# Automatic Annotation of Handwritten Document Images at Word Level

Ajoy Mondal
CVIT, IIIT Hyderabad
India
ajoy.mondal@iiit.ac.com

Krishna Tulsyan
CVIT, IIIT Hyderabad
India
krishna.tulsyan@research.iiit.ac.in

C. V. Jawahar
CVIT, IIIT Hyderabad
India
jawahar@iiit.ac.in

## ABSTRACT

Recent development in deep learning-based recognizers needs a large annotated corpus for creating the model. Manually annotating a large corpus is time-consuming, costly, and tedious. In this work, we propose a framework for automatic annotation at the word level for given handwritten data and corresponding text sequences (or corpora). The proposed framework consists of five modules (i) pre-processing, (ii) word detection, (iii) word recognition, (iv) alignment, and (v) manual correction and verification. The pre-processing module cleans the image and crops the text region from an image. Word detection and recognition modules localize and recognize words. It is necessary to align words in the sequence with the word images during detection and recognition because of errors in writing. The alignment module aligns words in text sequence to the word images. The human annotator will correct the errors in the automatic annotation process and verify the document. Finally, we created an annotated dataset containing word images and their corresponding ground truth transcriptions. In this work, we demonstrate the proposed tool for annotating 14 sets corresponding to 13 Indic languages and English. Each set contains 15000 handwritten document images. On an extensive collection of handwritten document images in 14 languages, 80% of words are correctly annotated by the automatic annotation tool, while the remaining 20% are corrected manually.

## CCS CONCEPTS

• **Human-centered computing → Graphical user interfaces**.

## KEYWORDS

Automatic annotation framework, handwritten documents, word detection, word recognition, alignment

**ACM Reference Format:**
Ajoy Mondal, Krishna Tulsyan, and C. V. Jawahar. 2022. Automatic Annotation of Handwritten Document Images at Word Level. In *Proceedings of the Thirteenth Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP'22), December 8–10, 2022, Gandhinagar, India,* Soma Biswas, Shanmuganathan Raman, and Amit K Roy-Chowdhury (Eds.). ACM, New York, NY, USA, Article 45, 9 pages. https://doi.org/10.1145/3571600.3571645


*ICVGIP'22, December 2022, Gandhinagar, India*

ACM ISBN 978-1-4503-9822-0/22/12.
https://doi.org/10.1145/3571600.3571645

## 1 INTRODUCTION

Annotation of handwritten documents covering a variety of writing styles is essential for developing and evaluating the recognition systems, which are mainly data-driven approaches [12, 34, 42, 47]. The lack of linguistic resources for many scripts, resulting in limited annotated handwritten datasets, is one of the significant challenges in building a recognizer for them. Annotation of such a large corpus is a time-consuming and error-prone process. In contrast, text transcriptions of handwritten documents may sometime be available in many cases, while handwriting is based on existing text. The handwritten data collection and annotation process can be done under various settings. Each of the settings has its properties.

Unrestricted handwritten data is collected under natural settings (e.g., classroom lecture notes) and is ideally suitable for building a real-world handwritten text recognizer. Sometimes, there is no restriction on what type of data can be collected. Annotation of the collected data can be done either manually [10] or semi-automatically [31] or automatically [14, 19]. Designed text is a pre-defined text to be written by the writers, which makes annotation simple and comprehensive. Usually, the exact text may be written by multiple writers. Dictation is a variation of designed text data collection. In this setting, the text is dictated to a group of writers who will all write the same text. Since the number of writers has increased, creating manual ground truths has become very difficult in time and cost. IAM handwritten text recognition dataset [31] is a good example in this direction. Data generation through various models is an alternative way of getting handwritten data and corresponding ground truth transcription. In this direction, several methods [2, 11, 16, 18, 24, 27] exist to create a large amount of handwritten data synthetically. However, the quality of synthetic data is limited to the performance of the recognition model on real handwritten data. Each of these three (unrestricted text, designed text, and dictated text) data collection strategies, there can have a parallel corpus of text due to two factors: (i) one can hire an experienced typist in a language to write transcriptions of the available handwritten documents, and (ii) many handwritten datasets are collected based on the text already available in the electronic form. However, the created text may not always align with the handwritten document because an error occurs either in handwriting or transcription.

In this work, we propose a model-based framework to automatically annotate handwritten document images at the word level. The proposed framework consists of five different modules — (i) pre-processing, (ii) word detection, (iii) word recognition, (iv) alignment of word image with text sequence (corpus), and (v) manual correction and verification. The pre-processing step mainly cleans the input document image and crops only the text region from the

