Textual Attribute Recognition for Documents

A thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Computer Science and Engineering by Research

by

Jyothi Swaroopa Jinka
2023701009
jinka.swaroopa@research.iiit.ac.in

Advisor: Dr. Ravi Kiran Sarvadevabhatla

International Institute of Information Technology Hyderabad
500 032, India

May 2026

International Institute of Information Technology Hyderabad
Hyderabad, India

CERTIFICATE

This is to certify that work presented in this thesis proposal titled ***Textual Attribute Recognition for Documents*** by *Jyothi Swaroopa Jinka* has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Advisor: Dr. Ravi Kiran Sarvadevabhatla

To my family, Guru and Lord Shiva

Acknowledgements

The work presented in this thesis has been made possible through the support and encouragement of my family, friends, colleagues, and above all, my advisor, Prof. Ravi Kiran Sarvadevabhatla. I have had the privilege of interacting with many individuals during this journey, each of whom has contributed to my growth and left a lasting impact on me.

First and foremost, I would like to express my sincere gratitude to my advisor, Prof. Ravi Kiran Sarvadevabhatla, for his invaluable mentorship and constant support. His guidance has been central to my research journey. During times when I struggled with confidence or direction, he was always patient and encouraging, helping me regain clarity and move ahead. In my initial days, when I had very little understanding of research, he invested significant time in mentoring me, teaching me not just how to approach problems, but how to think about them more deeply. His influence has shaped both my work and the way I approach research today.

I am deeply grateful to my family, who have been my strongest foundation. My parents, Venu Gopal Jinka and Nirmala Putha have always been my biggest supporters in choosing this path. While many questioned the need for pursuing research, especially when I already had a job, they always believed in me and encouraged me to explore higher studies. That belief meant everything to me. I am incredibly fortunate to have my two elder brothers, Sai Sagar Jinka and Uday Sagar Jinka. My eldest brother has been more than just a brother, he has been like a father to me and a mentor. He is the constant pillar of strength in both my personal and professional life. His guidance and support have influenced me in countless ways. My elder brother, Uday has always been my go-to person whenever I needed encouragement or a break from everything. Spending time with him has always been a source of comfort and happiness. All our small, meaningless fights are some of my most cherished memories. Without both of them, it is hard to imagine how I would have handled the stress and challenges of this journey. I would also like to thank my sisters-in-law, Meghana and Geethika, who have been more like sisters to me. The warmth, support, and the little joyful moments, especially our conversations and laughter made this journey much lighter and happier. I would also like to mention my niece and nephew, Indrakshi and Rudransh. Even a short conversation with them is enough to lift my mood and bring a sense of joy. Their innocent words and the way they call me “atha” always make me smile.

I am truly grateful to my wonderful friends, Lavanya, Kesav, Hrithik, Aryan, Srivatsa, Anirudh and Madhu for making this journey memorable. It has been a privilege to work and grow alongside them. I would especially like to thank Madhu for his constant encouragement and guidance. He always took the time to patiently address my questions and help me gain clarity. Beyond this, he consistently

encouraged me to explore new directions, think beyond conventional boundaries, and approach learning in a more strategic and structured manner. His guidance greatly improved both my understanding and my approach to research. I am grateful to Kesav for helping me during my initial days at BharatGen.

I would also like to thank my undergraduate friends, Bindhu, Shreya, and Sharvani for their constant support and encouragement throughout this journey.To my childhood friends, Saketh, Fahim, Vamsi, Vinay, Kaivalya, Navya, Chihnitha, and Likhitha, thank you for always bringing joy into my life. Spending time with you has always made me feel lively and happy, no matter how stressful things were.

I would like to express my heartfelt gratitude to Late B.V. Khadiravana, who was my mentor during my early days at CVIT. He played a crucial role in shaping my understanding of research. He taught me how to approach problems deeply rather than rushing toward solutions. During difficult times, especially when I struggled, his guidance and support made a lasting impact on me. I will always carry those lessons with me.

Finally, I am grateful to everyone who, in ways big or small, contributed to this journey and helped me become the person I am today.

Abstract

Understanding document content requires not only recognizing textual characters but also capturing textual attributes such as bold, italic, underline, and strikeout, which convey emphasis, hierarchy, and semantic intent. These attributes play a critical role in real-world documents, including legal records, administrative forms, educational materials, and historical manuscripts, where formatting often encodes meaning that cannot be inferred from text alone. Despite their importance, Textual Attribute Recognition (TAR) has received limited attention compared to traditional Optical Character Recognition (OCR), and existing approaches remain inadequate for handling the complexity of real-world, multilingual, and layout-rich documents.

A key limitation of prior TAR methods lies in their treatment of words as isolated visual units. Attribute recognition, however, is inherently context-dependent, as distinctions such as bold versus regular text or underline versus structural lines often require comparison with neighboring words or surrounding layout elements. Methods that ignore this context suffer from ambiguity and reduced accuracy. Conversely, approaches that attempt to incorporate full-document context using transformer-based architectures face significant computational challenges, due to the quadratic complexity of attention over large numbers of word instances. These issues are further exacerbated in multilingual settings, where script diversity, font variations, and document noise introduce additional variability.

This thesis addresses these limitations through TexTAR, a context-aware, multi-task Transformer framework for scalable and robust Textual Attribute Recognition. The central idea is to achieve an effective balance between contextual modeling and computational efficiency by operating on compact local contexts rather than entire documents. To this end, we introduce a Context Window (CW) based data selection pipeline, in which each target word is processed together with a small set of its spatially nearest neighbors, selected using a weighted Chebyshev distance. This formulation enables the model to capture essential contextual cues required for attribute discrimination, while avoiding the computational overhead associated with global attention.

Building on this representation, TexTAR employs a transformer architecture that processes sequences of word images within each context window. To effectively encode spatial relationships and document layout, the model incorporates a two-dimensional Rotary Positional Embedding (2D RoPE)-style mechanism, allowing it to model both horizontal and vertical positional dependencies. This design enables the network to reason about relative positioning and alignment patterns that are crucial for distinguishing visually similar attributes. Furthermore, TexTAR is trained in a multi-task learning setting, jointly

predicting multiple groups of attributes, which improves generalization and robustness across diverse document types and formatting styles.

In addition to the model architecture, this thesis introduces MMTAD (Multilingual Multi-domain Text Attribute Dataset), a large-scale dataset comprising over a million word-level annotations across multiple languages and document domains. The dataset captures a wide range of real-world variability, including different scripts, layouts, font styles, and noise conditions. To address the inherent class imbalance in attribute distributions, targeted data augmentation strategies are employed to simulate attribute-specific variations, enabling the model to learn more balanced and discriminative representations.

Extensive experimental evaluation demonstrates that TexTAR consistently outperforms existing approaches across multiple benchmarks and settings. The results highlight the importance of context-aware modeling for accurate attribute recognition, particularly in challenging scenarios involving ambiguous visual cues or complex layouts. At the same time, the proposed context window formulation ensures that the model remains computationally efficient and scalable to large document collections.

More broadly, this thesis argues that document understanding systems must move beyond text recognition toward richer representations that incorporate visual attributes as first-class signals. By integrating efficient context modeling, spatially-aware transformer architectures, and large-scale multilingual supervision, TexTAR advances the state of the art in Textual Attribute Recognition and provides a practical foundation for downstream applications such as document digitization, structured information extraction, and accessibility. In doing so, it contributes toward more comprehensive and semantically aware document intelligence systems.

Contents

List of Figures

List of Tables

List of Related Publications

[P1] Rohan Kumar, Jyothi Swaroopa Jinka and Ravi Kiran Sarvadevabhatla, **"TexTAR: Textual Attribute Recognition in Multi-domain and Multi-lingual Document Images"**, in proceedings of *International Conference on Document Analysis and Recognition, ICDAR*, 2025.

Chapter 1

Introduction

1.1 Motivation

The challenge of document understanding in computer vision is primarily driven by the dual goals of content extraction and structural fidelity. Although foundational methods in Optical Character Recognition (OCR) established core principles for recovering textual content from document images, the landscape has been reshaped by the growing demand for richer, semantically aware document intelligence. Systems capable of understanding not merely what text says, but how it is visually presented, have become increasingly essential across diverse real-world applications.

Modern documents convey meaning through a layered combination of linguistic content and visual presentation. Formatting cues such as **bold**, *italic*, underline, and ~~strikeout~~ are widely employed to express emphasis, hierarchy, correction, and semantic intent. These textual attributes are not merely decorative; they function as a layer of structural and semantic markup embedded directly within the visual form of a document. Consequently, any document intelligence system that aims to move beyond plain text extraction must be capable of recognizing not only what is written, but also how it is visually expressed. This gap between raw content recovery and attribute-aware document understanding is the primary motivation for this thesis.

1.1.1 The Dual Pursuit of Content and Presentation

A long-standing objective in document analysis has been the faithful digitization of document images into machine-readable representations. In recent years, significant progress has been made in transcribing textual content through deep learning-based OCR systems. The core pursuit within this field is twofold: achieving high-accuracy character recognition and enabling preservation of the visual structure that accompanies text. Although the challenge of content recognition has seen remarkable advancements, the quest for attribute-level, presentation-aware understanding remains an active and critical area of research. This thesis aims to address the specific need for word-level textual attribute recognition through a context-aware, multi-task framework.

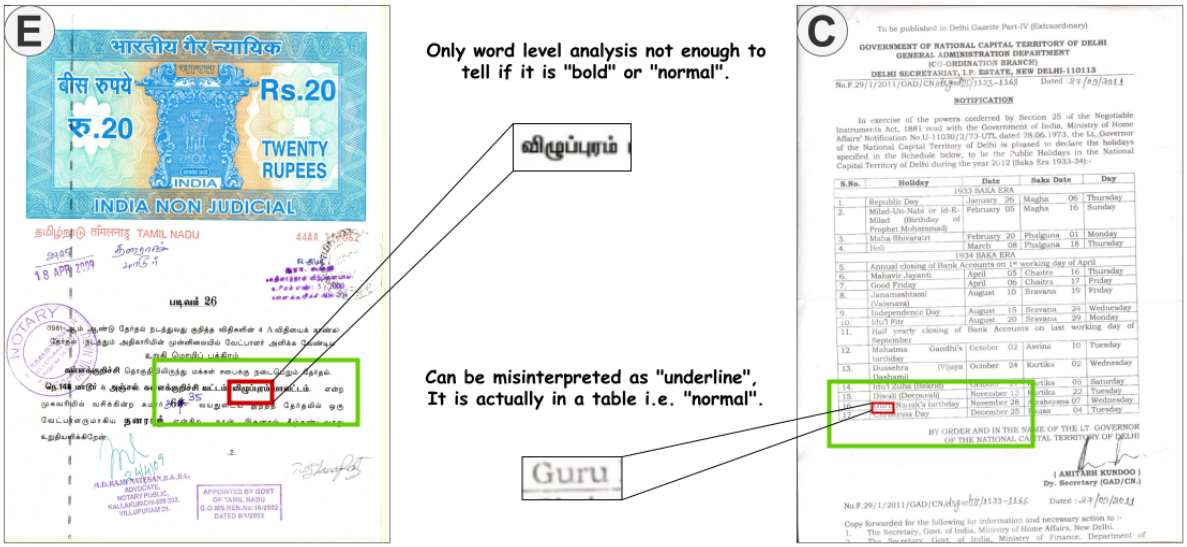


Figure 1.1 Challenges in Textual Attribute Recognition (TAR): (top) Challenge to distinguish `normal` and `bold` class from a single word level analysis. (bottom) Challenge to distinguish a `underline` word from a `normal` word in a table.

1.1.2 Evolution of Document Understanding: From Content to Structure

The evolution of document understanding systems is characterized by progressively more sophisticated methods of information recovery. Early approaches focused almost exclusively on recovering character sequences from document images, establishing OCR as the dominant paradigm. In these systems, presentation-level information was largely discarded, as the primary objective was transcription accuracy rather than structural fidelity.

A paradigm shift has occurred with the growing recognition that documents encode meaning at multiple levels simultaneously. Beyond raw text, documents employ formatting conventions — **bold** for emphasis, *italic* for distinction, underline for annotation, and ~~strikeout~~ for revision — that carry significant semantic and structural intent. This realization has prompted the development of richer document understanding frameworks that go beyond plain-text recovery. However, this progress has also exposed a fundamental gap: while systems can now transcribe text with high accuracy, recognizing the visual attributes attached to that text remains an underexplored and technically challenging problem. This structural deficit in existing document understanding systems defines the research frontier addressed by this thesis.

