information

Article

Gated Convolution and Stacked Self-Attention Encoder–Decoder-Based Model for Offline Handwritten Ethiopic Text Recognition

Direselign Addis Tadesse [1], Chuan-Ming Liu [2,*] and Van-Dai Ta [3]

[1] Institute of Technology, Debre Markos University, Debre Markos P.O. Box 269, Ethiopia; direselign_addis@dmu.edu.et
[2] Department of Computer Science and Information Engineering, National Taipei University of Technology (Taipei Tech), Taipei City 106, Taiwan
[3] Samsung Display Vietnam (SDV), Yen Phong Industrial Park, Bac Ninh 16000, Vietnam; t104999002@ntut.edu.tw
* Correspondence: cmliu@ntut.edu.tw

Abstract: Offline handwritten text recognition (HTR) is a long-standing research project for a wide range of applications, including assisting visually impaired users, humans and robot interactions, and the automatic entry of business documents. However, due to variations in writing styles, visual similarities between different characters, overlap between characters, and source document noise, designing an accurate and flexible HTR system is challenging. The problem becomes serious when the algorithm has a low learning capacity and when the text used is complex and has a lot of characters in the writing system, such as Ethiopic script. In this paper, we propose a new model that recognizes offline handwritten Ethiopic text using a gated convolution and stacked self-attention encoder–decoder network. The proposed model has a feature extraction layer, an encoder layer, and a decoder layer. The feature extraction layer extracts high-dimensional invariant feature maps from the input handwritten image. Using the extracted feature maps, the encoder and decoder layers transcribe the corresponding text. For the training and testing of the proposed model, we prepare an offline handwritten Ethiopic text-line dataset (HETD) with 2800 samples and a handwritten Ethiopic word dataset (HEWD) with 10,540 samples obtained from 250 volunteers. The experiment results of the proposed model on HETD show a 9.17 and 13.11 Character Error Rate (CER) and Word Error Rate (WER), respectively. However, the model on HEWD shows an 8.22 and 9.17 CER and WER, respectively. These results and the prepared datasets will be used as a baseline for future research.

Keywords: Ethiopic script; self-attention encoder–decoder; gated convolution; offline handwritten text recognition

check for updates

Citation: Tadesse, D.A.; Liu, C.-M.; Ta, V.-D. Gated Convolution and Stacked Self-Attention Encoder–Decoder-Based Model for Offline Handwritten Ethiopic Text Recognition. *Information* **2023**, *14*, 654. https://doi.org/10.3390/info14120654

Academic Editor: Katsuhide Fujita

Received: 26 September 2023
Revised: 25 November 2023
Accepted: 2 December 2023
Published: 9 December 2023

1. Introduction

Handwritten text recognition (HTR) has been one of the most attractive and challenging areas in the image processing and pattern recognition discipline. It is used as a user interface in numerous applications to transcribe a handwritten image into editable text. Increasing the recognition performance of the HTR system can improve the automation process.

According to the discussion in [1], a handwriting recognition system is essentially categorized into offline and online types. In offline recognition, handwritten features are extracted from scanned images, whereas, in online recognition, the features are extracted from both the pen trajectory and resulting images. Due to the different styles of writing handwritten characters by different writers, the visual resemblance of different characters, the overlap between characters, and the complex features of handwritten characters, the extraction of features from handwritten documents is difficult. Additionally, source document degradation is another challenge for offline HTR. This indicates that extracting features from pen trajectory input for online handwriting recognition is much better than extracting features from scanned images for offline HTR. As a result, offline HTR requires a more sophisticated method to precisely extract features and improve recognition performance.

Over the past few decades, many scholars have proposed different HTR systems and have made remarkable improvements. For instance, the Hidden Markov Model (HMM) [2] and an HMM-neural network hybrid [3] were implemented to recognize handwritten documents. However, due to the independence assumption of HMM, matching extracted features with labels has limitations, and there is a long-range input problem, even if it is slightly relaxed in the case of HMM-NN hybrid systems. As a result, deep neural network (DNN) approaches have been proposed to improve segmentation, feature extraction, and classification/recognition problems in shallow machine learning techniques [4]. Using deep learning approaches, offline HTR has been studied and has shown remarkable results for Latin, Arabic, Devanagari, and Chinese scripts, even for multilingual recognition [5–9]. In [10], a gated convolution neural network (Gated CNN) with a bidirectional gated recurrent unit (GRU) was proposed for offline handwritten text recognition using a publicly available dataset called IAM [11].

Even if these methods show remarkable recognition performance on several scripts, they have not been tested for Ethiopic script, which has a variety of characters. Ethiopic script is used as a writing system for more than 43 languages, including Amharic, Ge'ez, and Tigrigna. Of these languages, the Amharic and Ge'ez languages are more dominant because Amharic is used as the working language in Ethiopia, and Ge'ez is used as a liturgical language of the Ethiopian and Eritrean Orthodox Tewahedo and Catholic Churches. Moreover, there are many old manuscripts written in the Ge'ez and Amharic languages. Although many languages benefit from Ethiopic script, only few studies have been conducted on offline handwritten documents written using Ethiopic script.

In this paper, we propose a newly gated convolution and stacked self-attention encoder–decoder network (GCSEN) to recognize offline handwritten Ethiopic text. In our model, we apply gated CNN to extract features from handwritten images and a stacked self-attention encoder–decoder called Transformer to transcribe text. We integrate these models because gated CNN has high performance in feature extraction from complex datasets, and the stacked self-attention encoder–decoder outperforms in language modeling [12] by avoiding recursion. Our proposed model has feature extraction, encoder, and decoder layers. Additionally, we evaluate the performance of three other recently proposed recurrent-based models. These are gated CNN with long short-term memory (LSTM) [9], a convolutional neural network (CNN) with LSTM [8], and gated CNN with a gated recurrent unit (GRU) [10]. To simulate the selected and proposed models, we collect a dataset of offline handwritten text from volunteer high school students, university students, and university staff members. By preprocessing the collected datasets, we prepare a handwritten Ethiopic text-line dataset (HETD) and a handwritten Ethiopic word dataset (HEWD). Our system is evaluated using the HETD and HEWD databases. The overall contributions of this paper are summarized as follows:

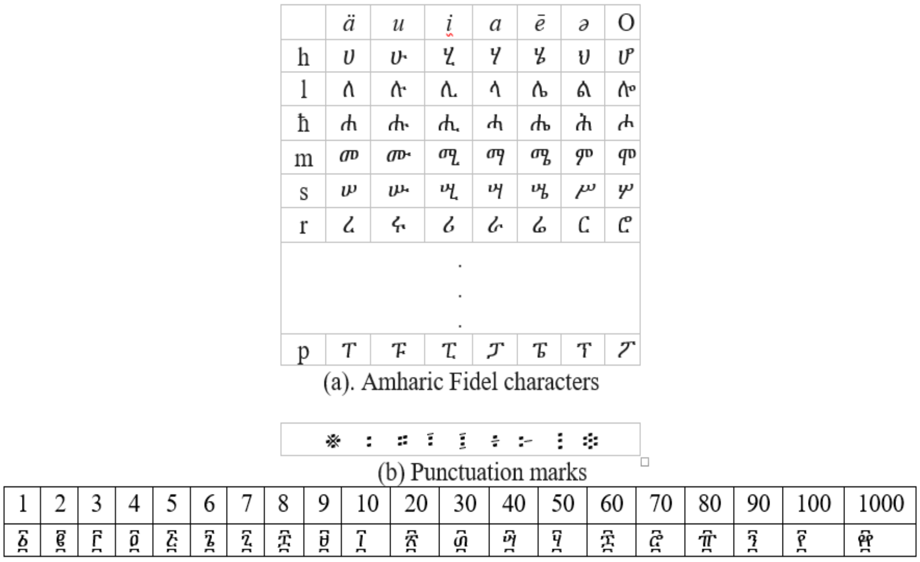
1. We prepare an offline handwritten database that is used for various applications, such as text recognition, writer identification, and signature verification. In addition, we prepare a synthetic dataset to pre-train the proposed model and adjust the weights and hyperparameters of the proposed model. The provided datasets are released and can be used by others in the future.
2. A comparative analysis between gated CNN-LSTM, CNN-LSTM, and HTR-flor++ (gated CNN-BGRU), which show state-of-the-art results on the IAM public dataset, and our proposed model, gated CNN-Transformer, is given.
3. The proposed model shows a promising recognition result on the collected HEWD and HETD and can be effectively used practically.

The rest of the paper is presented as follows: A short description of Ethiopic script and the Amharic language is given in Section 2. A brief discussion about dataset preparation, experiment setup and discussion of results are presented in Section 3. Finally, we conclude this paper in Section 4.

2. Materials and Methods

2.1. Ethiopic Script

The Ethiopic script originated from the Geez script, one of the world's ancient scripts [13]. It is used as a writing system for more than 43 languages, including Amharic, Geez, and Tigrigna. The script has been largely used by the Geez and Amharic languages, which are the liturgical and working languages of Ethiopia, respectively. Amharic is the second most spoken Semitic language in the world following Arabic. Fidel or Amharic script is a part of Ethiopic script, which is used to write the Amharic language. It is written from left to write. As shown in Figure 1, the script is written down in a tabular format, in which the first denotes the base character, and the other six columns are vowels derived from the base characters by slightly deforming or modifying the base character. The Amharic Fidel characters have 33 base characters and 6 derived vowels for each base character. Additionally, there are 20 labiovelar characters, with 5 orders, 20 numerals, and 9 punctuation marks.



	ä	u	i	a	ē	ə	O
h	ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
l	ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
ħ	ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ
m	መ	ሙ	ሚ	ማ	ሜ	ም	ሞ
s	ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ
r	ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ
			. . .				
p	ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ

(a). Amharic Fidel characters

፠ ፡ ። ፣ ፤ ፥ ፦ ፧ ፨

(b) Punctuation marks

1	2	3	4	5	6	7	8	9	10	20	30	40	50	60	70	80	90	100	1000
፩	፪	፫	፬	፭	፮	፯	፰	፱	፲	፳	፴	፵	፶	፷	፸	፹	፺	፻	፼

(c) Numerals

Figure 1. Ethiopic script alphabets.

Building an automatic Handwritten Text Recognition (HTR) system for the Ethiopic script, used in languages such as Amharic, Ge'ez, Tigrigna, and others, enables modern applications. These applications include the automatic identification of postal codes, digitization and recognition of text on paper documents, and automatic reading of bank checks, insurance forms, smart meters, and more. There are several research outputs for English, Chinese, and Arabic HCR and HTR. Additionally, there are also publicly available datasets for further research. However, languages that use Ethiopic script are not well studied, and there is no well-known public dataset. As a result, this paper prepares and presents a freely available public handwritten dataset with baseline results using selected state-of-the-art models and a newly introduced model.

2.2. Overview of Handwritten Text Recognition Techniques

The most popular sub-tasks performed to build an HTR system are segmentation (character, word, or text-line level), feature extraction, and classification tasks. The segmentation sub-task detects and segments the input image at the character, word, or text-line level. Character-level segmentation is the most challenging task for diacritic scripts. Ethiopic script has some ligatures, which are more difficult to segment at a character level. Due to this, we use word-level and text-line-level segmentation. Preprocessing techniques such as noise removal through median filtering [14] and Gaussian smoothing [15] play a crucial role in enhancing image quality and thereby improving the performance of the segmentation task.

Additionally, for feature extraction and classification sub-tasks, several machine learning techniques have been proposed, such as HMM, support vector machine (SVM), and neural networks [3,16–18]. Recently, DNN-based models have been introduced and have shown promising results for the high-dimensional automatic feature map extraction and recognition of handwritten texts [8,9,18–22]. The feature extraction and recognition tasks are performed in an end-to-end manner. Different from traditional machine learning methods, DNN-based approaches use minimal preprocessing [23]. For Ethiopic script offline HTR, some research works were studied. Most of them use conventional machine learning methods, and the recognition results need improving so for the methods to be used in practical applications. Some of the previous research outputs are as follows:

- Assabie and Bigun [24] presented a writer-independent offline HCR system using the characteristics and special relationships of primitive strokes. The accuracy of the proposed model was determined via the special relationships of primitives. If the relationships of primitives are poor, the recognition will fail; otherwise, they will be perfectly recognized. They used three different datasets that were collected from different sources, and the proposed approach achieved the recognition results of 87%, 76%, and 81% for each dataset.
- In [25], an HMM-based writer-independent offline handwritten Amharic word recognition system was designed using direction field tensor to detect text lines and extract features from the text lines. For each Ethiopic character, primitive structural features were stored as a feature list for the training and testing of the model. This work focused only on the Amharic language, which is one of the languages using half of Ethiopic script as the writing system.
- Assabie and Bigun [18] presented online handwritten Ethiopic script recognition by generating a unique set of primitive stroke sequences for each character using a special tree structure. For recognition, each stroke sequence was matched against a stored knowledge base. To improve the processing time and efficiency of recognition, structural similarity was used to classify a plausible set of unknown inputs. These approaches are limited to identifying a new entry with different characteristics from the stored ones.
- An unconstrained handwritten Amharic word recognizer was presented using the concatenated features of constituent characters and HMM [26]. Word features were formed by concatenating features of constituting characters from sample extracted features of characters. To build the HMM model, features were extracted from isolated handwritten characters. The models were trained and tested on good-quality and poor-quality data with 10, 100, and 10,932 training words. On both the good- and poor-quality training data, HMM recognition resulted in a better performance than a feature-level concatenation method. In the poor-quality dataset, HMM recognized 78%, 73%, and 41% of 10, 100, and 10,932 training words. Similarly, for good-quality data, HMM recognized 92%, 93%, and 66% of 10, 100, and 10,932 training words.
- In [27], a deep convolutional neural network was introduced to recognize Ethiopian ancient Ge’ez characters. This method considers only twenty-six characters.

In comparison with English, designing a robust offline HTR system for Ethiopic script has some challenges, such as the number of characters and the availability of visually similar characters. The Ethiopic script employs 466 characters in its writing system, which is over five times more than the English script. The number of characters affects the HTR system and demands memory and computation requirements. Additionally, there are visually similar characters that are very difficult for human beings to recognize.

In the previous research outputs of Ethiopic character/word recognition systems, the researchers use their private datasets, which are not publicly available. Moreover, the considered datasets contain only 265 characters, and the recognition performance needs to be improved. In this paper, we propose a gated CNN to extract features from handwritten text-line/word images and a Transformer network to transcribe the corresponding texts using the extracted feature maps as an input from the feature extraction layer. To train and test the proposed models, we prepare an isolated offline HEWD and HETD, which will be freely released to other researchers.

2.3. Proposed Methodology

This paper introduces a novel gated CNN architecture that incorporates a stacked self-attention encoder–decoder model for the recognition of offline handwritten Ethiopic text. Furthermore, we conduct an extensive exploration of the current state-of-the-art models for various scripts, including Bluche and Messina's gated CNN-1D-LSTM [9] for English, Puigcerver's CNN-1D-LSTM [8], and HTR Flor++ (gated CNN-1D-GRU) [10] for Arabic and other scripts.

The architecture of our proposed model is illustrated in Figure 2 and consists of three main components: a feature extraction layer, an encoder layer, and a decoder layer. The encoder–decoder layers are constructed using stacked multi-head self-attention, followed by position-wise fully connected networks known as Transformers. This architecture has gained prominence in recent years for natural language processing tasks [12] and has consistently achieved state-of-the-art results.

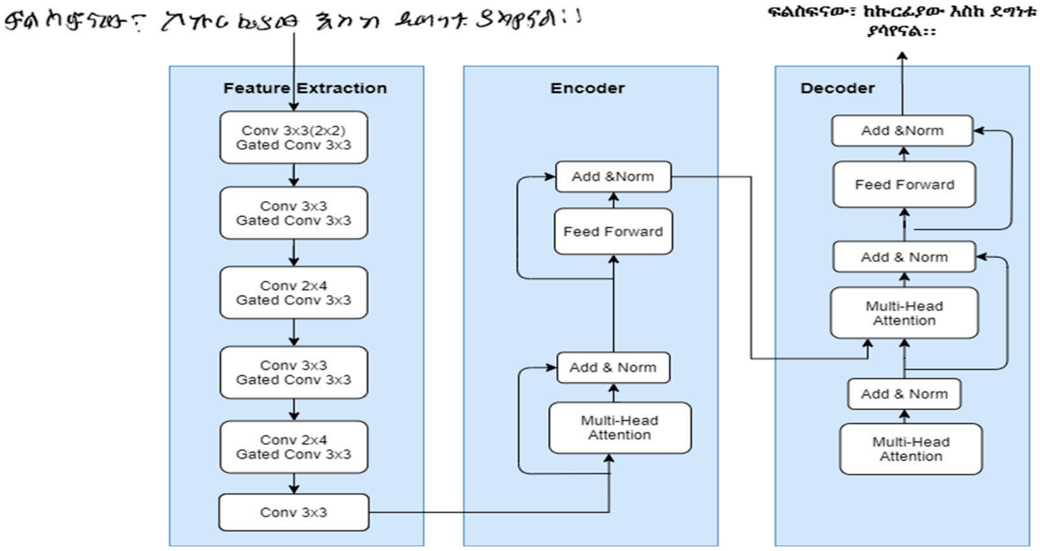


Figure 2. Proposed model.

One key distinction between the use of Transformers in natural language processing and their application in handwritten text recognition is the need to convert images into a sequential representation before feeding them into the Transformer network. To address this, we introduce a feature extraction unit that precedes the encoder layer, responsible for extracting sequences of feature maps from the input image. This preprocessing step ensures that the visual information from the image is appropriately structured for subsequent processing by the Transformer network.

While our proposed model consists of three fundamental layers, namely, feature extraction, encoder, and decoder layers, it is crucial to highlight that the model is trained in an end-to-end manner. Each of these layers plays a unique role in the recognition process, and we provide a concise overview of each layer in the following sections.

2.3.1. Feature Extraction Layer

As illustrated in Figure 2, the feature extraction layer serves as the initial component of our proposed model. As previously mentioned, the nature of input data for natural language processing and handwritten text recognition differs substantially. In the case of handwritten text recognition, the input, which is an image of handwritten text, must undergo a preprocessing step to convert it into spatial feature maps before it can be passed to the subsequent layers of the model.

To achieve this conversion from an input handwritten image to spatial feature maps, we employ a gated CNN architecture, as detailed in [10]. This architecture consists of a total of 11 convolutional neural networks, of which 5 are gated convolutional layers. In each of the convolutional layers, we apply the PReLU activation function and implement batch renormalization. Additionally, we incorporate a dropout technique with a rate of 0.2 in the last three gated convolutional layers to prevent overfitting and enhance the model's generalization.

Throughout the gated and convolutional layers, we extract feature maps with varying dimensions, including 16, 32, 40, 48, 56, and 64. The choice of kernel size also varies: 2×4 kernels are utilized on layers three and five, while 3×3 kernels are employed on the remaining convolutional layers. These choices of feature map dimensions and kernel sizes are strategically designed to capture and represent different spatial characteristics of the handwritten text, enabling the subsequent layers of the model to process this information effectively.

