technische universität
dortmund

Segmentation-Free Handwritten Text Recognition on Portuguese Death Certificates

Ahmed Hosam Aboelezz Hassan
October 13, 2025

Supervisors:
Prof. Dr.-Ing. Gernot A. Fink
Arthur Matei, M.Sc.

Fakultät für Informatik
Technische Universität Dortmund
http://www.cs.uni-dortmund.de

CONTENTS

1 MOTIVATION

1.1 MOTIVATION AND PROBLEM STATEMENT

The widespread adoption of digital technologies in the late 20th century marked a significant transition in the way documents are created and stored. However, this technological shift also left behind an enormous legacy: vast collections of handwritten historical documents (such as death certificates that predate the digital era) remain largely unprocessed and difficult to access for systematic analysis. These documents contain invaluable information about past societies, economies, and demographic patterns, but their handwritten nature creates major obstacles for large-scale digitization and analysis.

Historical documents such as census records and tax registers contain detailed information that can reveal long-term social trends, economic patterns, and demographic changes. For historians and demographers, these sources provide unique insight into the behavior of the population, the patterns of mortality, and the social structures of the past communities. However, extracting this information has traditionally required extensive manual labor, creating a significant bottleneck in historical research productivity.

The Portuguese death certificates[1] from the city of Porto, covering the period from 1834 to 1910, provide a compelling example of these archival challenges. These certificates represent a rich dataset for understanding mortality patterns and public health trends in Portugal of the 19th century. Each certificate contains structured information including personal details of the deceased, dates of death, causes of mortality, poverty status, and family relationships - data that could significantly advance our understanding of historical disease patterns and demographic transitions.

Despite their research value, manually annotating these documents presents considerable practical challenges. Current estimates show that processing a single year's worth of death certificates requires approximately six months of dedicated annotation work. This time-consuming manual process not only slows down research, but also represents an inefficient use of scholarly resources that could be better directed toward higher-level analytical tasks and historical interpretation.

[1] https://ciencia.iscte-iul.pt/projects/files/46310

1.2 AUTOMATED INFORMATION EXTRACTION

Automating the extraction of structured information from handwritten historical records requires methods that can reliably convert complex document images into machine-readable data. Over the years, two main paradigms have emerged for this task: traditional handwriting text recognition (HTR) pipelines based on segmentation, and more recent segmentation-free approaches. The following subsections outline their principles and challenges, and motivate the choice of the model for this thesis.

1.2.1 *Traditional HTR and its challenges*

The usual approach for similar annotation tasks was to break the task into two stages: Document Layout Analysis (DLA), where the text lines or words are segmented, and Handwritten Text Recognition (HTR), which generates transcriptions by mapping the parts from the DLA stage to predefined words and sentences. In practice, this also means that two separate models need to be trained and maintained, one for segmentation and one for recognition, each requiring its own annotated training data. This approach, however, has yet another disadvantage: if the segmentation was wrong, the entire process will be wrong because the second stage relies directly on it. Furthermore, relying on the output of DLA task can sometimes be a disadvantage, as the layout itself can help the model predict what text is at this position. Those problems can be better seen in documents with nested structures like table of contents for example. As we do not only want to recognize what sections we have, but also the relationship between sections and subsections.

1.2.2 *Segmentation-Free Approaches*

To address those limitations, segmentation-free approaches have been an interesting alternative in the field of HTR. These approaches merge the two steps (document layout and text recognition) into one end-to-end process, benefiting from the fact that layout and text can actually strengthen each other.

In Segmentation-free paradigms, neural architectures such as encoder-decoder models absorb all the contextual information from the document, including where they position and their relative structure. By skipping the segmentation stage, these methods can better handle complex document structure and can actually penalize the wrong layout recognition, in comparison to the traditional models, that had no access to the layout component.

They can be divided into models for just parsing the layout and text, and models for extracting specific information from the documents like the birthday of the decreased in the Portuguese death certificates.

Those models shine at most with complex layout structures like the table of content example or even the proposed Portuguese death certificates. Models using this strategy learn to focus on relevant parts of the document automatically, allowing more flexible recognition across different document types and layouts.

1.3 OBJECTIVES OF THE THESIS

The Portuguese death certificates have many challenges such as irregular handwriting, nested structures, and diverse layouts. Therefore, the choice of modeling paradigm is crucial. Segmentation-based models, while established, risk cascading errors and struggle with the variability of historical records. Segmentation-free approaches, by contrast, integrate layout and text in a single process and are therefore better suited to documents where structure and content are tightly interwoven.

As a result, this thesis takes segmentation-free models as its starting point and investigates how well they can support historians in extracting structured information from historical death records. In particular, it evaluates the performance of DONUT [Kim+22] model, which has been designed for, among other purposes, end-to-end document information extraction. Otherwise, the thesis has some other secondary objects as:

1. Training Strategy Comparison: Comparing how well the model performs when trained on multiple tasks sequentially versus individual tasks.
2. Preprocessing Effects: How eliminating irrelevant parts can affect the performance on some tasks.
3. Resources for Future Research: Developing Donut-formatted datasets from which future work can benefit.

The main research questions that are going to be addressed are:

RQ1 How accurately can DONUT [Kim+22] extract key information (names, dates, or causes of death) from this specific dataset?

RQ2 Does the model work better for this task when trained on multiple tasks or a single task?

RQ3 Is pre-processing necessary?

1.4 THESIS STRUCTURE

In the next five chapters, the segmentation-free recognition of Portuguese death certificates is going to be systematically addressed. Chapter 2 presents the fundamental concepts of handwritten text recognition, neural networks, transformers, and visual document understanding, establishing the theoretical background. Chapter 3 reviews related work, focusing on some other models and their applications. Chapter 4 details the methodology, including the model architecture, training process, and output parsing strategies. Chapter 5 describes the experiments, covering the dataset, evaluation metrics, experimental setup, baseline, core experiments, and result analysis. Finally, Chapter 6 concludes with a summary, key findings, limitations, and suggestions for future work, providing a comprehensive closure to the study.

2 FUNDAMENTALS

This chapter explains the fundamentals needed to understand the method used in chapter 4. The theoretical foundations in this chapter are primarily based on Bishop and Bishop [BB23] and lecture materials from [Fin25], with additional sources cited where applicable.

2.1 HANDWRITTEN TEXT RECOGNITION

Handwritten Text Recognition (HTR) belongs to the broader field of pattern recognition, which deals with the automatic processing and analysis of patterns in data. In the context of HTR, the goal is to convert images containing handwritten text into machine-readable digital text, enabling automated processing of handwritten documents.

The fundamental challenge in HTR lies in the variability of handwriting styles, document quality, and layout complexity. Unlike printed text recognition, handwritten text exhibits significant variations between different writers and even within the same writer's text, making automated recognition a complex computational problem.

Formally, HTR can be defined as learning a mapping function:

$$f_\theta : \mathcal{I} \to \mathcal{S}^* \tag{2.1.1}$$

where $\mathcal{I} = \mathbb{R}^{H \times W \times C}$ represents the space of input images with height H, width W, and C channels, $\mathcal{S}$ denotes the set of possible characters, and $\mathcal{S}^*$ represents sequences of characters from this set.

For document understanding tasks, this basic formulation can be extended to include question-answering capabilities:

$$f_\theta : \mathcal{I} \times \mathcal{Q} \to \mathcal{S}^* \tag{2.1.2}$$

where $\mathcal{Q}$ represents the space of questions about the document content. A specific instantiation of this formulation is Information Extraction (IE), where the task is to extract structured information from documents. The question-answering formulation naturally accommodates IE tasks by treating field identifiers as queries, enabling the extraction of predefined attributes from unstructured documents.

The model is trained to minimize the empirical loss over a dataset $\mathcal{D} = \{(\mathbf{I}_i, \mathbf{s}_i)\}_{i=1}^N$ (or $\{(\mathbf{I}_i, \mathbf{q}_i, \mathbf{s}_i)\}_{i=1}^N$ for question-answering):

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \ell(f_\theta(\mathbf{I}_i), \mathbf{s}_i) \tag{2.1.3}$$

The function f_θ in Equation 2.1.1 can be realized through various computational approaches, with neural networks being particularly effective for learning complex mappings from visual data to textual outputs. The following sections examine the neural network architectures and components that enable such learning.

2.2 NEURAL NETWORKS

Neural networks have proven particularly effective for pattern recognition tasks in computer vision, making them well-suited for handwritten text recognition and document understanding. By learning hierarchical features from raw data such as pixel intensities in document images, they can identify complex patterns like varying handwriting styles without explicit feature engineering.

2.2.1 *Basic Neural Network Components*

Artificial Neurons

The fundamental building block of a neural network is the artificial neuron, which computes a weighted sum of its inputs and applies a nonlinear transformation. Formally, a neuron computes:

$$y = \sigma\left(\sum_{i=1}^{n} w_i x_i + b\right)$$

where x_i are inputs, w_i are weights representing the strength of each connection, b is the bias term that shifts the decision threshold, and σ is an activation function that introduces nonlinearity.

The weights w_i determine how strongly each input influences the neuron's output, while the bias b allows the neuron to activate even when all inputs are zero. Without the activation function σ, the neuron would perform only linear transformations, severely limiting the network's representational capacity.

Activation Functions

Activation functions introduce nonlinearity into neural networks, enabling them to model complex patterns and relationships. Common activation functions include:

ReLU (Rectified Linear Unit):

$$\sigma(z) = \max(0, z)$$

ReLU outputs zero for negative inputs and the input value for positive inputs. This function is computationally efficient and helps mitigate the vanishing gradient problem during training.

Sigmoid Function:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

The sigmoid function maps any real number to the range $(0, 1)$, making it useful for binary classification tasks where outputs can be interpreted as probabilities.

Softmax Function: For multi-class classification, the softmax function converts a vector of real numbers into a probability distribution:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

where K is the number of classes. The softmax ensures that all outputs sum to 1 and can be interpreted as class probabilities.

Multi-Layer Architecture

Neurons are organized into layers, forming a network where each layer transforms its input and passes the result to subsequent layers as illustrated in Figure 2.2.1. A feedforward neural network with L layers implements the function:

$$f_\theta(\mathbf{x}) = \sigma^{(L)}\left(\mathbf{W}^{(L)}\sigma^{(L-1)}\left(\cdots\sigma^{(1)}\left(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}\right)\cdots\right) + \mathbf{b}^{(L)}\right)$$

where:

- $\mathbf{x} \in \mathbb{R}^{n_0}$ is the input vector
- $\theta = \{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}\}_{l=1}^{L}$ are the learnable parameters
- $\mathbf{W}^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$ is the weight matrix of layer l

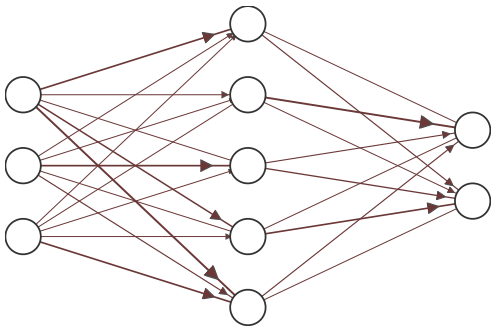


Figure 2.2.1: An example three-layer neural network

- $\mathbf{b}^{(l)} \in \mathbb{R}^{n_l}$ is the bias vector of layer l
- $\sigma^{(l)}$ is the activation function applied element-wise
- n_l is the number of neurons in layer l

Each layer extracts increasingly abstract patterns: early layers might detect edges and simple patterns, while deeper layers combine these into more complex representations such as character shapes or word structures.

2.2.2 *Input and Output Representations*

One-Hot Encoding

For classification tasks, categorical labels must be converted into numerical representations. One-hot encoding represents each class as a binary vector where exactly one element is 1 and all others are 0. For a vocabulary of size V, each character or token is represented as:

$$\mathbf{e}_i \in \{0,1\}^V \text{ where } \mathbf{e}_i[j] = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{otherwise} \end{cases}$$

This representation allows neural networks to process discrete categorical data while maintaining the property that no ordering relationship is implied between different categories.

