

Handwritten Text Recognition: A Survey

Carlos Garrido-Munoz, Antonio Rios-Vila, and Jorge Calvo-Zaragoza

Abstract—Handwritten Text Recognition (HTR) has become an essential field within pattern recognition and machine learning, with applications spanning historical document preservation to modern data entry and accessibility solutions. The complexity of HTR lies in the high variability of handwriting, which makes it challenging to develop robust recognition systems. This survey examines the evolution of HTR models, tracing their progression from early heuristic-based approaches to contemporary state-of-the-art neural models, which leverage deep learning techniques. The scope of the field has also expanded, with models initially capable of recognizing only word-level content progressing to recent end-to-end document-level approaches. Our paper categorizes existing work into two primary levels of recognition: (1) *up to line-level*, encompassing word and line recognition, and (2) *beyond line-level*, addressing paragraph- and document-level challenges. We provide a unified framework that examines research methodologies, recent advances in benchmarking, key datasets in the field, and a discussion of the results reported in the literature. Finally, we identify pressing research challenges and outline promising future directions, aiming to equip researchers and practitioners with a roadmap for advancing the field.

Index Terms—Handwritten Text Recognition, Document Image Analysis, Document Processing, Benchmarking.

I. INTRODUCTION

HANDWRITTEN Text Recognition (HTR) represents a cornerstone challenge within the fields of pattern recognition and machine learning. The task of converting handwritten text into a machine-encoded format has profound implications across several domains, including historical document preservation, automated data entry, digital note-taking, and accessibility for the visually impaired [1]. As we advance further into the digital age, the necessity for accurate and efficient HTR systems continues to grow, driven by the ever-increasing amount of handwritten data generated daily.

The complexity of HTR arises from the inherent variability in human handwriting. Unlike printed text, which adheres to standardized fonts and spacing, handwritten text exhibits a broad variety of styles, slants, sizes, and embellishments that are influenced by individual writing habits, cultural contexts, and even emotional states [2]. These variations present significant challenges for recognition systems, requiring algorithms capable of generalizing across diverse handwriting samples.

Historically, HTR systems have evolved through several technological waves. Early approaches relied on heuristic and rule-based methods, leveraging handcrafted features to recognize characters and words [3]–[7]. These systems were limited in their capacity to handle the high variability of handwritten text and often required significant preprocessing and segmentation of individual characters [8], [9]. The advent

Authors are with the Pattern Recognition and Artificial Intelligence Group, University of Alicante, Spain (e-mails: carlos.garrido@ua.es, antonio.rios@ua.es, jorge.calvo@ua.es)



Fig. 1. Overview of the different levels of granularity in Handwritten Text Recognition (HTR) systems, ranging from word-level to document-level transcription. The hierarchical structure illustrates the increasing complexity of HTR approaches as they seek to handle more extensive handwritten content, reflecting the advancements in this field.

of machine learning introduced a paradigm shift, enabling the development of more adaptable and robust recognition systems. Statistical methods, such as Hidden Markov Models (HMM) [5], [10], offered improved performance by learning from labeled data to recognize complete words or lines [11]–[14].

The emergence of deep learning [15] has revolutionized the field of HTR, bringing forth unprecedented advancements in recognition performance. Convolutional Neural Networks (CNN) [16] and Recurrent Neural Networks (RNN) [17] have demonstrated remarkable success in capturing spatial and sequential dependencies in handwriting. More recently, Transformer models have further pushed the boundaries [18]. These architectures have shown immense potential in HTR due to their ability to model long-range dependencies and handle variable-length sequences effectively.

In addition to advances in performance, these technological breakthroughs have also diversified the approaches to HTR. The scope of recognition has expanded in complexity over time, from transcribing words and lines [19]–[22] to approaches that directly output the content of entire paragraphs or documents [23]–[25]. Within this context, this survey aims to provide a comprehensive examination of the current state of HTR focusing on its different levels of granularity. We broadly categorize the possibilities into two main groups (see Fig. 1): *up to line-level*, comprising word-level and line-level; and *beyond line-level*, involving paragraph-level and document-level recognition. The need for this distinction is evident in the different ways of approaching the associated challenges, which we will discuss later.

Our survey focuses on the transcription process itself. We will not delve into auxiliary processes such as character segmentation or layout analysis, which are often necessary to prepare the input for HTR systems. This distinction is

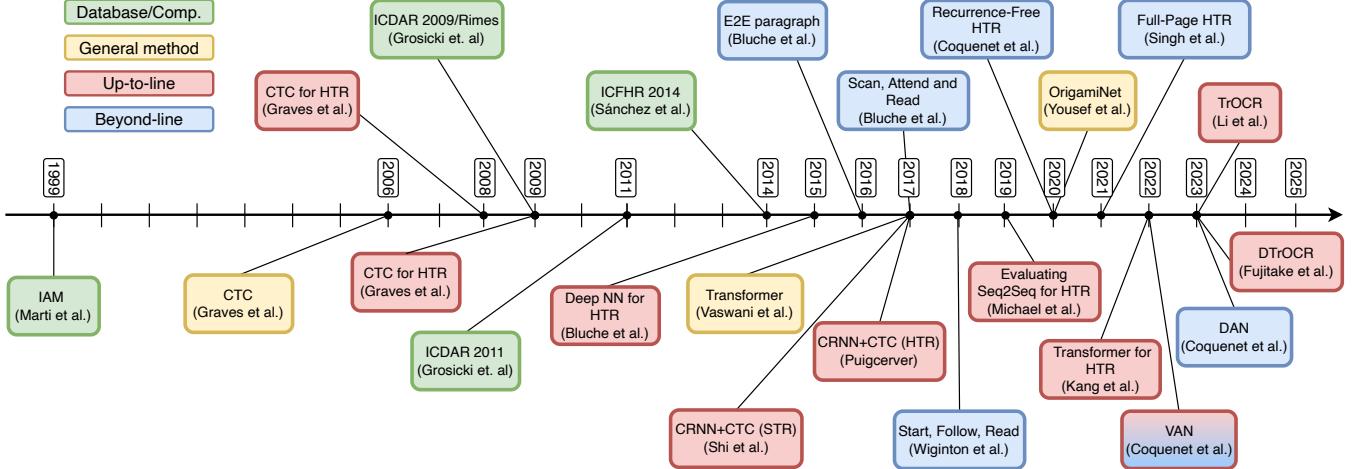


Fig. 2. Timeline of milestones in Handwritten Text Recognition (HTR). We categorize them into four levels: datasets and competitions (green), general methods/architectures (yellow), up-to-line models (red), and beyond-line models (blue).

crucial as it emphasizes methods that perform the handwriting recognition process. Therefore, when we discuss about *up to line-level HTR*, we refer to approaches that directly process an image containing, at most, one line of text. Similarly, when we review paragraph- or document-level literature, we refer to approaches that directly transcribe the handwritten content at those levels, without relying on prior segmentation. Following this perspective, we purposely exclude character-level recognition from this survey, as it essentially reduces to a classification task, an area already well-covered in the vast existing literature.

A. Purpose and contribution

HTR has seen significant progress in recent years, driven by the development of sophisticated deep learning techniques and the growing availability of diverse datasets (see Fig. 2). These advancements have transformed the field.

Existing surveys, such as those by Wang et al. [26] and AlKendi et al. [27] provide comprehensive overviews of text detection and recognition across various contexts. Wang et al. cover a wide range of text recognition tasks, including printed and scene text, addressing both detection and recognition phases. AlKendi et al. [27] offer a detailed examination of HTR systems with a specific focus on historical French documents and a broad range of languages. Additionally, Chen et al. [28] focus on text recognition in the wild, dealing with text in natural scenes, which presents different challenges compared to HTR (as discussed later). Therefore, it is both timely and necessary to conduct a comprehensive survey of the current state of HTR that reflects the new landscape of the field.

Our survey provides a specialized and focused examination of HTR, delving into the end-to-end transcription processes and categorizing methodologies based on the granularity of recognition. We provide a comprehensive review with a general and unifying view of the formulation. Our survey also explores the implications of these advancements on benchmarking practices, offering insights into the datasets and

metrics most relevant for evaluating HTR systems. In addition, this work unifies and discusses the results reported by the most notable advances in the field through a common benchmark, providing perspective on the evolution of methodological impact in the field.

It is important to emphasize that this evolution analysis is framed within the transcription of Latin scripts, as most technological advancements in the field have been reported on these sources.

B. Structure of the survey

The structure of this survey is organized as follows: Section II covers the foundational preliminaries, including the historical background of HTR, the formal problem formulation, and its connections to related fields. Section III explores the different methodologies employed in HTR, categorized by its inherent complexity levels, and examines the architectural designs and strategies used in each category. Section IV focuses on the benchmarking of HTR systems, providing an in-depth review of the datasets, evaluation metrics, and the performance reported in the literature. Finally, Section V concludes the survey by summarizing the key insights and identifying open research challenges, outlining potential directions for future work in HTR technologies.

II. PRELIMINARIES

A. Background of HTR

In this section, we provide an overview of the foundational concepts and historical development of HTR, covering the shift from early heuristic-based approaches to modern deep learning techniques. This background sets the stage for a detailed exploration of the methodologies and advancements that have emerged in recent years. Initial attempts in HTR focused on classifying individual characters [16]. However, within the scope of this survey, this task is considered a general classification challenge, that is extensively covered in the broader literature. Therefore, we will directly move on to the background of the sequential challenges of HTR.

Historically, HTR has adapted advancements from the field of Automatic Speech Recognition (ASR) [12], [29]–[31] because of the similarities in their problem formulations. Both fields deal with sequential data and require models that can handle variability and context dependencies. For years, the field of HTR relied on HMM [32]–[35]. Further advances favored the use of RNNs due to their capacity to model sequential data [36]–[39]. In particular, the use of bidirectional Long Short-Term Memory (LSTM) networks with Connectionist Temporal Classification (CTC) [36] dominated the state of the art in HTR for decades [40]–[43]. Despite this prevalence of CTC, attention-based encoder-decoder approaches [44] gained popularity for their competitive results [45]–[48]. The work of Michael et al. [49], provides a comprehensive comparison of the different sequence-to-sequence approaches for line-level HTR.

As occurred in many other areas, there has been a growing interest in scalable and parallelizable architectures such as the Transformer [18] by adapting the Vision Transformer [50] to the HTR field [51]–[56]. HTR has benefited from this adaptation, either in isolation with an encoder-decoder [57]–[60] or in combination with the CTC objective function [61]–[63]. Diaz et al. [55] explore universal architectures for text recognition, concluding that a CNN backbone with a Transformer encoder, a CTC-based decoder, plus an explicit language model, is the most effective strategy to date for line-level transcriptions. Regardless of this progress, however, the need for large labeled corpora as a pre-training strategy in Transformer-based models has become increasingly evident [55], [57], [64]–[66].

Given the increase in computational capacity, the paradigm has shifted towards end-to-end methods capable of reading beyond the line level. This current trend in the field began to develop with attention masking mechanisms that replace the explicit Layout Analysis (LA) [25], [67], [68]. Later advances also extended the single-line transcription state of the art to work with multiple lines. The CTC-based methodologies were extended to work in a multi-line setting, with reshaping mechanisms that alleviate its constraints [69], [70]. The Transformer architecture also emerged in this paradigm, enabling the adaptation of attention-based encoder-decoder architectures to transcribe documents of any structure, thanks to its unconstrained nature and the ability to produce large synthetic datasets [24], [71], [72]. This approach currently represents the state of the art of the field and lays the foundation for future development (see Fig. 2).

B. Levels of complexity

The field of HTR encompasses a wide range of challenges that vary significantly in their complexity. While existing literature provides various perspectives on these challenges, it often lacks clear differentiation between input types and their inherent difficulties. To advance our understanding and properly evaluate different methodologies, establishing a precise framework that defines and categorizes the different levels of complexity is essential.

The task of HTR involves converting handwritten text into an encoded format that can be interpreted by a machine.

However, the transcription challenge can be narrowed down in different ways. At the most fundamental level, we find individual *characters*,¹ which represent the simplest unit of recognition and complexity. As stated above, character recognition is outside the scope of this paper, as it can be reduced to a classification problem.

Beyond character recognition, the concept of Reading Order (RO) becomes an important aspect to consider. The RO is the sequence in which text elements are read to preserve the logical flow and conceptual meaning, following linguistic and cultural conventions. At the first level of complexity concerning RO, we focus on *words*, which are formed by the sequential arrangement of characters. For word-level recognition, the RO follows a single and consistent direction, which is determined by the reading rules of the writing system. For example, in Latin sources it is left-to-right, while it is top-to-bottom in traditional Japanese and Chinese books.

At a similar level of complexity, we also encounter lines, which are understood as a sequence of words separated by a delimiter character (commonly a space: ‘ ’). This increase in complexity does not alter the RO; instead, it lies in the need to transcribe a greater number of characters while incorporating this additional delimiter. In this context, HTR involves processing images that might span the full width of the page. These lines are read following the same RO type as words, remaining consistent and unidirectional, as illustrated in Fig. 3. Therefore, we establish the first level of complexity in the field.

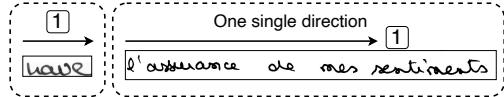


Fig. 3. Reading Order (RO) with a single direction at the line level.

Definition 1: Line-level HTR is the process of transcribing handwritten text from an input image containing a single line of text, which must be interpreted following a single RO, as determined by the rules of the language in which it is encoded. This level encompasses both *words* and *lines*.

Given the spatial limitations of paper, it is unfeasible to write complete texts in a single line. For this reason, the stacking of lines in an additional dimension is traditionally employed. In this case, an additional RO is introduced, where the text is interpreted first in a line-level direction and then typically follows a perpendicular direction to continue reading the subsequent lines, as depicted in Fig. 4. This structure is commonly known as *paragraphs* and establishes the next complexity level of HTR, since a new RO is introduced.

Definition 2: Paragraph-level HTR is the process of transcribing handwritten text from an input image containing multiple lines of text, where two predetermined RO directions must be followed: one for reading individual lines and another for traversing between lines. These directions are commonly perpendicular.

¹We emphasize that “character” here refers to the basic graphical unit that an HTR model learns to recognize, as the concept may vary across different writing systems.

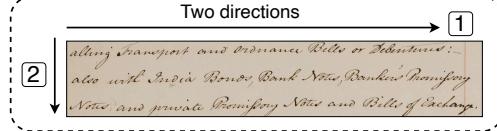


Fig. 4. Reading Order (RO) with two directions at the paragraph level.

Ultimately, the highest level of complexity arises when dealing with sources that contain multiple *paragraphs*. The arrangement of these blocks may not be straightforward, as shown in Fig. 5. In this case, a third RO across *paragraphs* is introduced, depending on the layout of each source.

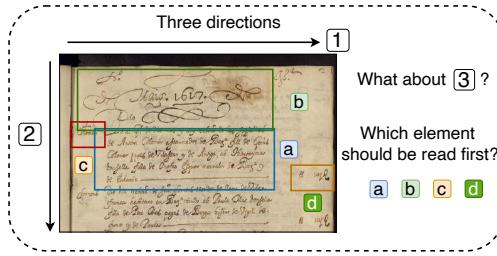


Fig. 5. Reading Order (RO) with three directions at the document level. In this case, the third RO is not necessarily consistent or trivial, and each manuscript may define its own rules regarding this aspect.