2.3.2. Encoder Layer

Following the feature extraction process, the extracted feature maps are seamlessly transitioned into the encoder layer, which plays a pivotal role in the proposed model. The encoder layer is composed of six stacks of similar layers, each designed to transform the input feature maps into higher-level representations that capture relevant information for subsequent processing.

Within each encoder layer, two major units are employed: a self-attention unit and a feed-forward neural network unit. These units work in tandem to progressively refine the feature representations.

The self-attention unit is a critical component of the encoder. It takes a collection of input encodings, which are derived from the previous encoder layer, and assesses the importance of each encoding in relation to all the others. This process of assigning weights to the input encodings based on their relevance to one another results in a set of output encodings that carry a refined understanding of the spatial relationships and contextual dependencies within the feature maps.

Subsequently, the output of each encoding is individually processed by the feed-forward neural network unit. This unit applies non-linear transformations to the feature encodings, enhancing their representational power and further refining the information contained within them.

The outputs of these units are then passed to the next encoder unit or, in the case of the final encoder unit, to the decoder layer. This flow of information from one unit to the next, layer after layer, allows the model to progressively build abstract and contextualized representations of the handwritten text, which is essential for accurate recognition and understanding. The encoder's capacity to analyze spatial and contextual information makes it a fundamental component in the model's ability to recognize handwritten text.

2.3.3. Decoder Layer

The decoder layer serves as the final piece of our proposed model, and it mirrors the structure of the encoder layer with six modules stacked on top of each other. Each of these modules shares common characteristics with the encoder layers, but with an additional sub-layer for conducting multi-head attention over the output of the encoder stack. This addition is crucial for the decoder's function, as it enables it to generate the final output sequence while considering the context and relationships encoded in the encoder's outputs.

In addition to our proposed model, we also explore the performance of three recently developed models for other languages. The first model, as presented in [8], leverages a convolutional neural network (CNN), bidirectional long short-term memory (BLSTM), and connectionist temporal classification (CTC). Referred to as CNN-1D-LSTM, this model is notable for its large parameter count, approximately 9.6 million. It features five convolution layers and five BLSTM layers, with each convolution layer employing a 3 × 3 kernel, batch normalization, leaky ReLU activation, and 2 × 2 max pooling. Additionally, a dropout with a probability of 0.5 is applied to mitigate overfitting. To update the model parameters, the RMSProp [28] algorithm is used, adjusting them incrementally based on the gradients of the CTC loss calculated for each batch of 16 images.

The second model adopts a different approach, utilizing a convolutional encoder for processing the input image and a bidirectional LSTM decoder for predicting character sequences. Known as gated convolutional recurrent neural networks (Gated CNN-BLSTM), this model was introduced by Bluche and Messina [9]. It distinguishes itself by having a more compact design with fewer parameters, approximately 730 thousand, compared to the previous model. Gated CNN-BLSTM incorporates eight convolution layers, including three gated convolution layers, and two BLSTM layers. This model is trained using a tanh activation function and an RMSProp optimizer with a standard learning rate, adjusting the parameters incrementally based on the gradients of the CTC loss.

The third model, named FLOR++ [10], features an architecture comprising 11 convolutional layers, including 6 gated convolutional layers and 2 bidirectional gated recurrent unit (BGRU) layers. Like the previously discussed models, FLOR++ extracts feature maps at different dimensions (16, 32, 40, 48, 56, and 64) in both gated and traditional convolution blocks. On the third and fifth convolution layers, a 2 × 4 kernel is applied, while a 3 × 3 kernel is used on the remaining convolution layers. This model incorporates the He uniform initializer, the PReLU [29] activation function, and batch renormalization [30] in all convolution layers. To address overfitting, a dropout rate of 0.2 is applied to the last three gated convolution layers. The optical model of FLOR++ consists of two bidirectional GRU layers with 0.5 dropouts, alternated by the dense layer, and it employs CTC to compute the loss and transcribe the expected characters. These diverse models contribute to our comprehensive exploration of approaches for handwritten text recognition in various languages.

3. Results

In this section, we delve into the comprehensive presentation and insightful analysis of the outcomes achieved by both the proposed model and the selected recent state-of-the-art models that were introduced in Section 2.3. This rigorous evaluation is essential to gauge the effectiveness of the models across the task of handwritten text recognition. The dataset used for experimentation, the specific experimental setups, and the findings obtained from each model are meticulously examined in the following sections. These results not only highlight the model's performance but also provide valuable insights into the strengths and potential areas for improvement, ultimately contributing to the advancement of handwritten text recognition technology.

3.1. Data Preparation

While publicly available datasets for languages like English [31], Arabic [32], and Swidish [33,34] have significantly aided research in their respective fields, the absence of such resources for Amharic handwritten text recognition posed a unique challenge for our research. In response, we meticulously undertook the task of dataset creation and preparation to facilitate our experiments. This section outlines the elaborate process involved in preparing the datasets that underpin our study.

To begin, we crafted forms with dedicated spaces for volunteers to contribute their handwritten Amharic text, thus generating much-needed ground truth data. This ground truth was sourced from various written materials, including blogs, books, and other relevant sources. An example of one such form is illustrated in Figure 3. Our data collection efforts spanned students from different educational levels, ranging from high school to university, as well as contributions from staff members. In total, we engaged with 250 users who collectively generated 300 pages of handwritten content.