Softmax for Probability Distributions

In multi-class classification problems such as character recognition, the network's final layer typically uses softmax activation to produce a probability distribution over possible classes. Given logits $\mathbf{z} \in \mathbb{R}^K$ from the final layer, softmax produces:

$$\mathbf{p} = \text{softmax}(\mathbf{z}) \text{ where } p_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

The resulting probability vector $\mathbf{p}$ satisfies $\sum_{i=1}^{K} p_i = 1$ and $p_i \geqslant 0$, making it suitable for representing confidence in class predictions.

2.2.3 *Training Procedure*

Training a neural network involves iteratively adjusting the network parameters to minimize prediction errors on a given dataset. This optimization process uses gradient-based methods to find parameter values that best approximate the desired input-output mapping.

Loss Functions and Gradient Descent

Given a training dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$, the training objective is to minimize the empirical risk:

$$\mathcal{J}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(f_\theta(\mathbf{x}_i), \mathbf{y}_i)$$

where $\mathcal{L}$ is a loss function that measures the discrepancy between predicted and target outputs. For example, Mean Squared Error is commonly used:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{2} \sum_{j=1}^{M} (d_j - f_j^{(L)})^2$$

where d_j is the desired output and $f_j^{(L)}$ is the actual network output for neuron j in the output layer.

To minimize this objective, parameters are updated iteratively using gradient descent:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{J}(\theta_t)$$

where $\eta > 0$ is the learning rate that controls the step size. The learning rate is typically initialized to a larger value (e.g., $\eta = 1.0$) and gradually decreased during training to achieve finer convergence to local optima.

Backpropagation Algorithm

Computing gradients $\nabla_\theta \mathcal{J}(\theta)$ efficiently requires the backpropagation algorithm, which applies the chain rule to propagate error signals backward through the network.

For a weight $w_{ij}^{(l)}$ connecting neuron i in layer $l-1$ to neuron j in layer l, the gradient is:

$$\frac{\partial \mathcal{L}}{\partial w_{ij}^{(l)}} = \frac{\partial \mathcal{L}}{\partial f_j^{(l)}} \cdot \frac{\partial f_j^{(l)}}{\partial y_j^{(l)}} \cdot \frac{\partial y_j^{(l)}}{\partial w_{ij}^{(l)}}$$

where $y_j^{(l)}$ is the pre-activation (weighted sum) of neuron j in layer l.

To compute these gradients efficiently, we introduce the local error $\delta_j^{(l)} = \frac{\partial \mathcal{L}}{\partial y_j^{(l)}}$, which simplifies the calculations by reusing intermediate results. For the output layer ($l = L$), this local error combines the loss gradient with the activation derivative:

$$\delta_j^{(L)} = \frac{\partial \mathcal{L}}{\partial f_j^{(L)}} \cdot \frac{\partial f_j^{(L)}}{\partial y_j^{(L)}}$$

When using sigmoid activation functions, this expression simplifies to:

$$\delta_j^{(L)} = -(d_j - f_j^{(L)}) \cdot f_j^{(L)} \cdot (1 - f_j^{(L)})$$

For hidden layers ($l < L$), the local error is computed by propagating errors backward from the subsequent layer:

$$\delta_j^{(l)} = \left(\sum_{k=1}^{M^{(l+1)}} w_{jk}^{(l+1)} \delta_k^{(l+1)} \right) \cdot f_j^{(l)} \cdot (1 - f_j^{(l)})$$

With these local errors computed, the weight update rule becomes:

$$w_{ij}^{(l)} \leftarrow w_{ij}^{(l)} - \eta \delta_j^{(l)} f_i^{(l-1)}$$

The complete backpropagation process operates in three phases. First, a forward pass computes network outputs layer by layer. Next, a backward pass computes the

local errors $\delta_j^{(l)}$ starting from the output layer and moving toward the input layers. Finally, these local errors are used to update the weights according to the rule above. This process repeats until convergence or a maximum number of iterations is reached.

The backpropagation algorithm enables efficient gradient computation by reusing intermediate calculations and avoiding redundant computations across the network layers.

2.3 TRANSFORMERS

The Transformer architecture [Vas+23] represents a fundamental shift in sequence processing with attention-based computations that enable parallel processing and improved modeling of long-range dependencies. Transformers have become the foundation for many state-of-the-art models in natural language processing and computer vision, including the document understanding model employed in this thesis.

2.3.1 *Input Representation*

Before examining the attention mechanisms that define transformer architectures, we must establish how raw data is converted into the vector representations that transformers operate on. This section details the three key components of transformer input representation: tokenization, embeddings, and positional encoding.

Tokenization

Tokenization converts raw text into discrete units that can be processed by neural networks. Let:

- Σ be a finite alphabet of characters
- $\mathcal{V}$ be a finite set of tokens, called the vocabulary
- $M : \Sigma^* \to 2^{\mathcal{V}^*}$ be a splitting procedure that maps a string to a set of possible token sequences

The tokenization function $T : \Sigma^* \to \mathcal{V}^*$ is defined as:

$$T(s) = \arg\max_{\mathbf{t} \in M(s)} F(\mathbf{t})$$

where $s \in \Sigma^*$ is the input string, $F : \mathcal{V}^* \to \mathbb{R}$ is a scoring function, and $\mathbf{t} = (t_1, t_2, \ldots, t_m)$ is the resulting sequence of token indices from $\mathcal{V}$.

The tokenization process addresses vocabulary size constraints by decomposing words into subword units. This approach handles rare words and out-of-vocabulary terms more effectively than word-level tokenization.

Embeddings

Token IDs are mapped to continuous vector representations via a learnable embedding matrix $\mathbf{E} \in \mathbb{R}^{D \times V}$, where D is the embedding dimension and V is the vocabulary size:

$$\mathbf{v}_n = \mathbf{E}\mathbf{x}_n$$

where $\mathbf{x}_n$ is the one-hot encoded token ID and $\mathbf{v}_n \in \mathbb{R}^D$ is the resulting vector representation.

The embedding matrix $\mathbf{E}$ is learned during training, allowing the model to discover semantic relationships between tokens. Tokens with similar meanings or functions tend to be closer in the vector space, enabling the model to generalize across related concepts.

Positional Embeddings

Transformer architectures process all input positions simultaneously, lacking the inherent positional awareness of recurrent models. To incorporate sequence order information, positional encodings are added to token embeddings.

Sinusoidal positional encodings $\mathbf{r} \in \mathbb{R}^{N \times D}$ are computed as:

$$r_{i,2j} = \sin\left(\frac{i}{U^{2j/D}}\right), \quad r_{i,2j+1} = \cos\left(\frac{i}{U^{2j+1/D}}\right)$$

where U is a hyperparameter (typically 10,000), $i \in \{1, \ldots, N\}$ indexes the sequence position, and $j \in \{0, \ldots, D/2 - 1\}$ indexes the embedding dimension.

The positional encoding alternates between sine and cosine functions across embedding dimensions. Even dimensions ($2j$) use sine functions with different frequencies, while odd dimensions ($2j + 1$) use cosine functions with corresponding frequencies. Each dimension pair $(2j, 2j + 1)$ represents a different frequency, with lower-indexed dimensions encoding faster-changing patterns and higher-indexed dimensions encoding slower-changing patterns. This creates a unique positional signature for each sequence position.

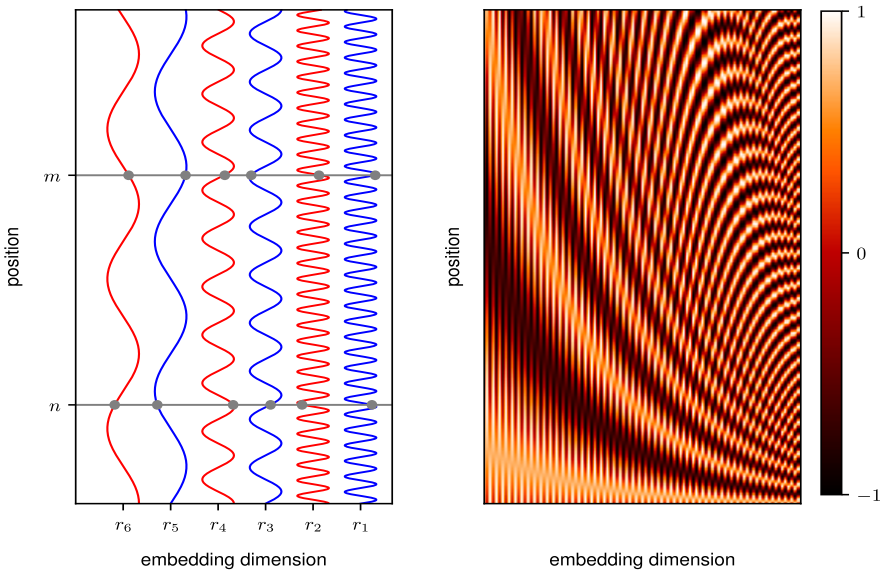


Figure 2.3.1: Sinusoidal positional encoding visualization. Left figure: Individual encoding functions for different embedding dimensions. Right figure: Heat map showing positional encoding values across positions and dimensions.

This encoding scheme has several important properties. It is deterministic, requiring no learnable parameters. All values remain bounded in $[-1, 1]$, and the model can learn to attend based on relative distances between positions. Additionally, the sinusoidal structure allows the model to handle sequences longer than those seen during training.

The final input representation combines token embeddings with positional encodings:

$$\mathbf{X} = [\mathbf{v}_1 + \mathbf{r}_1, \mathbf{v}_2 + \mathbf{r}_2, \ldots, \mathbf{v}_N + \mathbf{r}_N]$$

2.3.2 *Attention Mechanisms*

The core innovation of transformer architectures lies in their attention mechanisms, which enable models to dynamically focus on relevant parts of the input sequence when processing each element. This section examines the mathematical formulation and computational properties of self-attention.

Scaled Dot-Product Self-Attention

Self-attention mechanisms compute representations by relating different positions within a single sequence. Given a data matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$ representing N input vectors with D-dimensional features, the self-attention mechanism computes an output matrix $\mathbf{Y} \in \mathbb{R}^{N \times D_v}$.

The mechanism is parameterized by three learnable weight matrices:

- Query weights: $\mathbf{W}^q \in \mathbb{R}^{D \times D_k}$
- Key weights: $\mathbf{W}^k \in \mathbb{R}^{D \times D_k}$
- Value weights: $\mathbf{W}^v \in \mathbb{R}^{D \times D_v}$

where D_k and D_v are the dimensions of the key/query and value representations, respectively.

The attention computation proceeds as follows:

1. Linear Projections:

   $$\mathbf{Q} = \mathbf{X}\mathbf{W}^q, \quad \mathbf{K} = \mathbf{X}\mathbf{W}^k, \quad \mathbf{V} = \mathbf{X}\mathbf{W}^v$$

   where $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{N \times D_k}$ and $\mathbf{V} \in \mathbb{R}^{N \times D_v}$.

2. Attention Score Computation:

   $$\mathbf{A} = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{D_k}}\right)$$

   where $\mathbf{A} \in \mathbb{R}^{N \times N}$ contains attention weights. The dot product $\mathbf{Q}\mathbf{K}^\top$ measures similarity between queries and keys, while the scaling factor $\sqrt{D_k}$ prevents the dot products from growing too large and pushing the softmax function into saturation regions.

3. Weighted Value Aggregation:

   $$\mathbf{Y} = \mathbf{A}\mathbf{V}$$

   where $\mathbf{Y} \in \mathbb{R}^{N \times D_v}$ represents the attention-weighted combination of value vectors.