This layout may be either a static geometric arrangement, such as two-column sources, or a dynamic one, where artifacts such as images and tables may be introduced between text blocks. Therefore, we present the final definition.

Definition 3: Document-level HTR is the process of transcribing handwritten text from sources that contain multiple paragraphs arranged in an arbitrary, non-trivial RO.

Note that this definition excludes page images where a simple stack of paragraphs is found, which are sometimes understood as “full-page HTR” [70], [71]. In our view, the transcription challenge remains the same as for a single paragraph, since the RO for transitioning between lines also applies to transitioning between paragraphs. Therefore, from a technical perspective, such pages fall within the scope of the definition given for “paragraph-level HTR”.

Our hierarchical categorization of complexity levels provides a robust framework for understanding and addressing HTR challenges. These distinctions are fundamental to the comprehension of HTR methods, as different approaches may be better suited to different complexities. Accordingly, we structure the different methodologies based on where the most significant shift in complexity occurs due to the RO. Specifically, this refers to the transition from transcription *up to line level (words and lines)* to *beyond the line level (paragraphs and documents)*.

C. Formulation of HTR

Let us denote by \mathcal{X} the space of images containing handwritten text. Each element of \mathcal{X} is represented as a three-dimensional tensor in $\mathbb{R}^{H \times W \times C}$, where H , W , and C represent the height, width, and number of channels of the image,

respectively, and their dimensions may vary across instances. Given an input $\mathbf{x} \in \mathcal{X}$, the goal of HTR is to retrieve its corresponding textual representation $\mathbf{y} = (y_1, y_2, \dots, y_N)$, where $y_i \in \Sigma$ denotes individual *characters* from a predefined *alphabet* Σ . The alphabet Σ may include alphanumeric characters, punctuation symbols, or other special tokens such as spaces or line breaks. Statistically, this retrieval task can be formally defined as:

$$\mathbf{y}^* = \arg \max_{\mathbf{y} \in \Sigma^*} P(\mathbf{y} | \mathbf{x}), \quad (1)$$

where Σ^* is the Kleene closure over the alphabet Σ , representing all possible sequences of characters.

It is known that solving Eq. 1 exactly is computationally intractable [73]. Therefore, most HTR methods rely on approximate inference strategies to solve this problem efficiently. These methods differ in how they approximate Eq. 1 and how they estimate the probabilities involved.

Using Bayes’ theorem, Eq. 1 can be reformulated as:

$$P(\mathbf{y} | \mathbf{x}) = \frac{P(\mathbf{x} | \mathbf{y}) P(\mathbf{y})}{P(\mathbf{x})}. \quad (2)$$

Since $P(\mathbf{x})$ is independent of \mathbf{y} during the maximization process, the problem can be equivalently expressed as:

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y} \in \Sigma^*} P(\mathbf{x} | \mathbf{y}) P(\mathbf{y}), \quad (3)$$

where:

- $P(\mathbf{x} | \mathbf{y})$ represents the *likelihood*, which models how well the transcription \mathbf{y} explains the input image \mathbf{x} . This is implemented using a statistical model, such as a neural network.
- $P(\mathbf{y})$ represents the *prior probability* of the transcription \mathbf{y} , often estimated using a language model (LM).

The prior probability $P(\mathbf{y})$ can be modeled using statistical LM such as n -grams, where n defines the context size. A LM is formally defined as a function:

$$p : \Sigma^* \rightarrow [0, 1], \quad \text{such that} \quad \sum_{\mathbf{y} \in \Sigma^*} P(\mathbf{y}) = 1.$$

LMs provide prior knowledge about the structure of valid sequences and are often integrated into the recognition process through techniques such as re-scoring. Alternatively, in some domains, stricter constraints can be imposed by using a formal language or *lexicon*. Let $\mathbf{L} \subset \Sigma^*$ denote the set of valid transcriptions defined by such constraints. In this case, the HTR problem is reformulated as:

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y} \in \mathbf{L}} P(\mathbf{x} | \mathbf{y}) P(\mathbf{y}). \quad (4)$$

At a formulation level, the challenge in HTR is consistent across different levels of complexity, but the alphabet Σ changes depending on such level: at the *character level*, Σ represents the basic charset; at the *word level*, Σ includes additional symbols such punctuation marks; at the *line level*, Σ incorporates spaces; at the *paragraph level*, Σ includes line breaks; and at the *document level*, Σ might be further expanded to include paragraph breaks or other structural

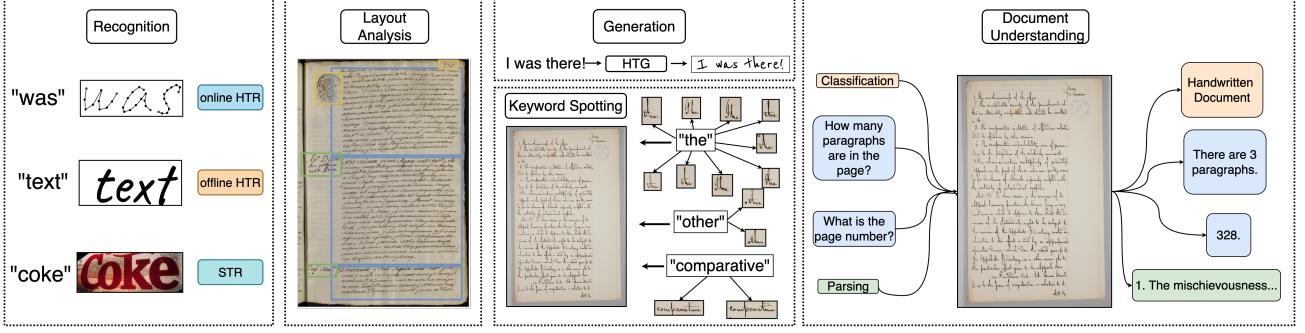


Fig. 6. Overview of related fields in Handwritten Text Recognition (HTR). The figure illustrates key areas closely related to HTR, including: (1) Text Recognition (online HTR, offline HTR, and Scene Text Recognition - STR); (2) Layout Analysis (LA) for segmenting and classifying document regions; (3) Handwritten Text Generation (HTG); (4) Keyword Spotting (KWS) for targeted retrieval of specific words; and (5) Document Understanding (DU), encompassing classification, parsing, and semantic analysis of handwritten documents to extract structured information such as paragraph counts, page numbers, and textual content.

tokens. However, regardless of the level, the goal of HTR is still to approximate $P(y | x)$.

D. Relation with other fields

HTR relates to diverse domains, many of which share both challenges and techniques. In this section, we introduce the fields most related to the focus of this survey, placing special emphasis on the differences that make HTR worthy of specific investigation. We illustrate the differences between HTR and the related fields in Fig. 6.

1) *Online HTR*: Online, or pen-based HTR [74]–[76], involves recognizing text as it is written using a digital pen or stylus, leveraging dynamic temporal information such as pen tip coordinates, pressure, and tilt. This data provides contextual clues about stroke order and direction, which are absent in “offline HTR”. While online HTR can be converted into an offline problem by rendering pen movements into images, the reverse is not possible. Offline HTR requires handling handwriting variability without access to temporal data, making it a distinct challenge with broader applicability, particularly for historical documents and scanned notes.

2) *Scene Text Recognition*: Scene Text Recognition (STR) [77]–[80] focuses on recognizing text in natural scenes, where varying fonts, distortions, and complex backgrounds pose unique challenges. While both leverage similar advances in pattern recognition and computer vision, the primary challenge in HTR lies in handling handwriting variability, while STR must address environmental noise and background interference, making them distinct fields despite their methodological overlaps.

3) *Keyword Spotting*: Keyword Spotting (KWS) [81]–[84] involves identifying and locating specific keywords within a set of document images. This field is particularly relevant for searching and retrieving information from large collections of handwritten documents. While KWS and HTR share the goal of identifying handwritten text, they approach the problem from different angles. HTR focuses on the comprehensive transcription of handwritten text into machine-readable format, while KWS is concerned with locating specific words or sentences within the text without necessarily transcribing the entire document.

4) *Layout Analysis*: Layout Analysis (LA) [85]–[87] focuses on segmenting and categorizing document components such as text blocks, tables, and images to understand their spatial arrangement. While both LA and HTR deal with document images, they address different aspects of document interpretation. LA focuses on the structural analysis and segmentation of the document, ensuring that various elements are correctly identified and classified. In contrast, HTR specifically concentrates on transcribing the handwritten text within these segmented areas.

5) *Document Understanding*: Document Understanding (DU) [88]–[91] encompasses a comprehensive set of tasks aimed at interpreting and extracting meaningful information from documents. This includes text recognition, entity extraction, or document question answering. While HTR provides the fundamental transcription of handwritten text, DU extends beyond this by analyzing relationships between text blocks, extracting key information, and generating structured representations. This requires additional layers of analysis, which are beyond the primary focus of HTR.

6) *Handwritten Text Generation*: Handwritten Text Generation (HTG) [92]–[94] focuses on synthesizing realistic handwritten text images from digital input, leveraging techniques such as Generative Adversarial Networks [92] and Transformers [93], [94]. This has applications in personalized handwriting synthesis, CAPTCHA systems, and augmenting datasets for training HTR models.

III. METHODOLOGIES

This section outlines the methodologies employed in the field of HTR, categorizing them based on the recognition level: up to line-level and beyond line-level. It examines the historical developments, methodological distinctions, and complementary techniques that enhance performance in the transcription of handwritten text images. HTR models are a class of probabilistic models that attempt to estimate the most probable transcription y given an image x , thus in the form of Eq. 4. Conceptually they all address the same task, that of aligning an input image with the most probable output sequence. The differences are encountered in (1) the alphabet

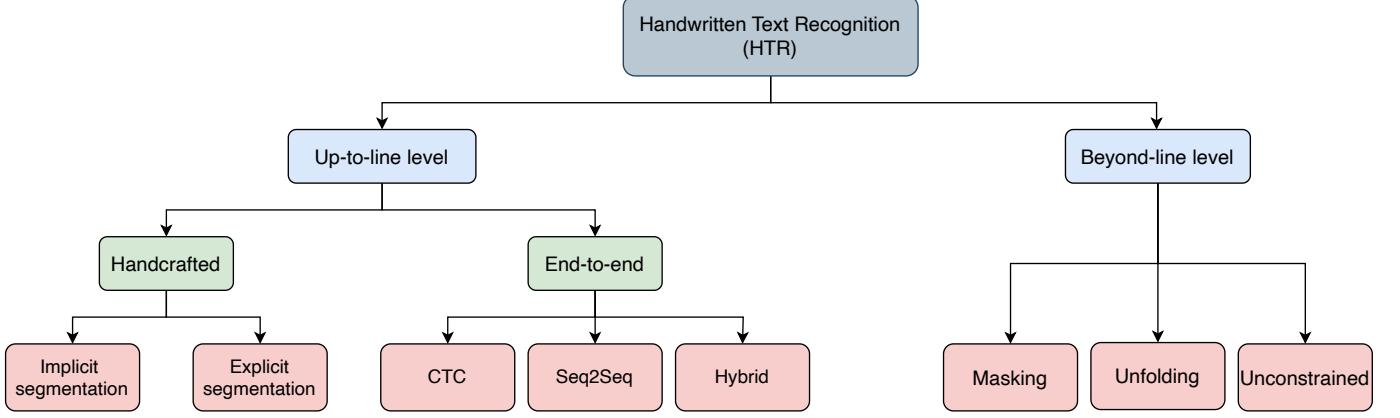


Fig. 7. Overview of Handwritten Text Recognition (HTR) methodologies. We divide the approaches into two main groups: up-to-line and beyond line-level. Up-to-line methods are further divided into Handcrafted approaches (explicit and implicit segmentation) and End-to-end techniques (CTC, Seq2Seq, and Hybrid models). Beyond line-level methods include SFR (Segment-Free Recognition) and Segmentation-free techniques, which are further specialized into Masking, Unfolding, and Unconstrained strategies.

(Σ), which differs depending on the level at which recognition is performed, and (2) how they align the input image with the output sequence.

We divide this section into two main categories: transcription models up to the line level (word and lines) and models beyond the line level (paragraph and documents).

A. Up-to-line level transcription

The challenges *up to line level* involve two categories: (1) word level, in which only characters without spaces are predicted or (2) line level, in which characters are predicted by adding the space character to Σ , which both share one single RO (one single direction). Before we start to unify the different approaches, we shall make a distinction between the models that go up to the line level. In the early days of HTR, due to the lack of computational and methodological resources, transcription was done in a more manual and laborious pipeline as shown in Fig. 8 and mainly focused on word recognition [13], [14], [95]–[102]. These steps consisted first of a pre-processing stage where the objective was to remove uninformative elements such as the background, lines or other elements and a posterior binarization to facilitate the extraction of information [103], [104]. The next step was normalization, which involved “rectifying” the image to keep the image invariant to the writer’s style. Typical operations in this case were skew [105]–[108] and slant correction [3], [10], [105], [108]–[111], smoothing (denoising) and image scaling. The next step was segmentation (also called fragmentation), in which the information appearing in the word was separated to be further recognized [112], [113].

At the methodological level, this does not change in that all still have to estimate Eq. 4, but these factors led to a series of work in which alignment had to occur after a number of manually extracted features. Specifically, these methods differed in whether the alignment was done after isolating the characters from the sequence (explicit segmentation) [4], [13], [97], [112], [114]–[124] or whether this separation was done implicitly (implicit segmentation) [5]–[7], [96], [98], [121],

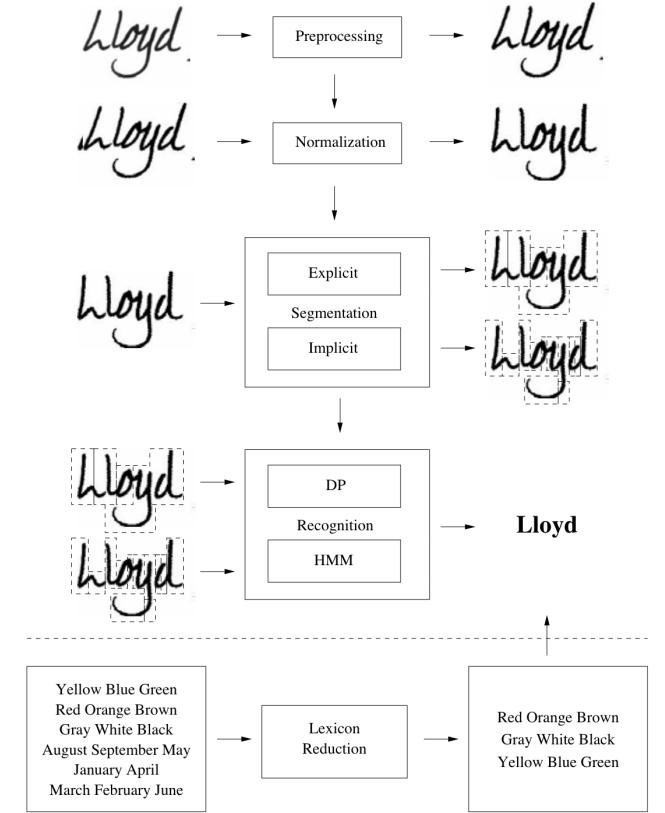


Fig. 8. Typical pipeline before the Deep Learning era. Image from [103].

[122], [125]–[135], so we consider this series of methods as two leaves of the same branch and group them into one: handcrafted methods. Therefore, we will divide up-to-line methodologies according to the “manual” learning process: handcrafted (manual) and end-to-end (automatic) approaches.