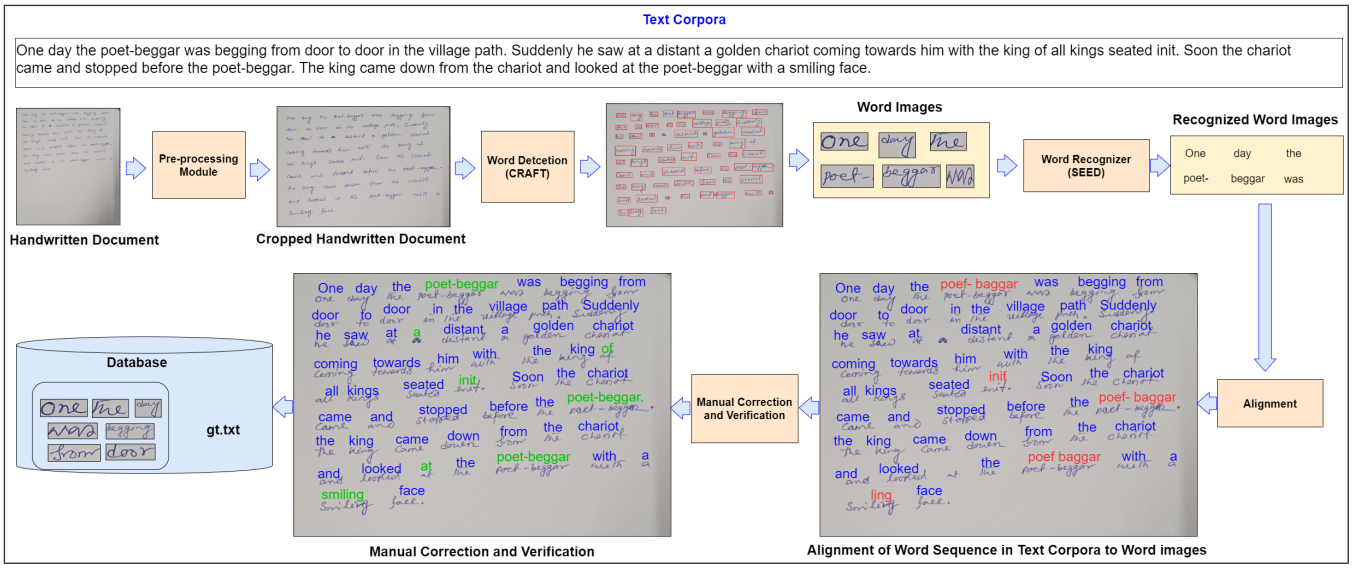


**Figure 1: Shows the proposed automatic annotation framework for handwritten documents at word level. The framework consists of five modules — (i) pre-processing, (ii) word detection, (iii) word recognition, (iv) alignment of word image with text sequence (corpus), and (v) manual correction and verification.**

document image. The word detection module localizes individual words present in the document image, and the recognition module recognizes the word images. The sequence of words in the text sequence (or text corpus) and the sequence of recognized words may differ. An error occurs during the writing, word localization, or recognition process. Therefore, it is necessary to align the word in the text sequence to the word image. The alignment module aligns the word in the text sequence to the word image. The automatic annotation tool may not always provide 100% accurate annotated results. The human annotator will correct the errors in the automated annotation process and then verify the annotation for the complete document image. Finally, we create a dataset containing word images and their corresponding ground truth text transcriptions. The annotation problem is only sometimes restricted to labeling the content (or text) information. Additional details like language/script, particular writer, writing condition, scanning process, etc., may have to be added for a specific application. We use our annotation tool to annotate 15000 English handwritten document images and 13 sets corresponding to 13 Indic languages, each set containing 15000 handwritten document pages. Experiments on an extensive collection of handwritten document images available in 14 languages show that 80% of the total words in documents are correctly annotated by the automatic annotation tool. In contrast, the remaining 20% of words are manually corrected.

The key contributions of this work are

- Develop a model-based framework consisting of five different modules — (i) pre-processing, (ii) word detection, (iii) word recognition, (iv) alignment of word image with text sequence (corpus), and (v) manual correction and verification, to automatically annotate handwritten document images at the word level.
- Experiments on a collection consisting of 21K of handwritten document images in 14 languages show that 80% of total words in documents are annotated correctly by the automatic annotation tool. The remaining 20% of words are manually corrected.

## 2 RELATED WORKS

In the space of English handwritten text recognition tasks, several high-performing models are available in the literature to recognize English handwritten text. Among them, few works [7, 39, 49] use convolutional neural networks entirely without using any recurrent architectures. While few recent works [9, 22] use a gating mechanism in CNNs to compensate for the dependency on Long Short-Term Memory (LSTM) cells, known as Gated Convolutional Neural Networks (GCN). These types of networks outperform fully convolutional networks, yet they lag behind transformer-based OCR models [13, 23, 30]. Recurrent Neural Networks (RNNs) are successfully applied to solve handwritten text recognition tasks. LSTM-based models can handle long-term context in sequences. The most common architectures [6, 46] are a combination of CNN and RNN, where CNN is used for feature extraction from word or line images and RNN is used for modeling sequential context. Several works [5, 25, 32, 44] use various attention mechanisms to improve the performance of CNN +RNN models. Recently, various deep encoder-decoder with attention frameworks [8, 26, 42, 48] have been developed to recognize complete handwritten pages. While transformer-based text recognizers [13, 23, 30] achieved state-of-the-art performance. Some of these works use a CNN-based backbone with self-attention as encoders to understand document images [30].

All these networks need a large number of annotated handwritten datasets. The large corpus can be annotated manually, semi-automatically, or automatically. The manual annotation of a

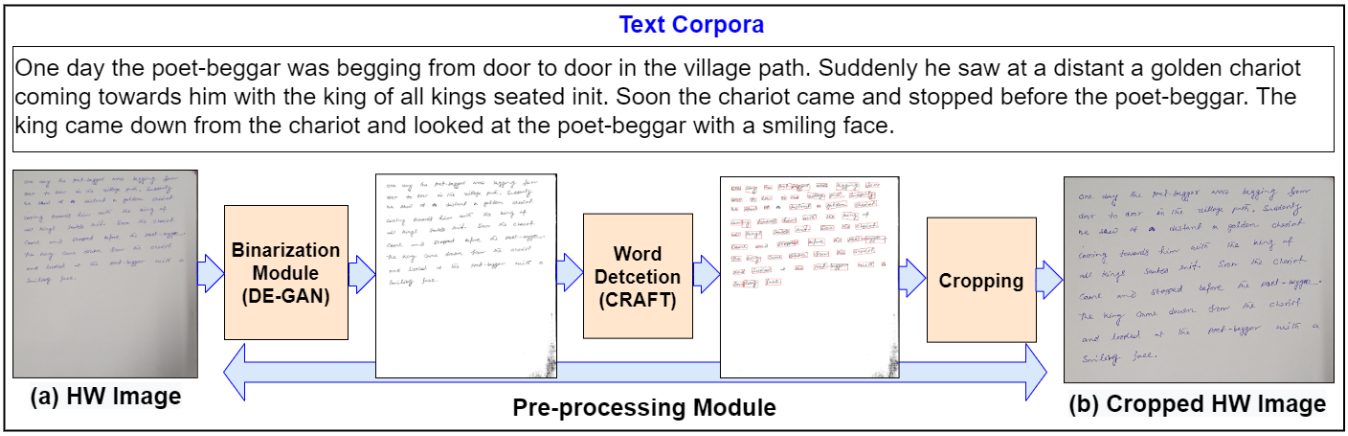


**Figure 2: Shows pre-processing operation on the input handwritten document image. (a) indicates input handwritten document image, and (b) indicates cropped image.**