The complete self-attention (illustrated in Figure 2.3.2) operation can be expressed as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{D_k}}\right)\mathbf{V}$$

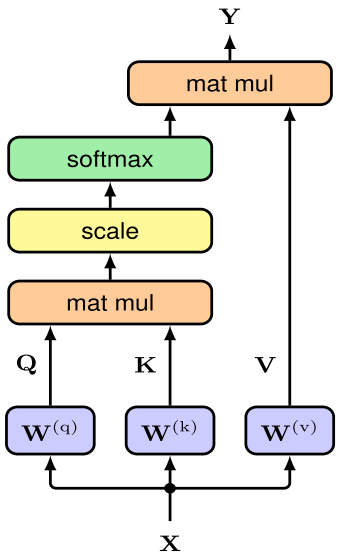


Figure 2.3.2: Scaled Dot-Product Attention mechanism. Input $\mathbf{X}$ is projected into query ($\mathbf{Q}$), key ($\mathbf{K}$), and value ($\mathbf{V}$) matrices. Attention is computed as $\text{softmax}(\mathbf{QK}^T/\sqrt{d_k})$ applied to $\mathbf{V}$, producing output $\mathbf{Y}$ [BB23].

Multi-Head Self-Attention

Multi-head attention extends the single attention mechanism by computing multiple attention functions in parallel, each focusing on different aspects of the input relationships.

Given input $\mathbf{X} \in \mathbb{R}^{N \times D}$, the multi-head self-attention mechanism employs H parallel attention heads, each with its own parameter matrices $\mathbf{W}_h^q, \mathbf{W}_h^k \in \mathbb{R}^{D \times D_k}$ and $\mathbf{W}_h^v \in \mathbb{R}^{D \times D_v}$.

For each head $h \in \{1, \ldots, H\}$:

$$\text{head}_h = \text{Attention}(\mathbf{XW}_h^q, \mathbf{XW}_h^k, \mathbf{XW}_h^v)$$

The outputs from all heads are concatenated and linearly transformed:

$$\mathbf{Y} = \text{Concat}(\text{head}_1, \ldots, \text{head}_H)\mathbf{W}^o \tag{2.3.1}$$

where $\mathbf{W}^o \in \mathbb{R}^{HD_v \times D}$ is a learned output projection matrix.

This architecture allows all heads to compute simultaneously, enabling parallel processing. Each head can learn different attention patterns, allowing the model to capture diverse relationships in the data. The presence of multiple attention paths also helps stabilize gradients during training, reducing optimization difficulties.

2.3.3 *Transformer layer*

The transformer layer combines multi-head attention with feed-forward processing through residual connections and layer normalization. A transformer layer outputs a matrix $\mathbf{Y}$:

$$\mathbf{Y} = \text{LayerNorm}(\mathbf{Z} + \text{FFNN}(\mathbf{Z})) \tag{2.3.2}$$

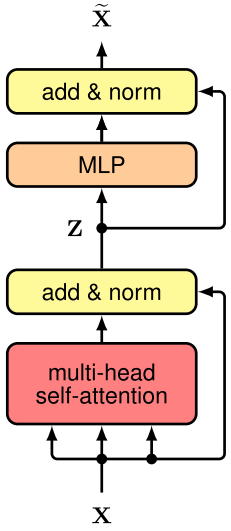


Figure 2.3.3: Single transformer layer architecture. Input $\mathbf{X}$ passes through multi-head self-attention, followed by a residual connection and layer normalization to produce $\mathbf{Z}$. This is then processed by a feed-forward network (MLP), with another residual connection and layer normalization producing the final output $\tilde{\mathbf{X}}$ [BB23].

as illustrated in Figure 2.3.3 where:

- $\mathbf{X} \in \mathbb{R}^{N \times D}$ is the input matrix
- $\mathbf{Z} = \text{LayerNorm}(\mathbf{X} + \text{MultiHead}(\mathbf{X}))$
- $\mathbf{Y} \in \mathbb{R}^{N \times D}$ is the final layer output
- LayerNorm normalizes activations along the feature dimension, stabilizing training
- FFNN is a feed-forward network applied independently to each position

This design enables parallel processing while modeling complex sequence relationships.

2.3.4 *Encoder-Decoder Model*

One variation is the encoder-decoder transformer, which excels in sequence-to-sequence tasks such as text translation. This architecture consists of two components that work together: an encoder that processes the input sequence and a decoder that generates the output sequence.

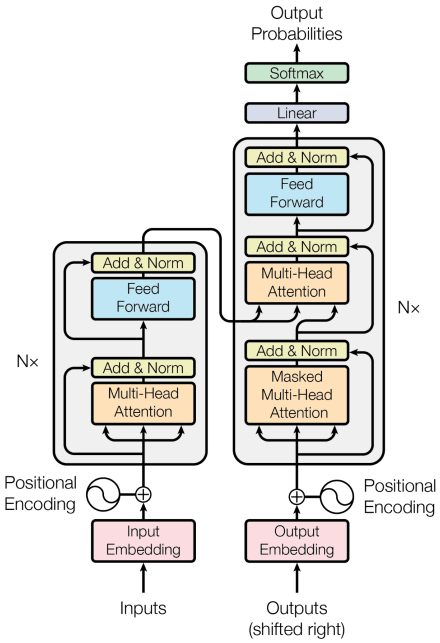


Figure 2.3.4: The Transformer model architecture. The encoder (left) processes input sequences through N stacked layers of multi-head self-attention and feed-forward networks. The decoder (right) generates outputs autoregressively using masked self-attention, cross-attention to encoder outputs, and feed-forward networks. A final linear layer and softmax produce output probabilities over the vocabulary. Positional encodings are added to input embeddings in both stacks [Vas+23].

The encoder processes the input sequence $\mathbf{X} \in \mathbb{R}^{N \times D}$ through L stacked transformer layers, producing a context-aware representation $\mathbf{Z} \in \mathbb{R}^{N \times D}$. This representation captures the complete input context and serves as the foundation for output generation.

The decoder then generates the output sequence using the encoder representation $\mathbf{Z}$ and previously generated tokens. Each decoder layer contains three main components. First, masked self-attention prevents positions from attending to future positions, ensuring that predictions depend only on previously generated tokens and maintaining the autoregressive property necessary for sequential generation. Second, cross-attention allows the decoder to access the encoder's output, with queries coming from the decoder while keys and values come from the encoder representation $\mathbf{Z}$. Finally, a feed-forward network is applied as in standard transformer layers.

This architecture (Figure 2.3.4) enables parallel processing during training while maintaining autoregressive generation, addressing fundamental limitations of sequential models. The original transformer [Vas+23] demonstrated that attention mechanisms alone could achieve state-of-the-art results while significantly reducing training time through parallelization.

2.3.5 *Vision Transformer*

Dosovitskiy et al. [Dos+21] demonstrated that transformer architectures, originally designed for sequential text data, can be applied to image classification with minimal architectural modifications by treating images as sequences of patches.

While images already possess a mathematical representation $\mathbf{I} \in \mathbb{R}^{H \times W \times C}$, directly feeding pixel values to transformers would be computationally prohibitive due to the quadratic complexity of self-attention with respect to sequence length.

To address this, the image is divided into $N = \frac{HW}{P^2}$ non-overlapping patches of size $P \times P$. Each patch is flattened into a vector of length P^2C and linearly projected to dimension D, forming the sequence $\mathbf{X} \in \mathbb{R}^{N \times D}$. A learnable classification token [CLS] is prepended to the sequence to aggregate information for the final classification decision.

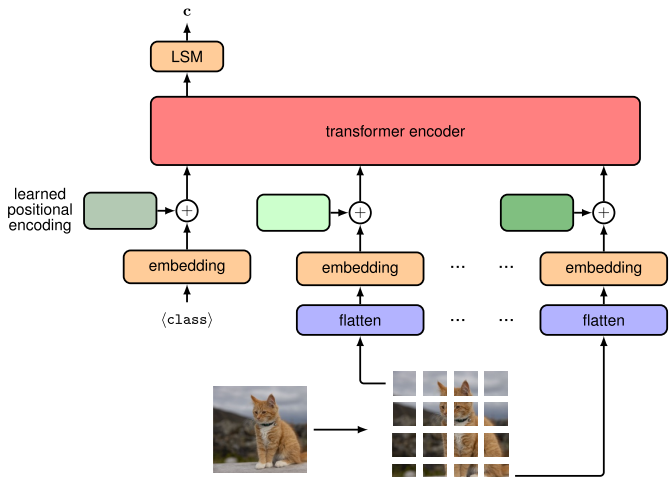


Figure 2.3.5: Vision Transformer architecture for image classification. The input image is divided into non-overlapping patches, which are flattened and linearly embedded. Learned positional encodings are added to preserve spatial information, and a learnable [CLS] token is prepended to the sequence. The resulting patch embeddings are processed by a transformer encoder, with the [CLS] token's output used for final classification [BB23].

This approach (Figure 2.3.5) enables transformers to process images using the same attention mechanisms developed for text, requiring only the addition of patch extraction and positional embeddings specific to spatial data.

3

RELATED WORK

The advent of deep learning has revolutionized handwritten text recognition (HTR), enabling the processing of complex visual data like historical documents during training. Early deep learning approaches relied on recurrent neural networks (RNNs) [GS08] and convolutional-recurrent hybrid [SBY15] architectures to handle the sequential and spatial nature of handwritten text. However, the introduction of transformer architectures [Vas+23] fundamentally changed sequence modeling by enabling parallel processing and superior long-range dependency capture through attention mechanisms.

This chapter examines transformer-based models that have advanced HTR and document understanding, tracing the evolution from encoder-decoder architectures to multimodal vision-language models. Each approach addresses specific limitations of its predecessors while introducing new capabilities relevant to processing historical documents like Portuguese death certificates.

3.1 TrOCR: Transformer-based Optical Character Recognition

Transformer-based Optical Character Recognition [Li+22] advanced HTR by extending the separation of visual understanding and sequence generation first seen in CRNN. This approach leverages transformers to overcome RNN limitations, offering a robust solution for handwritten and printed text recognition.

Architecture: Vision Encoder and Language Decoder

TrOCR builds on CRNN's principle of distinct visual and sequential tasks, replacing CNNs with a vision transformer (e.g., BEiT [Bao+22]) as the encoder to extract features from raw images, and a language transformer (e.g., RoBERTa [Liu+19]) as the decoder to generate text sequences. Both encoder and decoder employ transformer architectures with self-attention mechanisms. The encoder processes the entire image in parallel through patch-based attention, capturing spatial hierarchies, while the decoder uses attention to predict character sequences end-to-end in an autoregressive manner. Pre-training on synthetic data enhances its adaptability, making it segmentation-free and efficient for variable-length text. The pipeline is illustrated in Figure 3.1.1.

A key contribution of TrOCR is demonstrating that standard, pre-trained vision transformers can be directly applied to HTR without domain-specific architectural adaptations. Unlike prior approaches that relied on specialized CNN backbones or task-specific inductive biases, TrOCR uses vanilla ViT encoders (e.g., BEiT) with minimal modifications of simply treating text images as sequences of patches. This architectural simplicity proves that general-purpose transformer models, when properly pre-trained, can achieve state-of-the-art results on HTR tasks without requiring custom designs for the handwriting domain. This transferability represents a significant shift from domain-engineered architectures to more generic, scalable solutions.

Performance Impact

TrOCR demonstrated significant improvements on benchmarks like IAM and RIMES [Gro+09], outperforming traditional OCR systems with its transformer-based approach. Its ability to handle diverse scripts and noisy historical texts marks a leap forward, establishing transformers as a new standard in HTR.

Relevance and Challenges

TrOCR's relevance lies in its segmentation-free processing of raw images, enabling extraction of text fields like names or dates. However, its focus on word-level outputs limits layout understanding, and reliance on pre-trained models requires fine-tuning for domain-specific documents. These constraints suggest a need for models which integrate full-page context and structured representations.

