-sequence problem and uses an encoder-decoder Transformer architecture to learn to generate structured outputs, such as key-value pairs or JSON formats, directly from pixel inputs. This end-to-end approach simplifies the pipeline and improves robustness in scenarios where OCR performance is suboptimal.

Also published in the same year and very important as the most recognized model among the LayoutLM family, LayoutLMv3, the model chosen to be the focus of this research, represented a substantial advancement over its predecessors and by integrating text, layout, and image information through a unified, multimodal, pretraining framework.

However, unlike DONUT, LayoutLMv3 still relies on OCR-extracted text as an input modality. Its innovation lies in tighter fusion of visual and textual embeddings and a more advanced pre-training strategy that enhances the model's ability to align modalities. While both DONUT and LayoutLMv3 advance document understanding, they do so in fundamentally different ways.

The strong ability shown by LayoutLMv3 to predict KIE from invoices has set it as a key standard. This has led to the creation of many new methods one example is DocExtractNet (Yan et al., 2025). It is built on LayoutLMv3 and it introduces three new modules: ImageEnhance, PrecisionHints, and CrossModalFusion. The ImageEnhance module preprocesses the image to increase clarity, which improves recognition accuracy for low-quality or noisy scans. This step tackles common real-world problems like blurred or poorly scanned receipts. Moreover, the PrecisionHints module enhances the text by recovering missing key-value pairs, which often appear in unstructured or partially recognized OCR outputs. This improvement ensures better data integrity and reliability of the extracted information. Lastly, the CrossModalFusion mechanism combines the visual and textual streams, enabling the model to align and use complementary information from both sources. This collaboration allows for a more complete understanding of document content, significantly increasing extraction accuracy.

3. DATA AND METHODS

The methodology follows a structured workflow to develop the KIE from invoices and receipts using LayoutLMv3. The process is shown in Figure 3.1 and is detailed during this section.

Figure 3.1 – Methodology pipeline

The dataset that is used in the process consists in 813 invoices and receipts written in Portuguese and for each image the classified key information fields (Cruz & Castelli, 2022). This work only consider company, date, address and total amount as fields to be predicted. This dataset presents challenges such as poor paper quality, different resolutions and lack of classified entities. Figure 3.2 shows the respective image and annotations for one sample.


```
{
    "company": "copimatica",
    "date": "2017-11-11",
    "address": "oeiras parque, loja 1017 - paco arcos 2780-219 oeiras",
    "total": "5.43"
}
```


Figure 3.2 – Image and annotations

3.1. PRE-PROCESSING

This section explores the techniques used for efficiently converting the visual data into structured, machine-readable information. The methods are detailed in Figure 3.3 that illustrates the pipeline utilized for pre-processing and labelling, the objective of which was to prepare the images, text and bounding boxes to input in LayoutLMv3 (Huang et al., 2022).

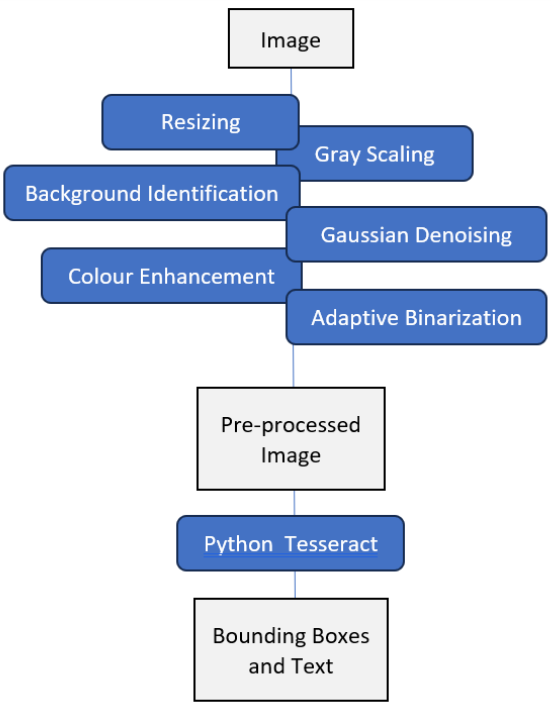

Figure 3.3 - Pre-processing pipeline

To standardize the dimensions of the images for subsequent processing, the resize function from the OpenCV library was used to resize all input images. This function performs interpolation, mapping the original pixel values to a new size while preserving spatial consistency and adjusting the resolution.

The conversion of colour images to grayscale involves a reduction of visual data from multiple channels (RGB) to a single-channel representation, where intensity values range from 0 (black) to 255 (white). This simplification enhanced the efficiency and effectiveness of OCR, as grayscale images emphasize critical structural and textural features such as edges and contours. Furthermore, gray scaling enhances robustness to lighting variations, ensures consistency across different colour spaces, (Kanan & Cottrell, 2012).

We employed a frequently used method named luminance, (Kanan & Cottrell, 2012) to perform gray scaling, this method combines the red, green, and blue channels into a single channel based on the human eye's sensitivity to each colour. This method is given by the equation bellow where R, G and B mean red, green and blue colour, respectively.

$$gray\ value = 0.3 * R + 0.59 * G + 0.11 * B$$

The invoice region within images was detected and isolated by employing Otsu's thresholding method, already reviewed in the literature review. This global thresholding method was used to convert the grayscale version of the image into a binary format. This binarization step is essential for highlighting foreground elements (potential invoice regions) against the background, facilitating subsequent structural analysis.

After the binarization process, the contour detection procedure was implemented by employing the OpenCV findcontours function, parametrized using the retr_external and chain_approx_simple parameters. This method identifies the external boundaries of all discrete foreground regions in the binary image. The algorithm proceeded to calculate the area of each contour using the OpenCV contourarea function and largest contour was selected. After identifying the dominant contour, the OpenCV drawcontours function is employed to generate a binary mask. This mask utilizes the function to isolate the invoice region, thereby filling the background region with white. The result of invoice background identification can be seen in Figure 3.4.