large corpus is tedious, time-consuming, and cost-ineffective. Semi-automatic and automatic is the alternative way for a cost-effective and faster annotation process. Many handwritten datasets are collected based on the text already available in electronic form (e.g., text corpora). In this setting, one major challenge is the alignment of the handwritten document and text corpora.

Zimmerman and Bunke [50] addressed the problem of alignment of handwritten and text corpora in the context of segmenting English text lines into words. Tomai *et al.* [45] proposed a framework to annotate words in historical handwritten documents. It mapped each word in the transcript to the associated word image in the document. The word images are matched to text based on the global properties of the word extracted from both handwritten images and text words rendered using a specific font. Rothfeder *et al.* [41] proposed a Hidden Markov Model (HMM) based alignment technique to align handwritten data to transcripts, which handles segmentation and transcription errors. In the case of online handwritten data, Guyon *et al.* [21] introduced UNIPEN standard for representing annotation of online handwritten data. Bhaskarabhatla *et al.* [4] discussed XML-based presentation scheme for annotation of online handwritten data. The authors also proposed a tool based on this presentation to annotate digital ink. In the same direction, Agarawal *et al.* [1] introduced UPX, an XML-based successor to the venerable UNIPEN format for representing annotation of handwritten data.

## 3 PROPOSED METHOD

Fig. 1 illustrates the proposed automatic annotation approach for handwritten document images at the word level. The input to the algorithm is a handwritten document image (containing one paragraph), referred to as the input handwritten document image, and the corresponding transcription, referred to as the text sequence/text corpora. The proposed approach consists of five steps — (i) pre-processing, (ii) word detection, (iii) word recognition, (iv) alignment with text sequence, and (v) manual correction and verification. The pre-processing step mainly cleans the input handwritten document image for further processing. While in the second step, an existing word detection algorithm is used to detect individual words in the document image. The error in the word detection module may occur due to an automatic word detection algorithm. Currently, we are not correcting errors during word detection. After detecting words, all word images go through the word recognizer to recognize word images. The Edit distance between the text sequence and the predicted text sequence indicates correct and incorrect prediction. Alignment between word sequence in the handwritten document image and text sequence is established. The manual annotators correct the errors from the automatic annotation tool and verify the annotated full document image. Finally, we create a handwritten dataset with word images and their ground truth transcriptions. Each of these steps is discussed in detail in the following subsection.

### 3.1 Pre-processing

The handwritten document images are collected from a natural setting. No constraint is given to the writers while writing. Due to the unconstrained setting, it contains several challenging issues like (i) multi-colored text, (ii) unwanted background, (iii) multi-colored document, and (iv) wrongly written text as garbage. A pre-processing step involves binarizing the input handwritten document image to remove all the issues mentioned above. We use Document Enhancement Generative Adversarial Networks (DE-GAN) [43] to binarize the handwritten document image to remove all such issues. We use trained DE-GAN model [1] on different DIBCO datasets [17, 33, 35–38] for this purpose.

Sometimes, the written text may occur only in a small part of a document. In such cases, we crop only the text region from the document image and create a cropped document image. We use an existing text detector, CRAFT [3], to localize individual words in the document image. We use the pre-trained CRAFT [2] model for this purpose. We apply DBscan [15, 28] clustering algorithm on the coordinates of word bounding boxes predicted by the CRAFT. The largest cluster corresponds to the text region in the document image. We find the lowest x coordinate ($x_{min}$), lowest y coordinate ($y_{min}$), largest x coordinate ($x_{max}$), and largest y coordinate ($y_{max}$)

[1] https://github.com/dali92002/DE-GAN/issues
[2] https://github.com/clovaai/CRAFT-pytorch