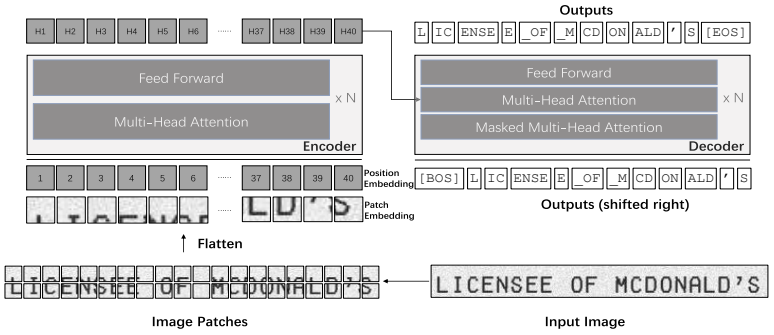


Figure 3.1.1: The architecture of TrOCR [Li+22].

3.2 DAN: DOCUMENT ATTENTION NETWORK - LAYOUT UNDERSTANDING EVOLUTION

Document Attention Network (DAN) [CCP23] extends transformer-based HTR from word-level recognition to full-page documents, building on the need for layout context beyond TrOCR's capabilities.

Architecture and Full-page Recognition Approach

DAN advances the separation of visual and sequential processing with a Fully Convolutional Network (FCN) encoder and transformer decoder, tailored for complex document layouts. In contrast to vision transformers that compute global attention across image patches, DAN employs an FCN encoder to better model local spatial dependencies inherent in document images. The FCN processes entire pages, generating dense feature maps that preserve spatial relationships and capture positional links between text elements. The transformer decoder uses attention to focus dynamically across the spatial feature map, enabling recognition of irregular layouts without pre-segmentation; a significant step from line-based approaches. The architecture is illustrated in Figure 3.2.1.

XML-like Structured Output

DAN generates XML-like output that captures both text and layout for structured representation. For example, it can produce:

```
<text_line>Patient Name: João Silva</text_line>
<text_line>Date of Death: March 15, 1869</text_line>
```

A more comprehensive example can be seen in Figure 3.2.2.

Performance on Complex Document Layouts

DAN excels on RIMES [Gro+09] and READ [Tos+18] 2016 datasets, handling full-page and double-page documents with irregular layouts and multiple columns. Its end-to-end training eliminates error propagation, showcasing the flexibility of attention-based processing for variable structures.

Limitations and Path to Vision-Language Models

Despite its layout focus, DAN targets handwritten text recognition, requiring task-specific training and lacking semantic understanding across document elements.

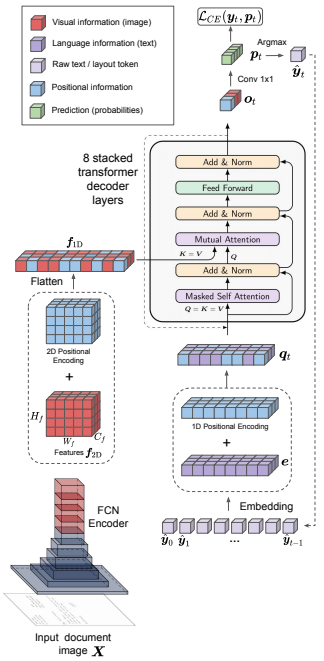


Figure 3.2.1: The DAN architecture is made up of an FCN encoder, for the extraction of 2D features $\mathbf{f}_{2D}$, and a transformer-based decoder for the recurrent prediction of the character/layout tokens $\hat{\mathbf{y}}_t$. At each iteration t, the model computes the representation $\mathbf{o}_t$ of the current character/layout token to recognize $\hat{\mathbf{y}}_t$, based on the flattened features $\mathbf{f}_{1D}$ and on the previous predictions. Positional encoding is added to these two modalities to preserve the spatial information through the transformer's attention mechanism [CCP23].

This suggests a need for vision-language models that integrate spatial and semantic capabilities.

3.3 LAYOUTLMV2: MULTI-MODAL PRE-TRAINING FOR VISUALLY-RICH DOCUMENT UNDERSTANDING

While DAN demonstrated the effectiveness of full-page processing for handwritten text recognition through OCR-free approaches, a parallel line of research explored multimodal document understanding by combining visual, textual, and layout information.

LayoutLMv2 [Xu+20] represents this alternative direction, focusing on information extraction from visually rich documents through the integration of multiple modalities.

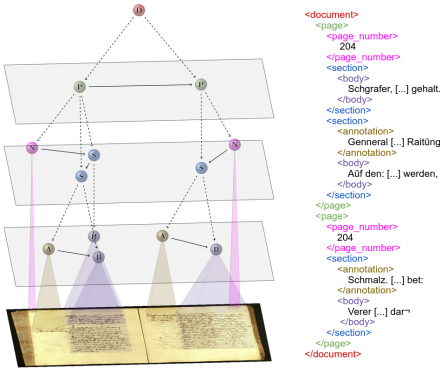


Figure 3.2.2: An example of structured output from DAN [CCP23].

Architecture: Multimodal Transformer Encoder

LayoutLMv2 uses a single transformer encoder that combines three modalities: image patches (extracted using a CNN or vision transformer), text tokens (from OCR), and 2D layout positions (bounding boxes). These inputs are aligned by the encoder using self-attention, and positional embeddings are used to capture spatial relationships, as illustrated in Figure 3.3.1.

LayoutLMv2's unified encoder, in contrast to DAN's FCN-Transformer split for recognition, has been pre-trained on sizable datasets like IIT-CDIP [Lew+06] with goals like text-image matching and masked visual-language modeling (predicting masked text/image regions).

Challenges and Limitations

Despite its improvements, LayoutLMv2 still uses bounding boxes and text generated by OCR as inputs, which could lead to errors in old, deteriorated documents where OCR isn't working. Its robustness for raw, unprocessed scans is limited by this dependency and the requirement for substantial pre-training resources. Additionally, compared to decoder-based models, its encoder-only design is less optimized for generative outputs. These drawbacks emphasize the necessity of completely OCR-free, end-to-end methods like DONUT [Kim+22], which improve upon multimodal transformers but do away with preprocessing for domain-specific fine-tuning in tasks like certificate extraction.

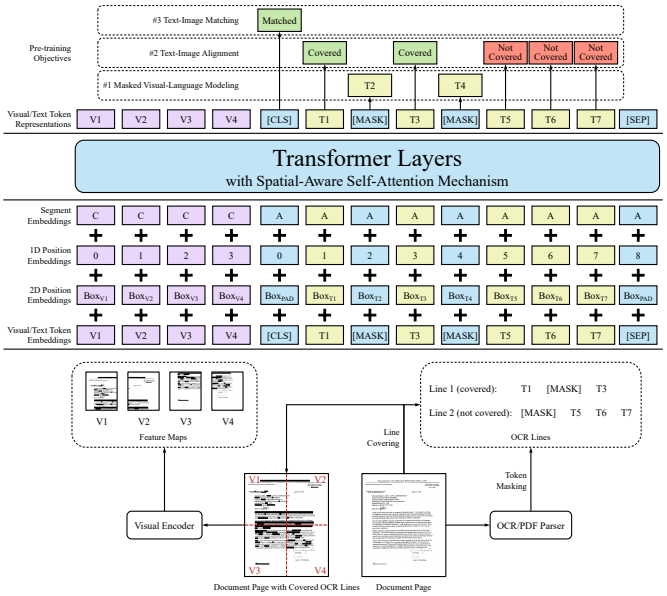


Figure 3.3.1: An illustration of the model architecture and pre-training strategies for LayoutLMv2 [Xu+20].

3.4 DESSURT: DOCUMENT END TO END SELF SUPERVISED UNDERSTANDING AND RECOGNITION TRANSFORMER

Document End to End Self Supervised Understanding and Recognition Transformer (DESSURT) [Dav+22] represents a significant departure from the two stage approaches of the LayoutLM family by eliminating the dependency on external OCR models entirely. This unified architecture performs both text recognition and document understanding in a single forward pass, addressing fundamental limitations of encoder only transformer approaches.

Architecture: Unified Recognition and Understanding

DESSURT employs a novel three stream architecture processing visual tokens, query tokens, and autoregressive response tokens simultaneously. Unlike LayoutLMv2's encoder-only design that requires precomputed OCR tokens, DESSURT's visual encoder (a modified CNN followed by Swin [Liu+21] transformer layers) directly processes document images while implicitly learning text recognition. The query stream encodes task specifications using standard transformer attention, while the response

stream generates arbitrary text output autoregressively through cross-attention to both visual and query features. The architecture is illustrated in Figure 3.4.1.

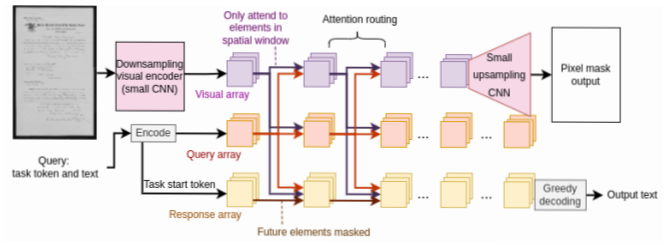


Figure 3.4.1: The DESSURT architecture with three token streams: visual tokens encoding document images, query tokens specifying tasks, and response tokens generating arbitrary text output [Dav+22].

Flexible Task Adaptation and Output Generation

A key innovation of DESSURT is its ability to generate arbitrary text output rather than being constrained to token level predictions like the LayoutLM family. By accepting task specific query strings, the model adapts to diverse document understanding tasks without architectural modifications, from question answering and form parsing to full page handwriting recognition and named entity recognition. This flexibility is demonstrated across different dataset task combinations including DocVQA [MKJ20], FUNSD [JET19], NAF [Dav+19], and IAM [MB02] datasets, with the model achieving competitive performance despite being a single unified architecture.

End to End Training Strategy

DESSURT's pretraining employs a multi-dataset curriculum including IIT CDIP documents [Lew+06], synthetic Wikipedia text, synthetic handwriting, and synthetic forms with diverse tasks like text infilling, reading, and parsing. This comprehensive pretraining enables the model to learn both recognition and understanding jointly. The elimination of external OCR dependencies is particularly valuable for challenging domains like historical handwritten documents where OCR quality is poor. The model achieves 4.8% CER on IAM full page recognition and successfully processes the historical NAF dataset containing mixed printed and handwritten text.

Limitations and Trade offs

While DESSURT's unified architecture offers flexibility and eliminates OCR dependencies, it comes with computational costs. The model has 127M parameters and requires processing full document images at 1152×768 resolution. Performance on some tasks lags behind specialized two stage approaches; for instance, achieving 63.2 ANLS on DocVQA compared to LayoutLMv2's 78.1. These limitations suggest that while end to end approaches offer architectural elegance and flexibility, task specific optimization through specialized pipelines may still yield superior performance for certain applications.

3.5 FULL-PAGE PROCESSING AT SCALE: THE SOCFACE PROJECT

The Socface project [Boi+24] demonstrates the practical viability of full-page OCR-free approaches for historical administrative records at unprecedented scale, providing empirical validation for the document understanding paradigm discussed in previous sections.

Project Scope and Challenges

The Socface project processes handwritten French census lists from 1836 to 1936, encompassing approximately 30 million images distributed across 100 departmental archives throughout France. These census documents present challenges characteristic of historical administrative records: 19th-century handwriting variability, evolving document templates across decades, physical degradation, and complex tabular layouts organizing individuals into household units.

The scale and diversity of the documents, combined with variations in table templates across census years (columns changed order and content, preservation quality varied substantially), made traditional multi-stage approaches impractical. Managing separate models for different document layouts or time periods would require maintaining dozens of specialized processing chains, motivating the adoption of a unified full-page recognition approach.

Full-Page Recognition Implementation

The project applied the DAN full-page architecture to French census documents, adapting its XML-like output format to capture both individual information and household structure simultaneously. Rather than developing a new model, the work

demonstrates how existing full-page architectures can be fine-tuned for domain-specific structured extraction tasks through careful ground truth formatting.