1.1.3 The Bottleneck of Isolated Word-Level Analysis

The recognition of textual attributes is complicated by the fact that many attributes cannot be determined reliably from individual word images in isolation. Existing approaches to Textual Attribute Recognition (TAR) largely treat the problem as a local classification task: given a cropped word image,

predict its associated attribute. While this formulation is simple, it is fundamentally limited because many textual attributes are inherently *relative* rather than absolute.

A word appears bold not solely because of its own stroke thickness, but because it is visually heavier than surrounding text. A horizontal line beneath a word may indicate underline, but it may equally represent a table border, separator, or annotation mark. Distinguishing between these possibilities requires access to spatial neighborhood and document-level context. However, a straightforward attempt to incorporate all words in a document introduces significant computational cost, especially when Transformer-based architectures are employed. These methods share a fundamental bottleneck: they solve either the problem of contextual richness or the problem of computational efficiency, but rarely both simultaneously. This defines the core technical challenge motivating the present work.

1.1.4 Context-Aware Attribute Recognition

The reliance on isolated word analysis defines the current research frontier and motivates a critical question: instead of classifying words independently, can a model learn to recognize textual attributes by reasoning over compact, spatially organized neighborhoods? This question frames the challenge of context-aware, part-sensitive document attribute recognition.

This level of understanding — the most semantically grounded in the spectrum of document analysis — aims to recover not just what text says, but how it is styled, through a learned awareness of local visual structure. As the historical trajectory indicates, the field consistently moves towards more explicit and detailed forms of document intelligence. The next logical step is to internalize spatial context into the attribute recognition process itself.

This thesis directly addresses this challenge. We propose a framework, **TexTAR**, that recognizes textual attributes through compact context windows centered on each anchor word, processed by a spatially aware Transformer architecture. By doing so, we aim to bridge the gap between isolated visual classification and structurally grounded document understanding, thereby addressing the critical bottleneck in the current state of the art.

1.2 Problem Statement

We explore the problem of recognizing word-level textual attributes in document images with high accuracy and contextual robustness, with three focus areas.

- **Context-aware recognition of textual attributes.** Our aim is to develop a compact neighborhood-based framework that predicts attributes such as bold, italic, underline, and strikeout by jointly reasoning over an anchor word and its spatial surroundings.
- **Efficient and scalable modeling for real-world documents.** Given the computational constraints of processing full documents, we aim to design an architecture that delivers contextual richness within a tractable computational budget.

- **Evaluation and benchmarking under multilingual and multidomain conditions.** To evaluate our proposed architecture, we introduce a large-scale dataset and a framework of metrics that assess both per-attribute recognition performance and generalization across diverse document types.

1.3 Research Challenges

- **Data scarcity and domain imbalance.** Fine-grained word-level attribute annotations are scarce, and existing datasets are limited in language coverage, document diversity, and attribute variety.
- **Visual ambiguity between attributes and document structure.** Attributes such as underline and strikeout are easily confused with table borders, separators, and incidental marks, requiring contextual disambiguation.
- **Balancing contextual richness with computational efficiency.** Incorporating neighborhood context is necessary for robust recognition, but naive full-document attention is prohibitively expensive for large documents.
- **Generalization across scripts, languages, and acquisition conditions.** Real-world documents vary substantially in font, layout, resolution, noise, and script, requiring models that are robust to multilingual and multidomain variation.

1.4 Thesis Contributions

As a contribution to this thesis, we propose a context-aware multi-task framework for textual attribute recognition and develop accompanying data and evaluation resources. Below is a summary.

- **TexTAR**, a context-aware Transformer-based architecture that processes compact local neighborhoods around each anchor word through a *Context Window* formulation for efficient and robust TAR.
- **A 2D RoPE-style spatially aware positional encoding** mechanism that enables the model to explicitly represent horizontal and vertical layout relationships among neighboring words.
- **MMTAD**, a Multilingual and Multidomain Textual Attribute Dataset providing large-scale word-level annotations across diverse languages, scripts, and document domains.
- **A multi-task training framework** that jointly recognizes multiple attribute categories within a unified architecture, promoting shared learning and improved generalization.
- **Novel evaluation metrics and protocols** for assessing per-attribute recognition performance, contextual sensitivity, and cross-domain generalization.

The code, model artifacts, and datasets for this project can be found at `https://textar.github.io/`

1.5 Structure of the Thesis

In this chapter, we introduce the problem domain and the motivation behind it, providing a brief overview of the research challenges and limitations of existing approaches.

Chapter 2 reviews related work in textual attribute recognition, document image understanding, context-aware visual modeling, and multilingual document analysis.

Chapter 3 presents the proposed TexTAR framework, including the problem formulation, context window construction, and model architecture.

Chapter 4 introduces the MMTAD dataset, detailing its annotation design, attribute taxonomy, and data preparation pipeline for multilingual and multidomain settings.

Chapter 5 describes the experimental setup, training strategy, baselines, evaluation protocol, and presents both quantitative and qualitative results along with ablation studies.

Chapter 6 explores the integration of TexTAR within the REPLICA pipeline, highlighting the role of textual attribute enrichment in improving document reconstruction quality.

Finally, Chapter 7 concludes the thesis by summarizing the key findings and outlining potential directions for future work.

Chapter 2

Related Work

This thesis lies at the intersection of document image understanding, fine-grained visual recognition, and context-aware modeling. In this section, we review prior work in textual attribute recognition (TAR), classical and OCR-based approaches, deep learning methods, context-aware modeling strategies, and challenges in multilingual real-world document settings.

2.1 Textual Attribute Recognition

Textual Attribute Recognition (TAR) is a fine-grained visual recognition task that aims to identify attributes such as bold, italic, underline, and strikeout for each word in a document image. Unlike conventional OCR, which focuses on recovering textual content, TAR seeks to capture presentation-level cues that contribute to semantic interpretation and document structure.

Early work in TAR primarily focused on exploiting low-level visual features of text, such as stroke thickness, orientation, and geometric properties of characters. Several studies [1–4] investigated the relationship between font characteristics and textual attributes, using handcrafted features or limited learned representations to distinguish between styles such as bold and italic. These methods demonstrated that textual attributes could be inferred from visual patterns in character shapes, but they were often restricted to controlled settings and lacked robustness to real-world variability.

2.2 Classical and OCR-based Approaches

Classical approaches to textual attribute recognition relied heavily on rule-based and image processing techniques. Methods based on morphological operations [5, 6] attempted to remove noise and isolate structural features such as underlines or stroke variations. Other works explored format conversion pipelines [7], where printed documents were transformed into editable representations by transferring style information along with text content.

Open-source OCR systems such as Tesseract [8] also provide limited support for textual attribute recognition. These systems typically infer attributes based on font properties and predefined heuristics. While effective in clean and well-formatted documents, such approaches struggle in scenarios involving

subtle attribute variations, noisy scans, or complex layouts. Moreover, they are not designed to handle multilingual or multidomain document settings, where font styles, scripts, and formatting conventions vary significantly.

A common limitation of classical and OCR-based approaches is their reliance on predefined rules or shallow features. As a result, they lack adaptability and fail to generalize to diverse document conditions, particularly in real-world applications.

2.3 Deep Learning Methods for TAR

Recent advances in deep learning have led to significant improvements in textual attribute recognition. Convolutional neural networks (CNNs) have been widely used for extracting visual features from word images. For example, DeepFont [9] leverages CNN-based architectures to learn discriminative representations for font recognition, which can be extended to attribute prediction tasks.

Multi-task learning approaches [10] have also been explored to jointly predict multiple textual attributes along with related properties such as resolution or print quality. These methods demonstrate the benefits of shared representation learning across related tasks. However, they typically operate on isolated word images and do not incorporate spatial context, limiting their ability to distinguish between visually similar attributes.

More recent approaches incorporate detection-based frameworks. The TaCo framework [11], for instance, combines contrastive learning with a detection architecture to jointly localize and classify text attributes. While this allows the model to capture context at the line level, the overall pipeline is complex and may not generalize well across diverse document types.

Transformer-based methods have also been explored for TAR. CONSENT [12] utilizes a transformer architecture to model relationships between neighboring words, demonstrating that contextual information can improve attribute recognition. However, this approach focuses primarily on a single attribute (bold) and does not address the full range of attributes encountered in real-world documents.

Overall, while deep learning methods have advanced the state of the art, most existing approaches either operate on isolated word images or are limited in scope, focusing on a subset of attributes or specific document settings.

2.4 Context-Aware and Transformer-based Approaches

A growing body of work highlights the importance of contextual information in document understanding tasks. In TAR, the attribute of a word often depends on its surrounding visual context rather than on the word image alone. For example, distinguishing underline from table borders or bold from naturally thick text requires comparison with neighboring elements.

Transformer-based architectures are particularly well-suited for modeling such relationships due to their ability to capture long-range dependencies. Recent work [12] has demonstrated that incorporating

neighborhood context can improve attribute prediction performance. However, existing methods face a trade-off between contextual richness and computational efficiency. Processing entire documents using transformers is computationally expensive due to the quadratic complexity of attention, especially when documents contain hundreds of words.

Some approaches attempt to address this by restricting context to line-level or region-level groupings. While this reduces computational cost, it may still include irrelevant information or fail to capture the most informative local neighborhood.

This thesis builds on these insights by proposing a *context window-based formulation*, where each word is processed together with a compact set of its nearest neighbors. This approach aims to retain essential contextual cues while maintaining computational efficiency, enabling scalable and effective TAR in real-world document settings.

2.5 Multilingual and Real-World Document Challenges

A key challenge in textual attribute recognition is the diversity of real-world documents. Documents vary widely across languages, scripts, fonts, layouts, and acquisition conditions. Multilingual settings introduce additional complexity due to differences in character shapes, writing systems, and typographic conventions.

Most existing TAR datasets and methods are limited to specific languages or controlled environments, and therefore do not adequately capture this variability. In practice, document images may contain noise, distortions, low resolution, skew, or complex backgrounds, all of which affect attribute recognition performance.

Another challenge is class imbalance. Certain attributes, such as bold, may be more frequent than others, while attributes like underline or strikeout may appear less often. This imbalance can bias models toward dominant classes and reduce performance on rare attributes.

These challenges highlight the need for large-scale, diverse datasets and robust modeling strategies that can generalize across multilingual and multidomain settings.

2.6 Summary of This Thesis

Prior work demonstrates significant progress in textual attribute recognition, particularly with the adoption of deep learning and transformer-based models. However, several key limitations remain. Classical and OCR-based approaches lack adaptability and struggle in noisy, real-world settings. Deep learning methods improve performance but often operate on isolated word images or focus on limited attribute sets. Context-aware approaches show promise but face challenges in balancing contextual modeling with computational efficiency. Furthermore, existing datasets do not adequately capture multilingual and multidomain variability.

This thesis addresses these limitations by introducing a context-aware formulation of TAR, a Transformer-based architecture that operates on compact neighborhood context, and a large-scale multilingual dataset designed to reflect real-world document diversity. By integrating efficient context modeling with spatially aware representations, the proposed approach advances textual attribute recognition toward a more robust and scalable solution for document understanding.

Chapter 3

TexTAR

Textual Attribute Recognition requires the model to determine not only the visual appearance of an individual word, but also the contextual cues that disambiguate that appearance within the document. A word that appears bold in isolation may not actually be bold relative to its neighboring words, and a line beneath a word may correspond either to an underline attribute or to a structural element such as a table border. For this reason, recognizing textual attributes from isolated word crops is often unreliable.

TexTAR addresses this challenge through a two-stage framework that combines efficient context selection with context-aware attribute prediction. In the first stage, the method constructs compact local neighborhoods, called *Context Windows* (CWs), around selected words in the document. These context windows are designed to capture the most relevant nearby words while avoiding the computational cost of processing the full document jointly. In the second stage, each context window is processed by a Transformer-based model that predicts textual attributes for all words within that window. Since the same word may appear in multiple overlapping context windows, the final word-level prediction is obtained by aggregating logits across all its occurrences.

The overall design is motivated by two requirements. First, the method must incorporate neighborhood context, because textual attributes are often visually relative and layout-dependent. Second, it must remain computationally efficient, since real-world documents often contain hundreds of words and cannot be processed naively as a single sequence. TexTAR satisfies both requirements through a context-window formulation, a spatially aware Transformer architecture, and a postprocessing module that consolidates predictions across overlapping local contexts.