Figure 3.4 – Image with background removed

The concluding stages of the preprocessing pipeline, namely denoising, contrast enhancement and adaptive binarization, were applied at the patch level rather than to the entire image. This decision was motivated by the necessity to account for local variations in illumination, background texture, and text density, which are commonly present in scanned document images. It has been demonstrated that global operations are incapable of adapting effectively to such localized changes, which consequently results in suboptimal binarization outcomes (Sauvola et al., 1997). The method involves the iterative extraction of square sub-regions from the original image by sliding a window of a specific size, defined by a patch size that adjusts according to the invoice size. By dividing the image into smaller patches and processing each

one individually, this approach ensures the preservation of local characteristics with higher precision.

After performing image division, denoising, contrast enhancement and binarization were implemented at the patch level, thereby enabling localized adjustments. Prior to the process of binarization, each patch undergoes a series of modifications. Initially, it is subjected to a form of blurring known as Gaussian blurring, the purpose of which is to attenuate noise. This is then followed by the implementation of a technique referred to as Contrast Limited Adaptive Histogram Equalization (CLAHE) (Reza, 2004).

The binarization process was executed through the implementation of the Sauvola thresholding method (Sauvola et al., 1997), a well-established adaptive thresholding technique that has demonstrated its resilience in the face of document degradation or non-uniformity. A suitable window size of 25 pixels was defined for Sauvola's algorithm, which calculates a local threshold for each pixel in the enhanced patch. Pixels that exceeded the computed threshold were allocated to the foreground, predominantly text. Finally, the binarized patches are seamlessly reassembled into a complete binarized image that can be seen in Figure 3.5.

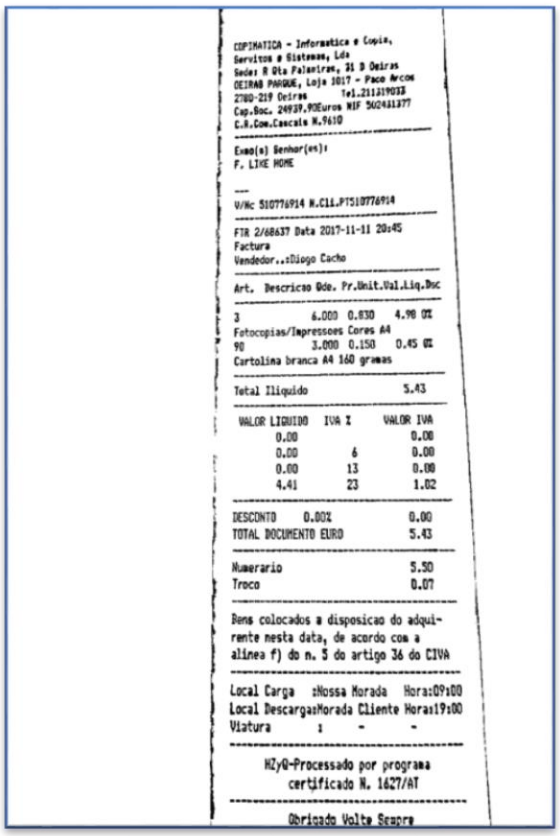

Figure 3.5 – Image after pre-processing phase

In the second stage of the process, both the text and the bounding boxes were extracted from the image using an OCR processor, named as Tesseract (Smith et al., 2007), its operation is facilitated by the use of pytesseract library and the available module to select the language.

3.2. LABELLING

This phase is responsible for converting the received text and bounding information. The data must be formatted according to the model's specifications and for that it is imperative that the extracted text is tokenized and labelled with the respective key information fields to ensure its integration into the model. This poses a challenge that is associated with the efficacy of our OCR and the initial pre-processing stage. To illustrate, consider Figure 3.6, a scenario in which our OCR extracted an address that contains a wrong letter.

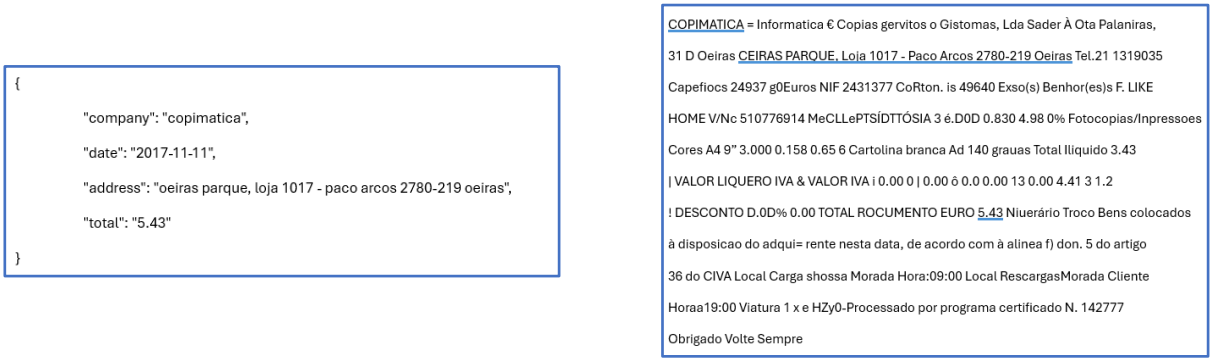

Figure 3.6 - Annotations and extracted text from OCR

Due to the impracticality of labelling the text based on exact matches, an alternative approach to labelling the tokens was employed. In this approach, all the tokens were compared with the annotations in exact match. Tokens that did not match were considered as a match if they were not a digit, they had more than 2 letters and the Levenshtein distance (Levenshtein, 1966) between the predicted and real word was equal to 1. Tokens that did not match with the approach previously mentioned were classified as "Other." Instances of the approach in action are shown in Table 3.1.