1) *Handcrafted approaches*: Early methods for HTR relied on handcrafted processes, focusing on segmenting word images into smaller components that could be matched to character models [112], [114]. We summarize this “classi-

cal” pipeline in Fig. 8. These segmentation-based approaches required manually designed processes to identify primitive segments, introducing significant challenges due to segmentation errors. Among these approaches, we identify two main categories: (1) explicit segmentation-based methods, which require the individual segmentation of each character, and (2) implicit segmentation-based methods, where the segments do not necessarily correspond to individual characters. Recognition performance depended heavily on accurately matching segments to corresponding characters. According to the segmentation methods, we find:

a) Explicit segmentation: Each character was segmented and used the following methods to perform the recognition.

- Dynamic Programming (DP). DP was a widely adopted technique for matching segments to letters [3], [131], [136]–[140]. By constructing an optimal path through a cost matrix, DP minimized the mismatch between observed segments and hypothesized letters. The method handled gaps and substitutions effectively but required carefully defined cost functions, often tailored to specific datasets or writing styles.
- Shortest path in graphs. Graph-based methods reformulated the segmentation problem as finding the shortest path in a graph, where nodes represented segment-letter pairs and edges encoded transition costs between them [115]–[117]. These methods evaluated all possible segment-to-letter matches in a structured way, leveraging graph traversal algorithms to find the least costly alignment. This approach allowed greater flexibility in handling overlapping or ambiguous segments.
- Hidden Markov Models (HMMs). HMMs became a prominent tool in HTR due to their capability to model sequential data with hidden states [4], [13], [197], [118]–[124]. Each letter was modeled as a sub-HMM, and these were concatenated to represent entire words. The Viterbi algorithm [141] was used to identify the most probable sequence of matches between observed segments and model states. HMMs were particularly effective for handling variability in writing styles and sequential dependencies between letters.

b) Implicit segmentation: While early methods relied on explicit segmentation, later approaches explored segmentation-free paradigms [103], where features were learned automatically rather than manually defined. However, these approaches still used handcrafted features, such as:

- Low-level features: Basic elements like word contours, strokes, and skeleton approximations of segments [5]–[7], [96], [125]–[130].
- Medium-level primitives: Aggregations of low-level features, forming more complex patterns like complete letters or stroke groups [133].
- High-level holistic features: Global characteristics of word images, such as ascenders, descenders, loops, or stroke patterns [98], [121], [122], [134], [135].

Despite these advancements, both segmentation-based and segmentation-free methods often relied on alignment techniques like Minimum Edit Distance [98], [127]–[129], [133],

[134] or HMMs [5]–[7], [121], [122], [125], [126], [130], [132], [135]. These techniques provided a framework for aligning observed features or segments to target words, regardless of the feature abstraction level or segmentation method. Due to the laboriousness of the process, the error-prone series of steps in the classical pipeline and the lack of computational resources, almost all the work described focused on word recognition.

2) End-to-end approaches: With the advent of Deep Learning (DL), all features are learned automatically in an end-to-end manner, which is what the field is based on [15]. Under DL-based methods, these differ essentially in how they automatically align the learned features with the output sequence, which we have previously referred to as “alignment” i.e. to approximate Eq. 4. Depending on the method employed to align the features with the output sequence, one can find (1) CTC-based methods [19], [20], [25], [40], [70], [77], [99], [142]–[144], (2) sequence-to-sequence approaches [21], [45], [54], [55], [57], [61] and (3) hybrid techniques [49], [51], [63]. For an outline of the Deep Learning-based methods up-to-line level, see Fig. 9. Note that, in the HTR literature, we adopt the terminology used in Machine Translation [145], referring to the encoder as the architectural component that encodes the maps the input signal into a set of features, while the decoder processes them, generally transcribing in an autoregressive manner. As we will see, some models employ encoder-only architectures, while others integrate both an encoder and a decoder. In the following sections we describe how these different methods align and approximate (4).

a) CTC-based methods: In order to align the image x with the sequence y , CTC introduces the notion of “valid alignments” of y , which are all the possible character sequences of the alphabet $\Sigma' = \Sigma \cup \{\epsilon\}$ such that collapsing results in the output sequence y . This collapse is produced by merging repeated characters and removing the ϵ (typically named blank) token. Therefore, with CTC, for each labelled sequence y there is a set of $\mathcal{B}^{(y)} \in \mathcal{B}$ valid alignments where \mathcal{B} represents the complete space of alignments. Note that the length of the alignments is given by $|x|$, which typically corresponds to the number of columns x_1, x_2, \dots, x_T of the image, typically referred to as frames.² In this section, the methodologies explored assume a frame-wise reading. This means that all the frames contained in the feature map represent exclusively one character. This allows up-to-line methods to only read in a single RO direction. To reach this assumption, all methods resort to an *image collapse* step, where data is transformed from an image space to a sequence one. This is defined as a function $r : \mathbb{R}^{w,h,c} \rightarrow \mathbb{R}^{l,z}$, where h is the height of the feature map, w its width, c the number of channels, l is the output sequence length and z the number of features. This function reshapes the input feature map to merge their features along the horizontal axis. Therefore, they

²In the case of images, each frame is typically associated with a column of the image. This convention originates from the field of ASR, where the continuous nature of audio signals led to their partitioning into overlapping segments known as frames.

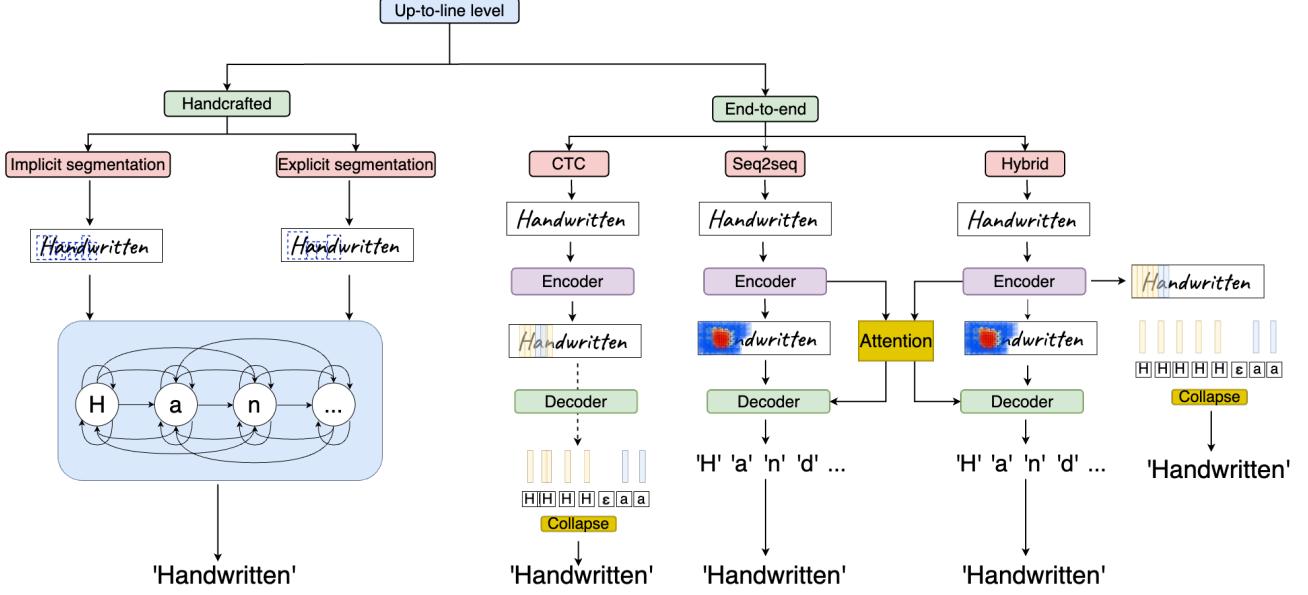


Fig. 9. Taxonomy of HTR approaches up to the line level. Methods are categorized into handcrafted and end-to-end approaches. Handcrafted methods rely on implicit or explicit segmentation, while end-to-end approaches include CT), sequence-to-sequence (Seq2Seq) models with attention, and hybrid architectures. Encoders extract visual features, and decoders reconstruct the textual sequence, with certain models employing attention mechanisms.

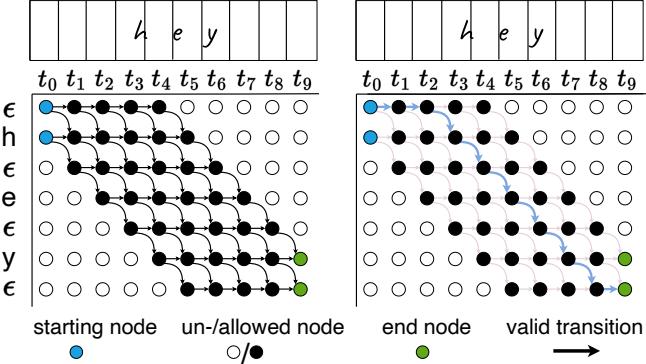


Fig. 10. Visualization of the Connectionist Temporal Classification (CTC) training mechanism for HTR. An input image x with the text sequence $y = \text{'hey'}$ is provided. For simplicity, we exemplify the process with a raw image. The left figure shows the valid transitions between nodes (allowed and end nodes) over time steps (t_0, t_9) while processing the handwritten input "hey". Starting nodes, unallowed nodes, allowed nodes, and end nodes are color-coded for clarity. The right figure shows how CTC optimizes the probabilities for the path that maximizes the alignment B .

address the following general formula:

$$\hat{y} = \arg \max_{y \in \mathcal{L}} P(y | r(f(x))) \quad (5)$$

where $f(\cdot)$ is the feature extraction process and $r(\cdot)$ the *image collapse* function. For simplicity, from now on we will assume that each frame x_t is the collapsed image into frames.

The approach for CTC then consists of dividing the input sequence into T frames and predicting, for each frame $t \in T$, the probability of each character of the target alphabet. Therefore, the alignment occurs monotonically from left to right, mapping one frame to a single character.

This process is illustrated in the Fig. 10, where the valid alignments correspond to those paths that pass through valid

nodes via the transitions established in the diagram. The transition probabilities can be used to infer the actual output in a greedy form by taking the most likely character at each time step t or by using beam search (see Section III-C). Formally, for a given pair of (x, y) CTC [36] computes the conditional probability

$$P(y|x) = \sum_{B^{(y)} \in \mathcal{B}} \prod_{t=1}^T P(a_t|x_t) \quad (6)$$

where $P(a_t|x)$ is the probability for a single alignment step-by-step (with T given by the length of the input to align), marginalized over the set $\mathcal{B}^{(y)}$ of valid alignments for that sequence, where $|\mathcal{B}^{(y)}| < T$. This probability is maximized throughout the log-loss $\sum_{(x,y) \in \mathcal{D}_{train}} -\log P(y|x)$, where \mathcal{D}_{train} is the training dataset, often consisting in pairs of images and transcriptions. For inference, it is selected the path that maximizes the alignment B :

$$B^* = \arg \max_B \prod_{t=1}^T P(a_t|x_t) \quad (7)$$

Typically, the CTC-based architectures used to estimate probabilities $P(a_t|x)$ in the HTR field have been LSTMs [17] and its bidirectional variant (Bi-LSTM) [146]. Specifically, in early work using LSTMs for HTR, the use of Multidimensional LSTMs (MDLSTM) was applied directly on HTR images to recognize words without using any prior feature extractor [19]. In later works [67], [147], [148], these MDLSTMs were applied on low-level features obtained by a convolutional feature extractor. The use of different MDLSTM layers with different heights allowed converting 2D signals to 1D allowing to apply CTC to this 1D output. The first attempt to recognize unconstrained lines without the pre-segmentation of words was in [19], where the authors used

a Bi-LSTM and CTC directly to recognize off-line English texts. Following [77], the work of Puigcerver [20] proposed to make this conversion from a 2D to 1D signal by collapsing the features in height by passing them to image channels to directly recognize sentences without the use of MDLSTMs. This allowed BLSTM-type networks to be applied directly to 1D signals, significantly reducing the computational cost and obtaining better results in terms of transcription. This has been the dominant pipeline for CTC-based models from then [20], [49], [70], [144], [149], [150].

Moving away from the use of recurrent networks as part of the encoder, there have been new trends to replace this part with Transformer encoder type attention based models as in [51], [63], [151] while still aligning with the CTC objective. Apart from this substitution, there is also literature using fully convolutional approaches to extract features and align directly on these with CTC as in [69], [70].

b) Sequence-to-sequence: Sequence-to-sequence models typically follow a decoupled architecture consisting of an encoder g , which reads the input signal and produces a representation $h = g(x)$, and a decoder f , which receives this representation and decodes the output as $y = f(g(x))$. In this case, unlike with CTC, the decoder directly outputs the sequence token by token. Generally, sequence-to-sequence models estimate the probability $P(x|y)$ as:

$$P(y|x) = \prod_{t=1}^T P(y_t|y_1, y_2 \dots y_{t-1}, x) \quad (8)$$

achieved by the chain rule of probability. More specifically, given the encoder g , the decoder f estimates the probability given by $P(y_t|y_1, y_2 \dots y_{t-1}, h)$, where $h = g(x)$. Typically, g is based on CNN [21], [150] or in combination with recurrence [143], [149], [152] or a self-attention mechanism [21], [57], [60]. The learning is achieved similarly to CTC with the log-loss $\sum_{(x,y) \in \mathcal{D}_{train}} -\log P(y|x)$, which is typically implemented using the Cross-Entropy (CE) loss. Mathematically, CE is defined as:

$$\mathcal{L}_{ce} = - \sum_{t=1}^T \sum_{k=1}^K y_{t,k} \log(\hat{y}_{t,k}),$$

where $y_{t,k}$ is a one-hot encoded vector representing the true label at timestep t for class k , $\hat{y}_{t,k}$ is the predicted probability for the same class, T is the number of timesteps, and K is the number of classes. CE measures the difference between the predicted probability distribution and the true labels, encouraging the model to assign higher probabilities to the correct outputs. In this approach, the alignment is established from all the frames/columns to a single token, unlike in CTC, where the alignment strictly follows a one-frame-per-token mapping.

In this case, the conditional probability is maximized throughout training using the Teacher Forcing technique [153], where the correct token y_t is fed to the model at the t -th timestep.

Generally, an attention mechanism is applied at the decoder side to gather contextual information from h and learn where to find the relevant content from the encoder. Depending on the manner in which they attend to h , attention can be mainly

divided into two types: (1) content-based attention [145], where the network is allowed to freely search for patterns to recognize the character to be predicted and (2) location-based attention [152], where the network also incorporates information about the locations previously consulted.

In the context of HTR, the first models using sequence-to-sequence models with only minor variation in the attention mechanism can be found in [143], [150], [154].

Beyond the “classic” sequence-to-sequence models, the architecture that has been dominating the sequence-to-sequence arena is the Transformer architecture [18]. While remaining conceptually the same as the classical models discussed above, the introduction of the self-attention layer as well as the cross-attention mechanisms turned this architecture into an improved, parallelizable and scalable version of the classical models using attention. However, the need for pre-training or synthetic data is crucial. Transformers were first used in the context of HTR in [21], in which the authors propose to train a large Transformer model with the features obtained by a ResNet as a convolutional extractor. In [63], they demonstrate that a considerable reduction of parameters both in convolutional extractor and Transformer architecture trained with CE performs equally or slightly better than a large model as in [21].