3.1 Overview

Given a document image as input, the objective of TexTAR is to predict the textual attributes associated with each word in the document. The framework proceeds in two stages.

In the first stage, word bounding boxes are obtained using a text detector. These word boxes are then processed through a novel data selection pipeline that constructs a set of context windows. Each context window contains a fixed number of spatially related words, thereby providing compact neighborhood context for attribute recognition.

In the second stage, each context window is passed to the proposed TexTAR model, which predicts the textual attributes of all words in that window. Since a word may belong to multiple overlapping windows, the model produces multiple predictions for the same word. These are combined in a final aggregation step using a context averaging module, which produces the final attribute label for each word.

3.2 Problem Formulation

Let $I \in \mathbb{R}^{h \times w \times 3}$ denote a document image, where h and w are its height and width, respectively. Suppose the document contains a set of detected word instances

$$W = \{w_1, w_2, \ldots, w_N\},$$

where each w_i corresponds to a word image extracted from a word bounding box.

The goal of Textual Attribute Recognition is to predict, for each word w_i, two attribute labels corresponding to two attribute groups:

$$\texttt{TexTAR}(w_i) \rightarrow (T_1, T_2),$$

where

$$T_1 \in \{\texttt{normal}, \texttt{bold}, \texttt{italic}, \texttt{bold\&italic}\}$$

and

$$T_2 \in \{\texttt{normal}, \texttt{underline}, \texttt{strikeout}, \texttt{underline\&strikeout}\}.$$

Instead of predicting these labels from each word in isolation, TexTAR predicts attributes over local context windows. Let a context window be defined as

$$CW = \{w_1, w_2, \ldots, w_S\},$$

where S denotes the fixed number of words in the window. Given such a context window, the model predicts a pair of attribute labels for every word in that window:

$$\texttt{TexTAR}(CW) \rightarrow \{(T_1^{(i)}, T_2^{(i)})\}_{i=1}^{S}.$$

Since the same word may appear in multiple context windows, the final prediction for each word is obtained by aggregating outputs from all windows in which it appears.

3.3 Stage I: Data Selection Pipeline

A central challenge in TAR is that the attribute of a word often cannot be determined solely from its isolated appearance. For instance, a word that appears underlined may actually lie above a table

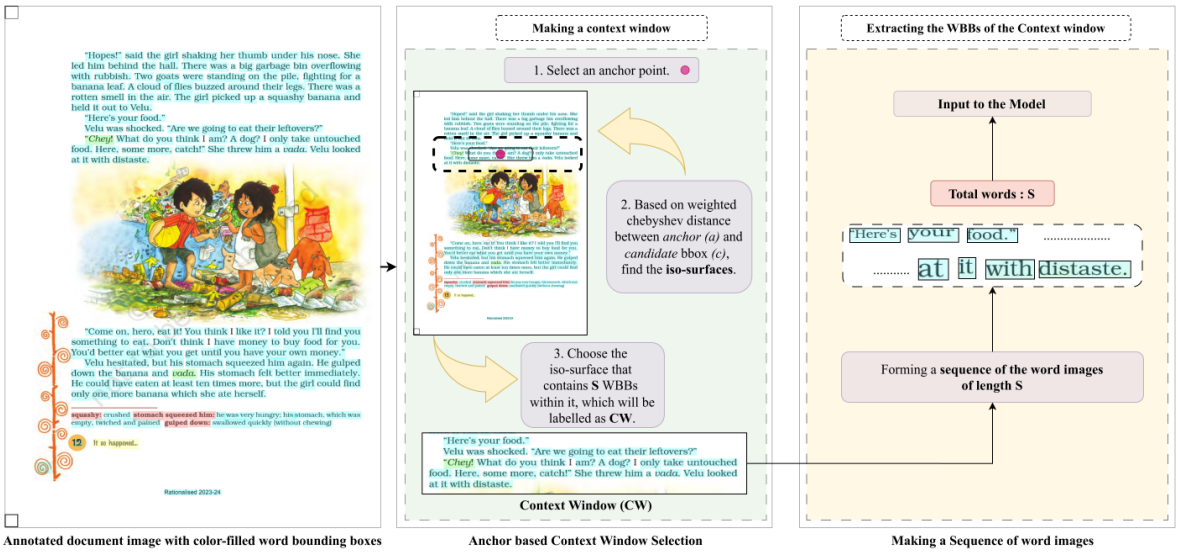


Figure 3.1 The proposed data selection pipeline: (a) Selection of anchor point. (b) Using the distance metric to get the nearest WBBs to the anchor point forming a `CW`. Here, the `iso-surface` is a set of WBBs equidistant from the anchor using the distance metric. Note that `CW` is a iso-surface containing S words. (c) Forming a sequence input of words in `CW` for TAR model.

separator line, and a word that appears bold may only be distinguishable as such relative to surrounding words. This makes neighborhood context essential for reliable attribute prediction.

At the same time, incorporating all words in a document as context is computationally impractical. Real documents often contain between 300 and 500 words, and Transformer-based self-attention scales quadratically with sequence length. Processing the full set of words jointly would therefore incur substantial computational overhead during both training and inference.

To address this, TexTAR introduces an efficient data selection pipeline that extracts only the most relevant local context for each prediction. Rather than constructing a global document representation, the method forms compact *Context Windows* around selected anchor words. These windows are designed to preserve neighborhood style and layout cues while keeping the sequence length fixed and manageable.

3.4 Context Window Construction

A Word Bounding Box (WBB) is defined as a rectangular region enclosing a single word in the document image. To incorporate local context, we define a *Context Window* (CW) around an *anchor word*. The anchor word is the word whose neighborhood is being constructed, while the remaining words inside the window are selected as contextual candidates.

For each anchor word, the context window contains the $S-1$ nearest neighboring WBBs according to a weighted Chebyshev distance:

$$D_{\text{Chebyshev}}(c,a) = \max\left(k|c_x - a_x|,\ m|c_y - a_y|\right),$$

where (c_x, c_y) and (a_x, a_y) denote the centers of the candidate and anchor WBBs, respectively, and k and m are constant weights.

This distance is chosen so that the resulting context window is rectangular, with greater horizontal extent than vertical extent. Such a design is appropriate for document images because text is typically organized along horizontal reading lines, and the most informative neighboring words often occur nearby within the same line or adjacent lines.

Each context window therefore consists of S words in total: the anchor word and its $S-1$ nearest contextual neighbors. The word images in the window are resized to fixed dimensions while preserving a predetermined aspect ratio, allowing them to be processed uniformly by the downstream model.

3.5 Sequential Context Windows

A naive approach would generate one context window for every word in the document. However, since a typical document may contain 300–500 words, this would create a large number of overlapping windows and incur substantial storage overhead.

To reduce redundancy while preserving coverage, TexTAR introduces the notion of *Sequential Context Windows*. The goal is to ensure that all words in a document are included in one or more context windows, without forcing every word to act as an anchor.

Initially, all word bounding boxes in the document are marked as unvisited. In each iteration, an anchor word is selected randomly from the set of unvisited words. A new context window is then formed by collecting the $S-1$ nearest neighboring words around that anchor. Once the context window is constructed, all words inside it are marked as visited. These words are therefore excluded from being selected as anchor words in subsequent iterations.

This process continues until all words in the document have been marked as visited. In effect, the algorithm produces a sequence of overlapping context windows that together cover the entire document while keeping the number of windows substantially smaller than the total number of words. This design strikes a balance between contextual coverage and storage efficiency.

3.6 Stage II: TexTAR Architecture

Given a context window as input, the objective of TexTAR is to predict the textual attributes of all words in that window. The architecture consists of four main components: a Feature Extractor Network (FEN), Transformer Encoder blocks (TEnc), RoPE-Mixed Attention Blocks (RoPE-MixAB), and dual

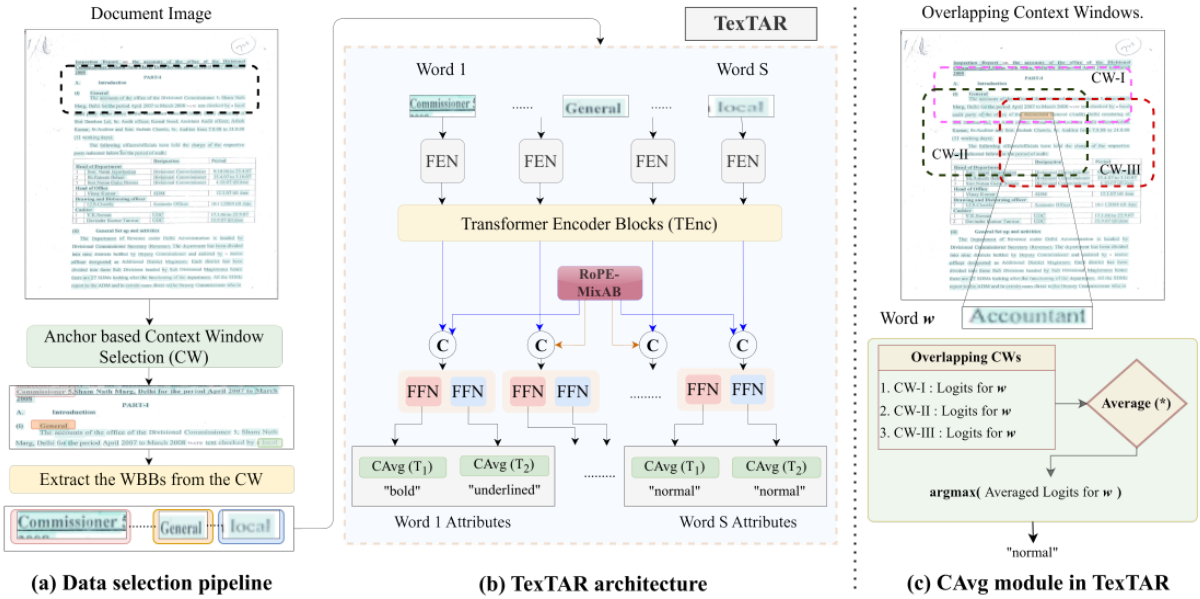


Figure 3.2 Overview of the method: (a) Describes the extraction of context window CW from the document. (b) Architecture of TexTAR, which takes in the sequence of word images (CW) and outputs the text attributes for each word in CW. (c) It decribes our postprocessing module i.e. `CAvg` for both T_1 and T_2.

classification heads. These are followed by a postprocessing module called `CAvg`, which aggregates predictions across overlapping windows.

3.7 Feature Extractor Network

Each word image in the context window is first passed through a convolutional neural network that acts as a feature extractor. This module, referred to as the *Feature Extractor Network* (FEN), produces an embedding for each word that captures its basic visual characteristics, such as stroke thickness, orientation, and local texture patterns.

For a context window

$$CW = \{w_1, w_2, \ldots, w_S\},$$

the resulting set of feature embeddings is

$$CW_{\text{emb}} = \{\texttt{FEN}(w_i) : i = 1, \ldots, S\}.$$

These embeddings serve as the input representation for subsequent contextual reasoning.

3.8 Transformer Context Encoder

The word-level embeddings produced by the FEN are passed through Transformer Encoder blocks, denoted by `TEnc`. The purpose of this module is to allow each word representation to attend to other words in the same context window, thereby incorporating neighborhood context into the prediction process.

This contextual reasoning is essential for disambiguating difficult cases. For example, the model may infer that a line beneath multiple neighboring words is more likely to be part of a table structure than an underline attribute, or that the visual prominence of a word depends on its contrast with adjacent words. In this way, the Transformer Encoder enables the model to move beyond isolated appearance and instead reason over the local structure of the document.

Formally, the contextualized embeddings are given by

$$T_{\text{emb}} = \{\texttt{TEnc}(CW^{(i)}_{\text{emb}}) : i = 1, \ldots, S\}.$$

Notably, absolute positional embeddings are not added directly to the input sequence, as this was found empirically to reduce performance for certain attributes.

3.9 RoPE-Mixed Attention Blocks

Rotary Positional Embedding (RoPE) is a positional encoding scheme that encodes position by applying a rotation to the query and key vectors within the self-attention mechanism. The rotation angle is proportional to the token's position, and as a result, the attention score between any two tokens naturally captures their relative positional distance rather than their absolute positions in the sequence. This makes RoPE more flexible and generalizable than conventional approaches such as sinusoidal absolute positional embeddings (APE) or learnable positional embeddings (LPE), which encode position as a fixed additive signal and do not inherently model relative relationships within the attention operation. Since document layout is two-dimensional, TexTAR adopts *RoPE-Mixed*, which extends RoPE to 2D by independently encoding the horizontal and vertical spatial coordinates of each word bounding box.