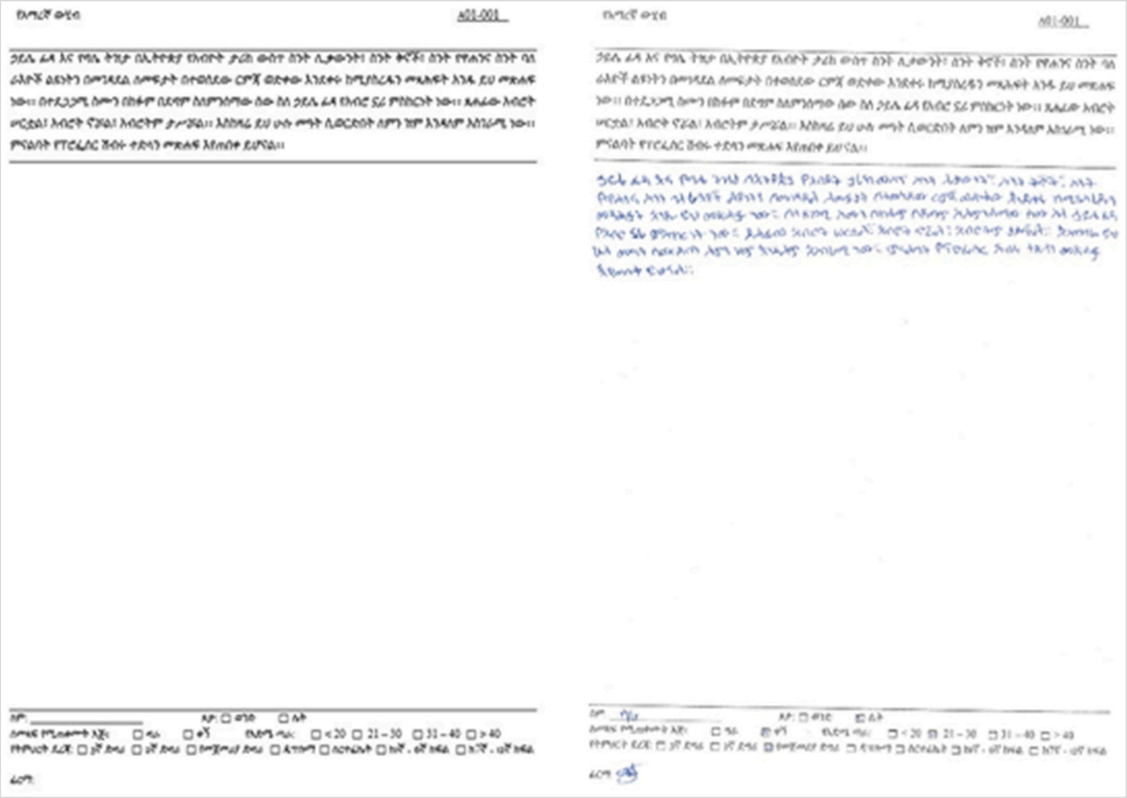


Figure 3. Handwriting collection form.

The collected handwritten forms were then subjected to digitization using an MX-M464N Sharp scanning and copying machine, ensuring the preservation of the original content. We leveraged the VIA annotation tool [35] to meticulously annotate the scanned images, labeling them at the paragraph level. Subsequently, the handwritten sections were cropped from the scanned images, isolating the core content for further analysis and processing.

For further refinement, the cropped images underwent a series of preprocessing steps, optimizing them for subsequent recognition tasks. This preprocessing involved converting the input cropped images into binary representations and optionally normalizing them into grayscale images. Additionally, deskewing operations were employed to correct any image skew. The OCRopus [36] toolbox was instrumental in this preprocessing phase, offering a comprehensive suite of tools for enhancing the input handwritten cropped images. An example of the cropped input and the corresponding binarized output image is presented in Figure 4.

Following the preprocessing phase, our attention turned to the vital tasks of word and text-line segmentation. These tasks aimed to extract meaningful text lines from the preprocessed images, and we employed a segmentation module from the OCRopus toolbox [36] to achieve this. This toolbox provided an array of Python-based tools for document analysis and recognition, effectively streamlining the segmentation process.

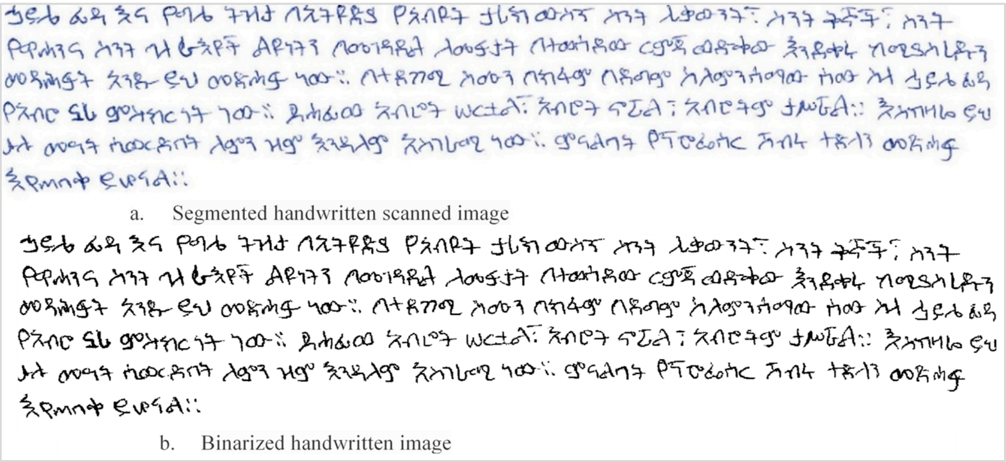

a. Segmented handwritten scanned image
b. Binarized handwritten image

Figure 4. Sample of a segmented and binarized handwritten image.

In the context of database preparation, the creation of ground truth data presented a significant challenge. To address this, we prepared ground truth text data for each segmented text-line image using the ocropus-gtedit HTML and extract commands, both of which are integral components of the OCRopus toolbox. A sample segmented text-line image alongside its corresponding ground truth data is exemplified in Figure 5.

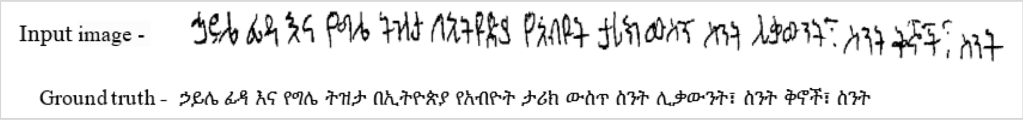

Input image -
Ground truth - ኃይሌ ፊዳ እና የግሌ ትዝታ በኢትዮጵያ የአብዮት ታሪክ ውስጥ ስንት ሊቃውንት፣ ስንት ቅኖች፣ ስንት

Figure 5. A sample of a segmented text-line image.

It is worth noting that, in cases where text lines were highly interconnected, leading to the segmentation of multiple lines as a single unit, such segmentation errors were discarded by leaving the ground truth empty. This rigorous quality control process ensured the integrity of the data. As a result, out of the 2910 segmented text lines, 2800 text-line images were included in the HETD database, while 110 text-line images were excluded due to the absence of ground truth.

To further enrich our HEWD dataset, we performed a secondary segmentation of the selected segmented text-line images using a contour-based algorithm. This algorithm performed the dual task of segmenting and labeling the text lines semi-automatically, utilizing the segmented text-line images and their ground truth data from the previous HETD as input. An exemplary detected word image from the segmented text line is displayed in Figure 6, with additional insights provided through the algorithm's pseudocode, which is presented in Algorithm 1. This meticulous data preparation process ensured the availability of high-quality datasets essential for our research on Amharic handwritten text recognition.