In [60], the authors train two Transformers, one from left-to-right and a right-to-left combining the predictions in a post-decoding process. Finally, relying heavily on synthetic data and pre-trained models, TrOCR [57] uses a fully Transformer-based architecture originally designed for general text recognition. With the same philosophy, DTrOCR [54] removes the encoder part and achieves higher recognition accuracy using nearly 2B of synthetic images for pre-training. The use of synthetic data generation will be studied in further sections.

c) Hybrid architectures: There have been a number of works that have opted to use a hybrid approach [49], [51], [63] by combining CTC with the sequence-to-sequence. Building on the work of [155], who introduced a hybrid method for speech recognition by combining CTC for the encoder and CE for the decoder, this approach was first adapted for HTR in [49]. In their study, the authors explore the integration of visual features learned by the encoder using CTC, with the output sequence predicted by the decoder utilizing these learned features. This is achieved by employing a hybrid loss function that merges both CTC and CE losses as follows:

$$\mathcal{L} = \lambda \mathcal{L}_{ctc} + (1 - \lambda) \mathcal{L}_{ce}, \lambda \in [0, 1] \quad (9)$$

where λ is a weighting parameter that controls the relative contribution of each loss to the overall optimization process and is typically set to 0.5. This combined loss approach has shown to be advantageous in HTR contexts, as demonstrated in [49]. In this latter approach, images are aligned with sequences using both CTC and sequence-to-sequence alignment mechanisms. In this setting, the encoder is typically unused for inference and the output is predicted as in a sequence-to-sequence approach. Recently, the concept of transcribing as a multitask learning problem has gained traction, particularly with architectures that incorporate lightweight CNNs and

Transformers. This trend is exemplified in the works [51], [63], which have explored these architectural combinations in addition to synthetic data generation for pre-training to further enhance transcription performance.

B. Beyond-line level transcription

The maturation of up-to-line level transcription techniques in Handwritten Text Recognition (HTR) has catalyzed a significant paradigm shift toward more ambitious challenges: the direct transcription of complete paragraphs and documents without requiring preliminary segmentation steps for isolating lines. This evolution represents not merely an incremental advancement but a fundamental reconceptualization of the transcription process.

A critical technical barrier in this transition stems from the vertical collapse function inherent to up-to-line level methods. While this approach effectively processes single lines by assuming that vertical features correspond to individual characters, it becomes problematic when scaling to multiple line scenarios. In paragraph or full-page transcription tasks, individual vertical slices of the image frequently contain several characters from different lines that must be independently recognized. The collapse function, by compressing these multiple characters into a single feature representation, fundamentally compromises the ability of the model to maintain proper RO and accurately distinguish between characters from different lines. Modern beyond-line level approaches specifically address this limitation through methodological adaptations and extensions. Note that the foundation of these methods is still the one that formulates the HTR field. Beyond line-level transcription methods mainly seek to address the $r(\cdot)$ function to retrieve a sequence-like data structure that represents the whole content of the page, aligned with ground truth.

Given the emerging nature of this paradigm and its profound implications for the field, we perform an individual analysis of current methodological approaches and their technical foundations.

Hereby, we find three different approaches: *attention masking* and *line unfolding*, where methods downscale the complexity of the problem into an up-to-line level challenge, either through attention-based segmentation or reshape functions, and (ii) *unconstrained* methods, where the already established sequence-to-sequence formulation from up-to-line level is adapted to learn more complex inputs. Figure 11 depicts a summary of the reviewed approaches.

1) *Attention masking*: The earliest publications beyond line-level applications found in the literature are those that apply attention matrices masking as the transcription method. These methods implement $r(\cdot)$ as $r(x) = \alpha \cdot x$, where α is a learned mask that is applied to the input feature map. These methods are applied “autoregressively”, as the mask has to be recalculated to extract all the lines from the input document, simplifying the multi-line complexity of the problem into a sequential line-by-line transcription. The presented methods in this particular approach propose different ways of computing α .

The first publications referred to these methods are the *Joint Line Segmentation and Transcription* (JLSAT) [25] and the

Scan, Attend and Read (SAAR) [67] models. Both methods are based on a combination of Convolutional Neural Networks and Multi Dimensional LSTM (MDLSTM) [100]. The first model, the SAAR, is a sequence-to-sequence approach where attention and inference are computed at the character level. The second one, JLSAT, is an optimization of the first approach that computes attention at the line level and training with the CTC loss function.

Although these approaches provided groundbreaking results—since they were the first segmentation-free models published—they were based in MDLSTM networks, which are inefficient in both training and inference. As a result, these approaches have been left behind over time, thanks to the income of Transformer-based autoregressive architectures.³ Despite this, the philosophy from these papers still prevails both in unconstrained methods, described in Section III-B3, and in the *Vertical Attention Network* (VAN) [24]. This network replaces the slow-to-train MDLSTM blocks with a convolutional network and a line-wise hybrid attention. The VAN specifically finds, approximately, the rows of a document that, at least, contain a character to transcribe. This method, although it is more efficient both in training and inference, is limited to transcribe only text paragraphs, as the rows are approximated in the vertical axis, assuming therefore that a row contains the full width of the image.

2) *Line unfolding*: In parallel to the development of masking methods, the unfolding systems started being developed. This approach works under a single-step end-to-end philosophy, as state-of-the-art line transcription does. In this case, the $r(\cdot)$ function is approached through a reshape operation that substitutes the vertical collapse between the feature extraction and the decoding stages.

The first publication that addresses this formulation is the Origaminet network [69]. The unfolding method is driven by a network composed of convolutional networks and upsampling operations. Authors assume that the produced line is long enough to contain the input document. Hereby, the network learns how to place its elements into the resulting feature map that is read in the correct RO, driven by the ground truth. This operation is done along the horizontal axis, forcing the network to implicitly learn to read from top to bottom and from left to right. Authors claim, with empirical evidence, that the model is able to transcribe full-page documents with non-trivial ROs. Although this network produced substantial results, it relies on user-defined hyperparameters, such as the line-like structure length and a fixed-size image, suggesting that the model performance is corpus-dependent.

The *Simple Predict and Align Network* (SPAN) network came briefly after Origaminet, and it is proposed as an unconstrained upgrade of this network [70]. Instead of relying on fixed-sized parameters, authors rely on an automatic unfolding operation that concatenates the rows from the feature map produced by the encoder. The concatenation step is performed from top to bottom. This operation is computed after the

³These early models were published around the same time as the Transformer, in 2017 (see Fig. 2)

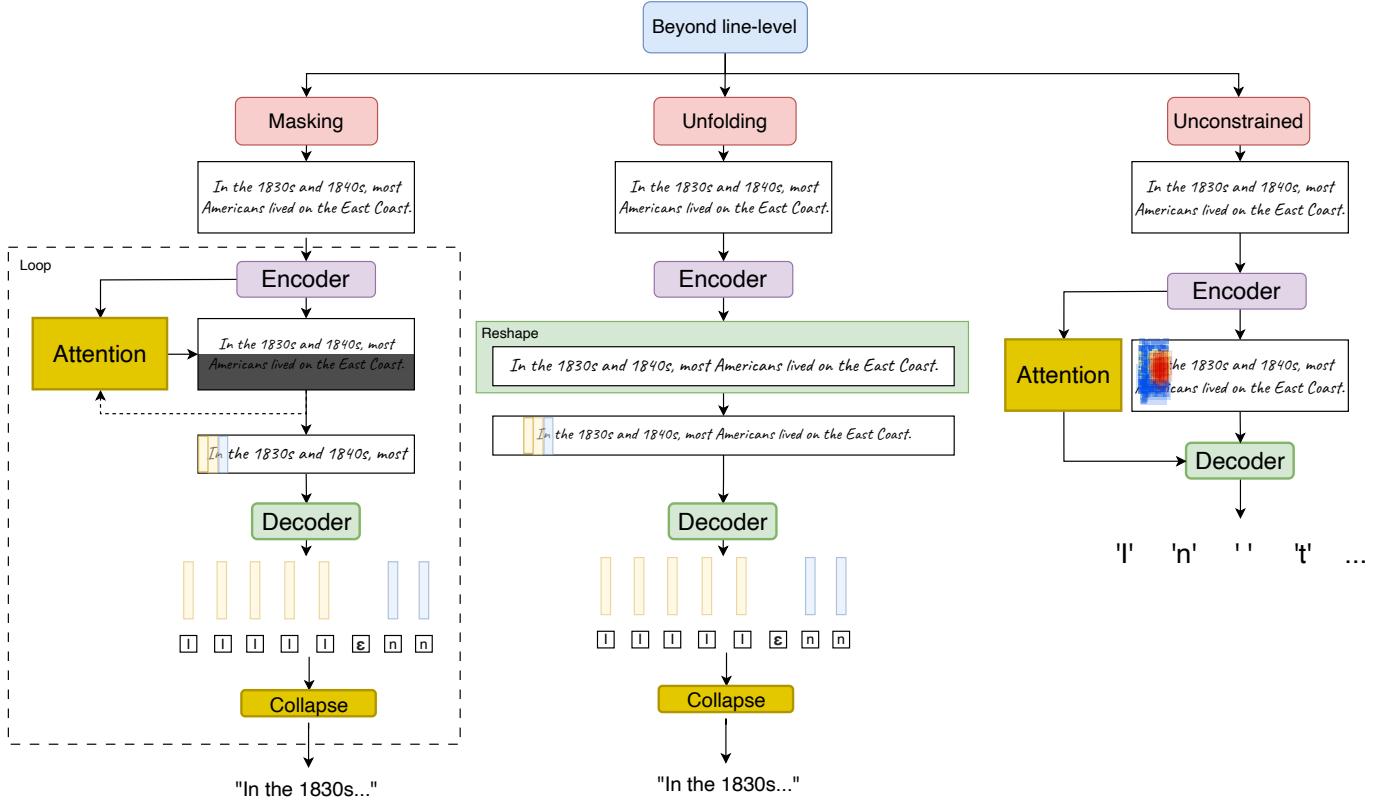


Fig. 11. Taxonomy of HTR approaches beyond line level. Methods are categorized into masking, unfolding, and unconstrained.

decision layer.⁴ This is a relevant feature of the SPAN model, as authors stress that, by implementing it this way, backpropagation is always applied in a two-dimensional context, which lets the model distinguish better the location of the different lines of the text. By sticking always to the features of the image, the SPAN method is independent of the image size, in contrast to the fixed-size images required by the Origaminet.

Despite setting a more generic unfolding method, SPAN is also limited. Specifically, the automatic unfolding operation only works on text paragraphs. When inputting more complex ROs, the network may be unable to transcribe them, as it is very difficult to find a correct alignment between the image and the ground truth. In the work of Parres et. al [156], they sort out the limitation by implementing a Swin Transformer with the pre-trained weights from DONUT to the SPAN concept, leveraging feature extraction to self-attention instead of convolutions. Although this is not deeply studied in the cited work, the authors empirically prove that this architecture is able to avoid this problem in some corpora.

3) *Unconstrained*: Unconstrained models approximate Eq. 8, but with multi-line input documents. These methods, as they are based on conditioned LMs, are free of any direct correlation between the input features and the ground truth, as in Machine Translation. Therefore, unconstrained models are able to learn an arbitrary RO. These methods arose after the conception of the Transformer architecture, as LSTM-based

models did not prove fruitful on the task.

The first published work about unconstrained methods is the *Full Page Handwritten Recognition model* (FPHR) [71]. This work proposed two key elements for these methods: (i) the usage of 2D Positional Encoding to hint the language model where each part of the image is bearing in mind spatiality and (ii) extensive synthetic pre-training to be able to learn arbitrary ROs. Nearby this publication, the TBPHTR [23] and the *Document Attention Network* (DAN) [68] were released, which holds on the same architecture concept, but with lighter implementations. The latter model is the most popular network in the community, as it is open-source, it is able to give structural details about the input document and it was proven state-of-the-art in three publicly-available datasets. Later, the MSdocTr-Lite was proposed with some optimizations of the DAN network [72]. In general, all of these systems follow the same implementation, where a generic CNN is placed as a feature extractor and a Transformer decoder acts as the language model that correlates the previous inputs with the feature map outputted by the encoder.

The most relevant differences lay in how these models are trained to handle multi-line documents. Transformers are models with a low inductive bias, meaning that they can learn any task if given enough data. However, most of the data sources HTR provides are scarce in quantity. As mentioned before, the FPHR established that a way of sorting this is by pre-training the model with synthetic data. This work, specifically, implements a brute-force training strategy, where synthetic documents are rendered with handwritten fonts,

⁴The layer that maps the feature embeddings into the output vocabulary of the network.

given WikiText content, or through the IAM database cropped words.

In the TBPHTR [23], authors resort to first pre-train on text-line images from the target dataset and, then, fine-tune the model to transcribe multi-line sources. This has a potential limitation, since it requires that the dataset labeling provides line-level information. However, it establishes an interesting takeaway: it is convenient to first pre-train on text lines, the simplest RO category, and then fine-tune into more complex structures seems to solve the task and leverage the data requirements for the model.

The DAN [68] expands further this idea by implementing a multi-stage Curriculum Learning (CL) training. First, the encoder network is trained to transcribe synthetic text lines with the CTC loss function. Then, the whole architecture is trained with synthetic documents of incrementally increasing text lines. As line quantity progresses, the synthetic generator creates increasingly complex documents. Once thirty lines are reached, a final CL is performed, where the model is fed with both synthetic and real samples. The quantity of synthetic and real data fed to the model is controlled by a linearly-scheduled probability.

The MSdocTr-Lite [72] simplifies the method proposed by the DAN, by first pre-training through text lines and, then, implementing a CL-based on text blocks complexity, which can be done either with a synthetic dataset or a layout-labeled real-world corpus.

C. Complementary techniques

In the evolving field of HTR, significant advancements in neural network architectures have been complemented by sophisticated techniques that enhance recognition accuracy and usability. Among these, decoding strategies are crucial, as they bridge the gap between the probabilistic outputs of neural networks and coherent text sequences. Decoding involves selecting the most plausible character sequences from the network's output probability distributions, a task made challenging by the variability in handwriting and the ambiguity in certain characters. Two prominent decoding methods are *greedy decoding* and *beam search decoding*, with the latter often augmented by lexicon-based constraints for improved performance.

1) *Greedy decoding*: Greedy decoding is a straightforward method for converting a neural network's output into a character sequence. At each timestep t , the method selects the character c_t with the highest probability, given the input x . This approach maximizes the likelihood of each individual character independently:

$$c_t = \arg \max_c P(c | x),$$

where $P(c | x)$ represents the probability of character c at timestep t . While greedy decoding is computationally efficient, it fails to consider dependencies between characters across timesteps. Consequently, it is prone to errors when local decisions lead to globally incoherent sequences. For example, ambiguities in handwriting might result in suboptimal predictions if the most probable character at a single timestep does

not align with the overall context of the sequence. There are several work that uses greedy decoding [21], [54], [57], [144], [149]

2) *Beam Search decoding*: Beam search improves upon greedy decoding by maintaining a fixed number of hypotheses, known as the "beam width" at each timestep. Unlike greedy decoding, which commits to a single character at each step, beam search explores multiple potential sequences in parallel, ranking them by their cumulative probabilities. The probability of a sequence $s = (c_1, c_2, \dots, c_T)$ is computed as:

$$P(s | x) = \prod_{t=1}^T P(c_t | x),$$

where T is the total number of timesteps. At each time step, the algorithm evaluates all possible extensions of the current hypotheses and retains the top- k sequences with the highest cumulative scores. This iterative process allows beam search to account for context and sequence-level dependencies, significantly improving transcription accuracy in scenarios with ambiguous handwriting. An example of this process is illustrated in Fig. 12. Works that use beam search in HTR can be found in [49], [152].