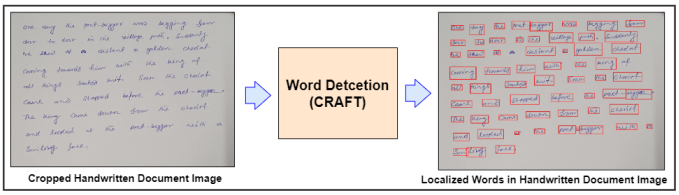


**Figure 3: Shows the localization of words in the document image using CRAFT.**

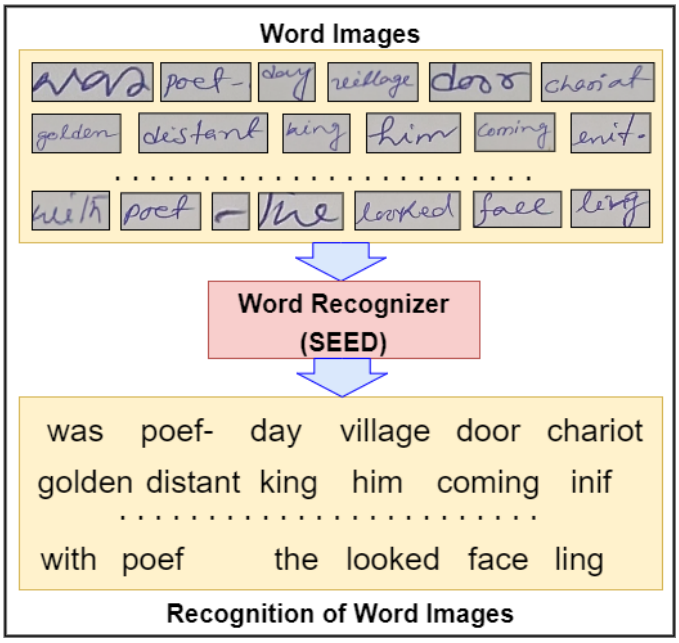


**Figure 4: Shows recognition output of word images.**

among all coordinates correspond to the bounding boxes within the most significant cluster. Finally, we crop the text region from the document image according to $x_{min}$, $y_{min}$, $x_{max}$, and $y_{max}$ and create a cropped document image. Fig. 2 shows cropped document image.

## 3.2 Word Detection

Since our goal is to generate annotation of the handwritten document at the word level, localizing individual words is an essential step in the proposed automatic annotation process. We use CRAFT [3] to localize individual words in the cropped document image. Fig. 3 shows the localized words in the cropped document image. We create a set of word images according to the word bounding boxes predicted by CRAFT. Since we use a pre-trained CRAFT model for word detection, detection/localization errors may occur in this step. The error may propagate to the next step.

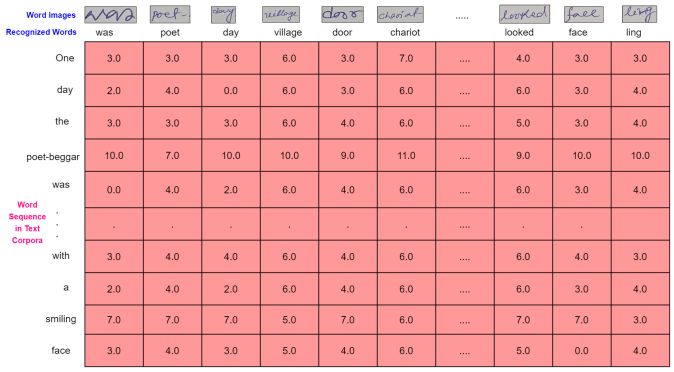


**Figure 5: Shows the alignment process of words in text corpora to word images based on Levenshtein distance between words in text corpora and recognized words.**

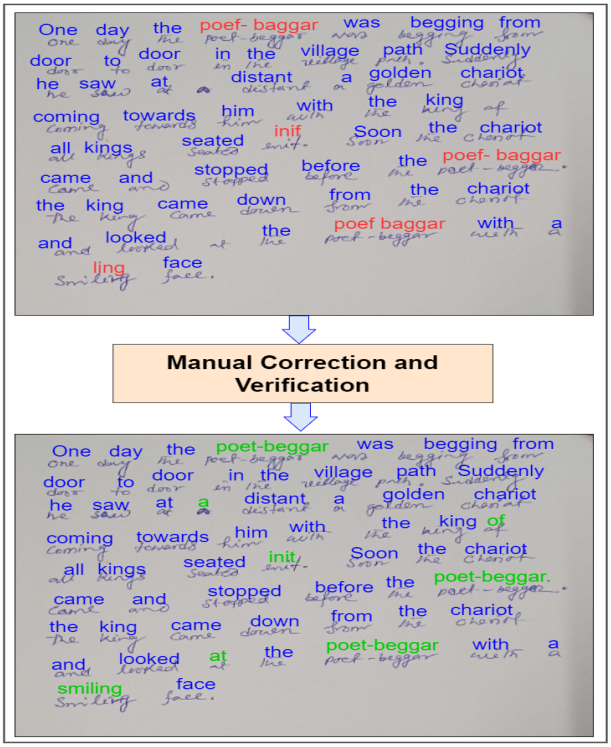


**Figure 6: Shows manual correction and verification on the automatic annotation output.**

## 3.3 Word Recognition

Recognition of word images is an essential step in our annotation process. To recognize word images, we use an existing text recognizer, SEED [40]. There are four modules in SEED — (i) the rectification module to rectify word images, (ii) the encoder to extract rich visual features, (iii) the semantic module to predict global semantic information from the visual features, and (iv) attention-based decoder to recognize the word image. Fig. 4 shows the recognized word images.