The Levenshtein distance is defined as the minimum number of operations, including insertion, deletion, and substitution, required to transform the predicted result into the ground truth. Under the assumption that the cost of each operation is equivalent to 1. Table 3.1 shows a Levenshtein distance equal to three when applied to a name of a company from a sample of our dataset.

Predicted token	Annotation token	Levenshtein Distance	Match
Pingu	Pingo	1	Yes
Doc	Doce	1	Yes
IK	IKEA	2	No
12	12	0	Yes
Levenshtein Distance = 3 MARKRO CASH & CARYR → MAKRO CASH & CARRY Insertion Deletion Substitution			

Table 3.1 – Labelling examples

This labelling approach constitutes a pivotal element of the present study. Therefore, it is imperative to deliberate on its strengths and limitations. While it corrects a significant portion of the errors produced by the OCR process, it does not eliminate them entirely. In the event that a number is not equal to the ground truth or segment of the OCR text output deviates significantly from the actual text, the algorithm fails to recognize it as a valid match. Consequently, the correct class may not be assigned to that part. These discrepancies result in the absence of certain classes from both the training and validation data sets. Consequently, during the validation process, performance metrics may be skewed. Specifically, results may appear worse if the model predicts a label correctly, but the labelling did not identify it.

3.3. Training Phase and Model Configurations

To assess the efficacy of the proposed methodology, the dataset was divided using hold-out, into two segments: 80% was allocated for training purposes (650 images), while the remaining 20% (163 images) was designated for testing. The selected ratio aligns with established best practices in machine learning, offering a trade-off between model training and unbiased performance assessment.

Google Colab notebook is utilized for training, facilitating access to graphical processing units (GPUs). While the specific hardware is not selected by the user, in this study, most of the computationally intensive tasks are executed on a NVIDIA A100-SXM4-40GB GPU.

The training process involved adjusting model parameters to minimize errors, as explained in the paragraphs bellow. Python scripts and LLM were used to convert the training and testing datasets into a format that is compatible with the model's requirements, for the implementation of evaluation metrics that will be employed during the subsequent phase of the project, and it also helped debugging issues during training and pre-processing phases.

In this work, the LayoutLMv3Tokenizer was employed, with the configuration set to apply maximum padding. This guarantees that all input sequences within a batch are of uniform length, thereby facilitating efficient batch processing and stable memory usage during training and evaluation.

In terms of model selection, the LayoutLMv3 Base variant was selected with the objective of achieving an equilibrium between performance and computational efficiency. By selecting the base version, experiments can be conducted within reasonable time and hardware constraints while still benefiting from the advanced capabilities introduced in the LayoutLMv3 architecture.

The training setup and hyper-parameters were inspired by the configurations presented in the original LayoutLMv3 paper, Table 3.2. However, there were some differences. The batch size was set to 4, the learning rate to 1e-5, and the number of epochs to 10. Furthermore, the metric for selecting the best-performing model during training was set to the best cross-entropy loss score on test set, ensuring that the version with the lowest evaluation loss was retained for downstream tasks.

Parameter	**Value**	**Parameter**	**Value**
vocab_size	50265	hidden_act	gelu
hidden_size	768	hidden_dropout_prob	0.1
num_hidden_layers	12	attention_probs_dropout_prob	0.1
num_attention_heads	12	max_position_embeddings	512
intermediate_size	3072	type_vocab_size	2
initializer_range	0.02	layer_norm_eps	1e-5
max_2d_position_embeddings	1024	coordinate_size	128
shape_size	128	has_relative_attention_bias	True
rel_pos_bins	32	max_rel_pos	128
max_rel_2d_pos	256	rel_2d_pos_bins	64
has_spatial_attention_bias	True	visual_embed	True
input_size	224	patch_size	16

Table 3.2 – Model Parameters

3.4. INFERENCE WITH DOCUMENT AI BY GOOGLE

Document AI is a sophisticated platform for the processing and interpretation of documents, designed to extract and convert unstructured information into structured, organized data. The identification and population of specific fields suitable for integration into databases is facilitated by the technology, thereby enabling more efficient comprehension, analysis, and utilization of the information contained within various types of documents, including invoices and receipts.

Initially, all images were imported into the Document AI platform with the precise same split as that used for training and evaluating LayoutLMv3. The labelling of documents was facilitated by employing a pre-trained model, utilizing the Gemini 2.0 Flash, with the objective of streamlining the process. Subsequently, manual corrections were applied to rectify extraction errors, ensuring the extracted values precisely correspond to those documented in the annotations. The extraction made by the pre-trained model for an invoice can be seen in Figure 3.7.