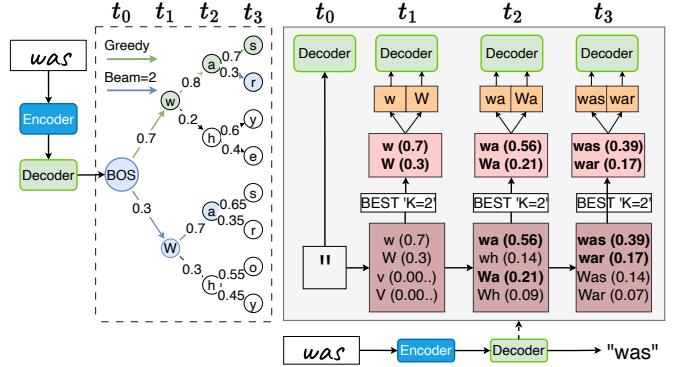


Fig. 12. Illustration of beam search with beam width $k = 2$, compared to greedy decoding. Greedy decoding selects the highest-probability character at each step ("was"), while beam search retains the top- k most probable sequences, evaluating and pruning hypotheses iteratively. This allows recovery from local ambiguities, yielding the most probable output ("was").

IV. BENCHMARKING

A. Datasets

In this section, we provide an overview of the most commonly used Latin-script datasets in HTR literature,⁵ highlighting their key features and contributions to the field. The field of HTR is rich with a variety of datasets although scarce in volume, with each dataset catering to different aspects of the recognition process. These datasets vary significantly in terms of language, historical period, number of writers, and the levels at which they provide annotations (word-level, line-level, or page-level). This diversity is critical for developing robust HTR systems capable of handling various handwriting

⁵This overview is not intended to be exhaustive. Instead, we focus on the datasets most frequently used in the HTR literature due to their significant impact, availability, and relevance for advancing the field.

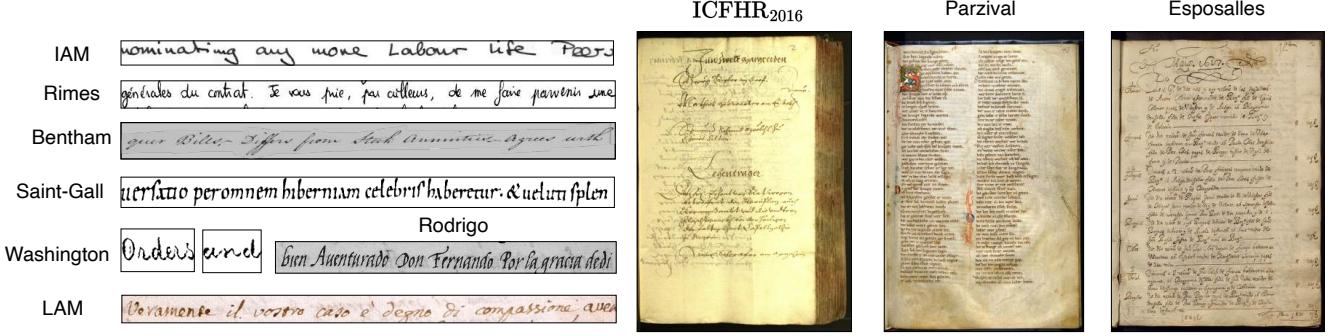


Fig. 13. Examples of handwritten text samples from various datasets commonly used in HTR research. The image showcases a range of challenges, including diverse handwriting styles, languages (e.g., English, French, Latin, Spanish, and German), and document types, from clean modern handwriting to complex historical manuscripts. These variations highlight the complexity of tasks and the need for robust recognition methods.

styles and document structures. Below, we provide an in-depth analysis on some of the prominent datasets used in HTR, highlighting their unique characteristics and contributions to the field. We provide a table summarizing the characteristics of each dataset in Tab. I. For clarity, we categorize these datasets into two main types: real data and synthetic data, reflecting their origins and usage in the HTR field.

1) *Real data*: We provide a comprehensive overview of widely used HTR datasets in Table I, covering various languages, historical periods, and annotation levels. IAM [157] remains the standard benchmark for English, while Rimes [158] is commonly used for French. Historical datasets like Bentham [159], Saint-Gall [160], and Rodrigo [161] support research on ancient scripts. Bozen [162] and Parzival [163] focus on medieval scripts. Esposalles [164] is particularly valuable for studying Catalan civil records. These datasets, with their distinct languages, time periods, and annotation levels, provide diverse challenges and opportunities for advancing technologies across different historical and linguistic contexts. Examples of each real dataset can be seen in Fig. 13.

2) *Synthetic data*: Synthetic data in HTR lacks a standardized dataset, leading each study to generate its own for pre-training. This has resulted in variations in text corpora, font choices, and dataset sizes, complicating direct comparisons across works. For synthetic data, there are two factors that characterize the generation of these data, being the text that is rendered and the synthetic fonts used. In this section, we describe the different datasets used by the different works at the two levels proposed. The first synthetic dataset for HTR was the IIIT-HWS [165]. The authors present the dataset containing 9M images as a result of rendering 90K words using the English dictionary Hunspell and 750 publicly available fonts. Kang et al. [166] introduce a synthetic dataset leveraging 387 fonts and a 430,000 word multilingual corpus (English, French, Catalan and German) to generate a total of 5.6 million images containing also words.

The first synthetic dataset on the line level is found in [21]. They rendered 138,000 handwritten lines from 387 fonts (probably the same as in [166]) but using a text corpus from unspecified online e-books. Although initially set for full pages, Singh and Karayev generated text lines from WikiText-

2 rendering 114 synthetic fonts [71]. In [63], the authors use texts from English Wikipedia articles using online available handwritten fonts. They generate a total of 10K synthetic text-line images for each training epoch. The same authors also used texts from Wikipedia and historical books alongside 21 manually selected fonts to generate training data [51]. To pre-train a Transformer, in [61] 16M text lines from news, webpages and Wikipedia are rendered from “several cursive computer fonts”. Using a very light CNN+Transformer encoder [151], the authors use the Brown corpus [167] and 400 handwriting-style fonts to render 50K lines of text. In TrOCR [57], they pre-train a large Transformer with different sizes with 684M generic lines (first stage) + 17.9M specifically for handwritten text (second stage) using 5247 synthetic fonts. For DTrOCR [54], they massively scale the pre-training of a decoder-only Transformer using 2B (2000M) lines and 5247 synthetic fonts.

For beyond-line level transcription, the first work that proposes synthetic generation was Singh et al. [71], where text was generated from the WikiText-2 dataset, as well as some images and mathematic formulae as artifacts to be detected by the system. The work of Coquenet et al. [24] composes synthetic images through the generation of lines with words from the IAM database, giving a more *realistic* image output, sacrificing vocabulary richness. The work of Constum et al. [168] acknowledges the limitations of this previous work—especially its tendency to overfit to this synthetic data—and proposes an extension of the method from Singh et al., composing images from the WikiText-3 [169] dataset in three languages: English, french and german. Moreover, this method curates a selection of synthetic handwritten fonts and is specially programmed to mimic the layout structure found in mentioned datasets, such as Rimes or Bozen.

B. Performance assessment

The assessment of transcription quality in HTR typically relies on computing edit distances. Edit distance, also known as Levenshtein distance, is a metric that quantifies how dissimilar two strings are. Specifically, it measures the number of insertions, deletions, or substitutions required to match a source string with a target one. The two well-known edit

TABLE I
DATASETS FOR THE DISTINCT LEVELS AT WHICH HTR IS PERFORMED.

Name	Language	Year	Writers	Word-level?	Line-level?	Page-level?	N. words	N. lines	N. Pages	Charset
IAM	English	1999	657	✓	✓	✓	115,520	13,355	1,539	79
Rimes	French	2009	1,300	✓	✓	✓	250,000	12,104	12,723	100
Bentham (ICFHR ₂₀₁₄)	English	1748–1832	1		✓	✓	110,000	11,537	433	86
Saint-Gall	Latin	9 th cent.	1	✓	✓		11,597	1,410	60	49
George Washington	English	1755	2	✓	✓		4,894	657	20	68
Rodrigo	Spanish	1545	1	✓	✓	✓	232,000	20,357	855	115
Bozen (ICFHR ₂₀₁₆)	German (mod.)	1470–1805	Unk.			✓	43,460	10,550	450	92
LAM	Italian	1672–1750	1		✓	✓	-	25,823	1,171	89
Parzival	German	1265–1300	3		✓	✓	23,478	4,447	45	96
Espasolles (total)	Catalan	1451–1905	1	✓	✓		67,311	7,010	202	85

distance-based metrics are the Character Error Rate (CER) and the Word Error Rate (WER). Let \hat{y} be the hypothesis and y be the ground truth sequence, and let us denote $L(\cdot, \cdot)$ as the function that calculates the Levenshtein distance between two character sequences. Then, these metrics are computed as:

$$E(\hat{Y}, Y) = \frac{\sum_{i=1}^n L(y_i, \hat{y}_i)}{\sum_{i=1}^n |y_i|} \quad (10)$$

where n is the number of samples in the test set and $|y_i|$ is the length of the sampled ground truth sequence in tokens. The only difference between CER and WER lies in the tokenization methods: characters or words, respectively. Lower values indicate better performance.

It is generally acknowledged that these metrics are suitable for up-to-line level transcription, since they effectively quantify character- and word-level errors, providing a direct measure of transcription accuracy in a constrained RO. However, as complexity increases, especially in document-level HTR, the system may make errors that are purely related to RO; for example, words that are correctly transcribed but not placed in their correct order according to the ground truth. In these situations, CER and WER may overly penalize such errors without providing insight into their cause.

This issue was firstly highlighted in the works of Antonacopoulos, Clausner and Pletschacher, where specific metrics—based on level of depth analysis—are proposed [170], [171]. These works, however, are not directly concerned with transcription quality evaluation. Some of them produce end-to-end results, while others deal with other related errors, such as LA.

An agnostic concept to solve the CER and WER strictness on alignment is the bag-of-words edit distance. This approach measures the error by comparing the frequencies of the terms of the vocabulary in both the transcript and the hypothesis. Thus, the problem is set up as a precision/recall measurement, one focusing on falsely predicted tokens and the other one on avoidance. This approach is found in several works of the literature [172]–[174], as well as it is a frequent benchmark in some competitions [174], [175].

Another concept that has been adopted to assess the transcription quality of documents is the *Hungarian Algorithm*, where the transcript words are matched with the ones in the ground truth that output the lowest edit distance. This concept has also been in several document recognition works [172],

[176], [177]. The most noteworthy approach is the “Flexible Character Accuracy” metric [172], [175].

Another idea that has been adopted in some works [142], [176], [178] is to resort to the “BLEU” measure [179]. This concept is borrowed from the Machine Translation field. It is based on matching the n-gram frequencies of the hypothesis with those of the reference sequence. Despite the avoidance of LA-related errors, this approach still suffers from the same problems that CER and WER have, which does not provide enough information about the source of the errors themselves.

Finally, the work of Vidal et al [180] proposes several modifications to the bag of words and the Hungarian Algorithm to measure adequately HTR beyond line-level transcription with regards to the RO. Specifically, they present, along with experimental insights, a set of three measures: traditional WER, the *hWER*—a Hungarian Algorithm-based WER—and Δ WER, which is the subtraction of the two previous metrics, which measures the proportion of error that is derived from RO misalignments in transcription.

C. Performance comparison

We compare state-of-the-art HTR methods across different levels of complexity, focusing on up-to-line transcription for words and lines, as well as beyond-line-level transcription for paragraphs and full documents. For the up-to-line level, we analyze the methodological evolution of these models, categorizing them based on their alignment strategies, such as CTC-based, sequence-to-sequence, and hybrid approaches, while also considering the impact of synthetic pre-training and external LMs. For beyond-line-level transcription, we explore the challenges posed by complex layouts and non-trivial RO, evaluating the effectiveness of segmentation-free models and unconstrained architectures. The comparison is structured around CER for up-to-line level (Table II) and beyond-line level (Table III). In both cases, we compare the results in terms of CER over the IAM dataset (for lines and paragraphs, respectively), as it is considered the standard benchmark for model comparison. Both tables report the architecture type along with the number of parameters (if reported in the original paper) and the amount of synthetic data used, expressed in terms of the number of examples and fonts employed. We report the CER results from the original papers for both levels. For up-to-line level transcription, we also report whether the method uses an LM and the vocabulary of the lexicon.

TABLE II

COMPARISON OF HTR MODELS AT THE WORD AND LINE LEVEL FOR THE IAM DATABASE. ‘PAR(M)’ STANDS FOR MILLION OF PARAMETERS AND ‘ALIG.’ FOR TYPE OF ALIGNMENT. †: RE-IMPLEMENTATION BY [181].

Model	Year	Par(M)	Synth	LM/Lex.	Alig.	CER
MLSTM-CTC [19]	2009	0.105	-	Yes/30k	\mathcal{L}_{ctc}	18.2
MLP/HMM [99]	2011	-	-	Yes/20k	HMM	9.8
GMM/HMM [33]	2011	-	-	Yes/50k	HMM	10.1
MLP/HMM [33]	2011	-	-	Yes/50k	Yes	12.4
BLSTM-CTC [182]	2012	-	-	Yes/-	\mathcal{L}_{ctc}	18.2
LSTM-HMM [183]	2013	-	-	Yes/50k	HMM	5.1
GMM/HMM [184]	2013	-	-	Yes/20k	HMM	8.2
LSTM-HMM [185]	2014	10.7	-	Yes/50k	HMM	4.7
MLSTM-CTC [147]	2014	0.142	-	Yes/50k	\mathcal{L}_{ctc}	5.1
MLSTM-CTC [22]	2015	24	-	Yes/50k	\mathcal{L}_{ctc}	4.4
LSTM-HMM [186]	2015	17	-	Yes/-	HMM	4.8
MLSTM-CTC [187]	2016	2.6	-	Yes/-	\mathcal{L}_{ctc}	3.5
MLSTM-Att. [25]	2016	-	-	Yes/50k	\mathcal{L}_{ce}	4.4
C-BLSTM-CTC [148]	2017	0.725	-	Yes/50k	\mathcal{L}_{ctc}	3.2
C-LSTM-CTC [20]	2017	9.3	-	Yes/-	\mathcal{L}_{ctc}	4.4
MLSTM-CTC [67]	2017	-	-	-	\mathcal{L}_{ctc}	6.6
MLSTM-CTC [188]	2017	0.686	-	-	\mathcal{L}_{ctc}	11.1
MLSTM-MLP/HMM [189]	2018	5.5	-	Yes/50k	HMM	3.6
C-BLSTM-Att.-CTC [190]	2018	-	-	-/50k	\mathcal{L}_{ctc}	5.1
C-BLSTM [144]	2018	-	90k-750f	-/90k	\mathcal{L}_{ce}	5.7
C-BGRU-GRU [149]	2018	-	-	-	\mathcal{L}_{ce}	6.8
C-BLSTM-LSTM [152]	2018	4.6	-	-	\mathcal{L}_{ce}	8.1
C-LSTM-Att. [143]	2018	-	-	-/50k	\mathcal{L}_{ce}	8.8
C-LSTM-CTC [191]	2018	-	Yes/90k	-	\mathcal{L}_{ctc}	9.8
C-LSTM-Att. [49]	2019	5	-	-	Hybr.	4.8
C-BGRU-GRU-Att. [150]	2019	-	250k-387f	-	\mathcal{L}_{ce}	
Transformer [21]	2020	100	130k-387f	-	\mathcal{L}_{ce}	4.6
Transformer [21]†	2020	100	130k-387f	-	\mathcal{L}_{ce}	7.6
C-CTC [69]	2020	3.4	-	-	\mathcal{L}_{ctc}	4.9
C-CTC [192]	2020	-	-	-	\mathcal{L}_{ctc}	6.1
C-LSTM-Att. [70]	2020	1.4	-	Yes	\mathcal{L}_{ce}	7.7
Transformer-CTC [55]	2021	8	Yes	Yes/-	\mathcal{L}_{ctc}	2.8
Transformer [60]	2021	-	1M-	-	\mathcal{L}_{ce}	5.67
C-BLSTM-GRU-Att. [56]	2021	-	-	-	\mathcal{L}_{ce}	16.6
Transformer [63]	2022	7.7	Yes/-	-	Hybr.	4.7
Transformer Enc. [151]	2022	1.74	40k-400f	-	\mathcal{L}_{ctc}	7.5
TrOCR (L) [57]	2023	558	702M-5.4kf	36-ch	\mathcal{L}_{ce}	2.9
TrOCR (M) [57]	2023	334	702M-5.4kf	36-ch	\mathcal{L}_{ce}	3.4
TrOCR (S) [57]	2023	62	702M-5.4kf	36-ch	\mathcal{L}_{ce}	4.2
TrOCR (M) [57]†	2023	385	702M-5.4kf	-	\mathcal{L}_{ce}	7.3
FCN-CTC (VAN) [62]	2023	1.7	-	-	\mathcal{L}_{ctc}	3
DTrOCR [54]	2024	105	2B-5.4kf	36-ch	\mathcal{L}_{ce}	2.4
Transformer [51]	2024	5.6	32M-21f	-	Hybr.	4.2
Transformer-CTC [193]	2024	53.5	-	-	\mathcal{L}_{ctc}	4.7