The adaptation employed specialized tokens to categorize extracted information: each piece of data is preceded by a token indicating its semantic type (surname, first-name, occupation, age, etc.). Critically, the labeling scheme distinguishes household heads with a unique token, enabling automatic reconstruction of household units from the sequential output. This encoding allows a single forward pass through the page to extract all individual attributes while simultaneously inferring household groupings without requiring separate segmentation or classification stages.

This approach leverages the key advantage of full-page processing: the model can attend to contextual markers (ditto marks, brackets spanning multiple rows, positional cues) that would be lost in segmentation-based pipelines where documents are split before recognition. The decoder learns to interpret both textual content and structural organization simultaneously.

Results and Scale Achievement

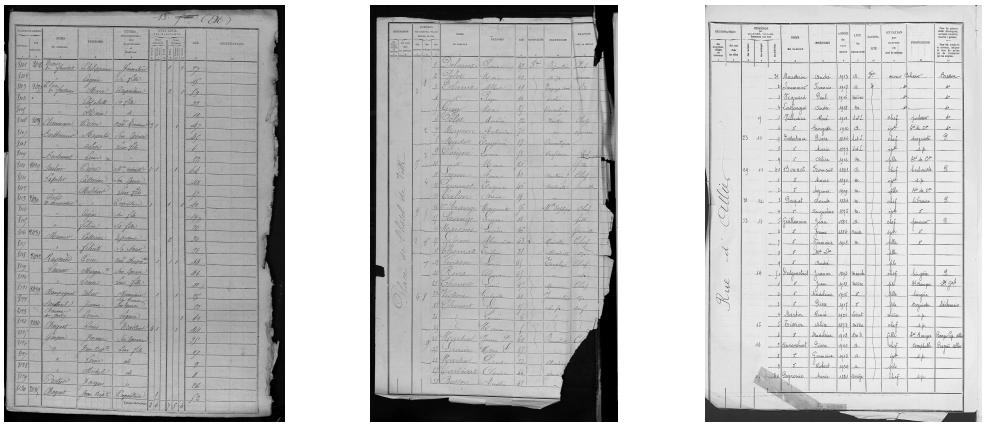
The model was trained on only 100 manually annotated census pages, achieving F1 scores ranging from 70% to 98% across different information categories (names, ages, occupations, household positions). This data efficiency demonstrates that full-page models can transfer effectively to new historical document domains with limited supervision.

The processing pipeline successfully handled 450,000 images in under 8 days using distributed computing resources, demonstrating both computational efficiency and robustness across the considerable diversity of document formats and handwriting styles present in the archival collection. The single model processed all template variants and time periods without requiring layout-specific configuration or retraining.

Implications for Full-Page Approaches

The Socface project provides empirical validation that full-page OCR-free recognition can handle real-world archival variability at production scale. The successful processing of documents spanning a century, with substantial evolution in table formats (Figure 3.5.1), confirms that end-to-end architectures adapt to structural variations through training rather than requiring manual template engineering for each document type.

The ability to simultaneously recognize text and reconstruct hierarchical relationships within a single processing stage eliminates error propagation from multi-stage



1836 1886 1936

Figure 3.5.1: Evolution of French census table formats from 1836 to 1936, showing variations in column organization, information categories, and document quality across different years (adapted from [Boi+24]). A single full-page model successfully processed all template variants without layout-specific modifications.

pipelines where segmentation failures corrupt subsequent extraction steps. The Socface deployment demonstrates that this integrated approach scales to millions of documents while maintaining acceptable accuracy, supporting the viability of OCR-free full-page models for historical document understanding tasks requiring structured information extraction from tabular layouts.

4

METHODOLOGY

From word-level outputs in TrOCR to layout-focused recognition in DAN and multimodal querying in LayoutLMv2, the development of transformer-based models, as described in the related work, has improved document understanding and handwritten text recognition (HTR). However, these methods frequently rely on optical character recognition (OCR) preprocessing, which restricts their applicability to historical documents.

This study leverages the Document Understanding Transformer Donut [Kim+22], an OCR-free model that takes end-to-end processing to the next level, acting as a unified "big black box" that transforms raw document images directly into structured information outputs. This section offers a thorough overview of its architecture, training paradigm, and evaluation framework, setting the stage for its use in the experiments that follow.

4.1 MODEL ARCHITECTURE

Donut's architecture follows a sequence-to-sequence framework, consisting of a vision encoder for visual feature extraction and a language decoder for text generation, that also receives a prompt. This design enables the model to process raw images and generate structured outputs without OCR, representing a significant advancement in document understanding.

4.1.1 *Encoder*

The encoder in Donut is based on the Swin Transformer [Liu+21], a hierarchical vision transformer that processes raw document images into visual tokens. Standard Vision Transformers (ViT) compute global self-attention across all image patches, resulting in quadratic computational complexity with respect to image resolution, which is a significant limitation for high-resolution documents. Swin Transformer addresses this by introducing a hierarchical representation and a shifted window self-attention mechanism, which reduces computational complexity to linear with image size while maintaining strong modeling power.

The Swin Transformer begins by dividing the input image into non-overlapping patches, treating them as tokens. After that, it employs a multi-step procedure that, like the hierarchical feature extraction in CNNs, combines patches at each stage to produce lower-resolution feature maps, enabling it to handle scale variations in documents, such as varying font sizes or layout variations.

The key innovation is the shifted window self-attention, which computes attention locally within non-overlapping windows to achieve efficiency. In each layer, attention is computed only within these fixed, non-overlapping window boundaries. However, to enable communication between different spatial regions, consecutive layers use different window partitioning schemes: the windows are shifted spatially, creating new groupings of patches as illustrated in Figure 4.1.1. This shifting strategy allows information to flow across the entire image without requiring global attention computation, as patches that were in different windows in one layer can be grouped together in the next layer. The approach maintains linear complexity while ensuring cross-window connectivity, making it suitable for high-resolution documents.

4.1.2 *Decoder*

The decoder in Donut is a BART [Lew+19] autoregressive transformer that generates text sequences based on the encoder's visual tokens and a prompt. The decoder uses a left-to-right generation process, where each token is predicted conditioned on previous tokens and the visual input. This setup allows Donut to produce structured outputs as XML-like tagged sequences, which are subsequently converted to JSON format for practical use.

The BART decoder employs multi-head self-attention and cross-attention mechanisms to integrate the encoder's features with the generated sequence. During generation, it uses teacher-forcing in training, where the ground-truth tokens are fed as input.

4.2 TRAINING PROCESS

Donut's training is divided into two steps:

1. Pre-training to be able to read, or better said: recognize characters.
2. Task-specific adaption as the model is capable of different tasks.

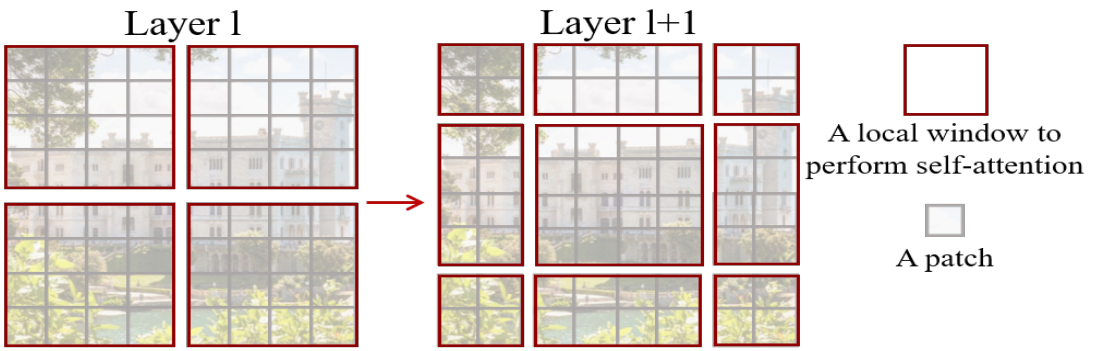


Figure 4.1.1: An illustration of the *shifted window* approach for computing self-attention in the proposed Swin Transformer [Liu+21] architecture. In layer l (left), a regular window partitioning scheme is adopted, and self-attention is computed within each window. In the next layer $l+1$ (right), the window partitioning is shifted, resulting in new windows. The self-attention computation in the new windows crosses the boundaries of the previous windows in layer l, providing connections among them.

4.2.1 *Pre-Training For OCR*

In this phase, the goal is for the model, given some pixels, to predict the next character(s). The objective of this phase is to minimize the cross-entropy loss function. One could say that the model in this phase is trained to associate the shape of certain words with their corresponding characters.

To account for the various differences in people's handwritten text, the model is trained on the IIT-CDIP dataset [Lew+06]. While this dataset is already large, it lacks non-Latin characters such as Chinese or Japanese.

Therefore, 2 million synthetic data samples were introduced using the SynthDoG method [Kim+22]. These samples include text in Japanese, Korean, English, and Chinese, 0.5 million each.

4.2.2 *Task-specific Adaption*

After the model has learned character recognition through pre-training, it is then trained to extract structured information from documents. In contrast to the pre-training phase which uses unlabeled or synthetically generated documents, this step requires task-specific labeled data where each document is annotated with the target

information to be extracted. However, it does not require as much data as the pre-training phase.

As Donut is capable of performing multiple tasks, the data is labeled according to the specific task. Task-specific prompts are fed to the decoder to help the model understand what should be generated:

- `<docvqa>` for visual question answering followed by the target question
- `<parsing>` for parsing the whole document
- `<class>` for classification of a document.

Different keywords could have been used.

4.3 ADAPTATION TO PORTUGUESE DEATH CERTIFICATES

Applying Donut to the Portuguese death certificates required several domain-specific adaptations to bridge the gap between the model's original training format and the structure of the available annotations.

4.3.1 *Data Format Conversion*

The original annotations were provided as an Excel spreadsheet mapping image filenames to extracted field values. To make this data compatible with Donut's training pipeline, the tabular annotations were converted into the DocVQA question-answering format expected by the model.

For each image, the annotation was transformed into a JSON structure:

```
{
 "ground_truth": "{"gt_parses": [
  {"question": "What is $(value_to_extract)",
   "answer": "$(answer)"}
  ]}"
}
```

This structure follows Donut's question-answering paradigm, where each field extraction task is framed as a question about the document content.

4.3.2 *Task-Specific Prompt Design*

Following Donut's task-specific adaptation approach (Subsection 4.2.2), the model receives a prompt combining the task token with the specific question:

```
<s_docvqa><s_question>What is paroq_cert</s_question><s_answer>
```

The decoder then generates the answer token sequence, terminated by the closing tags:

```
Campanhã</s_answer></s_docvqa>
```

Field-specific questions were formulated for each of the seven core information fields that will be described in Section 5.1.

4.4 DATA FLOW THROUGH THE MODEL

This section outlines the data flow through Donut, detailing how inputs are formatted and processed into structured outputs.

4.4.1 *Input Format for the model*

The data flow begins, of course, with preparation of raw document images, where they are padded to preserve aspect ratios. Then a task-specific string is fed to the decoder after tokenization as discussed in 4.2.2, along with the encoded visual features from the encoder.

4.4.2 *Output Format*

Based on the input, the decoder generates a sequence of tokens formatted as a tag-based structure using XML-like opening and closing tags. This approach enables the model to handle nested layouts effectively. Following the Transformer architecture, the generated output serves as a continuation of the input. Therefore, when the input includes a token like `<parsing>`, the output is expected to end with a corresponding closing tag, such as `</parsing>`.

4.4.3 *JSON-Conversion*

After the sequence is generated, it is then post-processed into JSON format, making it suitable for practical applications. Since the output is already in an `XML`-like format, it can be easily converted to JSON using regular expressions.

Missing opening or closing tags are simply ignored and treated as unrecognized. For example, the sequence `<name>someName</name><age>55` would produce the following JSON output: `{"name": "someName"}`.