Although contextual modeling is important, TAR also requires sensitivity to spatial arrangement. In documents, the relative position of neighboring words often affects attribute interpretation. For instance, whether a horizontal stroke is an underline may depend on its spatial relation to nearby words and lines.

To incorporate spatial information, TexTAR uses additional attention blocks equipped with *RoPE-Mixed* positional embeddings, referred to as `RoPE-MixAB`. Unlike conventional positional embeddings that are simply added to token representations, this mechanism encodes positional relationships within the attention computation itself.

RoPE-Mixed is well suited to the present setting because it supports positional awareness in continuous two-dimensional space and can effectively model spatial relationships among word bounding

boxes. This enables the model to reason not only about which words are nearby, but also about how they are arranged in the document layout.

The contextual embeddings from `TEnc` are therefore further processed through RoPE-based attention:

$$T_{\text{rope}} = \{\texttt{RoPE-MixAB}(T_{\text{emb}}^{(i)}) : i = 1, \ldots, S\}.$$

To preserve both contextual and spatial information, these position-enriched embeddings are concatenated with the original Transformer outputs:

$$T_{\text{final}} = \{T_{\text{emb}}^{(i)} \copyright T_{\text{rope}}^{(i)} : i = 1, \ldots, S\},$$

where © denotes per-token concatenation.

3.10 Dual Classification Heads

The final token representations are passed to two separate feed-forward classification heads. These heads correspond to two groups of attributes:

- T_1, which captures attributes affecting text emphasis and appearance, namely normal, bold, italic, and bold&italic;
- T_2, which captures attributes defined by explicit visual markings, namely normal, underline, strikeout, and underline&strikeout.

This dual-head design is motivated by the observation that these two groups reflect different visual phenomena. The first group changes the appearance of the text strokes themselves, while the second introduces additional graphical markings relative to the text. Separating the prediction into two heads allows the model to specialize for these different attribute families and was found to perform better than using a single joint classification head.

3.11 Context Averaging Module

Because the constructed context windows overlap, the same word may be processed multiple times under slightly different local neighborhoods. Rather than discarding these multiple predictions, TexTAR uses them as a source of robustness.

The logits produced for each occurrence of a word are collected across all context windows in which that word appears. These logits are then averaged separately for the two attribute groups through a postprocessing module called `CAvg`. The final predicted class for each group is obtained by taking the maximum averaged logit.

Thus, for each word, `CAvg` consolidates multiple context-dependent predictions into a single final label. This aggregation improves stability and reduces the impact of any one suboptimal local window, while still preserving the benefits of contextual reasoning.

3.12 Discussion

The design of TexTAR reflects a broader view of textual attribute recognition as a context-aware document understanding problem. The data selection pipeline ensures that the model sees the most relevant nearby words without incurring the cost of full-document processing. The Transformer-based architecture enables contextual reasoning across these local neighborhoods, while the RoPE-based attention blocks inject spatial layout awareness. Finally, the dual-head prediction and context averaging modules ensure that the output remains both semantically structured and robust across overlapping local contexts.

Taken together, these components allow TexTAR to recognize textual attributes more reliably than methods based on isolated word crops, while remaining computationally practical for real-world multilingual and multidomain documents.

Chapter 4

MMTAD Dataset

Existing approaches to textual attribute recognition are limited not only by modeling choices, but also by the lack of large-scale, realistic, and diverse benchmark datasets. Most prior resources for textual attribute analysis are restricted in one or more of the following ways: they focus on a narrow set of attributes, cover only a single language or script, are collected under relatively clean and controlled conditions, or do not reflect the visual diversity of real-world document images. As a result, they do not adequately capture the challenges that arise in practical multilingual and multidomain settings, where textual attributes must be recognized under noise, layout variation, and script diversity.

In real document images, textual attributes are affected by many factors beyond the local appearance of a word. These include illumination changes, low-resolution scans, background texture, distortions introduced during acquisition, variations in font style and typography, and differences across scripts and languages. Furthermore, attributes such as underline and strikeout are often visually entangled with surrounding layout structures such as table borders, separators, or annotations. A dataset intended to support meaningful progress in Textual Attribute Recognition must therefore go beyond clean, monolingual, or narrowly curated settings and instead represent the variability of real-world documents.

To address this need, this thesis introduces **MMTAD**, the *Multilingual and Multidomain Textual Attribute Documents* dataset. MMTAD is specifically designed for word-level recognition of multiple textual attributes in realistic document images. It provides large-scale annotation across diverse document domains, languages, and visual conditions, thereby enabling the study of TAR in a setting that more closely reflects practical deployment scenarios.

4.1 Motivation and Need for a New Dataset

A central challenge in TAR is that textual attributes are relatively sparse and highly imbalanced in naturally occurring documents. Most words in a page are unadorned, while only a small fraction exhibit attributes such as bold, italic, underline, or strikeout. Even among attributed words, the frequency distribution is uneven: some attributes such as bold appear much more often than others, while italic, underline, and strikeout are comparatively rare. This makes it difficult to train robust models unless the dataset is sufficiently large and diverse.

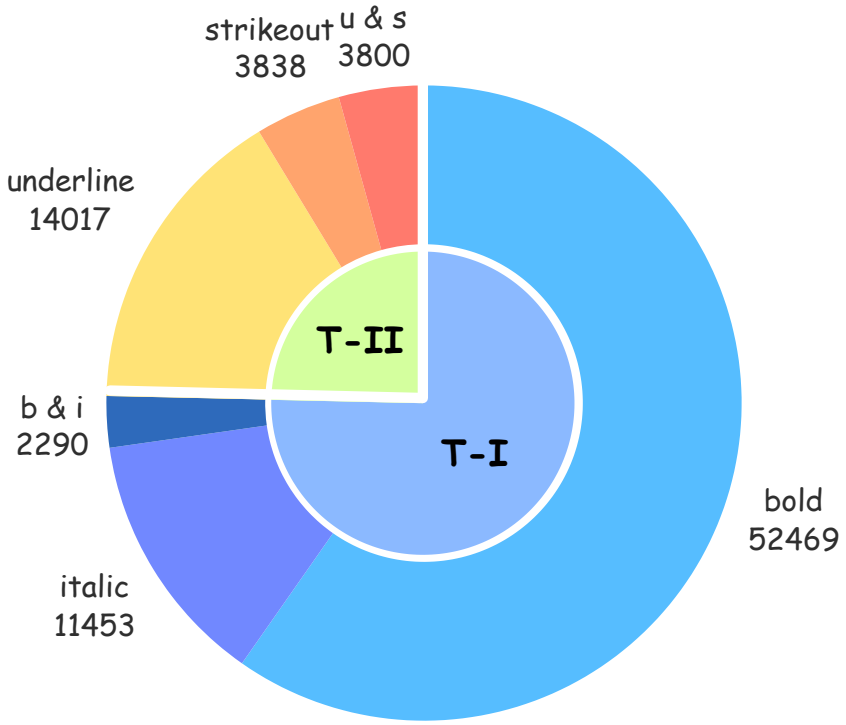


Figure 4.1 Distribution of annotated textual attributes in the MMTAD dataset.

Another important limitation of existing resources is their lack of multilingual and multidomain coverage. Document images encountered in practice span multiple languages, writing systems, layouts, and stylistic conventions. The same visual cue may manifest differently across scripts, and the same attribute may appear differently depending on domain, font, or document quality. Datasets confined to a narrow language or domain do not expose models to this variability and therefore limit generalization.

This thesis is motivated by the view that progress in TAR requires a dataset that simultaneously captures three properties:

1. **attribute diversity**, including multiple textual attributes and their combinations;
2. **language and script diversity**, so that models are not confined to a single writing system; and
3. **domain realism**, including noisy, scanned, layout-rich, and visually heterogeneous document images.

MMTAD is constructed with these requirements in mind.

4.2 Dataset Overview

MMTAD consists of 1623 real multilingual and multidomain document images collected from a real-world corpus. These documents exhibit substantial variability in illumination, layout, font usage,

background texture, color, scanning artifacts, resolution, and other common distortions encountered in document processing pipelines. Unlike synthetic or tightly controlled datasets, MMTAD is designed to reflect the complexity of realistic document imagery.

The dataset contains a total of 1,117,716 word-level annotations for textual attributes, making it a large-scale benchmark for multilingual and multidomain textual attribute recognition. The documents cover a range of practical document types, including notices, circulars, legislative documents, land records, textbooks, and notary documents. Each document contains, on average, approximately 300–500 annotated word bounding boxes, providing rich and dense supervision at the word level.

In addition to broad annotation coverage, MMTAD contains 87,867 annotated words belonging to non-`normal` attribute categories. This ensures that the dataset not only captures overall word-level structure, but also provides substantial supervision for the attribute classes of central interest.

4.3 Attribute Taxonomy and Annotation Design

Each word in MMTAD is annotated using a two-group attribute taxonomy. This grouping is motivated by the observation that different textual attributes affect the visual appearance of a word in different ways.

The first group, denoted by T_1, includes attributes that primarily alter the appearance of the text strokes themselves:

$$T_1 \in \{\texttt{normal}, \texttt{bold}, \texttt{italic}, \texttt{bold\&italic}\}.$$

The second group, denoted by T_2, includes attributes that introduce explicit visual markings relative to the text:

$$T_2 \in \{\texttt{normal}, \texttt{underline}, \texttt{strikeout}, \texttt{underline\&strikeout}\}.$$

This distinction is important because the two groups correspond to different visual phenomena. Attributes such as bold and italic are primarily expressed through changes to the text itself, including stroke thickness and slant. By contrast, underline and strikeout are characterized by additional graphical structures that may appear independently of the word's internal stroke pattern. Grouping the annotations in this way allows the modeling framework to reflect the underlying visual differences more naturally. In both groups, words that do not exhibit the relevant attribute are labeled as `normal`. This annotation design supports structured prediction of multiple attribute types while preserving interpretability and compatibility with the model architecture proposed in this thesis.

4.4 Multilingual and Multidomain Coverage

A major strength of MMTAD is its multilingual composition. In addition to English and Spanish, the dataset includes several South Asian languages, enabling evaluation in a linguistically diverse set-

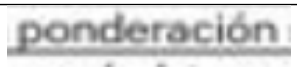
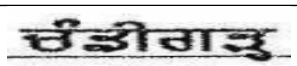
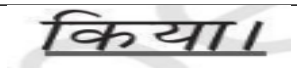
Image	T_1 group		T_2 group	
	`bold`	`italic`	`underline`	`strikeout`
	✗	✗	✗	✗
	✗	✗	✓	✗
	✓	✗	✓	✗
	✗	✗	✓	✓
	✓	✓	✗	✗
	✗	✓	✓	✗
	✓	✗	✗	✓

Table 4.1 Examples of textual attribute combinations in the `MMTAD` dataset. The table shows a subset of possible combinations across the T_1 and T_2 groups.

ting. The language distribution includes Hindi (67.35%), Telugu (8.23%), Marathi (7.95%), Punjabi (5.90%), Bengali (5.35%), and other languages including Gujarati, Tamil, and Spanish (5.22%), across both training and test splits.

This multilingual coverage is important because textual attributes interact with script-specific visual patterns. Character thickness, slant, spacing, and attribute manifestation may differ considerably across writing systems. A dataset that includes only one language cannot expose models to these variations. By contrast, MMTAD enables the study of attribute recognition under multilingual visual diversity.

The dataset is also multidomain in nature. It includes document categories such as notices, circulars, legislative documents, land records, textbooks, and notary documents. These domains differ not only in content but also in typography, formatting practices, density of annotations, and structural organization. This diversity is essential for evaluating whether a TAR model can generalize beyond a single document style.

4.5 Class Imbalance and Context-Aware Augmentation

As in most naturally occurring document corpora, the distribution of attributes in MMTAD is highly imbalanced. Words labeled as `normal` are far more frequent than words with explicit formatting attributes. Moreover, within the non-normal categories, attributes such as bold occur much more frequently than italic, underline, or strikeout.

This imbalance poses a challenge for learning, as models may become biased toward dominant classes and fail to adequately recognize rare but important attribute types. To address this issue, the dataset is augmented in a context-aware manner to increase the number of samples for underrepresented attributes, particularly italic, underline, and strikeout.

For italic augmentation, shear transformations are applied to word images in order to simulate realistic slant patterns. For underline and strikeout augmentation, noisy context-aware transformations are introduced so that the added markings resemble natural occurrences in real document settings rather than artificial overlays. The goal is not merely to increase class counts, but to preserve realism and maintain compatibility with surrounding visual context.