Early methods exhibited a high CER of up to 18.2 % [19]. However, these errors were significantly reduced over a span of just 4–5 years, reaching as low as 5.1 CER points [22], [147] or even lower [82], [187]. This significant improvement can largely be attributed to the integration of LMs and lexicons, which substantially facilitated the recognition process, alongside the increasing size and fine-tuning capacity of HTR models. In subsequent years, methodological advancements have focused on maintaining or surpassing these results without relying on LM and lexicons. This has come at the cost of a considerable increase in model parameters [21], [57] and computational expense, as well as the incorporation of large pre-training with synthetic data [54], [57]. These trends reflect the ongoing evolution of HTR, as discussed in the following sections.

In the case of beyond-line-level approaches, we observe a significant evolution in recognition accuracy. Early methods,

TABLE III

COMPARISON OF HTR MODELS BEYOND LINE-LEVEL FOR THE IAM DATABASE. THE ‘TYPE’ COLUMN REPRESENTS THE CATEGORIES IN SECTION III-B (MSK: MASKING; UNF: UNFOLDING; UNC: UNCONSTRAINED).

Model	Year	N. Params (M)	Synth	Type	CER
JLSAT [25]	2016	-	-	MSK	6.8
SAAR [67]	2017	-	-	MSK	16.2
Origaminet [69]	2020	14.0	-	UNF	4.7
SPAN [70]	2021	19.2	-	UNF	5.4
FPHR [71]	2021	27.9	58B-34f	UNC	6.3
TBPHTR [23]	2022	26.9	-	UNC	-
VAN [68]	2023	11.0	-	MSK	4.6
DAN [24]	2023	18.6	13k-0f	UNC	-
MSDocTr-Lite [72]	2023	26.9	-	UNC	6.4

such as the SAAR, initially reported high error rates, 16.2%.⁶ The integration of the CTC loss function marked a substantial improvement in the paradigm, as first demonstrated by the JLSAT, which reduced the error rate by 58%. The CTC approach continues to demonstrate superior performance for beyond line-level IAM. Current state-of-the-art models, the VAN and Origaminet, exhibit error rates of 4.7 and 4.6% respectively. These results contrast with the best-performant unconstrained approach, the MsDocTr-Lite, which yields a 6.4% CER. However, the specific approach to addressing this challenge appears less critical, as performance remains tightly clustered across methods. This suggests that the key factor lies in adapting the document into a line-level structure for transcription using traditional methods.

It is important to note that this comparison has limitations, as the IAM dataset only represents difficulty up to the paragraph level. This explains why models like the DAN, which reports state-of-the-art transcription performance, opt to exclude this dataset in favor of corpora such as Rimes or Bozen. These models excel at handling more complex difficulties, while paragraph-level recognition can be effectively addressed through line-level adaptation techniques, such as masking or unfolding, combined with CTC-based transcription.

V. CONCLUSIONS, CHALLENGES & FUTURE DIRECTIONS

In this work, we examined the methodological evolution of the field of Handwritten Text Recognition (HTR). We traced its progression from early handcrafted approaches relying on heuristics to transcribe individual words to modern systems capable of automatically transcribing entire documents. To structure our analysis, we divided the study into two main methodological categories: approaches that focus on transcription at the line level (words and lines) and those that extend beyond the line level (paragraphs, pages, and full documents). Additionally, we reviewed the most commonly used datasets in the literature and the evaluation methods applied to these systems. While the field has made significant advancements, there remains considerable room for improvement. In the following sections, we address the current challenges faced

⁶While JLSAT is historically recorded as the first beyond line-level method, evidence suggests that SAAR was developed earlier but published later due to publication delays.

by the field and propose potential future directions for its development.

A. Common comparison framework

A significant challenge in HTR research is the lack of a unified comparison framework, leading to three key issues: reliance on single datasets, inconsistencies in character set tokenization, and limited reproducibility. Many studies evaluate models only on datasets like IAM or Rimes, restricting generalizability and failing to assess robustness across diverse handwriting styles and languages. Furthermore, the use of synthetic data has begun to be the standard, making unfair comparisons. Additionally, advancements in tokenization, influenced by LLMs, have introduced disparities in character set sizes, with some models handling large charsets (e.g., multilingual alphabets) while others operate on smaller sets [54], [57], creating unfair comparisons due to differing task complexities. Compounding these problems, many papers do not provide code or sufficient implementation details, hindering reproducibility and reliable benchmarking. To address these issues, the field must prioritize cross-dataset evaluations, standardize tokenization practices, encourage code sharing, and adopt multilingual and domain-diverse benchmarks to enable fair and meaningful comparisons.

B. Generalization & Evaluation

In the early stages of HTR research, the focus in the literature was primarily on improving recognition accuracy on test splits from established competitions, measuring in-distribution (ID) generalization. As models have grown more capable of fitting the training distribution, the error rates on these test partitions have steadily decreased, as illustrated in Table II. However, this reduction in error often comes at the cost of overfitting to the specific test sets of datasets like IAM or Rimes. In this context, we argue that future benchmarks should prioritize evaluating performance on out-of-distribution (OOD) data to better assess models' capabilities in highly challenging scenarios. Specifically, models should demonstrate the ability to perform reasonably well in cross-lingual environments, where the training and test sets are in entirely different languages. While there has been an initial exploration of these generalization capabilities in [194], this study is limited to the HTR task at the line level, leaving broader aspects of generalization unexplored.

C. Fine-tuning & Adaptation

Many of the presented systems encounter difficulties when dealing with limited data [40], [142], which is often the case with historical sources. Most of the methods—specially the most modern ones—rely on synthetic data generation [51], [54], [57], [63], as it is a general-purpose solution that bypasses this obstacle and, sometimes, provides more general solutions. That is, most of the state-of-the-art are pre-trained models with synthetic data that are then fully-fine-tuned with the target corpora. Despite being proven an effective method, full fine-tuning may not be the most convenient technique for

this task. There exist many specific methods that may help fine-tuned HTR models to obtain better performance with few data [40], [166], [189], [195], [196].

D. Future directions

Future directions in HTR may increasingly explore the integration of Vision-Language Models (VLM) [197], [198], driven by advancements in computational capabilities and the availability of large-scale synthetic corpora, building on their successful application in Optical Character Recognition (OCR) [199], [200]. VLMs offer the potential to unify visual and linguistic information, which could enhance generalization across diverse handwriting styles and scripts. However, it remains uncertain how these models will be optimized for efficiency and real-world applicability, particularly given the high computational costs associated with large-scale training. Additionally, while synthetic data plays a crucial role in adapting real HTR datasets and improving performance [54], [57], its effectiveness in replicating real-world variability is still an open question. Future research will likely focus on assessing the feasibility and limitations of VLM in HTR, particularly for historical manuscripts and low-resource languages, where traditional methods still play a dominant role.

On the other hand, Self-Supervised Learning (SSL) represents one of the most recent and competitive paradigms within the deep learning field, aimed at alleviating the large amount of labeled data required by deep neural models for learning [201]. While traditional supervised learning relies on human-annotated corpora, SSL is designed to learn through pseudo-labeled data—where no human annotation is involved—before being applied to one or more downstream tasks (e.g., classification) [202]. Currently, this paradigm represents an effective solution to data-limited scenarios, providing multipurpose state-of-the-art models [203], [204]. SSL has already been tested in up-to-line transcription scenarios with remarkable results [205], although it has not been fully explored. It could represent a breakthrough in HTR, enabling training on real data without the need for synthetic datasets. Some adaptations could be proposed for image-to-sequence approaches—similar to how BEiT [204] is structured—as they better fit the classification pretext task.

ACKNOWLEDGMENTS

The first author is supported by grant CIACIF/2021/465 from “Programa I+D+i de la Generalitat Valenciana”. The second autor is supported by grant ACIF/2021/356 from the “Programa I+D+i de la Generalitat Valenciana”.

REFERENCES

- [1] G. Muehlberger, L. Seaward, M. Terras, S. A. Oliveira, V. Bosch, M. Bryan, S. Colutto, H. Déjean, M. Diem, S. Fiel *et al.*, “Transforming scholarship in the archives through handwritten text recognition: Transkribus as a case study,” *Journal of documentation*, vol. 75, no. 5, pp. 954–976, 2019.
- [2] J. Zakraoui, M. Saleh, S. Al-Maadeed, and J. M. AlJa’am, “A study of children emotion and their performance while handwriting arabic characters using a haptic device,” *Education and Information Technologies*, vol. 28, no. 2, pp. 1783–1808, 2023.

- [3] G. Kim and V. Govindaraju, "A lexicon driven approach to handwritten word recognition for real-time applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, 1997.
- [4] Y. He, M.-Y. Chen, and A. Kundu, "Off-line handwritten word recognition using hmm with adaptive length viterbi algorithm," *ICPR*, 1994.
- [5] D. Guilevic and C. Y. Suen, "Hmm word recognition engine," 1997.
- [6] T. Caesar, J. Glover, A. Kaltenmeier, and E. Mandler, "Handwritten word recognition using statistics," 1994.
- [7] H. Bunke, M. Roth, and E. G. Schukat-Talamazzini, "Off-line recognition of cursive script produced by a cooperative writer," 1994.
- [8] M. Y. Lu, "Character segmentation in handwritten words — an overview," *Pattern Recognition*, vol. 20, no. 1, pp. 77–96, 1996.
- [9] R. Casey and E. Lecolinet, "A survey of methods and strategies in character segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 690–706, 1996.
- [10] M. El-Yacoubi, M. Gilloux, R. Sabourin, and C. Suen, "An hmm-based approach for off-line unconstrained handwritten word modeling and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, 1999.
- [11] L. Rabiner and B. Juang, "An introduction to hidden markov models," *IEEE ASSP Magazine*, 1986.
- [12] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, 1989.
- [13] M.-Y. Chen, A. Kundu, and J. Zhou, "Off-line handwritten word recognition using a hidden markov model type stochastic network," *IEEE Trans. Pattern Anal. Mach. Intell.*, 1994.
- [14] A. Kundu, Y. He, and P. Bahl, "Recognition of handwritten word: first and second order hidden markov model based approach," *Pattern Recognition*, 1989.
- [15] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.
- [16] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: <https://doi.org/10.1162/neco.1997.9.8.1735>
- [18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.
- [19] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 855–868, 2009.
- [20] J. Puigcerver, "Are multidimensional recurrent layers really necessary for handwritten text recognition?" in *ICDAR*. IEEE, 2017, pp. 67–72.
- [21] L. Kang, P. Riba, M. Rusiñol, A. Fornés, and M. Villegas, "Pay attention to what you read: Non-recurrent handwritten text-line recognition," *Pattern Recognition*, vol. 129, p. 108766, 2022.
- [22] T. Bluche, "Deep Neural Networks for Large Vocabulary Handwritten Text Recognition," 2015.
- [23] A. C. Rouhou, M. Dhiaf, Y. Kessentini, and S. B. Salem, "Transformer-based approach for joint handwriting and named entity recognition in historical document," *Pattern Recognition Letters*, vol. 155, pp. 128–134, 2022.
- [24] D. Coquenet, C. Chatelain, and T. Paquet, "End-to-end handwritten paragraph text recognition using a vertical attention network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 508–524, 2023.
- [25] T. Bluche, "Joint line segmentation and transcription for end-to-end handwritten paragraph recognition," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 2016, pp. 838–846.
- [26] X.-F. Wang, Z.-H. He, K. Wang, Y.-F. Wang, L. Zou, and Z.-Z. Wu, "A survey of text detection and recognition algorithms based on deep learning technology," *Neurocomputing*, vol. 556, p. 126702, 2023.
- [27] W. AlKendi, F. Gechter, L. Heyberger, and C. Guyeux, "Advancements and challenges in handwritten text recognition: A comprehensive survey," *Journal of Imaging*, vol. 10, no. 1, p. 18, 2024.
- [28] X. Chen, L. Jin, Y. Zhu, C. Luo, and T. Wang, "Text recognition in the wild: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–35, 2021.
- [29] M. Benzeghiba, R. Mori, O. Deroo, S. Dupont, T. Erbes, D. Jouvet, L. Fissore, P. Laface, A. Mertins, C. Ris, R. Rose, V. Tyagi, and C. Wellekens, "Automatic speech recognition and speech variability: A review," *Speech Communication*, 2007.
- [30] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2014.
- [31] F. Jelinek, "Statistical methods for speech recognition," 1997.
- [32] S. Knerr and E. Augustin, "A neural network-hidden markov model hybrid for cursive word recognition," *14th ICPR (Cat. No.98EX170)*, 1998.
- [33] P. Dreuw, P. Doetsch, C. Plahl, and H. Ney, "Hierarchical hybrid mlp/hmm or rather mlp features for a discriminatively trained gaussian hmm: A comparison for offline handwriting recognition," *2011 18th IEEE International Conference on Image Processing*, 2011.
- [34] R. Seiler, M. Schenkel, and F. Eggemann, "Off-line cursive handwriting recognition compared with on-line recognition," *ICPR*, 1996.
- [35] Y. H. Tay, P.-M. Lallican, M. Khalid, C. Viard-Gaudin, and S. Knerr, "Offline handwritten word recognition using a hybrid neural network and hidden markov model," 2001.
- [36] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," *ICML*, 2006.
- [37] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv: Neural and Evolutionary Computing*, 2014.
- [38] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," *AAAI*, 2015.
- [39] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Recurrent neural networks," *Handbook on Neural Information Processing*, 2020.
- [40] J. C. Aradillas, J. J. Murillo-Fuentes, and P. M. Olmos, "Boosting offline handwritten text recognition in historical documents with few labeled lines," *IEEE Access*, 2021.
- [41] J.-A. Sánchez, V. Romero, A. Toselli, M. Villegas, and E. Vidal, "Icdar2017 competition on handwritten text recognition on the read dataset," *2017 14th IAPR ICDAR*, 2017.
- [42] J. A. Sánchez, V. Romero, A. Toselli, and E. Vidal, "Icfhr2014 competition (htts)," *2014 14th ICFHR*, 2014.
- [43] H. E. Abed, V. Märgner, and M. Blumenstein, "Icfhr 2010 - competitions overview," 2010.
- [44] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2015.
- [45] D. Kass and E. Vats, "Attentionhr: Handwritten text recognition based on attention encoder-decoder networks," 2022.
- [46] L. Kumari, S. Singh, V. V. S. Rathore, A. Sharma, L. Kumari, S. Singh, V. V. S. Rathore, and A. Sharma, "Lexicon and attention based handwritten text recognition system," 2022.
- [47] A. Abdallah, M. A. Hamada, and D. Nursetiov, "Attention-based fully gated cnn-bgru for russian handwritten text," *Journal of Imaging*, 2020.
- [48] D. Coquenet, C. Chatelain, and T. Paquet, "End-to-end handwritten paragraph text recognition using a vertical attention network," 2022.
- [49] J. Michael, R. Labahn, T. Grüning, and J. Zöllner, "Evaluating sequence-to-sequence models for handwritten text recognition," *ICDAR*, 2019.
- [50] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ArXiv*, vol. abs/2010.11929, 2020.
- [51] K. Barrere, Y. Soullard, A. Lemaitre, and B. Coüasnon, "Training transformer architectures on few annotated data: an application to historical handwritten text recognition," *IJDAR*, 2024.
- [52] S. Paul, G. Madan, A. Mishra, N. Hegde, P. Kumar, and G. Aggarwal, "Weakly supervised information extraction from inscrutable handwritten document images," *arXiv*, 2023.
- [53] D. V. Sang and L. T. B. Cuong, "Improving crnn with efficientnet-like feature extractor and multi-head attention for text recognition," *SoICT 2019*, 2019.
- [54] M. Fujitake, "Dtrocr: Decoder-only transformer for optical character recognition," *arXiv.org*, 2023.
- [55] D. H. Diaz, R. Ingle, S. Qin, A. Bissacco, and Y. Fujii, "Rethinking text line recognition models," *arXiv*, 2021.
- [56] J. Poulos and R. Valle, "Character-based handwritten text transcription with attention networks," *Neural Computing and Applications*, 2021.
- [57] M. Li, T. Lv, J. Chen, L. Cui, Y. Lu, D. Florencio, C. Zhang, Z. Li, and F. Wei, "Trocr: Transformer-based optical character recognition with pre-trained models," *AAAI*, 2023.