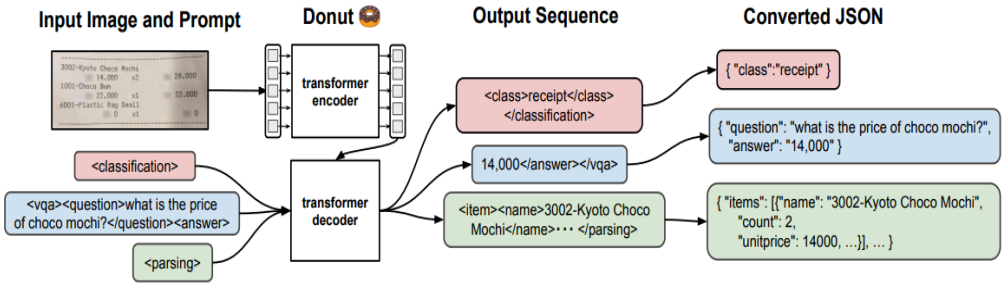


Figure 4.4.1: The pipeline of Donut [Kim+22]. The encoder maps a given document image into embeddings. With the encoded embeddings, the decoder generates a sequence of tokens that can be converted into a target type of information in a structured form

5 EXPERIMENTS

This chapter presents a comprehensive experimental evaluation of the Donut model for automated information extraction from historical Portuguese death certificates. The primary objective is to assess the model's capability to extract core information fields from handwritten 19th-century documents under various training configurations and preprocessing strategies.

Before conducting the main experiments, a baseline evaluation assesses the pre-trained model's zero-shot performance without any fine-tuning. Three experiments then investigate different training approaches. Single-field extraction evaluates performance when fine-tuning independently on each core field. Sequential training explores whether training on related fields in sequence improves extraction through transfer learning. Cropping experiments investigate whether focusing on relevant document regions improves extraction quality when images are downsampled to the model's fixed input resolution.

5.1 DATASET

The dataset consists of Portuguese death certificates from the Porto municipality, covering the period from January 1869 to January 1870. These historical documents are part of the Porto District Archive (Arquivo Distrital do Porto) collection, specifically from the Porto Civil Government fond, under the section of Assistance and Public Health.

It consists of 2,123 death certificate images from the Porto municipality covering January 1869 to January 1870. Of these, 1,635 certificates (77%) have been transcribed and are used in this work. The remaining certificates were excluded due to illegibility caused by physical degradation such as tears, water damage, or severe fading.

The death certificates follow printed form templates with handwritten entries. Two main form layouts exist:

- **Form 1**: Smaller horizontal format as in Figure 5.1.1.
- **Form 2**: Larger vertical format with two variants (2a and 2b) as illustrated in figure 5.1.2.

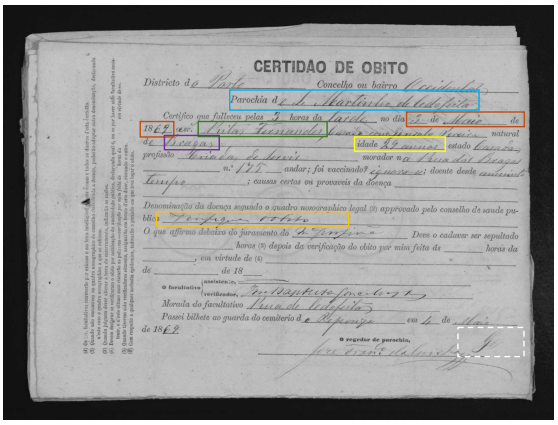


Figure 5.1.1: Example horizontal death certificate with colored bounding boxes marking core fields: parish name (light blue), date of death (dark orange), name (dark green), birthplace (purple), age (yellow), cause of death (orange), and poverty status (white).

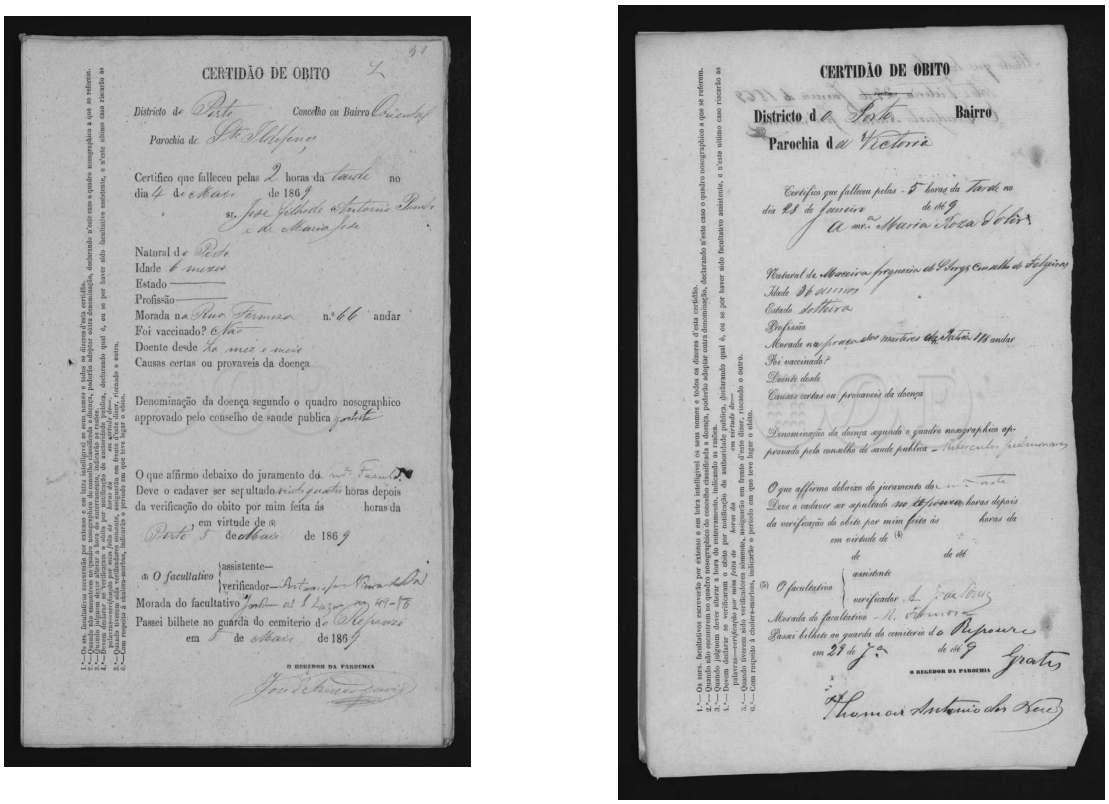
Despite layout differences, all forms contain identical information fields in Portuguese.

The dataset of 1,635 transcribed certificates was split into training (60%), validation (20%), and test (20%) sets using a systematic sampling approach. To ensure representative distribution across the temporal sequence of documents, the split was performed by partitioning every consecutive group of five certificates: the first three were assigned to the training set, the fourth to the validation set, and the fifth to the test set. This sequential partitioning strategy maintains the chronological distribution of documents across all splits while preventing data leakage between sets

This work focuses exclusively on the seven core information fields in table 5.1.1 identified as most important for historical research purposes.

5.2 EVALUATION METRICS

In our Information Extraction task, which involves extracting structured information from document images framed as question-answer pairs, we employ two primary metrics to assess model performance: Macro-F1 and Accuracy. These metrics are tailored to handle the JSON-formatted outputs, accounting for nested structures and



Form 2a *Form 2b*

Figure 5.1.2: Two different shapes of the vertical death certificates.

multiple field extractions per image. The ground truth is provided as a list of QA parses :

{"gt_parses" : [{"question" : "What is paroq_cert", "answer" : "Cedofeita"}, ...]}

representing key-value extractions. Model predictions are generated via prompts concatenating multiple <s_docvqa><s_question>q</s_question><s_answer> sequences, with outputs parsed into JSON dictionaries for evaluation.

Table 5.1.1: Core information fields extracted from death certificates.

Field	Description
Parish name	Parish where death occurred
Name	Full name of deceased (may include pronoun)
Age	Age of deceased in years
Birthplace	Place of birth
Date of death	Date when death occurred
Cause of death	Medical cause(s) of death
Poverty status	Economic status indicator

5.2.1 *Macro-F1*

The Macro-F1 score measures field extraction quality by comparing predicted and ground truth key-value pairs at the document level. This metric follows the standard evaluation protocol used in DocVQA tasks and the Donut framework.

Computation Process

For each document, keys are normalized by sorting by length then alphabetically and also converting single values to lists of single elements. Afterwards, the predicted and ground truth JSON structures are flattened into sets of (key, value) pairs. For example, `{"menu": [{"price": [10.5, 12.0]}, ]}` becomes the pairs `(menu.price, 10.5)`, `(menu.price, 12.0)`.

For each document, we compute:

- True Positives (TP): Matching (key, value) pairs between prediction and ground truth
- False Positives (FP): Predicted pairs not in ground truth
- False Negatives (FN): Ground truth pairs missing from prediction

The per-document F1 score is:

$$\text{F1}_{\text{doc}} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}}$$

The final Macro-F1 is the arithmetic mean across all documents:

$$\text{Macro-F1} = \frac{1}{N}\sum_{i=1}^{N} \text{F1}_{\text{doc}_i}$$

The values of the metrics are in the range $[0, 1]$ where 0 is worst. This metric is useful for evaluating extraction quality as it handles partial matches: a prediction that gets the key correct but value wrong receives partial credit, unlike strict exact-match accuracy.

5.2.2 *Tree Edit Distance (TED)-based accuracy*

The Tree Edit Distance (TED) based accuracy metric evaluates both structural and content similarity by representing JSON outputs as labeled trees. This metric follows the standard evaluation protocol used in DocVQA tasks and the Donut framework.

Computation Process

For each document, the predicted and ground truth JSON structures are normalized (as in Macro-F1) converted into labeled trees. Dictionary keys become nodes, with their values as child nodes; lists of dictionaries are represented as multiple child nodes under a `<subtree>` nodes, and primitive values are marked `<leaf>` nodes.

For example, consider the JSON:

```
{"menu": [{"item": "burger", "price": [10.5, 12.0]}, {"item": "fries"}]}
```

Its tree representation is shown in Figure 5.2.1.

The TED quantifies the minimum cost of operations to transform the predicted tree into the ground truth tree:

- Updates: String edit distance for leaf nodes; 1 for mismatched non-leaf labels (0 if matching)
- Insertions/Deletions: Label length for leaves; 1 for non-leaves

The normalized TED (nTED) is:

$$\text{nTED} = \frac{\text{TED}(\text{pred}, \text{gt})}{\text{TED}(\emptyset, \text{gt})}$$

The per-document accuracy is:

$$\text{Acc}_{\text{doc}} = \max(1 - \text{nTED}, 0)$$

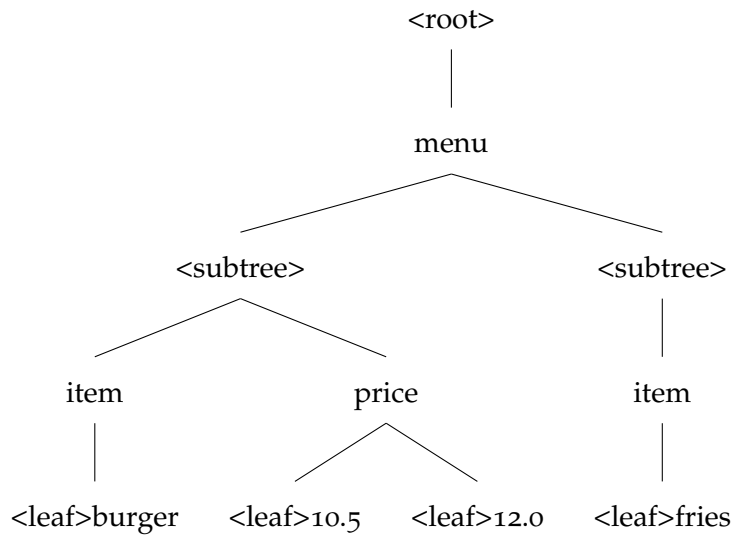


Figure 5.2.1: Tree representation of the JSON `{"menu": [{"item": "burger", "price": [10.5, 12.0]}, {"item": "fries"}]}`.