These augmentation strategies improve dataset diversity while remaining faithful to the appearance of real-world document distortions. In this way, MMTAD supports both large-scale supervision and a more balanced learning signal for rare attribute categories.

4.6 Discussion

The design of MMTAD reflects a broader position of this thesis: meaningful progress in textual attribute recognition requires not only stronger models, but also better data. A realistic TAR benchmark must capture the sparsity, imbalance, multilinguality, and domain diversity of real-world documents. It must also distinguish between attribute types that arise from different visual mechanisms, such as text-style changes versus independent graphical markings.

MMTAD addresses these requirements through large-scale word-level annotation, multilingual and multidomain coverage, and context-aware augmentation for rare attribute classes. As such, it provides a substantially richer benchmark for TAR than prior narrowly scoped datasets and forms the data foundation for the TexTAR framework proposed in this thesis.

Taken together, MMTAD fills an important gap in textual attribute recognition research. Existing datasets do not adequately reflect the multilingual, noisy, and structurally diverse conditions under which TAR must operate in practice. In contrast, MMTAD provides a large-scale benchmark with rich word-level supervision across multiple languages, domains, and attribute combinations.

This thesis uses MMTAD not only as a benchmark for evaluation, but as a means of reframing TAR itself as a realistic document understanding problem rather than a narrow style-classification task. By supporting the study of textual attributes under real-world conditions, MMTAD contributes toward more robust, context-aware, and practically useful document intelligence systems.

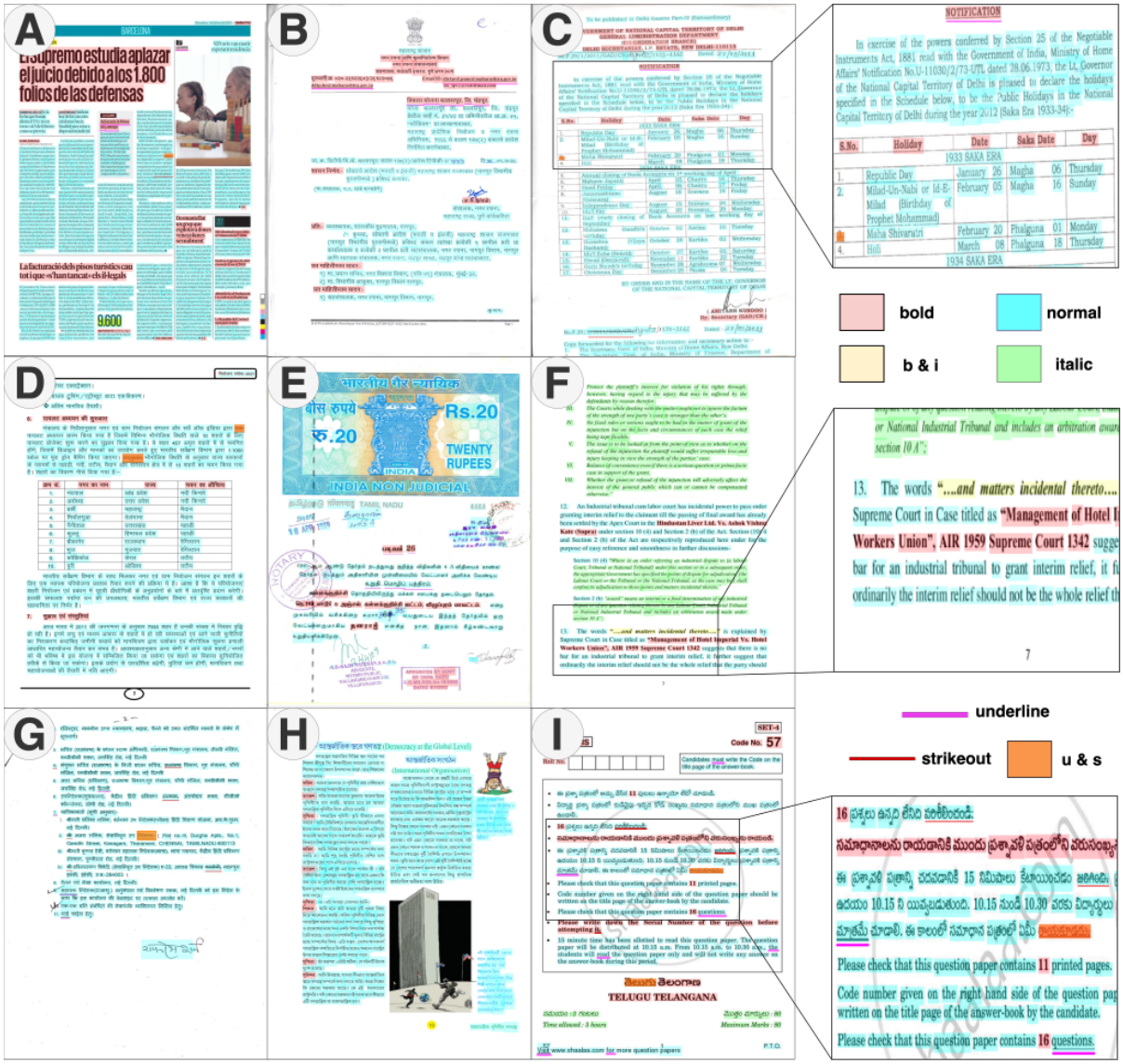


Figure 4.2 Our MMTAD Dataset: Examples of annotated documents from our multilingual and multidomain dataset. Note that u & s stands for `underline & strikeout` and b & i stands for `bold & italic` attribute categories.

Chapter 5

Experiments

This chapter presents the experimental setup and evaluation protocol used to assess TexTAR. We first describe the dataset split, implementation details, and training strategy, followed by the baseline methods used for comparison. We then explain the evaluation metric adopted for textual attribute recognition and summarize the main quantitative and qualitative findings. Finally, we present ablation studies that analyze the contribution of the principal design choices in the proposed framework.

5.1 MMTAD Split

All experiments are conducted on the proposed MMTAD dataset. The dataset is divided into 1005 training images, 137 validation images, and 481 test images. From these document images, we extract context windows using the data selection pipeline described in Chapter 3. These context windows form the input instances for the TexTAR model.

5.2 Implementation Details

Each word image within a context window is resized to a fixed spatial resolution of 128×96 pixels while maintaining an aspect ratio of $3 : 4$. Each context window contains a fixed number of words, with sequence size set to $S = 125$.

To reduce overfitting and improve robustness, we apply several data augmentations during training, including random rotation, Gaussian blur, color jitter, horizontal flip, and random affine transformations. In addition, dropout is used within the dual classification heads.

We train the model using the Adam optimizer with a learning rate of 10^{-4}. A weighted cross-entropy loss is employed, with empirically chosen class weights of 0.25 and 0.75 for the T_1 and T_2 groups, respectively. The model is trained for 100 epochs. We set the softmax temperature to 0.25 based on empirical validation.

Methods			T_1 group			T_2 group			**Average**
		normal	bold	italic	b & i	underline	strikeout	u & s	
Baselines	ResNet-18 [13]	0.97	0.75	0.88	0.77	0.68	0.97	0.99	0.86
	ResNet-50 [13]	0.97	0.74	0.89	0.69	0.73	0.98	0.99	0.86
	ResNeXt-101 [14]	0.97	0.77	0.91	0.74	0.78	0.99	0.99	0.88
	EfficientNet-b4 [15]	0.97	0.75	0.90	0.62	0.75	0.98	0.99	0.85
Variants	DeepFont [9]	0.97	0.72	0.80	0.44	0.64	0.93	0.98	0.78
	DropRegion† [16]	0.98	0.77	0.90	0.61	0.75	0.97	0.99	0.85
	MTL [10]	0.97	0.75	0.89	0.64	0.70	0.97	0.99	0.84
	TaCo† [11]	0.97	0.79	0.90	0.60	0.78	0.86	0.89	0.83
	CONSENT† [12]	0.98	0.86	0.93	0.84	0.81	0.96	0.98	0.91
	TexTAR	**0.99**	**0.92**	**0.95**	**0.90**	**0.87**	**0.99**	**0.99**	**0.94**

Table 5.1 Comparison with state-of-the-art approaches for `TAR`: (Section 5) Note that `u & s` is the short form for `underline & strikeout` attribute and `b & i` is the short form for `bold & italic` attribute. † Implemented by us due to unavailability of open-source code. All the above results are reported using F_1 metric.

5.3 Training Strategy

To stabilize learning and ensure effective integration of positional information, TexTAR is trained using a two-stage strategy.

In the first stage, we train a base architecture consisting of the Feature Extractor Network (`FEN`), Transformer Encoder blocks (`TEnc`), and the classification heads. This stage allows the model to learn strong visual and contextual representations from the input word images.

In the second stage, we freeze the encoder components learned in the first stage, namely `FEN` and `TEnc`, and introduce the `RoPE-MixAB` module. The outputs of `RoPE-MixAB` are concatenated with the pretrained contextual features, and the resulting architecture is fine-tuned while keeping the newly introduced positional module and classification heads trainable. This staged training process helps preserve the learned feature representations while allowing the model to incorporate spatial positional reasoning in a controlled manner.

Overall, this two-stage training strategy improves optimization stability and enables more effective integration of positional context into attribute prediction.

5.4 Baselines

We compare TexTAR against a diverse set of baseline and recent textual attribute recognition methods.

Single-word visual baselines. We include commonly used CNN feature extractors such as ResNet-18, ResNet-50, ResNeXt-101, and EfficientNet-B4. These models operate on isolated word images and therefore serve as strong baselines for attribute recognition without explicit contextual modeling.

Task-specific TAR baselines. We also compare against prior methods developed for font or textual attribute recognition, including DeepFont, DropRegion, MTL, TaCo, and CONSENT. These methods represent the most relevant prior work for TAR and span CNN-based, multi-task, detection-based, and transformer-based approaches.

To ensure fairness, all compared models are trained and evaluated on the same MMTAD data split. For methods whose official implementations were unavailable, we implement them following the published descriptions.

5.5 Evaluation Metric

We evaluate all methods using the F_1 score computed separately for each attribute class. Specifically, results are reported for the classes in the T_1 group (`normal`, `bold`, `italic`, `bold&italic`) and the T_2 group (`underline`, `strikeout`, `underline&strikeout`). We also report the average F_1 across attributes as an overall summary measure.

The use of per-class F_1 is important in this setting because the attribute distribution is highly imbalanced. Unlike simple accuracy, F_1 better reflects performance on minority classes such as italic, underline, and strikeout, which are central to the difficulty of the task.

5.6 Main Results

5.6.1 Comparison with Existing Approaches

The main comparison with prior approaches is presented in Table 5.1. TexTAR achieves the best overall performance, outperforming all baseline and recent state-of-the-art methods across nearly all attribute categories.

In particular, TexTAR attains strong improvements on visually subtle and context-dependent attributes such as `bold`, `underline`, and `bold&italic`. These gains are especially important because such attributes are precisely the ones that are difficult to recognize from isolated word crops alone. The final average F_1 of TexTAR is substantially higher than the competing methods, demonstrating the effectiveness of the proposed context-aware formulation.

5.6.2 Discussion of Comparative Results

The results reveal several important trends.

First, CNN-based baselines such as ResNet-18, ResNet-50, and EfficientNet-B4 perform reasonably on visually distinctive classes but struggle on attributes whose interpretation depends on neighborhood context. This is expected, since these methods operate only on individual word images and therefore cannot compare a word with its surrounding words or layout.

Figure 5.1 Qualitative Comparisions: Visualization of results for a subset of baselines and variants in comparision with TexTAR (ours).The red dotted circles highlight errors made by other models, while dark gray dotted circles indicate errors by ours, relative to the ground truth.

Second, earlier TAR-specific approaches such as DeepFont and DropRegion show limited performance in the multilingual and multidomain setting of MMTAD. Their representations are not sufficiently robust to the noise, script diversity, and structural complexity present in real-world documents.

Third, recent methods such as CONSENT improve performance by incorporating transformer-style contextual modeling, but remain limited in scope. In particular, CONSENT focuses primarily on a narrower attribute setting, while TaCo introduces additional complexity by casting TAR as a joint detection and classification problem. In our experiments, this added dependency on word and line detection introduces bottlenecks that reduce robustness on MMTAD.