- [58] S. Momeni and B. BabaAli, "A transformer-based approach for arabic offline handwritten text recognition," *arXiv.org*, 2023.
- [59] A. Mostafa, O. Mohamed, A. Ashraf, A. Elbehery, S. Jamal, G. Khoriba, and A. Ghoneim, "Ocformer: A transformer-based model for arabic handwritten text recognition," *2021 MIUCC*, 2021.
- [60] C. Wick, J. Zöllner, and T. Grüning, "Transformer for handwritten text recognition using bidirectional post-decoding," *ICDAR*, 2021.
- [61] C. Wick, J. Zöllner, and T. Grüning, "Rescoring sequence-to-sequence models for text line recognition with ctc-prefixes," in *Document Analysis Systems: 15th IAPR International Workshop, DAS 2022, La Rochelle, France, May 22–25, 2022, Proceedings*. Berlin, Heidelberg: Springer-Verlag, 2022, p. 260–274. [Online]. Available: https://doi.org/10.1007/978-3-031-06555-2_18
- [62] D. Coquenet, C. Chatelain, and T. Paquet, "Dan: a segmentation-free document attention network for handwritten document recognition," 2023.
- [63] K. Barrere, Y. Soullard, A. Lemaitre, and B. Couësnon, "A light transformer-based architecture for handwritten text recognition," 2022.
- [64] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," *Neural Information Processing Systems*, 2019.
- [65] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *arXiv: Computation and Language*, 2019.
- [66] Z. Lu, S. He, X. Zhu, L. Zhang, Y.-Z. Song, and T. Xiang, "Simpler is better: Few-shot semantic segmentation with classifier weight transformer," 2021.
- [67] T. Bluche, J. Louradour, and R. O. Messina, "Scan, attend and read: End-to-end handwritten paragraph recognition with MDLSTM attention," in *ICDAR*. IEEE, 2017, pp. 1050–1055.
- [68] D. Coquenet, C. Chatelain, and T. Paquet, "Dan: a segmentation-free document attention network for handwritten document recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.
- [69] M. Yousef and T. E. Bishop, "Origaminet: Weakly-supervised, segmentation-free, one-step, full page textrecognition by learning to unfold," in *CVPR*, June 2020.
- [70] D. Coquenet, C. Chatelain, and T. Paquet, "Span: A simple predict & align network for handwritten paragraph recognition," in *ICDAR*, ser. Lecture Notes in Computer Science, vol. 12823, 2021, pp. 70–84.
- [71] S. S. Singh and S. Karayev, "Full page handwriting recognition via image to sequence extraction," in *ICDAR*, ser. Lecture Notes in Computer Science, J. Lladós, D. Lopresti, and S. Uchida, Eds., vol. 12823. Springer, 2021, pp. 55–69.
- [72] "Msdoctr-lite: A lite transformer for full page multi-script handwriting recognition," *Pattern Recognition Letters*, vol. 169, pp. 28–34, 2023.
- [73] C. De la Higuera and J. Oncina, "The most probable string: an algorithmic study," *Journal of Logic and Computation*, vol. 24, no. 2, pp. 311–330, 2014.
- [74] T. Ghosh, S. Sen, S. Obaidullah, K. Santosh, K. Roy, and U. Pal, "Advances in online handwritten recognition in the last decades," *Computer Science Review*, 2022.
- [75] R. R. Ingle, Y. Fujii, T. Deselaers, J. Baccash, and A. C. Popat, "A scalable handwritten text recognition system," 2019.
- [76] A. Namboodiri and A. K. Jain, "Online handwritten script recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004.
- [77] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *Corr*, vol. abs/1507.05717, 2015. [Online]. Available: <http://arxiv.org/abs/1507.05717>
- [78] S. Fang, H. Xie, Y. Wang, Z. Mao, and Y. Zhang, "Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition," *Computer Vision and Pattern Recognition*, 2021.
- [79] X. Chen, L. Jin, Y. Zhu, C. Luo, and T. Wang, "Text recognition in the wild: A survey," *ACM Computing Surveys*, 2021.
- [80] F. Naiemi, V. Ghods, and H. Khaleqi, "Scene text detection and recognition: a survey," *Multimedia Tools and Applications*, vol. 81, no. 14, pp. 20 255–20 290, 2022.
- [81] I. Pratikakis, K. Zagoris, B. Gatos, J. Puigcerver, A. Toselli, and E. Vidal, "Icfhr2016 handwritten keyword spotting competition (h-kws 2016)," *2016 15th ICFHR*, 2016.
- [82] B. Wicht, A. Fischer, and J. Hennebert, "Deep learning features for handwritten keyword spotting," *2016 23rd ICPR*, 2016.
- [83] A. H. Toselli, E. Vidal, V. Romero, and V. Frinken, "Hmm word graph based keyword spotting in handwritten document images," *Information Sciences*, 2016.
- [84] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke, "A novel word spotting method based on recurrent neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.
- [85] M. Li, Y. Xu, L. Cui, S. Huang, F. Wei, Z. Li, and M. Zhou, "Docbank: A benchmark dataset for document layout analysis," *International Conference on Computational Linguistics*, 2020.
- [86] B. Pfitzmann, C. Auer, M. Dolfi, A. Nassar, and P. Staar, "Doclaynet: A large human-annotated dataset for document-layout segmentation," *Knowledge Discovery and Data Mining*, 2022.
- [87] G. M. Binmakhashen and S. Mahmoud, "Document layout analysis," *ACM Computing Surveys*, 2019.
- [88] S. Appalaraju, B. A. Jasani, B. Kota, Y. Xie, and R. Manmatha, "Docformer: End-to-end transformer for document understanding," *IEEE ICCV*, 2021.
- [89] J. Gu, J. Kuen, V. I. Morariu, H. Zhao, R. Jain, N. Barmpalias, A. Nenkova, and T. Sun, "Unidoc: Unified pretraining framework for document understanding," 2021.
- [90] J. V. Landeghem, R. P. Tito, L. Borchmann, M. Pietruszka, P. J'ozia, R. Powalski, D. Jurkiewicz, M. Coustaty, B. Ackaert, E. Valveny, M. B. Blaschko, S. Moens, and T. Stanislawek, "Document understanding dataset and evaluation (dude)," *IEEE ICCV*, 2023.
- [91] G. Kim, T. Hong, M. Yim, J. Nam, J. Park, J. Yim, W. Hwang, S. Yun, D. Han, and S. Park, "Ocr-free document understanding transformer," *ECCV*, 2021.
- [92] L. Kang, P. Riba, Y. Wang, M. Rusiñol, A. Fornés, and M. Villegas, "Ganwriting: Content-conditioned generation of styled handwritten word images," *Lecture Notes in Computer Science*, 2020.
- [93] V. Pippi, S. Cascianelli, and R. Cucchiara, "Handwritten text generation from visual archetypes," *CVPR*, 2023.
- [94] B. Vanherle, V. Pippi, S. Cascianelli, N. Michiels, F. V. Reeth, and R. Cucchiara, "Vatr++: Choose your words wisely for handwritten text generation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [95] S. Srihari and R. Bozinovic, "A multi-level perception approach to reading cursive script," *IJCAI*, 1987.
- [96] S. Edelman, T. Flash, and S. Ullman, "Reading cursive handwriting by alignment of letter prototypes," *International Journal of Computer Vision*, 1991.
- [97] Y. He, M.-Y. Chen, and A. Kundu, "Handwritten word recognition using hmm with adaptive length viterbi algorithm," *ICASSP*, 1992.
- [98] B. Plessis, A. Sicsu, L. Heutte, E. Menu, E. Lecolinet, O. Debon, and J. Moreau, "A multi-classifier combination strategy for the recognition of handwritten cursive words," *ICDAR '93*, 1993.
- [99] S. E. Boquera, M. J. C. Bleda, J. Gorbe-Moya, and F. Zamora-Martínez, "Improving offline handwritten text recognition with hybrid hmm/ann models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.
- [100] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in *Advances in Neural Information Processing Systems*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds., vol. 21. Curran Associates, Inc., 2008.
- [101] A. L. Koerich, R. Sabourin, and C. Y. Suen, "Large vocabulary off-line handwriting recognition: A survey," *Pattern Analysis and Applications*, 2003.
- [102] V. Lavrenko, T. Rath, and R. Manmatha, "Holistic word recognition for handwritten historical documents," *First International Workshop on Document Image Analysis for Libraries, 2004. Proceedings.*, 2004.
- [103] A. Vinciarelli, "A survey on off-line cursive word recognition," *Pattern Recognition*, 2002.
- [104] T. Steinherz, E. Rivlin, and N. Intrator, "Offline cursive script word recognition ? a survey," 1999.
- [105] R. M. Bozinovic and S. N. Srihari, "Off-line cursive script word recognition," 1995.
- [106] M. Morita, J. Facon, Flávio Bortolozzi, S. J. A. Garnés, and R. Sabourin, "Mathematical morphology and weighted least squares to correct handwriting baseline skew," *ICDAR '99 (Cat. No.PR00318)*, Sep. 1999. [Online]. Available: <https://doi.org/10.1109/ICDAR.1999.791816>
- [107] Myriam Côté, Eric Lecolinet, Mohamed Cheriet, and Ching Y. Suen, "Automatic reading of cursive scripts using a reading model and perceptual concepts," *International Journal on Document Analysis and Recognition*, Feb. 1998. [Online]. Available: <https://doi.org/10.1007/s100320050002>
- [108] Alessandro Vinciarelli and Juergen Luettin, "OFF-LINE CURSIVE SCRIPT RECOGNITION BASED ON CONTINUOUS DENSITY HMM," Jan. 2004.