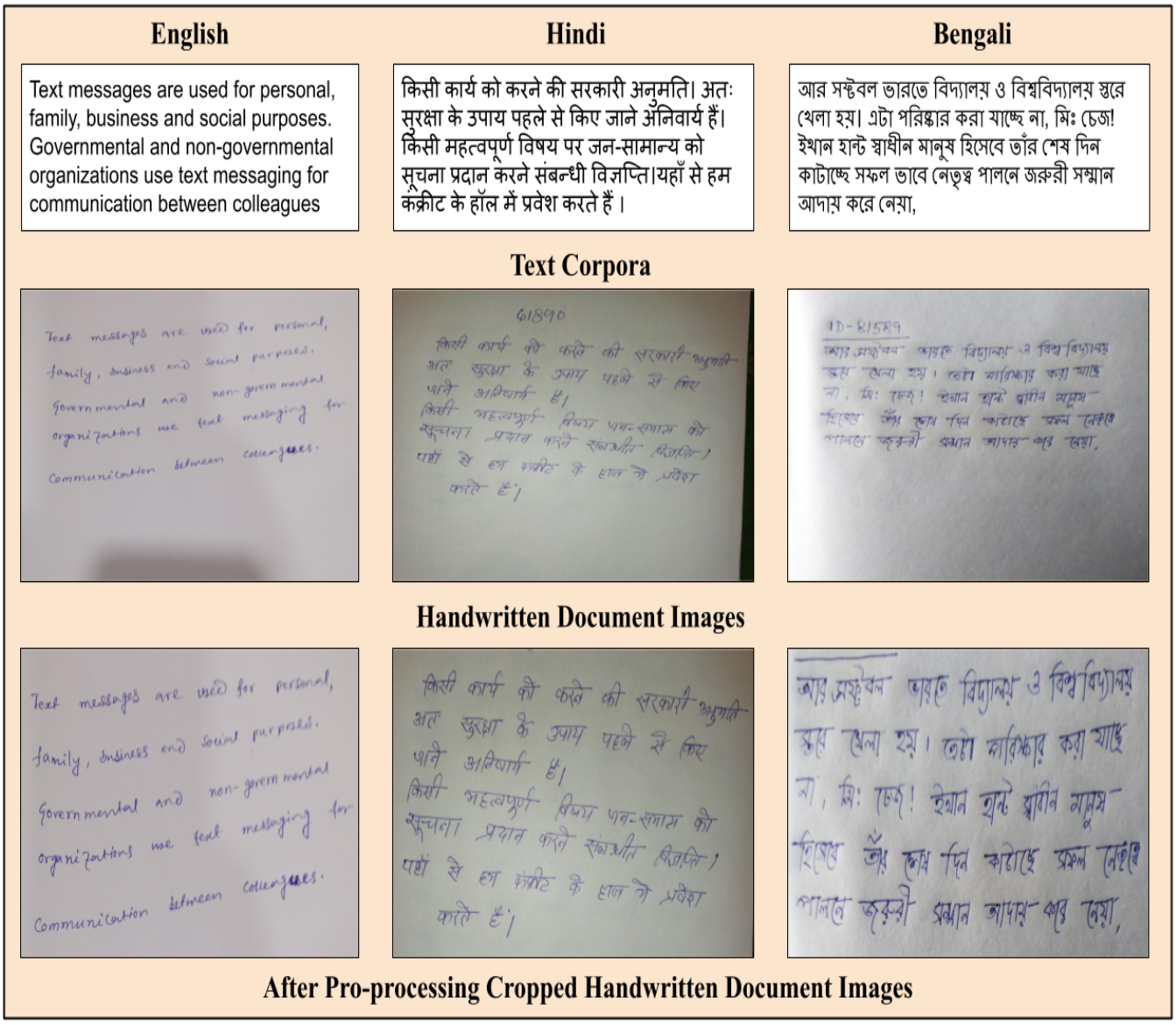


**Figure 7: Shows English, Hindi, and Bengali handwritten documents corresponding to text corpora and cropped handwritten document images after pre-processing.**

### 3.4 Align Recognized Text with Text Sequence

After recognizing word images of a handwritten document, it is necessary to align recognized text with text sequence to automatically create ground truth transcriptions corresponding to word images. Suppose text sequence (or text corpora), $T_1$ contains $n$ words, then $T_1 = \{w_1, w_2, ..., w_n\}$. Suppose $I$ denotes a set of word images of a handwritten document. Then $I = \{I_1, I_2, ..., I_m\}$, where $m$ is the number of word images, and $I_i$ denotes $i^{th}$ word image. Suppose, $w_i^r$ denotes the recognized text corresponding to $i^{th}$ word image $I_i$ by SEED. Suppose $T_2$ represents the recognized text sequence corresponding to the sequence of word images, $I$. Then $T_2 = \{w_1^r, w_2^2, ..., w_m^r\}$. Due to the error that occurs during either writing or word detection, the number of words in a text sequence and the number of recognized words may not be equal; and may not be a one-to-one correspondence. To align the text sequence to the word images, we calculate the Levenshtein distance [29] between every text pair in the sequences $T_1$ and $T_2$. We align text to the word image according to the: $w_i \rightarrow I_j$ if $lev(w_i, w_j^r) = 0$ and