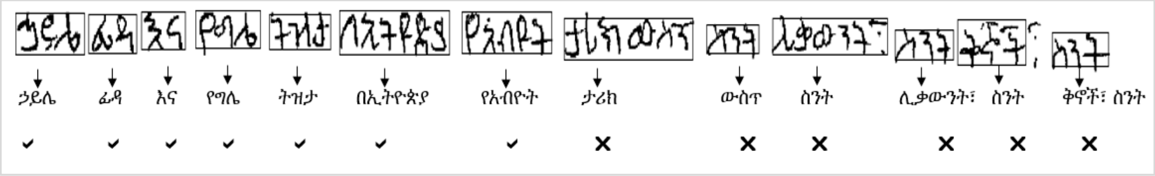


Figure 6. A sample of detected words from text-line input images.

Algorithm 1: Word Detection and Semi-Automatic Labeling.

```
1.  I ← Read text-line image
2.  gt ← Read ground truth text (GT)
3.  I ← Resize height I//in our case we set height = 50
4.  Kernel ← create anisotropic filter kernel//in our case we set kernels size, sigma and theta
    values 25, 11 and 7, respectively
5.  If ← 2D filtering on I//using the created anisotropic filter kernel and OpenCV filter2D
    library
6.  Ithr ← Apply threshold on If
7.  components ← detect connected components of Ithr//we use OpenCV findContours
8.  words ← gt.split(' ')//split ground truth text using space as a separator
9.  For (i, c) in enumerate (components):
        a.  If contour_area(c)> minArea//in our case min area = 100
                i.    box ← boundingRect(c)
                ii.   (x, y, w, h) ← box
                iii.  Iw ← I[y:y+h, x:x+w]
                iv.   Save image Iw with its cross ponding label words[i]
```

The effectiveness of the semi-automatic labeling method significantly hinges on the accuracy of the detection process. As depicted in Figure 6, the initial seven consecutive words are detected and labeled with precision, showcasing the method's potential to streamline the labeling process. However, challenges arise when two consecutive words are inaccurately detected as a single word, leading to subsequent labeling errors. To address these issues, a manual re-labeling process becomes necessary.

To ensure the dataset's overall quality and reliability, a meticulous manual re-labeling effort was undertaken. This involved correcting the labels for the words that were initially mislabeled due to the detection errors. As a result of this diligent curation process, a comprehensive dataset named HEWD was curated, comprising a total of 10,540 words.

Additionally, to enhance the robustness and versatility of our model, we conducted pre-training by synthetically generating text-line and word databases. These synthetic databases were instrumental in fine-tuning the model, adjusting its weights, and optimizing hyperparameters to enhance recognition performance.

For a concise overview of the prepared databases and their characteristics, please refer to Table 1. This table encapsulates the key statistics and features of the datasets, underscoring the importance of these meticulously curated resources in advancing our research in Amharic handwritten text recognition.

Table 1. Summary of prepared databases.

Database	Total	Training	Testing	Validation
HETD	2800	2240	560	-
HEWD	10,540	8432	2108	-
Synthetic text line	290,000	174,000	58,000	58,000
Synthetic word	500,000	300,000	100,000	100,000

3.2. Experiment Setup

The experiments were conducted on an Ubuntu machine equipped with an Intel Core i7-7700 (3.60 GHz) CPU, bolstered by 64 GB of RAM and a GeForce GTX 1080 Ti 11,176 MiB GPU. Our proposed system was implemented using Python 3.6 and the Keras library alongside TensorFlow. To ensure robustness, all the proposed networks underwent pre-training using synthetically generated text-line images inscribed in Ethiopic script.

For consistency and optimization, we employed the RmsProp optimizer [30] across all proposed networks, configuring a mini-batch size of 8. To prevent overfitting and enhance efficiency, an early stopping mechanism was set, triggering after 20 epochs without any discernible improvement in the validation loss value.

The evaluation of the proposed models' recognition performance was based on two key metrics: the Character Error Rate (CER) and the Word Error Rate (WER). These metrics serve as fundamental measures to assess the accuracy and fidelity of handwriting recognition systems.

CER quantifies the fidelity of character recognition by computing the Levenshtein distance, which measures the cumulative character-level operations—substitutions, insertions, and deletions—required to align the recognized text with the ground truth. Lower CER values indicate a higher accuracy in character-level recognition, signifying a closer match between the recognized and true characters.

Similarly, WER gauges the accuracy of word transcription in the recognized text by quantifying the disparity between recognized words and the actual content. It computes the collective word-level operations—substitutions, insertions, and deletions—necessary to align the recognized words with the ground truth. Lower WER values reflect a higher precision in transcribing the handwritten words, demonstrating a closer correspondence between the recognized words and the genuine content.

In the context of handwriting recognition, achieving lower CER and WER values indicates a more accurate and reliable system in deciphering handwritten text. These metrics are foundational in assessing the system's performance, providing insights into its ability to accurately transcribe handwritten characters and words, thus determining the overall effectiveness of the recognition model.

3.3. Experimental Results

The first experiment in our study aims to showcase the recognition performance of the proposed models, utilizing the text-line dataset. The results obtained from this experiment are presented in Table 2, offering valuable insights into the capabilities of each model in the context of handwritten text recognition.

Table 2. Experiment results of the proposed networks on HETD.

Network	WER	CER
Puigcerver	15.5	10.15
Bluche	14.8	9.12
Flor	14.51	8.92
Proposed	13.11	8.72

Upon reviewing the results in Table 2, it is evident that the proposed model stands out with a Word Error Rate (WER) of 13.11% and a Character Error Rate (CER) of 8.72%. This signifies the model's proficiency in recognizing handwritten Amharic text. In comparison, the GNN-GRU-CTC model, while still demonstrating respectable performance, yields a WER of 14.80% and a CER of 9.12%. Notably, the GNN-GRU-CTC model outperforms the CNN-LSTM-CTC model, showcasing the advantages of the former in the context of Amharic text recognition.

To further investigate the impact of using synthetically generated datasets on the performance of our recognition models, we conduct a training experiment from scratch using the handwritten datasets. The results from this experiment reveal a noteworthy reduction in performance for all models, as the WER and CER values increase by approximately 11% for the proposed model.

These results underscore the significant positive influence of employing synthetic datasets and pre-training the models on the overall recognition performance. The utilization of synthetic data not only enhances the models' adaptability but also contributes to a reduction in recognition errors, reinforcing the effectiveness of this approach in the realm of handwritten text recognition.