By contrast, TexTAR achieves superior performance by explicitly modeling compact local context, incorporating spatial reasoning through RoPE-Mix attention, and structuring the prediction problem through dual attribute heads. These results support the central thesis claim that textual attributes should be recognized as context-aware properties rather than isolated visual labels.

5.6.3 Qualitative Analysis

We further compare TexTAR qualitatively against representative baselines, as shown in Figure 5.1. The qualitative examples show that competing approaches often fail in context-sensitive cases, especially when a structural line is confused with underline, or when boldness must be inferred relative to neighboring words rather than from the isolated crop alone.

TexTAR produces more accurate predictions in such cases because it reasons jointly over local neighborhood context. The model is also more reliable on multilingual and noisy examples, where local appearance alone is insufficient for correct attribute recognition.

Additional qualitative examples for the ablation studies are shown in Figure 5.5. These visualizations further confirm that the full TexTAR model yields fewer errors than its ablated variants, particularly on attributes that depend strongly on the interaction between local appearance, surrounding context, and spatial positioning.

5.7 Ablation Studies

We perform a series of ablation experiments to evaluate the contribution of the major architectural and training components of TexTAR. The results are summarized in Table 5.2.

5.7.1 Effect of Dual Classification Heads

We first study the effect of replacing the dual-head design with a single classification head over the full Cartesian product of $T_1 \times T_2$ labels. This variant, denoted as `TexTAR-base`, performs consistently worse than the dual-head variant `TexTAR-base-DH`.

This result can be explained by the skewed distribution of cross-group attribute combinations in the training data. Some combinations appear rarely, which makes learning a single joint classifier difficult. By separating prediction into two heads, one for T_1 and one for T_2, the model benefits from better sample efficiency and cleaner gradient flow within each attribute family. The dual-head formulation therefore improves performance across both groups.

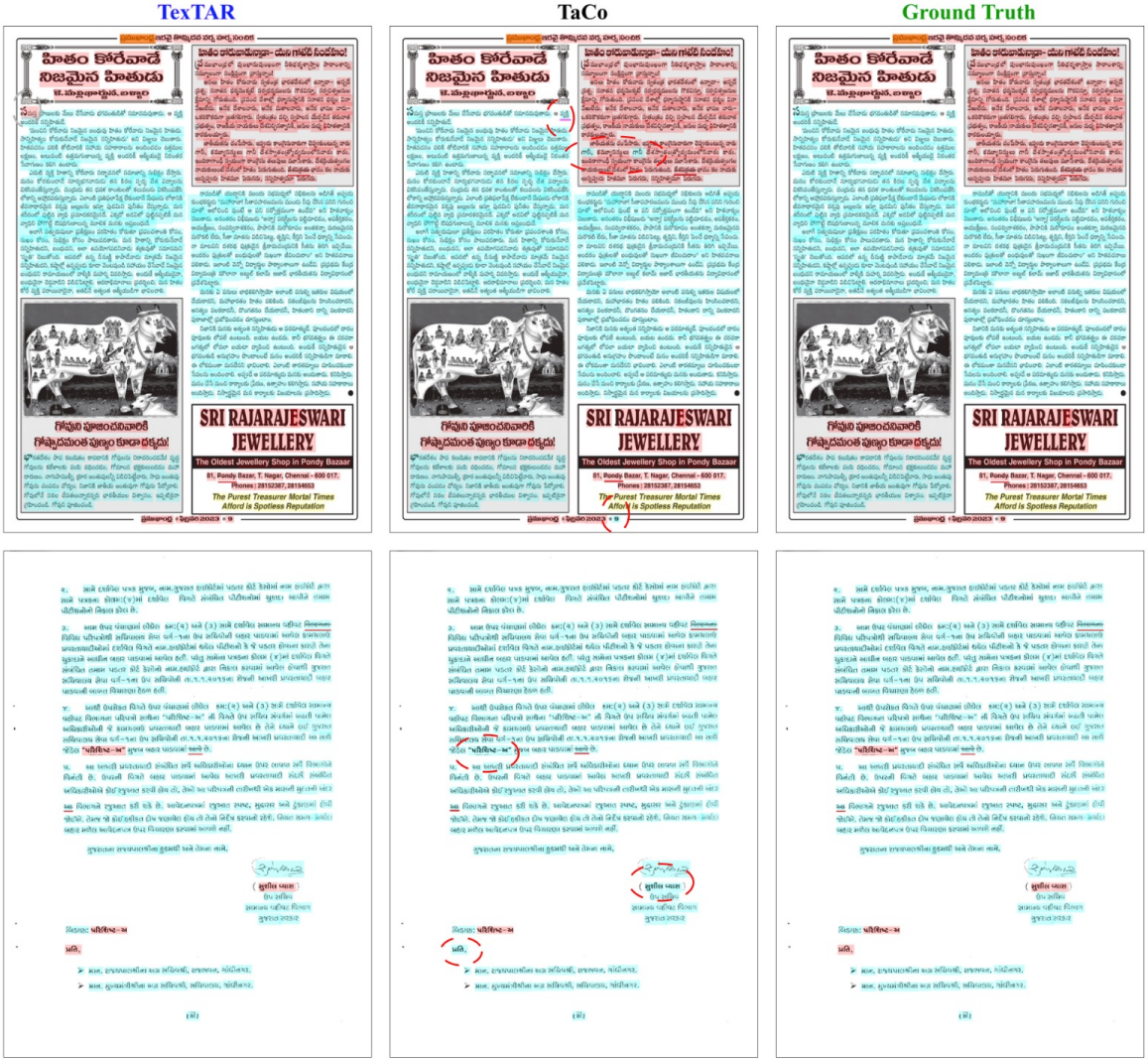


Figure 5.2 Qualitative Comparision of Taco and TexTAR: Visualization of results for a subset of baselines and variants in comparision with TexTAR (ours).The red dotted circles highlight errors made by models, relative to the ground truth.

5.7.2 Effect of RoPE-MixAB

We next analyze the role of positional encoding. We compare several positional embedding strategies, including Absolute Positional Embeddings (APE), Learnable Positional Embeddings (LPE), and RoPE-Mixed embeddings.

When trained end-to-end, the RoPE-Mixed variant performs better overall than the alternatives, but still shows some degradation on certain attributes such as underline. To address this, we adopt the two-stage training strategy described earlier, in which the base encoder is pretrained first and the positional module is introduced during fine-tuning.

Figure 5.3 Qualitative Comparision of Consent and TexTAR: Visualization of results for a subset of baselines and variants in comparision with TexTAR (ours).The red dotted circles highlight errors made by models, relative to the ground truth.

Under this setting, the final TexTAR model achieves the best qualitative and quantitative performance, outperforming both APE and LPE variants. These results suggest that RoPE-Mixed is the most effective way to introduce spatial positional information, provided that it is integrated in a controlled manner rather than through direct end-to-end optimization from scratch. This controlled integration helps stabilize training and ensures that spatial cues are incorporated without disrupting learned representations. Furthermore, it highlights the importance of carefully designing positional encoding strategies when dealing with complex document layouts and context-dependent attributes. Overall, these findings emphasize that both architectural choices and training strategies play a crucial role in effectively leveraging spatial information.

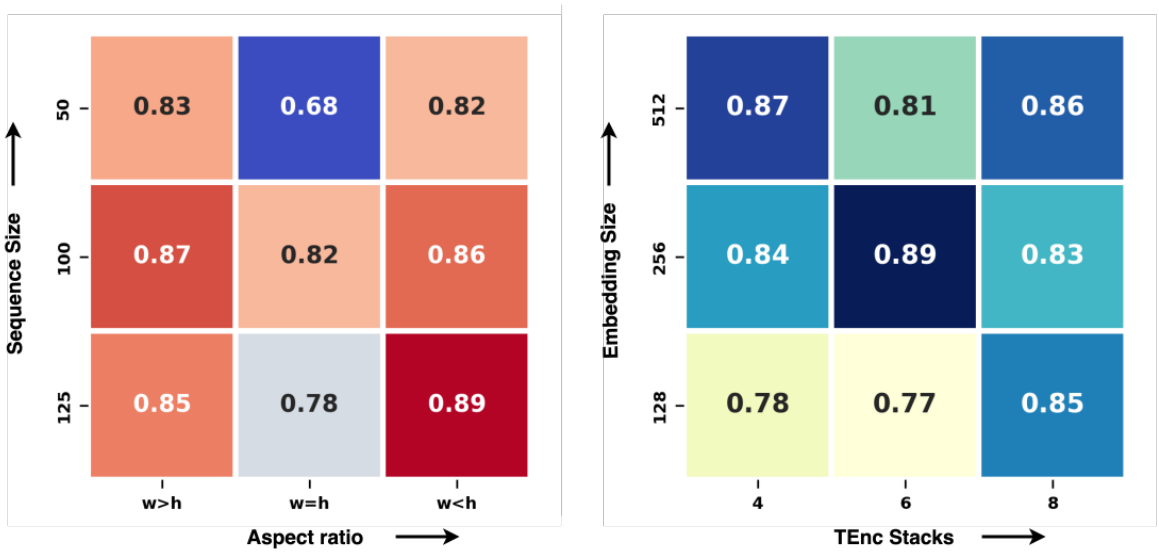


Figure 5.4 Ablation experiments for `TexTAR-base-DH` model: (left) The heatmap shows the F_1-scores obtained by varying Sequence size S and Aspect Ratio of the word image in CW. (right) The heatmap shows the F1-scores obtained by varying Embedding size for word representation and the number of stacks in `TEnc`.

5.7.3 Effect of Concatenation

We further study the importance of concatenating the pretrained contextual embeddings with the outputs of the RoPE-Mix attention blocks. Removing this concatenation, in the variant `TexTAR w/o ©`, leads to reduced performance, especially for `italic` and `bold&italic`.

This indicates that positional information alone is insufficient. If the model relies too heavily on the RoPE-based features, attribute-specific visual cues learned by the pretrained encoder may be weakened. Concatenation provides a more balanced representation by preserving both the learned contextual features and the newly introduced spatial information.

5.7.4 Effect of End-to-End Training

We also compare the proposed two-stage training strategy with a fully end-to-end version of the same architecture. The end-to-end variant performs substantially worse, particularly on difficult attribute classes.

This result supports the use of staged training. Freezing the pretrained encoder while fine-tuning the positional module allows the model to preserve its learned visual-contextual features and prevents them from being disrupted by the introduction of positional bias too early in training.

	Ablations on TexTAR-base		T_1 group			T_2 group		
		normal	bold	italic	b & i	underline	strikeout	u & s
I	TexTAR-base	0.98	0.85	0.93	0.74	0.79	0.97	0.99
	TexTAR-base-DH	0.99	0.90	0.94	0.86	0.87	0.98	0.99
II	TexTAR-APE-DH	0.98	0.89	0.92	0.72	0.87	0.99	0.99
	TexTAR-LPE-DH	0.99	0.90	0.93	0.84	0.84	0.98	0.99
	TexTAR-RoPE-Mix-DH	0.99	0.90	0.93	0.86	0.83	0.99	0.99
III	TexTAR-base-APE-DH	0.99	0.91	0.95	0.88	0.87	0.98	0.99
	TexTAR-base-LPE-DH	0.99	0.91	0.94	0.87	0.87	0.98	0.99
IV	TexTAR (E2E)	0.98	0.82	0.89	0.65	0.83	0.98	0.99
	TexTAR (w/o ©)	0.99	0.92	0.94	0.86	**0.88**	0.99	0.99
	TexTAR	**0.99**	**0.92**	**0.95**	**0.90**	0.87	**0.99**	**0.99**

Table 5.2 Ablative studies for TexTAR: (Section 5.7) Based on the notations in table, TexTAR is equivalent to `TexTAR-base-RoPE-Mix-DH`. Note that `u & s` is the short form for `underline & strikeout` attribute and `b & i` is the short form for `bold & italic` attribute. All the above results are reported using F_1 metric.

5.7.5 Effect of Augmentation

Finally, we examine the contribution of data augmentation. Removing augmentations leads to a consistent drop in performance across both attribute groups. In particular, bold drops from 0.92 to 0.87, italic from 0.95 to 0.93, and underline from 0.87 to 0.81.

These results show that augmentation is especially important for robust recognition of rare and visually subtle attribute classes. Since MMTAD reflects real-world visual variability, augmentation helps the model better generalize to changes in distortion, style, and acquisition conditions.