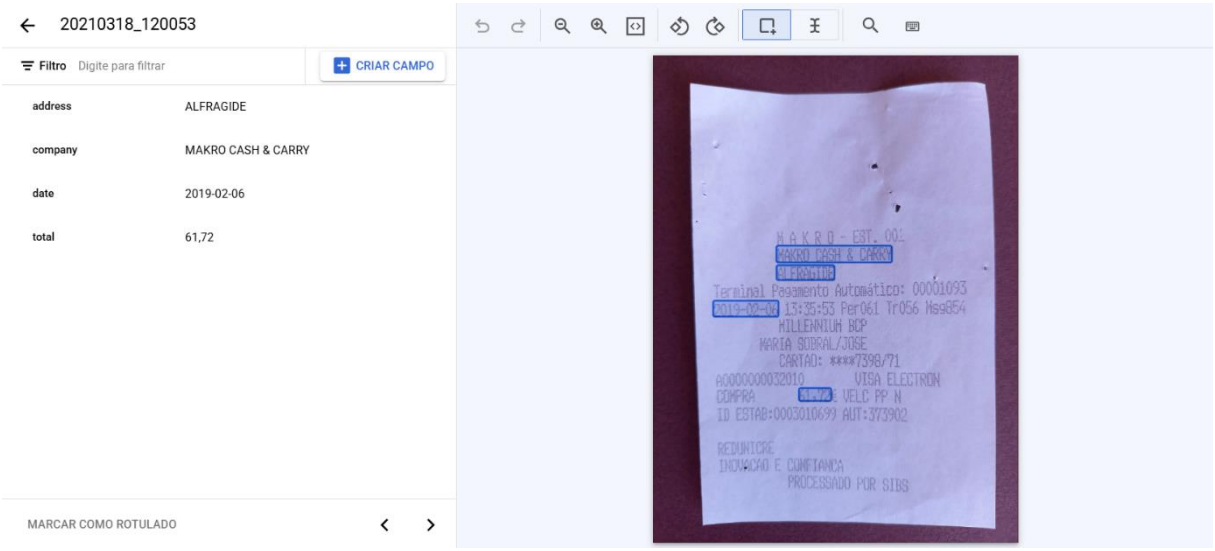

Figure 3.7 – KIE by Document AI pre-trained model

The training step was with a custom Document AI model with no information available about its internal configuration or parameters.

3.5. EVALUATION

The objective of this stage is to quantify the outcomes in order to assess whether meaningful improvements have been achieved. This evaluation phase is important for judging how well the proposed method solves the information extraction problem. It is valuable for both developers and end users, as it offers insights for system improvement and helps with making informed decisions. In this section, three key evaluation metrics were used: precision, recall, and F1-score.

Given that for our work no specific class is accorded a higher priority than others. Consequently, the 'micro' averaging method is employed. This approach involves the aggregation of metric scores across all classes, thereby effectively conducting the evaluation at the token level while disregarding correct predictions pertaining to the "Other" category.

However, it is important to note that if this work is turned into a practical strategy for use in an accounting office, the method of evaluation would need to be reconsidered. In a real operational context, certain key values, like total amounts, tax identifiers, or invoice numbers, usually matter more for accounting tasks than less critical information like address for example. Therefore, a micro-level evaluation might not fully show the practical usefulness or business impact of the system.

Precision is defined as the proportion of positive predictions that are correctly identified. It is calculated by dividing the number of true positives by the sum of true positives (TP) and false positives (FP).

$$Precision = \frac{TP}{TP + FP}$$

In contrast, the recall function calculates the percentage of actual positives that the model has identified. The calculation is performed by dividing the true positives by the sum of true positives and false negatives (FN).

$$Recall = \frac{TP}{TP + FN}$$

It is important to note that precision and recall are often reciprocal, enhancing one can often diminish the other. To address this F1-score is utilised, representing the harmonic mean of precision and recall. A high F1-score is indicative of a model's efficacy in identifying true positives, whilst concomitantly minimising false positives.

$$F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

In the context of KIE from invoices, the terms true positive, false positive, and false negative serve as fundamental indicators of the model's performance in identifying and classifying relevant data fields.

A true positive is defined as the occurrence of a model's accurate detection and labelling of a piece of information that is genuinely present in the document and corresponds with the annotated ground truth. For instance, the receipt in Figure 3.7 that contains the company name "MACRO CASH & CARRY" if the model accurately extracts and assigns it to the company class, this prediction is counted as a true positive.

On the other hand, a false positive is indicative of an incorrect extraction or misclassification. This arises when the model predicts information that is either non-existent or incorrectly

labels an existing entity. To illustrate this point, consider the invoice in Figure 3.6 and a scenario in which the model predicts as address “CEIRAS” instead of “OEIRAS”.

Finally, a false negative outcome indicates that the model has failed to identify a piece of information that should have been extracted according to the annotations. To clarify this point, consider the invoice in Figure 3.6 that includes a date, but the model disregards this information entirely.

4. RESULTS AND DISCUSSION

A comprehensive comparative analysis was undertaken to rigorously assess the performance enhancements afforded by the LayoutLMv3 and the customer Document AI architecture in the context of relevant field prediction on invoices. The evaluation process entailed the establishment of two distinct baseline computations.

The baseline employed our primary model, LayoutLMv3, in conjunction with the commercial custom Document AI solution, which has been specifically engineered for invoice processing, thereby establishing a tangible benchmark.

The comparisons were conducted by training and testing on the publicly available Portuguese invoice dataset, which comprises a total of 813 distinct invoice samples. The objective of the classification task was to accurately identify and extract data for four critical fields. The fields in question include the company name, company address, date and total amount. This methodological approach facilitates a robust and multi-faceted understanding of LayoutLMv3 efficacy in this domain.

Table 4.1 provides a synopsis of the global precision, recall, and F1-score for LayoutLMv3 and Google Document AI custom model.

Model	Precision	Recall	F1-score
LayoutLMv3 (Lv3)	0.79	0.18	0.30
Google Document AI (DAI)	0.80	0.75	0.78

Table 4.1 – Comparison between LayoutLMv3 and Google Document AI custom model

As demonstrated in Table 1, a marked contrast is evident between the two models. LayoutLMv3 achieved a relatively high precision of 0.79, signifying that when the model predicted a field, it was usually correct. However, the model demonstrated a substantially lower recall rate of 0.18, suggesting that it was unable to detect a significant proportion of the relevant fields present in the documents. Consequently, the F1-score, a metric that harmonizes precision and recall, was only 0.30 for LayoutLMv3, underscoring the considerable gap between accurate predictions and comprehensive extraction.

In contrast, Document AI demonstrated a more balanced and robust performance profile. The precision of the model was marginally higher at 0.80, and the recall was notably improved at 0.75. This high recall can be taken as evidence that Document AI successfully detected the majority of target fields, thereby maintaining a high level of correctness, which resulted in an overall F1-score of 0.78. This performance profile indicates that Document AI can handle the

variability and complexity inherent in Portuguese invoices and receipts, delivering both accurate and comprehensive extractions.