- [109] Jinhai Cai and Zhi-Qiang Liu, "OFF-LINE UNCONSTRAINED HANDWRITTEN WORD RECOGNITION," *International Journal of Pattern Recognition and Artificial Intelligence*, May 2000. [Online]. Available: <https://doi.org/10.1142/s0218001400000180>
- [110] A. Senior and A. J. Robinson, "An off-line cursive handwriting recognition system," *IEEE Trans. Pattern Anal. Mach. Intell.*, 1998.
- [111] Ergina Kavallieratou, Nikos Fakotakis, and G. Kokkinakis, "Slant estimation algorithm for OCR systems," *Pattern Recognition*, Dec. 2001. [Online]. Available: [https://doi.org/10.1016/s0031-3203\(00\)00153-9](https://doi.org/10.1016/s0031-3203(00)00153-9)
- [112] Y. Lu and M. Shridhar, "Character segmentation in handwritten words — an overview," *Pattern Recognition*, 1996.
- [113] G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis, "Text line and word segmentation of handwritten documents," *Pattern Recognition*, 2009.
- [114] R. Casey and E. Lecolinet, "A survey of methods and strategies in character segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, 1996.
- [115] J. Favata, "General word recognition using approximate segment-string matching," 1997.
- [116] J. T. Favata and S. Srihari, "Offline recognition of handwritten cursive words," *Electronic Imaging*, 1992.
- [117] C. Burges, J. Ben, J. Denker, Y. LeCun, and C. Nohl, "Off line recognition of handwritten postal words using neural networks," *Int. J. Pattern Recognit. Artif. Intell.*, 1993.
- [118] M.-Y. Chen, A. Kundu, and S. Srihari, "Handwritten word recognition using continuous density variable duration hidden markov model," *ICASSP*, 1993.
- [119] M.-Y. Chen and A. Kundu, "A complement to variable duration hidden markov model in handwritten word recognition," *Proceedings of 1st International Conference on Image Processing*, 1994.
- [120] A. Senior and F. Fallside, "An oline cursive script recognition system using recurrent error propagation networks," 2012.
- [121] M. Gilloux, "Writer adaptation for handwritten word recognition using hidden markov models," 1994.
- [122] M. Gilloux, M. Leroux, and J. Bertille, "Strategies for handwritten words recognition using hidden markov models," *ICDAR '93*, 1993.
- [123] M.-Y. Cheii and N. Command, "Variable duration hidden markov model and morphological segmentation for handwritten word recognition," 1993.
- [124] M. Chen, A. Kundu, and J. Zhou, "Off-line handwritten word recognition (hwr) using a single contextual hidden markov model," 1992.
- [125] G. Saon and A. Belaid, "Off-line handwritten word recognition using a mixed hmm-mrf approach," 1997.
- [126] H. Bunke, M. Roth, and E. G. Schukat-Talamazzini, "Off-line cursive handwriting recognition using hidden markov models," *Pattern Recognition*, 1995.
- [127] S. Madhvanath and V. Govindaraju, "Holistic lexicon reduction for handwritten word recognition," *Electronic Imaging*, 1996.
- [128] C. Parisse, "Global word shape processing in off-line recognition of handwriting," *IEEE Trans. Pattern Anal. Mach. Intell.*, 1996.
- [129] S. Madhvanath and V. Krpasundar, "Pruning large lexicons using generalized word shape descriptors," *ICDAR*, 1997.
- [130] W. Cho, S.-W. Lee, and J. H. Kim, "Modeling and recognition of cursive words with hidden markov models," *Pattern Recognition*, 1995.
- [131] M. Mohamed and P. Gader, "Handwritten word recognition using segmentation-free hidden markov modeling and segmentation-based dynamic programming techniques," *IEEE Trans. Pattern Anal. Mach. Intell.*, 1996.
- [132] M. Gilloux, "Hidden markov models in handwriting recognition," 1994.
- [133] T. Paquet and Y. Lecourtier, "Automatic reading of the literal amount of bank checks," *Journal of Machine Vision and Applications*, 1993.
- [134] W. P. d. Waard, "An optimised minimal edit distance for hand-written word recognition," *Pattern Recognition Letters*, 1995.
- [135] M. Gilloux, M. Leroux, and J. M. Bertille, "Strategies for cursive script recognition using hidden markov models," *machine vision applications*, 1995.
- [136] P. Gader and W.-T. Chen, "Word-level optimization of dynamic programming-based handwritten word recognition algorithms," *Electronic Imaging*, 1999.
- [137] P. D. Gader, M. A. Mohamed, and J. M. Keller, "Dynamic programming-based handwritten word recognition using the choquet fuzzy integral as the match function," *Journal of Electronic Imaging*, 1996.
- [138] G. Kim and V. Govindaraju, "Handwritten word recognition for real-time applications," *ICDAR*, 1995.
- [139] F. Kimura, N. Kayahara, Y. Miyake, and M. Shridhar, "Machine and human recognition of segmented characters from handwritten words," *ICDAR*, 1997.
- [140] G. Kim and V. Govindaraju, "Recognition of handwritten phrases as applied to street name images," *CVPR*, 1996.
- [141] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.
- [142] C. Wigington, C. Tensmeyer, B. Davis, W. Barrett, B. Price, and S. Cohen, "Start, follow, read: End-to-end full-page handwriting recognition," in *ECCV*, 2018, pp. 367–383.
- [143] J. Sueiras, V. Ruiz, A. Sanchez, and J. F. Velez, "Offline continuous handwriting recognition using sequence to sequence neural networks," *Neurocomputing*, 2018.
- [144] K. Dutta, P. Krishnan, M. Mathew, and C. V. Jawahar, "Improving cnn-rnn hybrid networks for handwriting recognition," *ICFHR*, 2018.
- [145] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, Y. Bengio and Y. LeCun, Eds., 2015.
- [146] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602–610, 2005, iJCNN 2005.
- [147] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour, "Dropout improves recurrent neural networks for handwriting recognition," in *ICFHR*, 2014, pp. 285–290.
- [148] T. Bluche and R. O. Messina, "Gated convolutional recurrent neural networks for multilingual handwriting recognition," *2017 14th IAPR ICDAR*, 2017.
- [149] L. Kang, J. I. Toledo, P. Riba, M. Villegas, A. Fornés, and M. Rusiñol, "Convolve, Attend and Spell: An Attention-based Sequence-to-Sequence Model for Handwritten Word Recognition," *Lecture Notes in Computer Science*, 2018.
- [150] L. Kang, P. Riba, M. G. Villegas, A. Fornés, and M. Rusiñol, "Candidate fusion: Integrating language modelling into a sequence-to-sequence handwritten word recognition architecture," 2019.
- [151] R. d'Arce, T. Norton, S. Hannuna, and N. Cristianini, "Self-attention Networks for Non-recurrent Handwritten Text Recognition," in *Frontiers in Handwriting Recognition*, U. Porwal, A. Fornés, and F. Shafait, Eds. Cham: Springer International Publishing, 2022, vol. 13639, pp. 389–403, series Title: Lecture Notes in Computer Science. [Online]. Available: https://link.springer.com/10.1007/978-3-031-21648-0_27
- [152] A. Chowdhury and L. Vig, "An efficient end-to-end neural model for handwritten text recognition," in *British Machine Vision Conference*, 2018. [Online]. Available: <https://api.semanticscholar.org/CorpusID:49907144>
- [153] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, vol. 1, no. 2, pp. 270–280, 1989.
- [154] J. Poulos and R. Valle, "Character-based handwritten text transcription with attention networks," *Neural computing & applications (Print)*, 2017.
- [155] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *ICASSP*, 2017, pp. 4835–4839.
- [156] D. Parres, D. Anitei, and R. Paredes, "Handwritten document recognition using pre-trained vision transformers," in *Document Analysis and Recognition - ICDAR 2024*. Cham: Springer Nature Switzerland, 2024, pp. 173–190.
- [157] U.-V. Marti and H. Bunke, "The iam-database: an english sentence database for offline handwriting recognition," *International Journal on Document Analysis and Recognition*, 2002.
- [158] C. Kermorvant and J. Louradour, "Handwritten mail classification experiments with the rimes database," in *ICFHR*. IEEE Computer Society, 2010, pp. 241–246. [Online]. Available: <https://doi.org/10.1109/ICFHR.2010.45>
- [159] T. Causer and V. Wallace, "Building a volunteer community: Results and findings from transcribe bentham," *Digital Humanities Quarterly*, vol. 6, no. 2, 2012.
- [160] G. Scherrer, *Verzeichniss der Handschriften der Stiftsbibliothek von St. Gallen*. Halle, 1875.
- [161] N. Serrano, F. Castro, and A. Juan, "The RODRIGO database," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapia, Eds. Valletta, Malta: European Language Resources

- Association (ELRA), May 2010. [Online]. Available: http://www.ijrc-conf.org/proceedings/ijrc2010/pdf/477_Paper.pdf
- [162] J. A. Sánchez, V. Romero, A. H. Toselli, and E. Vidal, “Icfhr2016 competition on handwritten text recognition on the read dataset,” in *ICFHR*, 2016, pp. 630–635.
- [163] A. Keller, V. Frinken, and H. Bunke, “Lexicon-free handwritten word spotting using character hmms,” *Pattern Recognition Letters - PRL*, vol. 33, p. 934–942, 05 2012.
- [164] V. Romero, A. FornéS, N. Serrano, J. A. SáNchez, A. H. Toselli, V. Frinken, E. Vidal, and J. LladóS, “The esposalles database: An ancient marriage license corpus for off-line handwriting recognition,” *Pattern Recognit.*, vol. 46, no. 6, p. 1658–1669, Jun. 2013.
- [165] P. Krishnan and C. V. Jawahar, “Generating synthetic data for text recognition,” *CoRR*, vol. abs/1608.04224, 2016. [Online]. Available: <http://arxiv.org/abs/1608.04224>
- [166] L. Kang, M. Rusinol, A. Fornes, P. Riba, and M. Villegas, “Unsupervised adaptation for synthetic-to-real handwritten word recognition,” *workshop on applications of computer vision*, 2020.
- [167] W. N. Francis and H. Kucera, “Brown corpus manual,” Department of Linguistics, Brown University, Providence, Rhode Island, US, Tech. Rep., 1979.
- [168] T. Constum, P. Tranouez, and T. Paquet, “Daniel: A fast document attention network for information extraction and labelling of handwritten documents,” 2024. [Online]. Available: <https://arxiv.org/abs/2407.09103>
- [169] S. Merity, C. Xiong, J. Bradbury, and R. Socher, “Pointer sentinel mixture models,” 2016.
- [170] C. Clausner, S. Pletschacher, and A. Antonacopoulos, “The significance of reading order in document recognition and its evaluation,” in *ICDAR*. IEEE, 2013, pp. 688–692.
- [171] S. Pletschacher, C. Clausner, and A. Antonacopoulos, “European newspapers OCR workflow evaluation,” in *Proceedings of the 3rd International Workshop on Historical Document Imaging and Processing*. New York, NY, USA: Association for Computing Machinery, 2015, p. 39–46.
- [172] C. Clausner, S. Pletschacher, and A. Antonacopoulos, “Flexible character accuracy measure for reading-order-independent evaluation,” *Pattern Recognit. Lett.*, vol. 131, pp. 390–397, 2020.
- [173] P. B. Ströbel, S. Clematide, and M. Volk, “How much data do you need? about the creation of a ground truth for black letter and the effectiveness of neural OCR,” in *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020. [Online]. Available: <https://aclanthology.org/2020.ijrc-1.436>
- [174] C. Clausner, A. Antonacopoulos, and S. Pletschacher, “ICDAR2017 competition on recognition of documents with complex layouts - RDCL2017,” in *2017 14th ICDAR*, vol. 1. IEEE, 2017, pp. 1404–1410.
- [175] ——, “ICDAR2019 competition on recognition of documents with complex layouts - RDCL2019,” in *ICDAR*. IEEE, 2019, pp. 1521–1526.
- [176] C. Tensmeyer and C. Wigington, “Training full-page handwritten text recognition models without annotated line breaks,” in *ICDAR*. IEEE, 2019, pp. 1–8.
- [177] S. Long, S. Qin, D. Panteleev, A. Bissacco, Y. Fujii, and M. Raptis, “Towards end-to-end unified scene text detection and layout analysis,” in *CVPR*, 2022, pp. 1049–1059.
- [178] J. A. Sánchez, V. Romero, A. H. Toselli, M. Villegas, and E. Vidal, “ICDAR2017 competition on handwritten text recognition on the READ dataset,” in *2017 14th ICDAR*, vol. 01, 2017, pp. 1383–1388.
- [179] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.
- [180] E. Vidal, A. H. Toselli, A. Ríos-Vila, and J. Calvo-Zaragoza, “End-to-end page-level assessment of handwritten text recognition,” *Pattern Recognition*, vol. 142, p. 109695, 2023.
- [181] S. Cascianelli, M. Cornia, L. Baraldi, and R. Cucchiara, “Boosting modern and historical handwritten text recognition with deformable convolutions,” *IJDAR*, vol. 25, pp. 207 – 217, 2022.
- [182] M. Liwicki, A. Graves, and H. Bunke, “Neural Networks for Handwriting Recognition,” *Computational Intelligence Paradigms in Advanced Pattern Classification*, 2012.
- [183] M. Kozielski, P. Doetsch, and H. Ney, “Improvements in RWTH’s System for Off-Line Handwriting Recognition,” *ICDAR*, 2013.
- [184] M. Kozielski, D. Rybach, S. Hahn, R. Schlüter, and H. Ney, “Open vocabulary handwriting recognition using combined word-level and character-level language models,” *ICASSP*, 2013.
- [185] P. Doetsch, M. Kozielski, and H. Ney, “Fast and Robust Training of Recurrent Neural Networks for Offline Handwriting Recognition,” *ICFHR*, 2014.
- [186] P. Voigtlaender, P. Doetsch, S. Wiesler, R. Schlüter, and H. Ney, “Sequence-discriminative training of recurrent neural networks,” *ICASSP*, 2015.
- [187] P. Voigtlaender, P. Doetsch, and H. Ney, “Handwriting Recognition with Large Multidimensional Long Short-Term Memory Recurrent Neural Networks,” *2016 15th ICFHR*, 2016.
- [188] Z. Chen, Y. Wu, F. Yin, and C.-L. Liu, “Simultaneous script identification and handwriting recognition via multi-task learning of recurrent neural networks,” in *2017 14th IAPR ICDAR*, vol. 01, 2017, pp. 525–530.
- [189] D. Castro, B. Bezerra, and M. Valençā, “Boosting the Deep Multidimensional Long-Short-Term Memory Network for Handwritten Recognition Systems,” *ICFHR*, 2018.
- [190] L. Gui, X. Liang, X. Chang, and A. Hauptmann, “Adaptive Context-aware Reinforced Agent for Handwritten Text Recognition,” *British Machine Vision Conference*, 2018.
- [191] P. Krishnan, K. Dutta, and C. V. Jawahar, “Word spotting and recognition using deep embedding,” *2018 13th DAS*, 2018.
- [192] X. Huang, L. Qiao, W. Yu, J. Li, and Y. Ma, “End-to-end sequence labeling via convolutional recurrent neural network with a connectionist temporal classification layer,” *International Journal of Computational Intelligence Systems*, vol. 13, pp. 341–351, 2020. [Online]. Available: <https://doi.org/10.2991/ijcis.d.200316.001>
- [193] Y. Li, D. Chen, T. Tang, and X. Shen, “HTR-VT: Handwritten text recognition with vision transformer,” *Pattern Recognition*, 2024.
- [194] C. Garrido-Munoz and J. Calvo-Zaragoza, “On the generalization of handwritten text recognition models,” 2024. [Online]. Available: <https://arxiv.org/abs/2411.17332>
- [195] J. Kohút and M. Hradis, “Fine-tuning is a surprisingly effective domain adaptation baseline in handwriting recognition,” *Lecture Notes in Computer Science*, 2023.
- [196] A. K. Bhunia, S. Ghose, A. Kumar, P. N. Chowdhury, A. Sain, and Y.-Z. Song, “Metahtr: Towards writer-adaptive handwritten text recognition,” *Computer Vision and Pattern Recognition*, 2021.
- [197] L. B. et al., “Paligemma: A versatile 3b vlm for transfer,” *ArXiv*, vol. abs/2407.07726, 2024.
- [198] H. Liu, C. Li, Q. Wu, and Y. J. Lee, “Visual instruction tuning,” *ArXiv*, vol. abs/2304.08485, 2023.
- [199] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, and M. Zhou, “Layoutlm: Pre-training of text and layout for document image understanding,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD ’20. Association for Computing Machinery, 2020, p. 1192–1200. [Online]. Available: <https://doi.org/10.1145/3394486.3403172>
- [200] G. Kim, T. Hong, M. Yim, J. Nam, J. Park, J. Yim, W. Hwang, S. Yun, D. Han, and S. Park, “Ocr-free document understanding transformer,” in *ECCV*, 2021.
- [201] Y. LeCun, “Self-supervised learning: The dark matter of intelligence.” [Online]. Available: <https://ai.facebook.com/blog/self-supervised-learning-the-dark-matter-of-intelligence/>
- [202] U. Ozbulak, H. J. Lee, B. Boga, E. T. Anzaku, H. Park, A. Van Messem, W. De Neve, and J. Vankerschaver, “Know your self-supervised learning: A survey on image-based generative and discriminative training,” *arXiv preprint arXiv:2305.13689*, 2023.
- [203] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” in *NAAACL-HLT 2019, Vol. 1*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 4171–4186.
- [204] H. Bao, L. Dong, and F. Wei, “Beit: BERT pre-training of image transformers,” in *10th ICLR, Apr 2022, Virtual, France*, 2022.
- [205] A. Aberdam, R. Litman, S. Tsiper, O. Anschel, R. Slossberg, S. Mazor, R. Manmatha, and P. Perona, “Sequence-to-sequence contrastive learning for text recognition,” in *CVPR 2021*. Computer Vision Foundation / IEEE, 2021, pp. 15 302–15 312.