5.8 Analysis and Insights

The experimental results demonstrate that TexTAR achieves state-of-the-art performance on the MMTAD benchmark. The proposed framework consistently outperforms both generic CNN baselines and recent TAR-specific methods, particularly on context-dependent attributes. The ablation studies validate the importance of the dual-head design, RoPE-Mix attention, concatenation strategy, staged training, and augmentation. Taken together, these findings support the main claim of this thesis: effective textual attribute recognition requires context-aware and spatially grounded modeling rather than isolated word-level classification.

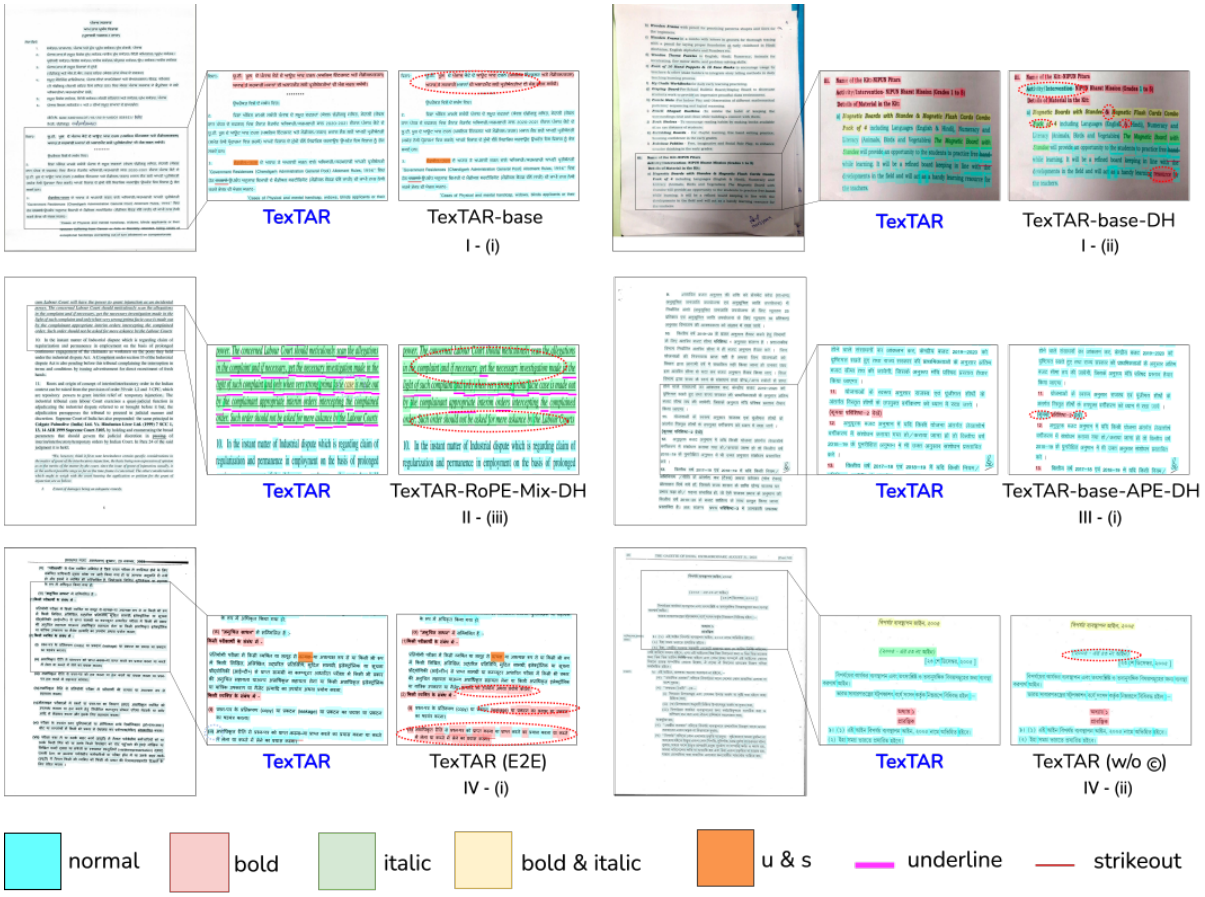


Figure 5.5 Qualitative examples for ablation studies: Examining the qualitative improvements for `TexTAR` over its ablation experiments. The red dotted circles highlight errors made by other ablations, while dark gray dotted circles indicate errors by ours, relative to the ground truth. Note that the numbering `i-(j)` (below ablation name) denotes the `j`th row of `i`th section in Table 5.2.

Furthermore, qualitative analysis reveal that TexTAR produces more coherent and semantically consistent predictions, especially in visually cluttered or ambiguous scenarios. The model demonstrates strong generalization across diverse layouts and textual variations, highlighting its robustness in real-world settings. These results reinforce the effectiveness of integrating contextual reasoning with spatial grounding for reliable and scalable textual attribute recognition. Overall, this work establishes a strong foundation for future research in presentation-aware document understanding.

Chapter 6

Textual Attribute Enrichment in REPLICA using TexTAR

6.1 Overview of REPLICA Pipeline

REPLICA is a document reconstruction framework designed to convert document images into high-fidelity, layout-aware HTML representations. Unlike traditional OCR pipelines that produce flat text sequences, REPLICA preserves structural hierarchy, spatial relationships, and visual semantics of the document. This enables downstream systems to operate on a representation that closely resembles the original document layout rather than a linearized approximation.

The REPLICA pipeline is composed of four major stages:

- **Segment:** This stage detects and groups layout regions such as paragraphs, tables, headers, and figures. It leverages a combination of models including DocLayout-YOLO-v10 [17] for flat layouts, IndicDLP-YOLO-v10 [18] for hierarchical grouping, and Hi-SAM [19] for fine-grained text segmentation. While DocLayout-YOLO and Hi-SAM provide precise region boundaries, IndicDLP-YOLO merges semantically related regions to produce coherent layout blocks.

- **Localize:** In this stage, fine-grained textual and visual elements are extracted. A Set-of-Mark prompting strategy is employed, where paragraph- and line-level regions obtained from Hi-SAM are color-indexed and passed to a vision-language model (VLM). The pipeline integrates multiple sources of information, including OCR outputs (Google OCR [20], DocTr [21]) for text extraction and TexTAR for textual attribute prediction. This stage enriches the representation with semantic tags, textual content, and stylistic attributes.

- **Assemble:** The extracted sub-components are merged into structured HTML representations. Sub-HTML fragments are arranged in reading order using absolute spatial coordinates, ensuring that the reconstructed layout matches the original document structure.

- **Refine:** The final stage improves visual fidelity. Font sizes are adjusted using optimization techniques (e.g., binary search with textFit.js), background regions are restored using image inpainting, and a VLM-driven refinement loop iteratively aligns the rendered HTML with the original document image. This step ensures consistency in both appearance and structure.

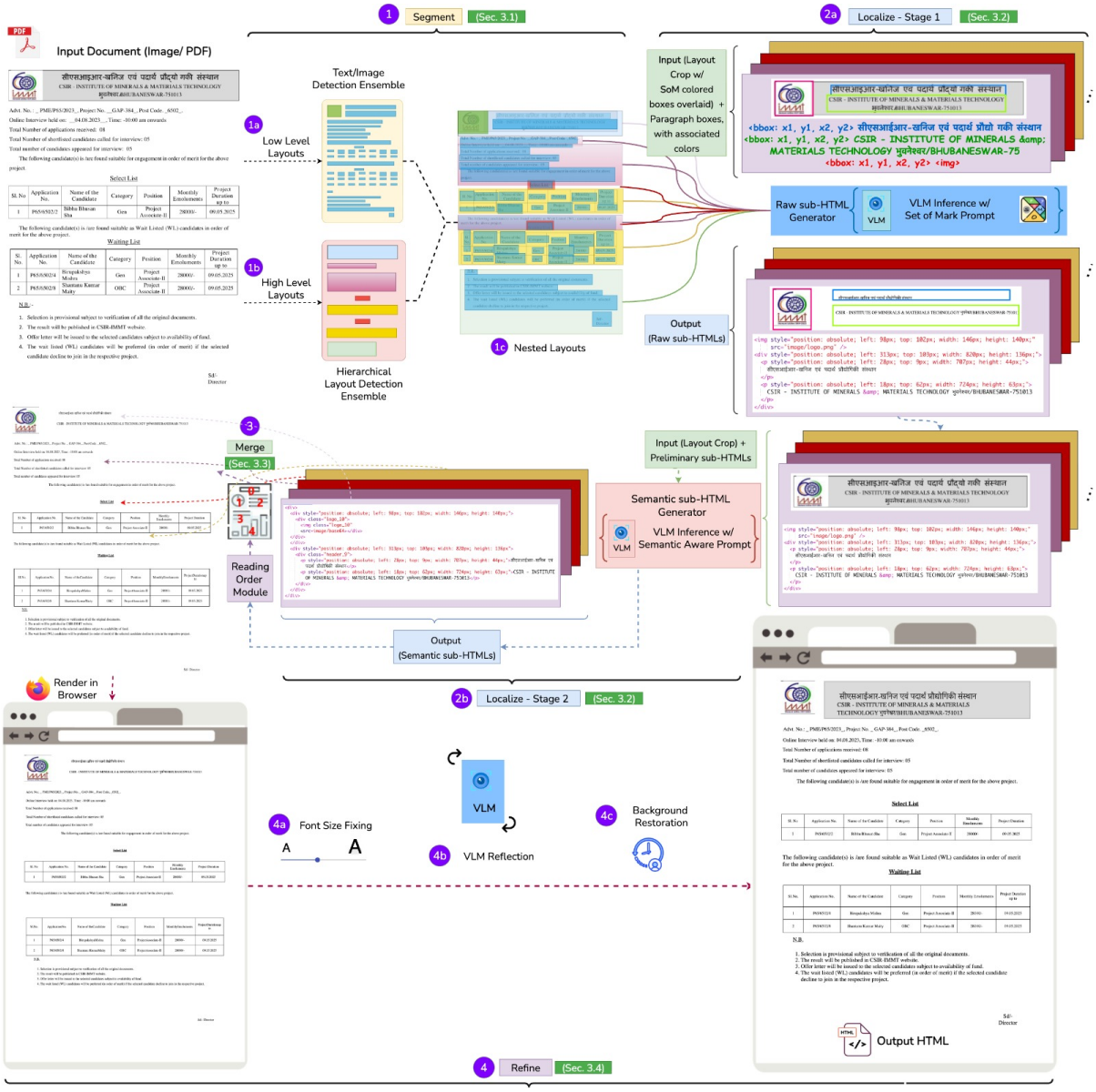


Figure 6.1 Overview of REPLICA, our four-stage framework for high-fidelity document-to-HTML conversion.(1 Segment) Build a nested layout tree using an ensemble of flat, hierarchical, and text-segmentation models to capture both fine-grained and structural regions. (2 Localize) Generate region-level sub-HTMLs via two-stage layout-aware Set-of-Mark prompting, enriched with OCR and fine-grained attributes, enabling parallel and semantically rich generation. (3 Assemble) Merge sub-HTMLs by aligning absolute bounding boxes with predicted reading order, preserving layout fidelity and document flow. (4 Refine) Enhance visual quality via font-size fixing, background restoration, color correction, and an iterative VLM reflection loop for progressive alignment with the source document. Refer supplementary for more details.

Overall, REPLICA transforms document images into semantically rich and visually faithful HTML representations, bridging the gap between raw visual input and structured digital documents.

6.2 Limitations of Text-Only Reconstruction

Despite its strong structural reconstruction capabilities, REPLICA initially relies on OCR-centric representations for textual content. These representations capture the textual string but ignore typographic attributes such as bold, italic, underline, and strikeout.

This limitation is significant for several reasons. First, textual attributes often encode semantic intent. For example, bold text is commonly used for headings or emphasis, while italic text may indicate titles or special terminology. Similarly, underline and strikeout are frequently used to denote annotations, corrections, or important highlights.

Second, the absence of attribute modeling leads to a loss of visual fidelity. Even if the textual content is correct, the reconstructed HTML may not visually match the source document. This discrepancy becomes more pronounced in structured documents such as forms, legal documents, and educational materials, where styling plays a critical role in interpretation.

Finally, OCR-based pipelines treat each word independently, making it difficult to infer subtle attribute variations that depend on surrounding context. As a result, relying solely on OCR leads to incomplete document understanding.

6.3 TexTAR for Textual Attribute Recognition

To address these limitations, we integrate **TexTAR**, a context-aware Transformer architecture designed for word-level textual attribute recognition.

TexTAR operates on *Context Windows (CWs)*, where each word is analyzed along with its neighboring words. This design enables the model to capture both local visual features and contextual patterns, which are essential for distinguishing subtle typographic variations.