The second experiment in our research focuses on evaluating the recognition performance of the proposed models, this time using HEWD. The results of this experiment are detailed in Table 3, shedding light on the models' capabilities in the context of recognizing handwritten text from this specific dataset.

Table 3. Experiment results of proposed networks on HEWD.

Network	WER	CER
Puigcerver	13.75	9.55
Bluche	11.24	8.46
Flor	11.08	8.41
Proposed	9.17	8.22

Upon reviewing the results presented in Table 3, it becomes evident that the proposed model continues to demonstrate its superiority in the realm of Amharic handwritten text recognition. It outperforms the other previously proposed models, delivering superior recognition accuracy and efficiency.

This outcome underscores the robustness and versatility of the proposed model when applied to HEWD. It reaffirms the model's ability to handle diverse styles and forms of handwritten text, further solidifying its position as a leading solution for Amharic handwritten text recognition.

The positive results from this second experiment reinforce the promise of the proposed model in enhancing the field of handwritten text recognition, especially in the context of Amharic script, and they highlight its potential for various applications that rely on accurate text recognition from handwritten documents.

The results obtained from our experiments highlight the notable advantages of the Transformer network over the previously proposed state-of-the-art models. Beyond its improved recognition performance, the Transformer-based model also offers the compelling advantage of being more parameter-efficient than its counterparts. This means that it achieves remarkable results while requiring fewer computational resources, making it a more efficient and scalable solution.

Nonetheless, it is essential to acknowledge that Amharic script, like many other scripts, presents its own unique challenges. The script contains characters that share similar phonetic sounds but have distinct visual shapes. This discrepancy between the ground truth and the input image presented a challenge for the recognition performance. The model's ability to differentiate between visually similar characters is an ongoing area of improvement.

In both the text-line and word-based experiments, we observed that the proposed network faced limitations when dealing with characters that share similar shapes or when dealing with characters with a limited number of training samples. Furthermore, the presence of ligatures, which are combinations of characters with specific shapes and meanings, also had a notable impact on the recognition performance of the model.

To address these limitations and further enhance the model's capabilities, it is evident that an expanded and more diverse training dataset is required. This dataset should encompass a broader range of samples for each character, particularly focusing on characters that present challenges due to their visual similarities or the scarcity of training data. The continuous refinement and expansion of the dataset will contribute to improved character discrimination and recognition accuracy, ultimately advancing the performance of the model in recognizing handwritten Amharic text.

4. Conclusions and Discussion

Offline handwritten text recognition (HTR) has achieved remarkable results in recent years, owing to the advancements in machine learning techniques and the availability of large-scale datasets that facilitate the design of highly effective recognition models. However, when it comes to languages based on Ethiopic script, such as Amharic, the field of HTR remains relatively underexplored and in need of significant improvement. In this paper, we take a significant step towards addressing this gap by meticulously preparing two crucial datasets: the 10,540-word HEWD and the 2800-text-line HETD, thoughtfully collected from volunteers. Utilizing these datasets, we conducted a series of experiments to recognize both individual words and complete text lines. Our approach leverages gated convolution as the feature extraction layer, followed by the powerful Transformer network to transcribe the extracted features into text. Furthermore, we conducted a thorough analysis of recently proposed models, including CNN-LSTM, GNN-LSTM, and GNN-GRU networks. The results from our experiments are promising, demonstrating the viability of our model for handwritten Ethiopic words and text-line recognition. With the prepared handwritten testing dataset, the proposed GCSEN model achieved impressive results, exhibiting a Character Error Rate (CER) of 8.22 and a Word Error Rate (WER) of 9.17 on HEWD, and a CER of 8.72 and a WER of 13.11 on HETD. Looking forward, we plan to further enhance the recognition of handwritten words and text lines by integrating language modeling techniques and expanding our dataset. We are committed to sharing our implementation code and the prepared datasets to facilitate further research and advancements in this field, and they are accessible at https://github.com/direselign/HETR-HEWR (accessed on 6 December 2023).

Author Contributions: Conceptualization, D.A.T. and C.-M.L.; methodology, D.A.T., C.-M.L. and V.-D.T.; software, D.A.T.; validation, D.A.T., C.-M.L. and V.-D.T.; formal analysis, C.-M.L.; investigation, D.A.T. and V.-D.T.; resources, D.A.T.; data curation, D.A.T.; writing—original draft preparation, D.A.T.; writing—review and editing, C.-M.L. and V.-D.T.; visualization, D.A.T.; supervision, C.-M.L.; project administration, C.-M.L.; funding acquisition, C.-M.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Science and Technology Council of Taiwan grant number 112-2221-E-027-106.

Data Availability Statement: Words and text lines by integrating language modeling techniques and expanding our dataset. We are committed to sharing our implementation code and the prepared da-tasets to facilitate further research and advancements in this field, and they are acces-sible at https://github.com/direselign/HETR-HEWR (accessed on 6 December 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, C.-L.; Yin, F.; Wang, D.-H.; Wang, Q.-F. Online and Offline Handwritten Chinese Character Recognition: Benchmarking on New Databases. *Pattern Recognit.* **2013**, *46*, 155–162. [CrossRef]
2. Natarajan, P.; Saleem, S.; Prasad, R.; MacRostie, E.; Subramanian, K. Multi-Lingual Offline Handwriting Recognition Using Hidden Markov Models: A Script-Independent Approach. In *Arabic and Chinese Handwriting Recognition*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 231–250. [CrossRef]
3. España-Boquera, S.; Castro-Bleda, M.J.; Gorbe-Moya, J.; Zamora-Martinez, F. Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 767–779. [CrossRef] [PubMed]
4. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process Syst.* **2012**, *25*, 1–9. [CrossRef]
5. Zhao, Y.; Zhang, X.; Fu, B.; Zhan, Z.; Sun, H.; Li, L.; Zhang, G. Evaluation and Recognition of Handwritten Chinese Characters Based on Similarities. *Appl. Sci.* **2022**, *12*, 8521. [CrossRef]
6. Hu, M.; Qu, X.; Huang, J.; Wu, X. An End-to-End Classifier Based on CNN for In-Air Handwritten-Chinese-Character Recognition. *Appl. Sci.* **2022**, *12*, 6862. [CrossRef]
7. Graves, A.; Schmidhuber, J.J. Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks. Available online: https://proceedings.neurips.cc/paper_files/paper/2008/hash/66368270ffd51418ec58bd793f2d9b1b-Abstract.html (accessed on 1 December 2023).
8. Puigcerver, J. Are Multidimensional Recurrent Layers Really Necessary for Handwritten Text Recognition? In Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, Kyoto, Japan, 2 July 2017; IEEE Computer Society: Washington, DC, USA; Volume 1, pp. 67–72.