The final TED-based accuracy is the arithmetic mean across all documents:

$$\text{TED-Acc} = \frac{1}{N} \sum_{i=1}^{N} \text{Acc}_{\text{doc}_i}$$

The metric yields values between 0 and 1, where 0 is worst. This metric is useful for evaluating extraction quality as it penalizes hierarchical mismatches (e.g., incorrect nesting or list orders) alongside content errors, providing a more comprehensive assessment than flat metrics like Macro-F1.

5.3 EXPERIMENTAL SETUP

5.3.1 *Hardware Configuration*

All experiments were conducted on a single NVIDIA GeForce RTX 3070 GPU. Due to memory constraints, input resolution and batch size were adjusted from the original Donut training configuration.

5.3.2 *Training Configuration*

Training was performed by fine-tuning the pre-trained Donut model[1], which had been previously trained on the DocVQA [MKJ20] dataset. Following sections will reference this checkpoint as DocVQA checkpoint. The training hyperparameters were adapted from the original Donut configuration[2] with the following key modifications:

[1] https://huggingface.co/naver-clova-ix/donut-base-finetuned-docvqa accessed on 02.08.2025.
[2] https://github.com/clovaai/donut/blob/master/config/train_docvqa.yaml

Model Input:

- Input resolution: 640×320 pixels (reduced from 2560×1920 to accommodate GPU memory)

Training Hyperparameters:

- Batch size: 1 (training and validation)
- Maximum epochs: 12
- Warmup steps: 10% of total training steps
- Early stopping patience: 3 epochs
- Gradient clipping: maximum gradient norm of 1.0

Training Strategy: Validation was performed after each epoch, with early stopping triggered if validation performance did not improve for 3 consecutive epochs. The model checkpoint with the best validation performance was retained for testing.

5.4 BASELINE PERFORMANCE

To establish a performance baseline, the pre-trained Donut DocVQA checkpoint was evaluated directly on the test set without any fine-tuning on the Portuguese death certificates. This checkpoint had been trained on the DocVQA [MKJ20] dataset, which consists primarily of modern English documents with varied layouts and question types.

When applied to extract core fields such as parish name, deceased name, age, birthplace, date of death, cause of death, and poverty status from the Portuguese historical certificates; the model achieved 0% accuracy for both TED-based accuracy and Macro-F1 metrics across all fields.

This complete failure is attributable to several domain mismatches:

- **Language barrier:** The model was trained exclusively on English, Japanese and Chinese documents, while the death certificates contain 19th-century Portuguese text with archaic vocabulary and spelling conventions.
- **Historical handwriting:** The certificates feature cursive handwriting styles characteristic of the 1860s, which differ substantially from the modern printed and handwritten text in DocVQA.

- **Document degradation:** Physical deterioration including fading, staining, and paper damage significantly affects text legibility.

These results demonstrate that despite Donut's strong performance on modern document understanding tasks, direct transfer to historical Portuguese documents requires domain-specific fine-tuning.

5.5 SINGLE-FIELD EXTRACTION

Following the baseline evaluation, the pre-trained Donut checkpoint was fine-tuned independently on each core field using the training split, then evaluated on the corresponding test split. This approach allows assessment of field-specific extraction difficulty and model adaptability to different information types within the documents.

5.5.1 *Experimental Design*

For each core field, a separate fine-tuning experiment was conducted:

1. Each training sample consisted of a single question-answer pair targeting the specific field
2. Fine-tuning was performed using the configuration described in Section 5.3
3. The best checkpoint based on validation performance was selected for test evaluation

5.5.2 *Results and Analysis*

Table 5.5.1 presents the TED-based accuracy and Macro-F1 scores for each field extraction task.

Table 5.5.1: Single-field extraction performance after fine-tuning on individual core fields.

Field	**TED Accuracy**	**Macro-F1**
Parish name	98%	49%
Birthplace	60%	30%
Poverty status	55%	28%
Name	6%	3%

Parish name achieved the highest performance (98% TED accuracy), likely due to the limited vocabulary of parish names (only 12 values) within Porto municipality and their consistent spatial positioning within the certificate layout.

Birthplace and poverty status demonstrated moderate performance (55-60% TED accuracy), suggesting that the model can learn to extract these fields with reasonable accuracy despite handwriting variability and document degradation.

Name extraction exhibited dramatically lower performance (6% TED accuracy), representing a near-complete failure. This poor performance is attributable to severe class imbalance in the dataset, as illustrated in Figure 5.5.1. The distribution is heavily dominated by a small set of common names: "Maria" alone accounts for approximately 14% of all samples, while "António", "José", "Manuel", and "Joaquina" together with "Maria" represent the majority of training examples. In stark contrast, the vast majority of names in the dataset appear with extremely low frequency, most occurring at most three times in the training data, with many appearing only twice or once. This extreme long-tail distribution prevents the model from learning robust name extraction patterns, as it lacks sufficient examples to generalize beyond the most frequent names. The model essentially memorizes the common names, but fails to recognize the orthographic and spatial patterns that would enable extraction of rare or previously unseen names.

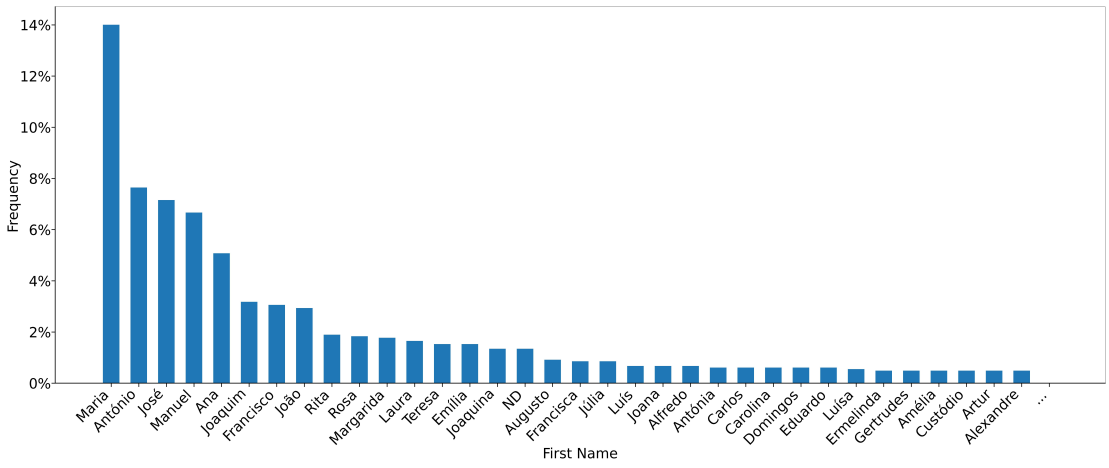


Figure 5.5.1: Distribution of first names in the dataset showing severe class imbalance. The top five names (Maria, António, José, Manuel, Joaquina) dominate the dataset, while the majority of names appear three times or fewer. Y-axis shows frequency as proportion of total samples.

The substantial performance gap between fields demonstrates that extraction difficulty varies significantly based on vocabulary diversity and training example frequency, with high-cardinality fields like names requiring substantially larger datasets or alternative training strategies.

5.6 MULTI-FIELD SEQUENTIAL TRAINING

To examine how to effectively prompt and fine-tune an integrated model like Donut, an experiment was conducted using sequential fine-tuning. The hypothesis was that training the model on multiple fields sequentially might improve performance through exposure to more document variations and potentially shared visual patterns.

5.6.1 *Experimental Design*

The model was fine-tuned in two stages:

1. Fine-tuning on parish name extraction using the parish name training split
2. Continued fine-tuning on name extraction using the name training split

Both fields were then evaluated on their respective test splits after the complete two-stage training process.

5.6.2 *Results and Analysis*

Table 5.6.1 presents the performance comparison between sequential training and single-field training approaches.

Table 5.6.1: Sequential training performance compared to single-field baseline.

Field	**TED Accuracy (Sequential)**	**TED Accuracy (Single-field)**
Parish name	67%	98%
Name	6%	6%

Sequential training yielded no measurable improvement over single-field training for either field, as shown in Table 5.6.1. Parish name performance degraded substantially from the single-field of the last experiment, while name extraction performance remained unchanged, showing no benefit from prior parish name training. This absence of performance gains, combined with the degradation in parish name extraction,

suggests that sequential training on related extraction tasks did not lead to improved generalization between them and may have introduced negative transfer effects.

The lack of transfer can be attributed to persistent data imbalance issues. Exposing the model to additional training data through parish name fine-tuning does not address the fundamental class imbalance problem in name extraction. The model still lacks sufficient examples of rare names to generalize effectively, and simply increasing overall document exposure through sequential training cannot compensate for the extreme long-tail distribution of names in the dataset.

Furthermore, these results highlight limitations of the sequential training approach itself. Simple sequential fine-tuning may fail to create shared representations between tasks, as the model appears to treat each field extraction as an independent problem rather than learning transferable visual or linguistic patterns. Alternative approaches such as multi-task learning with simultaneous training on both fields might yield different results by encouraging the development of shared feature representations. These findings indicate that field-specific challenges, particularly severe class imbalance, cannot be overcome simply by increasing overall training exposure through sequential training on related fields.

5.7 PREPROCESSING: DOCUMENT CROPPING

To investigate whether reducing irrelevant visual information could improve extraction performance, particularly for the poorly performing name field, experiments were conducted using cropped document images that focus on the upper region containing core information fields.

5.7.1 *Cropping Methodology*

A fixed cropping strategy was applied to all documents: 20% removed from the left edge and 50% removed from the bottom.

The motivation for testing this cropping approach was to investigate whether removing irrelevant document regions might improve the quality and legibility of handwriting when images are scaled down to the fixed input resolution of 640×320 pixels. By reducing the visual field to focus on relevant content, cropping could potentially preserve more fine-grained details of the handwritten text that would otherwise be lost during downsampling of full documents.

This specific cropping ratio was selected to retain the upper portion of the certificates where parish name, deceased name, birthplace, age, and date of death fields are lo-

cated, while removing the lower section containing boilerplate text and administrative information. The cropping parameters were found to work consistently across both horizontal (Form 1) and vertical (Form 2a/2b) document layouts.

Figure 5.7.1 illustrates the effect of cropping on a sample certificate, with colored boxes indicating the locations of parish name (blue) and deceased name (green) fields.

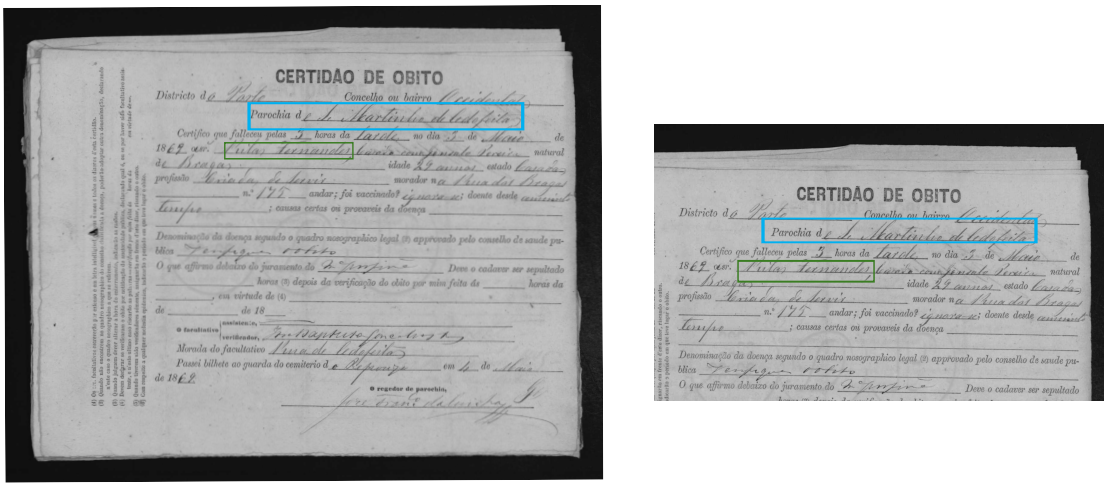


Figure 5.7.1: Death certificate before (left) and after (right) cropping. Blue box indicates parish name field location, green box indicates deceased name field location. The cropping removes 20% from left and 50% from bottom while preserving the core information fields in the upper region.