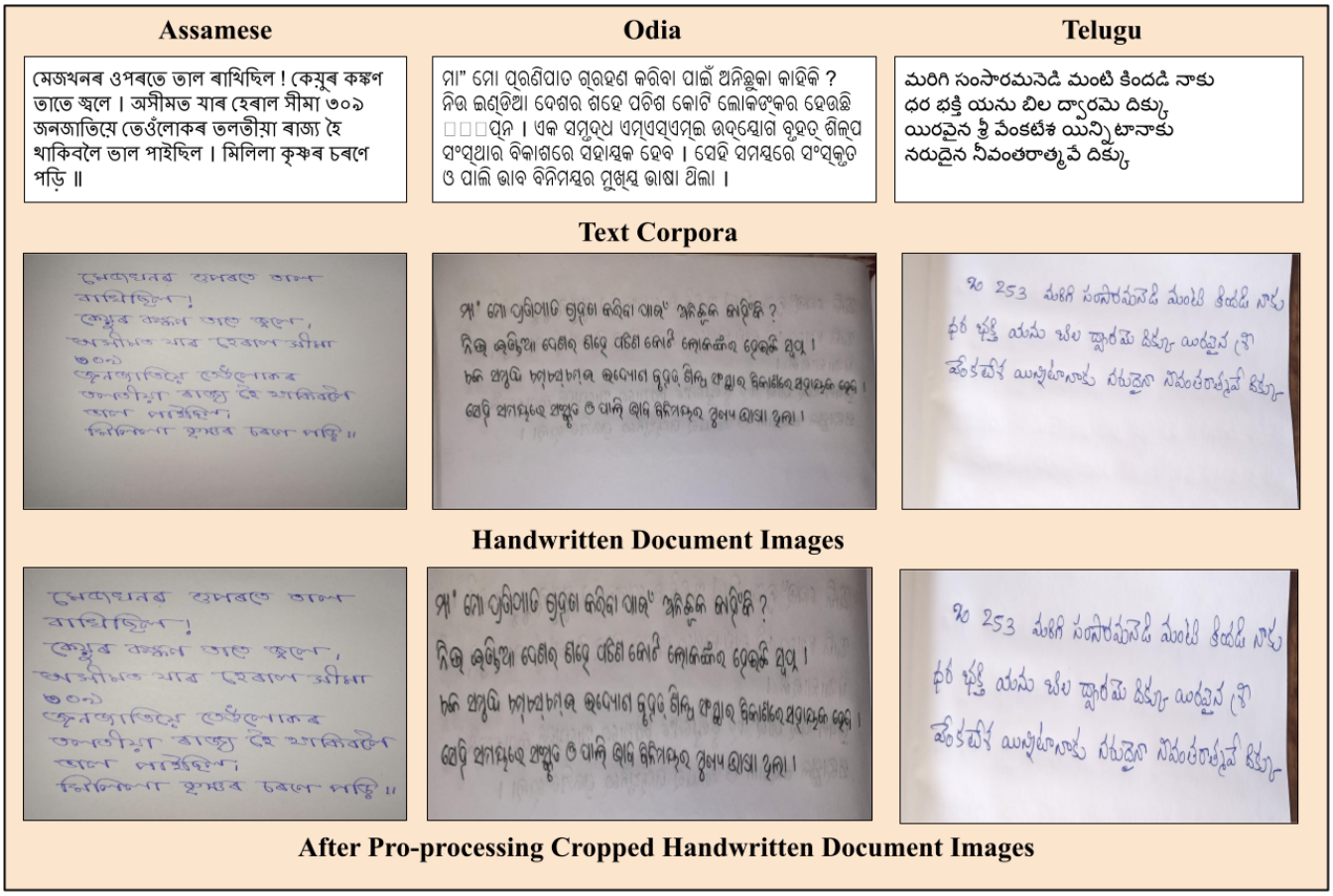


**Figure 8: Shows Assamese, Odia, and Telugu handwritten documents corresponding to text corpora and cropped handwritten document images after pre-processing.**

$flag(w_j^r) = 0$, $\forall w_i \in T_1$, and $\forall w_j^r \in T_2$, where $lev(a, b)$ is Levenshtein distance between $a$ and $b$, $flag(w_j^r) = 0$ indicates not match with other words. Fig. 5 visually illustrates the alignment of text words in text corpora to the recognized words.

### 3.5 Manual Correction and Verification

After getting automatic annotation outputs, the human annotators verify the ground truths; if there are errors, the annotators correct these errors. Fig. 6 shows the manual correction and verification.

### 3.6 Storage and Reuse of Annotation

Preservation of the word images and their corresponding ground truth transcriptions is essential for creating a newly trained model and updating the existing model. A folder consists of all word images in ".png" format; a text file such as "gt.txt" contains a word image path, and ground truth transcription corresponds to the word image separated by a particular delimiter (e.g., tab) in each line.

## 4 EXPERIMENTAL RESULTS AND DISCUSSION

We use our developed tool to annotate handwritten documents in 14 languages - English, Assamese, Bengali, Gujarati, Gurumukhi, Hindi, Kannada, Malayalam, Manipuri, Marathi, Odia, Tamil, Telugu, and Urdu. Text corpora corresponding to language are collected from different sources. 100-150 different writers write 15K pages for each language. Each page contains 30-40 words. Therefore for a language, we have a total of 500K word images to annotate using annotation tools. Fig. 7 and Fig. 8 show handwritten documents in English, Hindi, Bengali, Assamese, Odia, and Telugu languages based on corresponding text corpora and cropped images after applying to pre-process.

For experiments, we used trained DE-GAN model[3] to binarize the handwritten documents. We used trained CRAFT model[4] for word localization in the document image. We used the SEED model[5] on IAM [31] dataset for English word recognition. While we train SEED on Indic datasets [20] to recognize word images of Indic languages.

[3]https://github.com/dali92002/DE-GAN
[4]https://github.com/clovaai/CRAFT-pytorch
[5]https://github.com/Pay20Y/SEED

**Figure 9: Left Image: shows word detection, word recognition, and word alignment using the automatic annotation tool in English, Hindi, and Bengali handwritten document images, respectively. Right Image: shows manual correction in the output of the automatic annotation process. Blue Colored Text: correctly recognized by the recognizer. Red Colored Text: wrongly recognized by the recognizer. Green Colored Text: manually corrected text.**

Left images of Fig. 9 show the localized words, recognized words, and aligned word images to the sequence of the words in the text corpora. The red-colored rectangular boxes indicate the word bounding boxes predicted by CRAFT. The recognizer correctly recognizes texts in blue. In comparison, texts in red are wrongly recognized by the recognizer.

In the case of the English document image (left image of 1st row of Fig. 9), most of the words are correctly localized except 'messages', 'Governmental', 'non-governmental', 'organizations', and 'messaging'. For these words, instead of a single box, the word detector, CRAFT predicts more than one box. The recognizer, SEED recognizes most of the words correctly (e.g., the blue colored text) except 'messages', 'personal,', 'family,', 'purposes,' 'Governmental', 'non-governmental', 'organizations', 'messaging' and 'colleagues'. Among them, a few words (e.g., 'messages', 'Governmental', 'non-governmental', 'organizations', 'messaging') are wrongly recognized by the recognizer due to an error in the word localization process. Human annotators verify and correct the errors (both in word localization and word recognition processes) that occur in the automatic annotation process. In the case of the English document image, the right image of the 1st row of Fig. 9 shows the manually corrected and verified image. The green colored bounding boxes and texts are corrected manually. The 2nd and 3rd rows of Fig. 9 and Fig. 10 show the automatic annotation and manual correction

**Figure 10: Left Image: shows word detection, word recognition, and word alignment using the automatic annotation tool in Assamese, Odia, and Telugu handwritten document images, respectively. Right Image: shows manual correction in the output of the automatic annotation process. Blue Colored Text: correctly recognized by the recognizer. Red Colored Text: wrongly recognized by the recognizer. Green Colored Text: manually corrected text.**

of Hindi, Bengali, Assamese, Odia, and Telugu handwritten documents, respectively. Among 15K document images containing 500K word images for each of 14 languages, on average (language), 80% words are correctly annotated by automatic annotation, and the remaining 20% wrongly annotated words are corrected manually. We experimentally validate that the proposed automated annotation tool is cost-effective for annotating handwritten documents.