The global results obtained reveal a significant distinction between the two approaches: when a class is found LayoutLMv3 is equally good as Document AI on correctly identifying the characters to be correct but tends to find less classes than Document AI, likely due to limitations in OCR extraction and the model's sensitivity to layout irregularities. Document AI pipeline demonstrates a superior capacity for generalization, attaining a high detection rate with minimal compromise to precision.

In order to gain a more granular understanding of the models behavior, performance metrics were also evaluated individually for each target field: company, date, address, and total. The results for each class are summarized in Table 4.2.

Class	Precision (Lv3)	Precision (DAI)	Recall (Lv3)	Recall (DAI)	F1-score (Lv3)	F1-score (DAI)
Company	0.77	0.75	0.08	0.64	0.14	0.69
Address	0.76	0.59	0.26	0.55	0.39	0.56
Date	0.87	0.96	0.26	0.95	0.40	0.96
Total	0.84	0.89	0.73	0.88	0.78	0.89

Table 4.2 - Comparison by key field between LayoutLMv3 and Google Document AI custom model

In the case of the 'Company' field, LayoutLMv3 demonstrates a marginally higher level of precision (0.77) in comparison to Google Document AI (0.75). This finding indicates that when LayoutLMv3 predicts a company name, there is a marginal increase in the probability of its accuracy in comparison to Google Document AI. However, the recall metrics present a more nuanced picture, with LayoutLMv3 achieving a significantly low recall of 0.08, while Google Document AI shows 0.64. This finding suggests that LayoutLMv3 experiences significant challenges in accurately identifying all the company names present on the invoices, with a substantial proportion being overlooked.

The most salient discrepancy observed in the 'Company' class pertains to the F1-score. LayoutLMv3 displays an extremely low F1-score of 0.14, which is in stark contrast to the robust 0.69 score achieved by Google Document AI. The marked difference in the F1-score, which is chiefly due to the very low recall of LayoutLMv3, indicates a significant problem with LayoutLMv3's capacity to extensively extract company names. The F1-score of 0.13, while demonstrating relatively high precision, suggests that the model is producing a negligible number of accurate positive predictions in relation to the total number of actual positive instances. Conversely, Google Document AI F1-score of 0.69 indicates a much healthier

balance between its precision and recall, making it a far more reliable model for extracting company names.

The results of the 'Address' field analysis demonstrate a more pronounced divergence between the models. LayoutLMv3 achieves a high precision of 0.76, indicating that when extracting an address, it is frequently accurate. Conversely, Google Document AI precision for 'Address' is notably lower at 0.59, suggesting that it makes more false positive errors when attempting to extract addresses.

However, the recall values present a contrasting narrative. LayoutLMv3 recall is a mere 0.26, which suggests that it is unable to recognize a significant majority of the actual addresses present in the documents. This low recall value signifies that, while its predictions are frequently accurate, it is unable to identify a significant proportion of the relevant data. Conversely, Google Document AI exhibits a substantially superior recall of 0.55, signifying its remarkable capacity to identify and extract a greater proportion of authentic addresses.

Consequently, when examining the F1-score, Google Document AI (0.56) substantially outperforms LayoutLMv3 (0.39). This outcome underscores the fact that, despite the enhanced precision exhibited by LayoutLMv3, its markedly deficient recall capabilities have a deleterious effect on its overall efficacy regarding address extraction. It is evident that Google Document AI has achieved a superior equilibrium between precision and recall, thus providing a more pragmatic solution for this exacting field. The complexity and variability of address formats across invoices are likely to be significant contributing factors to these challenges, especially for LayoutLMv3 low recall, indicating an inability to generalize across different address structures.

The 'Date' field demonstrates the most distinct performance differential, with Google Document AI consistently outperforming LayoutLMv3 by a significant margin. Google Document AI achieves an impressive precision of 0.96 and a recall of 0.95, culminating in an F1-score of 0.96. The findings of this study indicate that Google Document AI demonstrates a high degree of accuracy and comprehensiveness in the identification and extraction of dates from invoices. This suggests that there is a near-perfect balance between the avoidance of false positives and the minimization of false negatives.

In stark contrast, while LayoutLMv3 demonstrates a commendable precision of 0.87, its recall for the 'Date' field experiences a precipitous decline to 0.26. Analogous to the 'Address' field, this indicates that LayoutLMv3, despite being accurate when it makes a prediction, fails to identify the vast majority of dates present on the invoices. This low recall has a significant impact on the model's F1-score, which stands at a mere 0.40. The marked difference in the recall rate suggests that Google Document AI employs a significantly more robust mechanism for date detection, which may be more adaptable to different date formats and document locations.

For the "Total" field, both models demonstrate robust performance, typically attaining elevated scores across all metrics. LayoutLMv3 demonstrates a precision of 0.84 and a recall of 0.73, yielding an F1-score of 0.78. Google Document AI demonstrates marginal superiority across all metrics, exhibiting a precision of 0.89, which approaches the LayoutLMv3 threshold. It also shows a recall of 0.88 and an F1-score of 0.89.

While both models are highly effective at extracting the total amount, Google Document AI demonstrates a clearer advantage, primarily due to its significantly higher recall. This finding suggests that Google Document AI demonstrates a higher degree of success in capturing nearly all instances of the total amount, while maintaining a level of precision that is comparable to that of LayoutLMv3. The high performance of both models in this field is likely attributable to the relatively consistent formatting and prominent placement of the total amount on most invoices, making it a more straightforward extraction task compared to the more variable "Address" field.