Given a word and its surrounding context, TexTAR predicts textual attributes across two groups:

- T_1**:** Bold, Italic, Bold & Italic
- T_2**:** Underline, Strikeout, Underline & Strikeout

Unlike traditional approaches that rely solely on font features, TexTAR leverages contextual dependencies through Transformer-based attention mechanisms and incorporates spatial information using 2D positional embeddings. This allows it to accurately classify attributes even in challenging scenarios such as noisy scans, multi-lingual documents, and visually complex layouts.

6.4 Integration of TexTAR within REPLICA

TexTAR is integrated into the **Localize** stage of the REPLICA pipeline, where it enhances the extracted word-level representations with stylistic information.

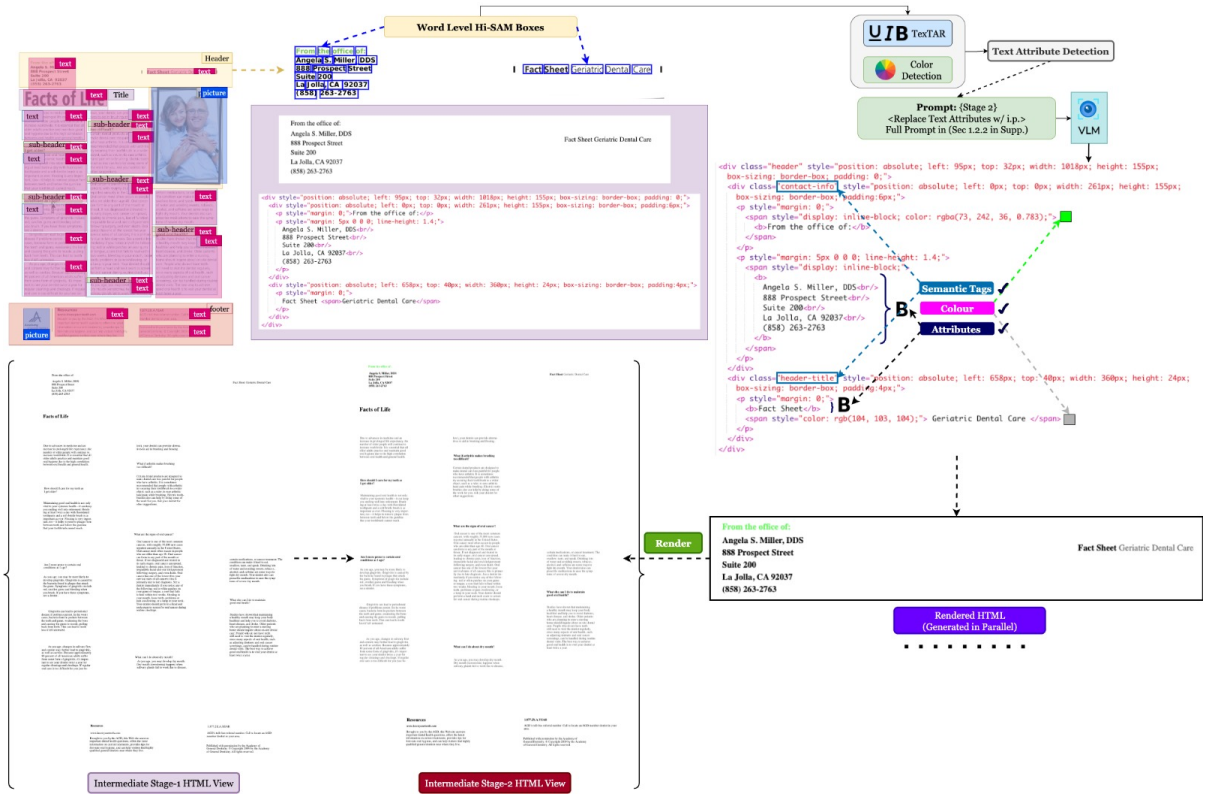


Figure 6.2 TexTAR integration within the REPLICA pipeline: The sub-HTML representations are enriched with semantic structure and stylistic information during the localization stage. Word-level regions obtained from Hi-SAM [19] provide fine-grained text crops, which are processed by TexTAR to predict typographic attributes such as bold, italic, underline, and strikeout. In parallel, color information is estimated using heuristic clustering. These attribute predictions are incorporated into layout-aware prompts, enabling the VLM to generate HTML with correct inline styles and semantic tags (e.g., headings, lists, and tables). This integration allows REPLICA to jointly model textual content and presentation, resulting in more faithful and visually consistent document reconstruction.

6.4.1 Word-Level Attribute Extraction

After obtaining word-level regions using OCR and Hi-SAM segmentation, each word image is passed through TexTAR. The model predicts the corresponding textual attributes based on both the word appearance and its contextual neighborhood.

6.4.2 Prompt Enrichment

The predicted attributes are incorporated into the layout-aware prompts provided to the VLM (Gemini 2.5 Pro). These attributes are encoded as inline style cues, enabling the model to generate HTML outputs with correct formatting, such as:

- <b> for bold text

- `<i>` for italic text
- `<u>` for underline
- `<s>` for strikeout

This enrichment allows the VLM to produce semantically and visually consistent HTML representations.

6.4.3 Impact on Reconstruction Pipeline

By integrating TexTAR, REPLICA transitions from a purely text-and-layout reconstruction system to a *style-aware document understanding pipeline*. The reconstructed output now captures not only what is written, but also how it is presented.

6.5 Importance of Textual Attributes

Textual attributes play a crucial role in document interpretation. They provide additional semantic signals that are not captured by text alone.

For instance:

- Bold text often indicates section headings or key information.
- Italic text may represent emphasis or domain-specific terminology.
- Underline is commonly used for highlighting or annotations.
- Strikeout indicates corrections or removed content.

Ignoring these attributes results in a loss of important contextual information. By incorporating TexTAR, REPLICA preserves these signals, leading to improved interpretability and usability of reconstructed documents.

6.6 Quantitative Impact

We evaluate the contribution of textual attribute modeling using the Visual Fidelity Score (VFS), which measures how closely the reconstructed HTML matches the original document in terms of structure and appearance.

Incorporating TexTAR results in an absolute improvement of **15%** in VFS. This demonstrates that textual attributes are a critical component for achieving high-fidelity document reconstruction.

Configuration	VFS ↑
Base w/ OCR + Color	0.60
+ Text Attributes (TexTAR)	0.75

Table 6.1 Impact of TexTAR on document reconstruction quality.

6.7 Summary

In summary, REPLICA provides a strong foundation for layout-aware document reconstruction, but its effectiveness is limited without modeling typographic attributes. By integrating TexTAR, the system gains the ability to capture both textual content and stylistic information.

This integration significantly improves reconstruction fidelity, enhances semantic understanding, and enables more accurate representation of complex real-world documents. The results highlight the importance of combining context-aware attribute recognition with layout-aware reconstruction for comprehensive document understanding.

Chapter 7

Conclusion

7.1 Discussion

This thesis studied the problem of *Textual Attribute Recognition* (TAR) in real-world document images. Unlike conventional OCR systems that focus only on textual content, this work emphasized the importance of visual attributes such as bold, italic, underline, and strikeout, which encode semantic structure, emphasis, and document hierarchy. Accurately recognizing these attributes is essential for richer document understanding and faithful reconstruction of documents into structured formats.

A key observation of this work is that textual attributes cannot be reliably determined from isolated word images. Instead, they depend on local visual context and spatial relationships within the document. To address this, we introduced **TexTAR**, a context-aware Transformer-based framework that operates on compact *Context Windows* to efficiently capture neighborhood information while remaining computationally scalable.

We further proposed a spatially-aware attention mechanism using `RoPE-MixAB`, enabling the model to incorporate 2D positional information. The use of dual classification heads allows the model to separately learn text-dependent and visually distinct attribute groups. In addition, the `CAvg` module aggregates predictions across overlapping context windows, improving robustness at the word level.

To support realistic evaluation, we introduced **MMTAD**, a multilingual and multidomain dataset with large-scale word-level annotations for textual attributes. Experiments on this dataset demonstrate that TexTAR consistently outperforms existing methods, particularly on attributes that require contextual reasoning. Ablation studies further validate the effectiveness of the proposed architectural and training design choices.

Overall, this thesis shows that incorporating context and spatial structure is crucial for accurate textual attribute recognition, and highlights the importance of moving beyond text-only document understanding toward richer, visually grounded representations.

7.2 Future Work

Several directions remain for future research. First, extending the attribute space beyond the current set (bold, italic, underline, strikeout) to include richer typographic features such as font styles, colors, and sizes would enable more comprehensive document understanding.

Second, integrating TAR with end-to-end document pipelines, including text detection and recognition, could improve robustness and reduce dependency on external detectors. Third, exploring hierarchical context modeling that combines local and global document structure may further enhance performance in complex layouts.

Finally, expanding multilingual coverage and developing more diverse datasets, especially for low-resource scripts and domains, will be important for building more general and practical document intelligence systems.

In summary, this work provides a strong foundation for context-aware textual attribute recognition and opens avenues for integrating visual semantics into next-generation document understanding systems.

Bibliography

[1] L. Zhang, Y. Lu, and C. L. Tan, "Italic font recognition using stroke pattern analysis on wavelet decomposed word images." vol. 4, 01 2004, pp. 835–838.

[2] R. KantYadav and B. Mazumdar, "Detection of bold and italic character in devanagari script," *International Journal of Computer Applications*, vol. 39, pp. 19–22, 02 2012.

[3] H. Singh, "Detection of bold and italic character in gurmukhi script," *IOSR Journal of Computer Engineering*, vol. 1, pp. 28–31, 01 2012.

[4] A. Zramdini and R. Ingold, "Optical font recognition using typographical features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 877–882, 1998.

[5] A. Ramakrishnan, "Script independent detection of bold words in multi font-size documents," 12 2013.

[6] B. Chaudhuri and U. Garain, "Automatic detection of italic, bold and all-capital words in document images," in *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No.98EX170)*, vol. 1, 1998, pp. 610–612 vol.1.

[7] S. Ajward, N. Jayasundara, S. Madushika, and R. Ragel, "Converting printed sinhala documents to formatted editable text," in *2010 Fifth International Conference on Information and Automation for Sustainability*, 2010, pp. 138–143.

[8] R. Smith, "An overview of the tesseract ocr engine," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 2, 2007, pp. 629–633.

[9] Z. Wang, J. Yang, H. Jin, E. Shechtman, A. Agarwala, J. Brandt, and T. S. Huang, "Deepfont: Identify your font from an image," 2015.

[10] T. Mondal, A. Das, and Z. Ming, "Exploring multi-tasking learning in document attribute classification," 2021.

[11] C. Nie, Y. Hu, Y. Qu, H. Liu, D. Jiang, and B. Ren, "Taco: Textual attribute recognition via contrastive learning," 2022.

[12] I.-C. Sandu, D. Voinea, and A.-I. Popa, "Consent: Context sensitive transformer for bold words classification," 2022.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[14] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," 2017. [Online]. Available: https://arxiv.org/abs/1611.05431

[15] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," 2020. [Online]. Available: https://arxiv.org/abs/1905.11946

[16] S. H. Z. Zhong, L. Jin, S. Zhang, and H. Wang, "Dropregion training of inception font network for high-performance chinese font recognition," 2017. [Online]. Available: https://arxiv.org/abs/1703.05870

[17] Z. Zhao, H. Kang, B. Wang, and C. He, "Doclayout-yolo: Enhancing document layout analysis through diverse synthetic data and global-to-local adaptive perception," *arXiv preprint arXiv:2410.12628*, 2024.

[18] O. Nath, S. Kukkala, M. Khapra, and R. K. Sarvadevabhatla, "Indicdlp: A foundational dataset for multi-lingual and multi-domain document layout parsing," in *Proceedings of ICDAR 2025 (Oral)*, 2025, dataset spans 11 Indic languages plus English, covering 12 document domains. [Online]. Available: https://indicdlp.github.io/

[19] M. Ye, J. Zhang, J. Liu, C. Liu, B. Yin, C. Liu, B. Du, and D. Tao, "Hi-sam: Marrying segment anything model for hierarchical text segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 03, pp. 1431–1447, 2025.

[20] G. Jaume, H. Kemal Ekenel, and J.-P. Thiran, "Funsd: A dataset for form understanding in noisy scanned documents," in *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, vol. 2, 2019, pp. 1–6.

[21] Mindee, "doctr: Document text recognition," https://github.com/mindee/doctr, 2021.