5.7.2 *Results and Analysis*

Three experimental configurations were evaluated to assess the impact of cropping on model performance:

- Models fine-tuned on cropped images and evaluated on cropped test images
- Models fine-tuned on cropped images and evaluated on full uncropped test images
- Models fine-tuned on uncropped images (from Section 5.5) and evaluated on cropped test images

Name Field Experiments

Table 5.7.1 presents the performance of name extraction with cropped training across different testing formats.

Table 5.7.1: Impact of image cropping on name extraction performance.

Training Format	Testing Format	TED Accuracy
Cropped	Cropped	6%
Cropped	Uncropped	6%
Uncropped	Uncropped	6% (Table 5.6.1)
Uncropped	Cropped	6%

The model was fine-tuned using cropped images for name extraction and evaluated on both cropped and uncropped test images. No improvement was observed over the baseline single-field performance, regardless of whether test images were cropped.

This result suggests that the fundamental challenge of extreme class imbalance cannot be addressed through spatial focus alone. Reducing the visual field does not help the model generalize to rare names that lack sufficient training examples.

Parish Name Experiments

Table 5.7.2 presents the performance of parish name extraction across different cropping configurations. The results reveal strong sensitivity to training-testing format mismatches, with the most dramatic degradation occurring when models trained on full documents are tested on cropped images.

Table 5.7.2: Impact of image cropping on parish name extraction performance.

Training Format	Testing Format	TED Accuracy	Macro-F1
Uncropped	Uncropped	98% (Table 5.6.1)	49% (Table 5.6.1)
Cropped	Cropped	88%	44%
Cropped	Uncropped	66%	33%
Uncropped	Cropped	22%	11%

The cropping experiments reveal several important findings about the model's behavior. Most notably, the results demonstrate strong spatial context dependence: when the test-time image format mismatches the training format, performance drops

dramatically. For the parish name field, accuracy plummets from 98% to 22% when a model trained on full documents is tested on cropped images. This indicates that Donut learns strong spatial priors about field locations within the document layout, and removing spatial context at test time violates these learned expectations, which indicates that data augmentation may have benefited the training.

This spatial dependence manifests asymmetrically across training configurations. Models trained on cropped images retain some ability to handle full documents (degrading to 66% accuracy), while models trained on full documents cannot adapt to unexpected cropping at test time (collapsing to 22% accuracy). This asymmetry suggests that exposure to the full document layout during training provides more robust spatial representations than training on cropped images alone.

Even when training and testing conditions remain consistent, cropping introduces modest performance degradation for high-performing fields. Parish name extraction decreases from 98% to 88% accuracy when both training and testing use cropped images, suggesting that broader document context provides subtle but useful cues for extraction that are lost when the visual field is reduced.

Importantly, cropping provided no benefit for the challenging name extraction field, confirming that data scarcity and class imbalance are the fundamental limitations rather than spatial distraction from irrelevant document regions. These results demonstrate that preprocessing strategies like cropping do not address the core challenges of historical document processing and that maintaining consistency between training and deployment conditions is critical for model performance.

6 CONCLUSION AND OUTLOOK

6.1 SUMMARY

This thesis investigated the application of segmentation-free transformer-based models for extracting structured information from historical Portuguese death certificates. Specifically, it evaluated the Donut model's capacity to perform end-to-end information extraction from handwritten 19th-century administrative documents without requiring optical character recognition preprocessing or layout segmentation. The experimental results demonstrate both the potential and current limitations of applying pre-trained document understanding models to historical archival materials. The baseline evaluation confirmed that direct transfer from modern documents to historical Portuguese certificates is infeasible, with the pre-trained model achieving 0% accuracy across all extraction tasks. This complete failure underscores the substantial domain gap between current document understanding benchmarks and degraded historical handwriting.

Domain-specific fine-tuning yielded highly variable results depending on field characteristics. Parish name extraction achieved 98% TED-based accuracy, demonstrating that Donut can successfully learn extraction patterns when vocabulary is constrained and spatial positioning is consistent. Birthplace and poverty status showed moderate performance (55-60% accuracy), indicating the model can handle fields with greater variability given sufficient training data. However, name extraction exhibited near-complete failure (6% accuracy), revealing a critical limitation: severe class imbalance in the training data fundamentally prevents the model from generalizing beyond frequently occurring values.

The analysis identified class imbalance as the primary barrier to successful extraction of high-cardinality fields. The extreme long-tail distribution of personal names prevents the model from learning generalizable orthographic and spatial patterns. This finding has significant implications for applying deep learning to historical records, where many fields naturally exhibit high cardinality and sparse distributions.

Sequential training experiments found no evidence of knowledge transfer between related extraction tasks, suggesting that simple curriculum learning approaches cannot compensate for fundamental data scarcity issues.

Similarly, document cropping experiments revealed that the model develops strong spatial context dependencies during training, with performance degrading dramatically when test-time image formats differ from training conditions. Notably, cropping provided no benefit for challenging fields, confirming that data distribution rather than spatial distraction constitutes the limiting factor.

These results establish that while OCR-free transformer models represent a promising direction for historical document processing, their effectiveness depends critically on training data characteristics. Fields with limited vocabularies and consistent formatting can be extracted reliably with modest training sets, while high-cardinality fields require substantially larger and more balanced datasets to achieve acceptable performance.

6.2 LIMITATIONS

Several limitations constrain the generalizability and completeness of this work. The dataset encompasses only 1,635 transcribed certificates from a single municipality (Porto) covering January 1869 to January 1870, limiting both the volume of training data and the diversity of handwriting styles, document conditions, and administrative practices represented. GPU memory constraints necessitated reducing input resolution from 2560×1920 to 640×320 pixels, potentially degrading the model's ability to discriminate fine-grained handwriting details.

6.3 SUGGESTIONS FOR FUTURE WORK

Several directions could address the identified limitations and advance segmentation-free processing of Portuguese death certificates. The most critical need is mitigating class imbalance for personal names in death certificates. Synthetic data generation could augment the training set by rendering additional Portuguese name variations in 19th-century handwriting styles, replicating rare names to balance their representation against frequently occurring names like Maria, António, and José.

Alternatively, strategic oversampling of underrepresented names or class-balanced batch sampling during training may improve model generalization to names appearing only once or twice in the original dataset. The training paradigm warrants reconsideration for certificate processing. Rather than framing extraction as single question-answer pairs, adopting a multi-answer format where each certificate provides ground truth for all seven fields simultaneously could enable the model to develop

shared representations across related information types. This approach may facilitate knowledge transfer that was absent in the sequential training experiments.

BIBLIOGRAPHY

[Bao+22] Hangbo Bao et al. *BEiT: BERT Pre-Training of Image Transformers*. 2022. arXiv: 2106.08254 [cs.CV]. URL: https://arxiv.org/abs/2106.08254.

[BB23] Christopher M. Bishop and Hugh Bishop. *Deep Learning: Foundations and Concepts*. 1st ed. Springer Cham, 2023. ISBN: 978-3-031-45467-7. DOI: 10.1007/978-3-031-45468-4.

[Boi+24] Mélodie Boillet et al. *The Socface Project: Large-Scale Collection, Processing, and Analysis of a Century of French Censuses*. 2024. arXiv: 2404.18706 [cs.CV]. URL: https://arxiv.org/abs/2404.18706.

[CCP23] Denis Coquenet, Clément Chatelain, and Thierry Paquet. "DAN: A Segmentation-Free Document Attention Network for Handwritten Document Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.7 (July 2023), pp. 8227–8243. ISSN: 1939-3539. DOI: 10.1109/tpami.2023.3235826. URL: http://dx.doi.org/10.1109/TPAMI.2023.3235826.

[Dav+19] Brian Davis et al. "Deep Visual Template-Free Form Parsing". In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019, pp. 134–141. DOI: 10.1109/ICDAR.2019.00030.

[Dav+22] Brian Davis et al. *End-to-end Document Recognition and Understanding with Dessurt*. 2022. arXiv: 2203.16618 [cs.CV]. URL: https://arxiv.org/abs/2203.16618.

[Dos+21] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV]. URL: https://arxiv.org/abs/2010.11929.

[Fin25] Gernot A. Fink. *Skriptum zur Vorlesung "Mustererkennung"*. https://web.patrec.cs.tu-dortmund.de/lectures/WS25/mustererkennung/mustererkennung.pdf. Accessed: 2025-10-13. 2025.

[Gro+09] Emmanuèle Grosicki et al. "Results of the RIMES Evaluation Campaign for Handwritten Mail Processing". In: *2009 10th International Conference on Document Analysis and Recognition*. 2009, pp. 941–945. DOI: 10.1109/ICDAR.2009.224.

[GS08] Alex Graves and Jürgen Schmidhuber. "Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by D. Koller et al. Vol. 21. Curran Associates, Inc., 2008. URL: https://proceedings.neurips.cc/paper_files/paper/2008/file/66368270ffd51418ec58bd793f2d9b1b-Paper.pdf.

[JET19] Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. "FUNSD: A Dataset for Form Understanding in Noisy Scanned Documents". In: *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*. Vol. 2. IEEE, 2019, pp. 1–6. DOI: 10.1109/ICDARW.2019.10029.

[Kim+22] Geewook Kim et al. *OCR-free Document Understanding Transformer*. 2022. arXiv: 2111.15664 [cs.LG]. URL: https://arxiv.org/abs/2111.15664.

[Lew+06] D. Lewis et al. "Building a test collection for complex document information processing". In: *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '06. Seattle, Washington, USA: Association for Computing Machinery, 2006, pp. 665–666. ISBN: 1595933697. DOI: 10.1145/1148170.1148307. URL: https://doi.org/10.1145/1148170.1148307.

[Lew+19] Mike Lewis et al. *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. 2019. arXiv: 1910.13461 [cs.CL].

[Li+22] Minghao Li et al. *TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models*. 2022. arXiv: 2109.10282 [cs.CL]. URL: https://arxiv.org/abs/2109.10282.

[Liu+19] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 [cs.CL]. URL: https://arxiv.org/abs/1907.11692.

[Liu+21] Ze Liu et al. *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*. 2021. arXiv: 2103.14030 [cs.CV].

[MB02] U.-V. Marti and H. Bunke. "The IAM-database: an English sentence database for offline handwriting recognition". In: *International Journal on Document Analysis and Recognition* 5.1 (2002), pp. 39–46. DOI: 10.1007/s100320200071.

[MKJ20] Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. *DocVQA: A Dataset for VQA on Document Images*. 2020. arXiv: 2007.00398 [cs.CV].

[SBY15] Baoguang Shi, Xiang Bai, and Cong Yao. *An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition*. 2015. arXiv: 1507.05717 [cs.CV]. URL: https://arxiv.org/abs/1507.05717.

[Tos+18] A.H. Toselli et al. *HTR Dataset ICFHR 2016 (Version 1.2.0)*. 2018. DOI: 10.5281/zenodo.1297399. URL: https://zenodo.org/record/1297399.

[Vas+23] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[Xu+20] Yiheng Xu et al. "LayoutLM: Pre-training of Text and Layout for Document Image Understanding". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '20. ACM, Aug. 2020, pp. 1192–1200. DOI: 10.1145/3394486.3403172. URL: http://dx.doi.org/10.1145/3394486.3403172.