The comparative analysis of the LayoutLMv3 and Google Document AI models for KIE from invoices reveals several critical insights. A comprehensive evaluation of the Google Document AI model reveals its consistent superiority over LayoutLMv3, particularly regarding recall and, consequently, the F1-score, across a wide range of classes. While LayoutLMv3 occasionally exhibits competitive or even slightly higher precision in certain fields, such as "Company" and "Address," its recall frequently manifests as a significant weakness, particularly in the "Company," "Address," and "Date" fields. This finding indicates that, while LayoutLMv3 is effective in ensuring the accuracy of its positive predictions, it frequently fails to identify all relevant instances, resulting in a high rate of false negatives.

The significant disparities in F1-scores, particularly for "Company," "Address," and "Date," underscore Google Document AI superior balanced performance. The F1-score, defined as the harmonic mean of precision and recall, has been shown to provide a more comprehensive metric for evaluating the practical utility of a model. Google Document AI higher F1-scores imply that it is generally more effective for real-world applications where both accuracy (precision) and completeness (recall) are crucial.

The challenges observed for LayoutLMv3, particularly its low recall, could stem from various factors, including limitations in the OCR and entity matching processes used. Inaccurate OCR can result in erroneous text that the model cannot correctly interpret, while flawed entity matching can misclassify or fail to recognize valid data points. Additionally, Tesseract or LayoutLMv3 model architecture may exhibit reduced robustness or flexibility in handling variations in document layouts and textual representations of data, resulting in missed extractions.

5. CONCLUSIONS AND FUTURE WORKS

The final chapter of this research project aims to bring together the findings, discuss their implications, recognize the challenges encountered during the research process, and suggest possible future directions for the work presented in this thesis. The research focuses on developing and evaluating an important information extraction system for invoices, using a multimodal LayoutLMv3 model. This project meets a key need for efficient and accurate automation in document processing.

The main goal of this work was to develop and comprehensively evaluate a key information extraction system for invoices utilizing a multimodal LayoutLMv3 model. The research specifically examined the efficacy of a model that integrates visual, textual, and layout information in identifying and extracting significant data fields. This is of particular pertinence for documents characterized by diverse layouts like invoices and receipts.

This objective emerged from the inherent challenges associated with automating invoice processing and this work proposes a multimodal deep learning approach to transcend these limitations, thereby offering a robust and adaptable solution that mimics the human ability to understand document context. The successful achievement of this objective promised not only to demonstrate the practical applicability of advanced AI models in document understanding but also to provide a foundational system for improving efficiency in business operations.

Although the theory behind multimodal understanding is strong, using LayoutLMv3 on this Portuguese dataset of invoices and receipts did not achieve the level of effectiveness that was expected. Findings showed that LayoutLMv3, even with its multimodal design, struggled to reliably and accurately detect and extract important details such as company, date, and address from documents with diverse and complex layouts. This was especially true when compared to the established Google Document AI system. In essence, while a multimodal approach shows promise for document understanding, LayoutLMv3, in this specific experimental configuration, did not demonstrate itself to be the most effective solution for invoice information extraction.

Consequently, while the automation of financial processes remains a significant concern, the LayoutLMv3 model employed did not yield the anticipated robust solution. Nonetheless, this work has revealed important points about multimodal models in a real-world situation. It also shows where we can improve, particularly for domain-specific datasets. Although automating financial processes is a key goal, this experiment has made the current limitations and strengths of the research topic and it provided a useful reference for future research and development.

5.1. RESEARCH LIMITATIONS

The principal limitations of this research are considered to be as follows:

- The study used the hold-out method for both training and evaluation. It is widely recognized that the results from this approach depend on how the dataset is split, which makes it less reliable than other methods like cross-validation. However, because of the costs associated with computation, we chose to use the hold-out method in this study.
- The ability to make accurate predictions depends a lot on factors like OCR and the quality of the pre-processing of the image.
- The assumptions made by assigning labels to the tokens that were wrongly generated by Tesseract are conducive to errors.
- The use of basic parameter settings might have affected the results.

5.2. FUTURE WORK

The findings of this thesis, especially the comparative results, create many exciting opportunities for future research and practical use. Building on the foundational work presented here and recognizing the performance difference between LayoutLMv3 and the document AI model, we can identify several key areas for further exploration.

Future work should focus on diversifying datasets to improve model strength and general use, this includes exploring new ways to create synthetic invoices. This involves developing support for multiple languages by fine-tuning models on larger and more varied multilingual invoice datasets.

Since the quality of predictions relies heavily on text recognition and pre-processing, future work should look into using better OCR engines and improved pre-processing methods. This includes language-specific models and techniques designed for noisy scans. Also, to fix errors that occur during token labeling, we should develop a stronger annotation strategy, adding a more robust post-processing approach like semi-automated for example.

Finally further research into deploying the model in production and methods for improving the understanding of multimodal transformer models and other document AI models would offer important insights into why certain extraction decisions occur.

By pushing the limits of document intelligence, this work lays the foundation for smarter, more adaptable systems that can revolutionize how businesses process and understand information at scale. As a strategic next step, this research can grow into a production-ready pipeline by combining scalable data augmentation, ongoing evaluation, and modular deployment with monitoring.