## 5 CONCLUSION AND FUTURE WORKS

In this paper, we propose a model-based framework for automatically annotating handwritten data (available in 14 languages) at a word level when text corpora are available for the data. The proposed framework consists of five modules (i) pre-processing to clean and crop the text region of the image, (ii) word detection, (iii) word recognition, (iv) aligning words in text corpora to the word images, and (v) correction and verification of automatic annotation. Experiments on a large set of handwritten document images available in 14 languages show that 80% of the total words in documents are correctly annotated by the automatic annotation tool. In contrast, the remaining 20% of words are manually corrected. The experiment illustrates that we can use the proposed framework to annotate extensive handwritten data with the availability of corresponding text corpora. We will also extend this framework for annotating printed and scene text document images.

# REFERENCES


[1] Mudit Agrawal, Kalika Bali, Sriganesh Madhvanath, and Louis Vuurpijl. 2005. UPX: A new XML representation for annotated datasets of online handwriting data. In *ICDAR*. 1161–1165.

[2] Eloi Alonso, Bastien Moysset, and Ronaldo Messina. 2019. Adversarial Generation of Handwritten Text Images Conditioned on Sequences. In *ICDAR*. 481–486.

[3] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. 2019. Character region awareness for text detection. In *CVPR*. 9365–9374.

[4] Ajay S Bhaskarabhatla, Sriganesh Madhvanath, MNSSKP Kumar, A Balasubramanian, and C V Jawahar. 2004. Representation and annotation of online handwritten data. In *IWFHR*. 136–141.

[5] Théodore Bluche. 2016. Joint line segmentation and transcription for end-to-end handwritten paragraph recognition. *Advances in neural information processing systems* 29 (2016), 838–846.

[6] Théodore Bluche and Ronaldo Messina. 2017. Gated convolutional recurrent neural networks for multilingual handwriting recognition. In *ICDAR*, Vol. 1. 646–651.

[7] Kartik Chaudhary and Raghav Bali. 2022. Easter2.0: Improving convolutional models for handwritten text recognition. *ArXiv* (2022).

[8] Denis Coquenet, Clément Chatelain, and Thierry Paquet. 2020. End-to-end Handwritten Paragraph Text Recognition Using a Vertical Attention Network. *CoRR* (2020).

[9] Denis Coquenet, Clément Chatelain, and Thierry Paquet. 2020. Recurrence-free unconstrained handwritten text recognition using gated fully convolutional network. In *ICFHR*. 19–24.

[10] Christian M Dahl, Torben Johansen, Emil N Sørensen, and Simon Wittrock. 2021. HANA: A HAndwritten NAme database for offline handwritten text recognition. *arXiv* (2021).

[11] Brian L. Davis, Chris Tensmeyer, Brian L. Price, Curtis Wigington, Bryan S. Morse, and Rajiv Jain. 2020. Text and Style Conditioned GAN for Generation of Offline Handwriting Lines. In *BMVC*. 1–13.

[12] Gideon Maillette De Buy Wenniger, Lambert Schomaker, and Andy Way. 2019. No padding please: Efficient neural handwriting recognition. In *ICDAR*. 355–362.

[13] Daniel Hernandez Diaz, Siyang Qin, Reeve Ingle, Yasuhisa Fujii, and Alessandro Bissacco. 2021. Rethinking text line recognition models. *arXiv* (2021).

[14] Kartik Dutta, Praveen Krishnan, Minesh Mathew, and CV Jawahar. 2018. Offline handwriting recognition on devanagari using a new benchmark dataset. In *DAS*. 25–30.

[15] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *KDD*. 226–231.

[16] Sharon Fogel, Hadar Averbuch-Elor, Sarel Cohen, Shai Mazor, and Roee Litman. 2020. ScrabbleGAN: Semi-supervised varying length handwritten text generation. In *CVPR*. 4324–4333.

[17] Basilis Gatos, Konstantinos Ntirogiannis, and Ioannis Pratikakis. 2009. ICDAR 2009 document image binarization contest (DIBCO 2009). In *ICDAR*. 1375–1382.

[18] Shivansh Gaur, Siddhant Sonkar, and Partha Pratim Roy. 2015. Generation of synthetic training data for handwritten Indic script recognition. In *ICDAR*. 491–495.

[19] Santhoshini Gongidi and CV Jawahar. 2021. IIIT-INDIC-HW-WORDS: A Dataset for Indic Handwritten Text Recognition. In *ICDAR*. 444–459.

[20] Santhoshini Gongidi and C. V. Jawahar. 2021. IIIT-INDIC-HW-Words: A Dataset for Indic Handwritten Text Recognition. In *ICDAR*. 444–459.

[21] Isabelle Guyon, Lambert Schomaker, Réjean Plamondon, Mark Liberman, and Stan Janet. 1994. UNIPEN project of on-line data exchange and recognizer benchmarks. In *ICPR*. 29–33.

[22] R Reeve Ingle, Yasuhisa Fujii, Thomas Deselaers, Jonathan Baccash, and Ashok C Popat. 2019. A scalable handwritten text recognition system. In *ICDAR*. 17–24.

[23] Lei Kang, Pau Riba, Marçal Rusiñol, Alicia Fornés, and Mauricio Villegas. 2022. Pay attention to what you read: non-recurrent handwritten text-line recognition. *Pattern Recognition* 129 (2022), 108766–108778.

[24] Lei Kang, Pau Riba, Yaxing Wang, Marçal Rusinol, Alicia Fornés, and Mauricio Villegas. 2020. GANwriting: content-conditioned generation of styled handwritten word images. In *ECCV*. 273–289.

[25] Lei Kang, J Ignacio Toledo, Pau Riba, Mauricio Villegas, Alicia Fornés, and Marçal Rusinol. 2018. Convolve, attend and spell: An attention-based sequence-to-sequence model for handwritten word recognition. In *German Conference on Pattern Recognition*. 459–472.

[26] Dmitrijs Kass and Ekta Vats. 2022. AttentionHTR: Handwritten Text Recognition Based on Attention Encoder-Decoder Networks. In *DAS*. 507–522.

[27] Praveen Krishnan and CV Jawahar. 2016. Generating synthetic data for text recognition. *arXiv* (2016).

[28] K Mahesh Kumar and A Rama Mohan Reddy. 2016. A fast DBSCAN clustering algorithm by accelerating neighbor searching using Groups method. *Pattern Recognition* 58 (2016), 39–48.

[29] Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*. 707–710.

[30] Minghao Li, Tengchao Lv, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. 2021. Trocr: Transformer-based optical character recognition with pre-trained models. *arXiv* (2021).

[31] U-V Marti and Horst Bunke. 2002. The IAM-database: an English sentence database for offline handwriting recognition. *IJDAR* 5, 1 (2002), 39–46.

[32] Johannes Michael, Roger Labahn, Tobias Grüning, and Jochen Zöllner. 2019. Evaluating sequence-to-sequence models for handwritten text recognition. In *ICDAR*. 1286–1293.

[33] Konstantinos Ntirogiannis, Basilis Gatos, and Ioannis Pratikakis. 2014. ICFHR 2014 competition on handwritten document image binarization (H-DIBCO 2014). In *ICFHR*. 809–813.

[34] Jason Poulos and Rafael Valle. 2021. Character-based handwritten text transcription with attention networks. *Neural Computing and Applications* 33, 16 (2021), 10563–10573.

[35] Ioannis Pratikakis, Basilis Gatos, and Konstantinos Ntirogiannis. 2012. ICFHR 2012 competition on handwritten document image binarization (H-DIBCO 2012). In *ICFHR*. 817–822.

[36] Ioannis Pratikakis, Basilis Gatos, and Konstantinos Ntirogiannis. 2013. ICDAR 2013 document image binarization contest (DIBCO 2013). In *ICDAR*. 1471–1476.

[37] Ioannis Pratikakis, Konstantinos Zagoris, George Barlas, and Basilis Gatos. 2016. ICFHR 2016 Handwritten Document Image Binarization Contest (H-DIBCO 2016). In *ICFHR*. 619–623.

[38] Ioannis Pratikakis, Konstantinos Zagoris, George Barlas, and Basilis Gatos. 2017. ICDAR 2017 competition on document image binarization (DIBCO 2017). In *ICDAR*. 1395–1403.

[39] Raymond Ptucha, Felipe Petroski Such, Suhas Pillai, Frank Brockler, Vatsala Singh, and Paul Hutkowski. 2019. Intelligent character recognition using fully convolutional neural networks. *Pattern recognition* 88 (2019), 604–613.

[40] Zhi Qiao, Yu Zhou, Dongbao Yang, Yucan Zhou, and Weiping Wang. 2020. Seed: Semantics enhanced encoder-decoder framework for scene text recognition. In *CVPR*. 13528–13537.

[41] Jamie Rothfeder, R Manmatha, and Toni M Rath. 2006. Aligning transcripts to automatically segmented handwritten manuscripts. In *DAS*. 84–95.

[42] Sumeet S Singh and Sergey Karayev. 2021. Full page handwriting recognition via image to sequence extraction. In *ICDAR*. 55–69.

[43] Mohamed Ali Souibgui and Yousri Kessentini. 2020. DE-GAN: a conditional generative adversarial network for document enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), 1180–1191.

[44] Jorge Sueiras, Victoria Ruiz, Angel Sanchez, and Jose F Velez. 2018. Offline continuous handwriting recognition using sequence to sequence neural networks. *Neurocomputing* 289 (2018), 119–128.

[45] Catalin I Tomai, Bin Zhang, and Venu Govindaraju. 2002. Transcript mapping for historic handwritten document images. In *ICFHRW*. 413–418.

[46] Curtis Wigington, Seth Stewart, Brian Davis, Bill Barrett, Brian Price, and Scott Cohen. 2017. Data augmentation for recognition of handwritten words and lines using a CNN-LSTM network. In *ICDAR*. 639–645.

[47] Curtis Wigington, Chris Tensmeyer, Brian Davis, William Barrett, Brian Price, and Scott Cohen. 2018. Start, follow, read: End-to-end full-page handwriting recognition. In *ECCV*. 367–383.

[48] Curtis Wigington, Chris Tensmeyer, Brian Davis, William Barrett, Brian Price, and Scott Cohen. 2018. Start, follow, read: End-to-end full-page handwriting recognition. In *ECCV*. 367–383.

[49] Mohamed Yousef, Khaled F Hussain, and Usama S Mohammed. 2020. Accurate, data-efficient, unconstrained text recognition with convolutional neural networks. *Pattern Recognition* 108 (2020), 107482–107492.

[50] Matthias Zimmermann and Horst Bunke. 2002. Automatic segmentation of the IAM off-line database for handwritten English text. In *ICPR*. 35–39.