9. Bluche, T.; Messina, R. Gated Convolutional Recurrent Neural Networks for Multilingual Handwriting Recognition. In Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, Kyoto, Japan, 2 July 2017; IEEE Computer Society: Washington, DC, USA; Volume 1, pp. 646–651.
10. Flor, A.; Neto, D.S.; Leite, B.; Bezerra, D.; Toselli, A.H. *HTR-Flor++: A Handwritten Text Recognition System Based on a Pipeline of Optical and Language Models*; Association for Computing Machinery: New York, NY, USA, 2020.
11. Marti, U.V.; Bunke, H. The IAM-Database: An English Sentence Database for Offline Handwriting Recognition. *Int. J. Doc. Anal. Recognit.* **2003**, *5*, 39–46. [CrossRef]
12. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Transformer: Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems 30, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
13. Mendisu, B.S.; Efforts, C.B. The Ethiopic Script: Linguistic Features and Socio-Cultural Connotations. *Oslo. Stud. Lang.* **2017**, *8*, 137–172.
14. Huang, T.S.; Yang, G.J.; Tang, G.Y. A Fast Two-Dimensional Median Filtering Algorithm. *IEEE Trans. Acoust.* **1979**, *27*, 13–18. [CrossRef]
15. Praveen, K.S.; Babu, K.P.; Sreenivasulu, M. Implementation of Image Sharpening and. *Int. Sci. Eng. Appl. Sci.* **2016**, *2*, 7–14.
16. Xu, S.; Wu, Q.; Zhang, S. *Application of Neural Network in Handwriting Recognition*; IEEE Transactions on International Conference of Stanford University: Stanford, CA, USA, 2020.
17. Sadri, J.; Suen, C.Y.; Bui, T.D. Application of Support Vector Machines for Recognition of Handwritten Arabic/Persian Digits. In Proceedings of the Second Conference on Machine Vision and Image Processing & Applications (MVIP 2003), Tehran, Iran, 13–14 February 2003; Volume 1, pp. 300–307.
18. Assabie, Y.; Bigun, J. Online Handwriting Recognition of Ethiopic Script. In Proceedings of the Eleventh International Conference on Frontiers in Handwriting Recognition (ICFHR2008), Montreal, QC, Canada, 19–21 August 2008; pp. 153–158.
19. Bluche, T.; Louradour, J.; Messina, R. Scan, Attend and Read: End-To-End Handwritten Paragraph Recognition with MDLSTM Attention. In Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, Kyoto, Japan, 9–15 November 2017; Volume 1, pp. 1050–1055. [CrossRef]
20. Graves, A. Offline Arabic Handwriting Recognition with Multidimensional Recurrent Neural Networks. In *Guide to OCR for Arabic Scripts*; Springer: London, UK, 2012; pp. 297–313. [CrossRef]
21. Moysset, B.; Messina, R. Are 2D-LSTM Really Dead for Offline Text Recognition? *Int. J. Doc. Anal. Recognit.* **2019**, *22*, 193–208. [CrossRef]
22. Stuner, B.; Chatelain, C.; Paquet, T. Handwriting Recognition Using Cohort of LSTM and Lexicon Verification with Extremely Large Lexicon. *Multimed. Tools Appl.* **2020**, *79*, 34407–34427. [CrossRef]
23. Soomro, M.; Farooq, M.A.; Raza, R.H. Performance Evaluation of Advanced Deep Learning Architectures for Offline Handwritten Character Recognition. In Proceedings of the 2017 International Conference on Frontiers of Information Technology, FIT, Islamabad, Pakistan, 18–20 December 2017; pp. 362–367. [CrossRef]
24. Assabie, Y.; Bigun, J. Writer-Independent Offline Recognition of Handwritten Ethiopic Characters. In Proceedings of the 11th International Conference on Frontiers in Handwriting Recognition (ICFHR), Montréal, QC, Canada, 19–21 August 2008; pp. 652–657.
25. Assabie, Y.; Bigun, J. HMM-Based Handwritten Amharic Word Recognition with Feature Concatenation. In Proceedings of the 2009 10th International Conference on Document Analysis and Recognition, Barcelona, Spain, 26–29 July 2009; IEEE: Toulouse, France; pp. 961–965.
26. Assabie, Y.; Bigun, J. Offline Handwritten Amharic Word Recognition. *Pattern Recognit. Lett.* **2011**, *32*, 1089–1099. [CrossRef]
27. Demilew, F.A.; Sekeroglu, B. Ancient Geez Script Recognition Using Deep Learning. *SN Appl. Sci.* **2019**, *1*, 1315. [CrossRef]
28. Tieleman, T.; Hinton, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA Neural Netw. Mach. Learn.* **2012**, *4*, 26–31.
29. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *arXiv* **2005**, arXiv:1502.01852.
30. Ioffe, S. Batch Renormalization: Towards Reducing Minibatch Dependence in Batch-Normalized Models. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 1946–1954.
31. Cohen, G.; Afshar, S.; Tapson, J.; van Schaik, A. EMNIST: An Extension of MNIST to Handwritten Letters. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017.
32. Kiessling, B.; Ezra, D.S.B.; Miller, M.T. BadAM: A Public Dataset for Baseline Detection in Arabic-Script Manuscripts. In *ACM International Conference Proceeding Series*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 13–18. [CrossRef]
33. Yavariabdi, A.; Kusetogullari, H.; Celik, T.; Thummanapally, S.; Rijwan, S.; Hall, J. CArDIS: A Swedish Historical Handwritten Character and Word Dataset. *IEEE Access* **2022**, *10*, 55338–55349. [CrossRef]
34. Cheddad, A.; Kusetogullari, H.; Hilmkil, A.; Sundin, L.; Yavariabdi, A.; Aouache, M.; Hall, J. SHIBR—The Swedish Historical Birth Records: A Semi-Annotated Dataset. *Neural Comput. Appl.* **2021**, *33*, 15863–15875. [CrossRef]
35. Dutta, A.; Zisserman, A. The {VIA} Annotation Software for Images, Audio and Video. In Proceedings of the 27th ACM International Conference on Multimedia, Nice, France, 1 January 2021; ACM: New York, NY, USA, 2019.

36. Breuel, T.M. The OCRopus Open Source OCR System. In Proceedings of the Document Recognition and Retrieval XV, SPIE, San Jose, CA, USA, 27 January 2008.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.