BIBLIOGRAPHICAL REFERENCES

Ahlawat, S., Choudhary, A., Nayyar, A., Singh, S., & Yoon, B. (2020). Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN). Sensors, 20(12), 3344. https://doi.org/10.3390/s20123344

Al-Khatatneh, A., Pitchay, S. A., & Al-qudah, M. (2015). A Review of Skew Detection Techniques for Document. 2015 17th UKSim-AMSS International Conference on Modelling and Simulation (UKSim), 316–321. https://doi.org/10.1109/UKSim.2015.73

Amin, A., & Fischer, S. (2000). A Document Skew Detection Method Using the Hough Transform. Pattern Analysis & Applications, 3(3), 243–253. https://doi.org/10.1007/s100440070009

Arica, N., & Yarman-Vural, F. T. (2001). An overview of character recognition focused on off-line handwriting. IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews), 31(2), 216–233. https://doi.org/10.1109/5326.941845

Arora, S., Bhattacharjee, D., Nasipuri, M., Malik, L., Kundu, M., & Basu, D. K. (2010). Performance Comparison of SVM and ANN for Handwritten Devnagari Character Recognition. https://doi.org/10.48550/ARXIV.1006.5902

Ben-Hur, A., & Weston, J. (2010). A User's Guide to Support Vector Machines. In O. Carugo & F. Eisenhaber (Eds.), Data Mining Techniques for the Life Sciences (Vol. 609, pp. 223–239). Humana Press. https://doi.org/10.1007/978-1-60327-241-4_13

Bloomberg, D., & Kopec, G. (n.d.). Method and apparatus for identification of document skew (Patent No. EP0431962B1).

Borges Oliveira, D. A., & Viana, M. P. (2017). Fast CNN-Based Document Layout Analysis. 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), 1173–1180. https://doi.org/10.1109/ICCVW.2017.142

Chen, H., Perozzi, B., Al-Rfou, R., & Skiena, S. (2018). A Tutorial on Network Embeddings (Version 1). arXiv. https://doi.org/10.48550/ARXIV.1808.02590

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21–27. https://doi.org/10.1109/TIT.1967.1053964

Cruz, F., & Castelli, M. (2022). Dataset of invoices and receipts including annotation of relevant fields [Dataset]. Zenodo. https://doi.org/10.5281/ZENODO.6371710

De, R., Chakraborty, A., & Sarkar, R. (2020). Document Image Binarization Using Dual Discriminator Generative Adversarial Networks. IEEE Signal Processing Letters, 27, 1090–1094. https://doi.org/10.1109/LSP.2020.3003828

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (Version 2). arXiv. https://doi.org/10.48550/ARXIV.1810.04805

Divakar, N., & Babu, R. V. (2017). Image Denoising via CNNs: An Adversarial Approach. 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 1076–1083. https://doi.org/10.1109/CVPRW.2017.145

Dongre, V. J., Mankar, V. H., & Suganya, G. (2010). A Review of Research on Devnagari Character Recognition. International Journal of Computer Applications, 12(2), 8–15. https://doi.org/10.5120/1653-2224

Donoho, D. L. (1995). De-noising by soft-thresholding. IEEE Transactions on Information Theory, 41(3), 613–627. https://doi.org/10.1109/18.382009

Due Trier, Ø., Jain, A. K., & Taxt, T. (1996). Feature extraction methods for character recognition-A survey. Pattern Recognition, 29(4), 641–662. https://doi.org/10.1016/0031-3203(95)00118-2

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian Network Classifiers. Machine Learning, 29(2/3), 131–163. https://doi.org/10.1023/A:1007465528199

Hossain, M. Z., Amin, M. A., & Yan, H. (2012). Rapid Feature Extraction for Optical Character Recognition (Version 1). arXiv. https://doi.org/10.48550/ARXIV.1206.0238

Hussein, T., Omar, H., & Jihad, K. (2021). A study on image noise and various image denoising techniques. 11, 27–42. https://doi.org/10.17605/OSF.IO/87XGJ

Huang, Y., Lv, T., Cui, L., Lu, Y., & Wei, F. (2022). LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking (Version 3). arXiv. https://doi.org/10.48550/ARXIV.2204.08387

Jaekyu Ha, Haralick, R. M., & Phillips, I. T. (1995). Recursive X-Y cut using bounding boxes of connected components. Proceedings of 3rd International Conference on Document Analysis and Recognition, 2, 952–955. https://doi.org/10.1109/ICDAR.1995.602059

Jain, A. K., & Zhong, Y. (1996). Page segmentation using texture analysis. Pattern Recognition, 29(5), Article 5. https://doi.org/10.1016/0031-3203(95)00131-X

Kanan, C., & Cottrell, G. W. (2012). Color-to-Grayscale: Does the Method Matter in Image Recognition? PLoS ONE, 7(1), e29740. https://doi.org/10.1371/journal.pone.0029740

Katti, A. R., Reisswig, C., Guder, C., Brarda, S., Bickel, S., Höhne, J., & Faddoul, J. B. (2018). Chargrid: Towards Understanding 2D Documents. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 4459–4469. https://doi.org/10.18653/v1/D18-1476

Kim, G., Hong, T., Yim, M., Nam, J., Park, J., Yim, J., Hwang, W., Yun, S., Han, D., & Park, S. (2022). OCR-Free Document Understanding Transformer. In S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, & T. Hassner (Eds.), Computer Vision – ECCV 2022 (Vol. 13688, pp. 498–517). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-19815-1_29

Lauer, F., Suen, C. Y., & Bloch, G. (2007). A trainable feature extractor for handwritten digit recognition. Pattern Recognition, 40(6), 1816–1824. https://doi.org/10.1016/j.patcog.2006.10.011

Lebourgeois, F., Bublinski, Z., & Emptoz, H. (1992). A fast and efficient method for extracting text paragraphs and graphics from unconstrained documents. Proceedings., 11th IAPR International Conference on Pattern Recognition. Vol.II. Conference B: Pattern Recognition Methodology and Systems, 272–276. https://doi.org/10.1109/ICPR.1992.201771

Levenshtein, V. I. (1966). Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Soviet Physics Doklady, 10, 707.

Memon, J., Sami, M., Khan, R. A., & Uddin, M. (2020). Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR). IEEE Access, 8, 142642–142668. https://doi.org/10.1109/ACCESS.2020.3012542

Mor, B., Garhwal, S., & Kumar, A. (2021). A Systematic Review of Hidden Markov Models and Their Applications. Archives of Computational Methods in Engineering, 28(3), 1429–1448. https://doi.org/10.1007/s11831-020-09422-4

Nguyen, T. T. H., Jatowt, A., Coustaty, M., & Doucet, A. (2022). Survey of Post-OCR Processing Approaches. ACM Computing Surveys, 54(6), 1–37. https://doi.org/10.1145/3453476

Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man, and Cybernetics, 9(1), 62–66. https://doi.org/10.1109/TSMC.1979.4310076

Papandreou, A., Gatos, B., Louloudis, G., & Stamatopoulos, N. (2013). ICDAR 2013 Document Image Skew Estimation Contest (DISEC 2013). 2013 12th International Conference on Document Analysis and Recognition, 1444–1448. https://doi.org/10.1109/ICDAR.2013.291

Pavlidis, T. (1993). Threshold selection using second derivatives of the gray scale image. Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR '93), 274–277. https://doi.org/10.1109/ICDAR.1993.395733

Portilla, J., Strela, V., Wainwright, M. J., & Simoncelli, E. P. (2003). Image denoising using scale mixtures of gaussians in the wavelet domain. IEEE Transactions on Image Processing, 12(11), Article 11. https://doi.org/10.1109/TIP.2003.818640

Reza, A. M. (2004). Realization of the Contrast Limited Adaptive Histogram Equalization (CLAHE) for Real-Time Image Enhancement. The Journal of VLSI Signal Processing-Systems for Signal, Image, and Video Technology, 38(1), 35–44. https://doi.org/10.1023/B:VLSI.0000028532.53893.82

Sauvola, J., Seppanen, T., Haapakoski, S., & Pietikainen, M. (1997). Adaptive document binarization. Proceedings of the Fourth International Conference on Document Analysis and Recognition, 1, 147–152. https://doi.org/10.1109/ICDAR.1997.619831

Smith, R. (2007). An Overview of the Tesseract OCR Engine. Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2, 629–633. https://doi.org/10.1109/ICDAR.2007.4376991

Taghva, K., & Stofsky, E. (2001). OCRSpell: An interactive spelling correction system for OCR errors in text. International Journal on Document Analysis and Recognition, 3(3), 125–137. https://doi.org/10.1007/PL00013558

Wang, H., Pan, C., Guo, X., Ji, C., & Deng, K. (2021). From object detection to text detection and recognition: A brief evolution history of optical character recognition. WIREs Computational Statistics, 13(5), e1547. https://doi.org/10.1002/wics.1547

Wei, H., Baechler, M., Slimane, F., & Ingold, R. (2013). Evaluation of SVM, MLP and GMM Classifiers for Layout Analysis of Historical Documents. 2013 12th International Conference on Document Analysis and Recognition, 1220–1224. https://doi.org/10.1109/ICDAR.2013.247

Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., & Zhou, M. (2020). LayoutLM: Pre-training of Text and Layout for Document Image Understanding. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 1192–1200. https://doi.org/10.1145/3394486.3403172

Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., Lu, Y., Florencio, D., Zhang, C., Che, W., Zhang, M., & Zhou, L. (2021). LayoutLMv2: Multi-modal Pre-training for Visually-rich Document Understanding. 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), 2579–2591. https://doi.org/10.18653/v1/2021.acl-long.201

Yan, Z., Ye, Z., Ge, J., Qin, J., Liu, J., Cheng, Y., & Gurrin, C. (2025). DocExtractNet: A novel framework for enhanced information extraction from business documents. Information Processing & Management, 62(3), 104046. https://doi.org/10.1016/j.ipm.2024.104046

Yang, J., Wang, K., Li, J., Jiao, J., & Xu, J. (2012). A fast adaptive binarization method for complex scene images. 2012 19th IEEE International Conference on Image Processing, 1889–1892. https://doi.org/10.1109/ICIP.2012.6467253

Yang, X., Yumer, E., Asente, P., Kraley, M., Kifer, D., & Giles, C. L. (2017). Learning to Extract Semantic Structure from Documents Using Multimodal Fully Convolutional Neural Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4342–4351. https://doi.org/10.1109/CVPR.2017.462

Ying Liu, & Srihari, S. N. (1997). Document image binarization based on texture features. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(5), 540–544. https://doi.org/10.1109/34.589217

Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., Lu, Y., Florencio, D., Zhang, C., Che, W., Zhang, M., & Zhou, L. (2021). LayoutLMv2: Multi-modal Pre-training for Visually-rich Document Understanding. 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), 2579–2591. https://doi.org/10.18653/v1/2021.acl-long.201

Yang, X., Yumer, E., Asente, P., Kraley, M., Kifer, D., & Giles, C. L. (2017). Learning to Extract Semantic Structure from Documents Using Multimodal Fully Convolutional Neural Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4342–4351. https://doi.org/10.1109/CVPR.2017.462

Yu, B., & Jain, A. K. (1996). A robust and fast skew detection algorithm for generic documents. Pattern Recognition, 29(10), Article 10. https://doi.org/10.1016/0031-3203(96)00020-9

Yu, W., Lu, N., Qi, X., Gong, P., & Xiao, R. (2021). PICK: Processing Key Information Extraction from Documents using Improved Graph Learning-Convolutional Networks. 2020 25th International Conference on Pattern Recognition (ICPR), 4363–4370. https://doi.org/10.1109/ICPR48806.2021.9412927

Zhang, K., Zuo, W., Chen, Y., Meng, D., & Zhang, L. (2017). Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. IEEE Transactions on Image Processing, 26(7), 3142–3155. https://doi.org/10.1109/TIP.2017.2662206

Zhang, K., Zuo, W., & Zhang, L. (2018). FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising. IEEE Transactions on Image Processing, 27(9), 4608–4622. https://doi.org/10.1109/TIP.2018.2839891

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa