Learning by Injection: Attention Embedded Recurrent Neural Network for Amharic Text-image Recognition

Birhanu Belay*+, Tewodros Habtegebrial*, Gebeyehu Belay †, Million Meshesha‡, Marcus Liwicki§ and Didier Stricker ¶
*Dept. of Computer Science, University of Kaiserslautern, Kaiserslautern, Germany
Email:{ b_belay18@cs.uni-kl.de}
†+Faculty of Computing, Bahir Dar Institute of Technology, Bahir Dar, Ethiopia
‡Faculty of Informatics, Addis Ababa University, Addis Ababa, Ethiopia
§Department of Computer Science, Lulea University of Technology, Lulea, Sweden
¶DFKI-German Research Center for Artificial Intelligence, Kaiserslautern, Germany

***Abstract*—In the present, the growth of digitization and world-wide communications make OCR systems of exotic languages a very important task. In this paper, we attempt to develop an OCR system for one of these exotic languages with a unique script, Amharic. Motivated with the recent success of the Attention mechanism in Neural Machine Translation (NMT), we extend the attention mechanism for Amharic text-image recognition. The proposed model consists of CNNs and attention embedded recurrent encoder-decoder networks that are integrated following the configuration of the seq2seq framework. The attention network parameters are trained in an end-to-end fashion and the context vector is injected, with the previous predicted output, at each time steps of decoding. Unlike the existing OCR model that minimizes the CTC objective function, the new model minimizes the categorical cross-entropy loss. The performance of the proposed attention-based model is evaluated against the test dataset from the ADOCR database which consists of both printed and synthetically generated Amharic text-line images, and achieved a promising results with a CER of 1.54% and 1.17% respectively.**

***Index Terms*—Amharic script , Attention mechanism, OCR, Encoder-decoder, Text-image**

I. Introduction

Amharic is an official working language of the Federal Democratic Republic of Ethiopia. As many as 100 million people around the world speak Amharic, making it the second most spoken Semitic language next to Arabic and it has a rich collection of documents ranging from historical to modern, from vallum written to the paper printed and from simple to complex layouts. Amharic is widely spoken in different countries like Eritrea, USA, Israel, Somalia, and Djibouti [1], [2], [3], [4].

Amharic has been the working language of the courts, the language of trade and everyday communications, the military, dated back from the late 12^{th} century and remains the official language of Ethiopia today [5]. Since then there are multiple documents, containing religious and academic contents, written in Amharic script [6]. Since then, these documents are stored in different places such as Ethiopian Orthodox Tewahdo Churches, public and academic libraries in the form of hardcover books, and preserved in a manual catalog [7].

In the Amharic script, there are about 317 different alphabets, including 238 core characters, 50 labialaize characters, 9 punctuation marks, and 20 numerals which are written and read, like English, from left to right [1], [8], [9]. All vowels and labialized characters in Amharic script are derived, with a small change, from the 34 consonant characters.

As shown in Table I, the 1^{st} column is the base symbol with no explicit vowel indicator usually called consonants. In the 2^{nd} and 3^{rd} columns, the corresponding consonants are modified by a projection half-way down the right leg and the base of the right leg respectively. The 4^{th} column has a short left leg while the 5^{th} column has a loop on the right leg. The 6^{th} and 7^{th} columns are less systematic, but some regularity happened on the left and right leg of the base characters. Due to these small modifications on the consonants, Amharic characters have similar shapes which may make the task of recognition hard for machines as well as humans [3], [10] .

Even though there is considerable shape modification regularity of characters across some columns, there are also unpredictable modification patterns in some columns. Some of them, such as the 3^{rd} and 5^{th} column are more consistent than others, such as the 2^{nd} and 4^{th} columns, while others, such as 6^{th} and 7^{th} columns are completely inconsistent. These features are particularly interesting in research on character recognition because a small change in the basic physical features may affect the orthographic identities of letters.

Numerous works, in the area of Optical Character Recognition (OCR) and Document Image Analysis (DIA), have been done and widely used for decades to digitize various historical and modern documents [11], [12], [13]. Many of the well-known scripts have OCR systems with sufficiently high performance that enables OCR applications to be applied in industrial/commercial settings. However, OCR systems yield very-good results only on a narrow domain and very specific use cases. Thus, it is still considered as challenging task

Table I
SHAPE FORMATION OF SAMPLE BASIC AMHARIC CHARACTERS [3]. ORDERS OF CONSONANT-VOWEL VARIANTS (34 × 7). CHARACTERS IN THE FIRST COLUMN ARE CONSONANTS AND THE OTHERS ARE DERIVED VARIANTS. VOWELS ARE DERIVED BY ADDING DIACRITICS AND/OR REMOVE PART OF CONSONANTS AND THE ORTHOGRAPHIC IDENTITIES OF EACH CHARACTER VARY ACROSS ROWS AS MARKED WITH THE VIOLET COLOR.

	0	1	2	3	4	5	6
0	ሀ hâ	ሁ hu	ሂ hí	ሃ ha	ሄ hē	ህ hi	ሆ ho
1	ለ lâ	ሉ lu	ሊ lí	ላ la	ሌ lē	ል li	ሎ lo
3	ሐ hâ	ሑ hu	ሒ hí	ሓ ha	ሔ hē	ሕ hi	ሖ ho
4	መ mâ	ሙ mu	ሚ mi	ማ ma	ሜ mé	ም mi	ሞ mo
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
33	ቨ vâ	ቩ vu	ቪ ví	ቫ va	ቬ vē	ቭ vi	ቮ vo

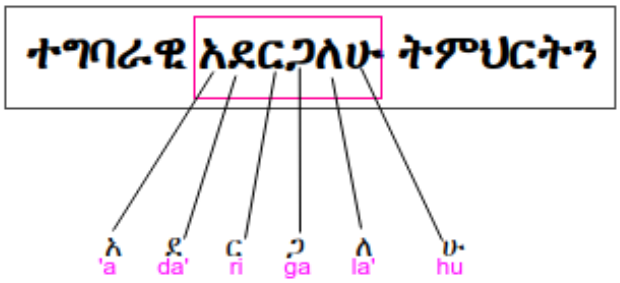

Figure 1. Sample Amharic text-line image. A word marked by violet color is composed of six individual Amharic characters and the corresponding sounds of each character is described with English letters.

and there are other indigenous scripts for which no well developed OCR systems exist [1]. In the present, the growth of digitization and world wide communications make OCR systems of exotic languages, like Amharic, a very important task.

In literature, attempts made for Amharic script recognition so far are based on the classical machine learning techniques and they are very limited in addressing the issues for Amharic OCR. Moreover, these attempts are neither shown results on large datasets nor considering all possible characters used in the Amharic writing system. Recently published work [1], introduced an Amharic OCR database called ADOCR. We took a sample text-line image from the ADOCR database whose word formation and character arrangements in a sample word are illustrated in Figure 1.

RNNs that are trained with CTC objective function has been employed for text-image recognition of multiple scripts. It is also a state-of-the-art approach and widely applied for sequence-to-sequence learning tasks to date. Recently, another sequence-to-sequence learning technique has been emerged, from the filed of NMT, and has often been shown to improve the performance over the existing approaches. Therefore, this paper presents an attention-based encoder-decoder model for Amharic OCR.

Text-line image recognition has recently been widely treated as a sequence-to-sequence learning task while a traditional segmentation based character recognition has applied for a longer time. Later an LSTM-CTC based techniques have been used for recognition of multiple scripts including Amharic script which impair the valuable spatial and structural information of text-line images. This work also aims to push the limits of such techniques using attention-based text-line image recognition which is based on neural networks. Unlike the CTC, the attention model explicitly uses the history of the target sequence without any conditional independence assumptions.

In addition, attention enables the networks to potentially model language structures, rather than simply mapping an input to an output [14]. Moreover, encouraged by recent attention given by researchers for Amharic document digitization and inspired by its success in employing attention for neural machine translation [15], [16] and speech recognition [17], [18], we investigate models that can attend to a salient part of a text-image, in the context of Amharic script, while generating a character.

As a continuation of our previous work [1], [3], which employed the CTC as a cost function to train and tune the parameters of an LSTM network this paper presents the first result using the concept of attention mechanism with the following additional contributions:

1) We propose an attention-based OCR framework for Amharic text-image recognition for the first time.
2) The proposed method is trained by injection learning strategy which allows the model to learn from its error and reduce exposure of bias, unlike the existing attention mechanisms that are usually trained by teacher forcing techniques.
3) Different from the existing attention-based encoder-decoder model that uses the last hidden state of the encoder as an initial hidden states of the decoder, the proposed model uses independent and randomly initialized hidden states for both encoder and decoder.
4) The proposed model is designed by leveraging the architecture of the Seq2Seq framework, used in NMT, and then stacking CNN layers before the encoder network as a feature extractor. During training, the overall model components are treated as a unified framework.
5) We validate the proposed attention mechanisms on Amharic text-image recognition task and show the advantage of attention mechanism against the methods employed for the recognition of Amharic text-images so far through empirical analysis.

The rest of the paper is organized as follows: Section II reviews the relevant methods and related works. The proposed method and training strategies are described in section III. Section IV presents all experiments, empirical analysis, and

results obtained from the experiment. Finally, conclusions and future works are described in the last section.

II. Related work

The existing OCR models can utilize either traditional or holistic techniques. Methods belongs to the first category were mainly applied before the introduction of deep learning and follows step-wise routines. By contrast, the holistic approach integrated the feature extraction and sequential translation steps in a unified framework that is trained from end-to-end. Therefore, in this section, we review the research trends of Amharic OCR development and the existing state-of-the-art techniques that are applied for sequence-to-sequence learning tasks.

Even though OCR research for the Amharic script started in 1997 [19], it is still in its infancy and it is still an open area of research. Since then, attempts have been made to develop Amharic OCR [8], [19], [20], [21] using different statistical machine learning techniques.

Recently, following the success of deep learning, other attempts are also made to develop a model for Amharic OCR and achieved relatively promising results; such as Belay et al [22] proposed CNN for Amharic character image recognition and a year later a factored convolutional neural network [9] model was also proposed for Amharic character image recognition. This new model was designed by leveraging the arrangement of Amharic characters in Fidel-Gebeta. It consists of two classifiers that have shared layers at the lower stage and task-specific layer at their last stage (one as a column detector while the other is the row component detector), where both classifiers are trained jointly. A parallel work by Gondere et al [23] was proposed for Handwritten Amharic character image recognition. The work of Gondere is based on the architecture proposed by Belay. Unlike the previous work, the work of Gondere is mainly focused on handwritten Amharic character recognition.

The standard OCR tasks have been investigated based on RNNs [1], [3], and the CTC [24] objective function. For example, an LSTM network together with CTC objective function has been proposed for Amharic text-line image recognition [1]. In this work, benchmark datasets called ADOCR were introduced. A model for Amharic text-line image recognition was also proposed by stacking CNNs before LSTM networks [3] as a feature extractor, where training is done in an end-to-end fashion. However, CTC-based architectures are subject to inherent limitations like strict monotonic input-output alignments and an output sequence length that is bound by the subsampled input length while the the attention-based sequence-to-sequence model is more flexible, suits the temporal nature of the text and can focus on the most relevant features of the input by incorporating attention mechanisms [15].

In literature, most CTC-based architectures for text-image recognition were LSTM networks, in many cases using the Bi-LSTM such as Bidirectional-LSTM architecture for online handwritten recognition [25], an LSTM-based online handwritten recognition for 102 languages in 26 scripts [26], combined Connectionist Temporal Classification (CTC) with Bidirectional LSTM (BLSTM) for unconstrained online handwriting recognition [24], and Multidimensional-LSTM for Chinese handwritten recognition [27].

Others, like [3], [28], [29], integrate CNNs for improved low-level feature extraction prior to the recurrent layers. This approach has been mainly applied for printed text-image recognition [30], handwritten recognition [31], and license plate recognition [32]. A convolutional recurrent neural network is also employed for Japanese handwritten recognition [29], and Chinese handwritten text recognition [33], fully convolutional neural networks for detection of watermarking region [34] and handwritten text-line segmentation [35], a scattering feature maps integrated with convolutional neural networks has been applied for Malayalam handwritten character recognition [36].

The main challenge in sequence-to-sequence learning tasks is to find an appropriate alignment between input and output sequences of variable length. For text-image recognition, one needs to identify the correct characters at each time step without any prior knowledge about the alignment between the image pixels and the target characters. The current two major methods that overcome this problem are attention-based and CTC-based sequence-to-sequence learning approaches.

The CTC-based models compute a probability distribution over all possible output sequences, given an input sequence. They do so by dividing the input sequence into frames and emitting, for each frame, the likelihood of each character of the target alphabet. The probability distribution can be used to infer the actual output greedily, either by taking the most likely character at each time step.

Attention-based models are an alternative and recent idea of sequence-to-sequence architectures that follow the encoder-decoder framework is to decouple the decoding from the feature extraction. The model consists of an encoder module, at the bottom layer, that reads and builds a feature representation of the input sequence, and a decoder module, at the top layer, which generates the output sequence one token at a time. The decoder uses an attention mechanism to gather context information and search for relevant parts of the encoded features.

Even though attention mechanism was introduced to address the problem of long sequences in Neural Machine Translation(NMT), nowadays attention becomes one of the most influential ideas in the deep learning community and it is widely applied in many sequence-to-sequence learning tasks. Bahdanua et al. [15] and Luong [16] proposed attention-based encoder-decoder model for machine language translation.

The most works so far with an attention mechanism has focused on neural machine translation. However, researchers have recently applied attention to different research areas. Therefore, it becomes popular and a choice of many researchers in the area of OCR. For example, Doetsch et al [37] proposed an attention neural network to the output of

a Bi-LSTM network that is operating on frames extracted from a text-line image with a sliding window and Bluche [38] proposed a similar technique for an end-to-end handwritten paragraph recognition, recognizing handwritten texts [39], [40], characters in the wild [41], handwritten mathematical expression [42], Chen et al [43] proposed an adaptive embedding gate controlled attention module for scene text recognition.

Finally, our system is quite similar to Bahdanua et al. [15] attention network architecture the one proposed for neural machine translation. The main difference is that we apply the attention for text-line image recognition where the decoder network outputs character by character given the decoder history and the expected input from the attention mechanism. Further, CNN layers are integrated with the encoder network as a feature descriptor and the decoder LSTM initializes with a Keras default weight initializer, $Xavier\ uniform\ initializer$ [44], instead of the final state of the encoder LSTM network. The following section gives a detailed overview of the proposed attention-based encoder-decoder Amharic OCR model.

III. The Proposed Approach

In this section, we elaborate on the proposed attention-based network for Amharic OCR. In the text-image recognition task, the raw data is 32 by 128 text-line images which should be processed and encoded to a sequence of high-level features. To do so, as illustrated in Figure 2, our Amharic OCR model follows the standard encoder-decoder framework with attention mechanism [15]. The model consists of three basic modules: an encoder module that combines a CNN as a generic feature extractor with recurrent neural network layers to introduce temporal contexts in the feature representation, a decoder module that utilizes a recurrent network layer to interpret those features and the third module is an attention mechanism that enables the decoder to focus on the most relevant encoded features at each decoding time step.

A. Encoder

Our encoder module is equipped with CNNs; thus the segmented text-line images are converted into a sequence of visual feature vectors. CNN layers integrated into encoder network are pre-trained model weights adopted from an Amharic text-image recognizer proposed by Belay [3]. Then two Bidirectional-LSTM layers, which reads the sequence of convolutional features to encode temporal context between them, are employed.

The Bidirectional-LSTM processes the sequence in opposite directions to encode both forward and backward dependencies and capture the natural relationship of texts. All the configurations and corresponding network parameters of the proposed encoder module are presented in Table II and III, where the input text-line image size is $32 \times 128 \times 1$.

As shown in Figure 2, the encoder network takes an Amharic text-line image as an input and encapsulates the information as the internal state vectors which are later used by the decoder. The hidden state and cell state (h_0 and c_0) of the encoder are initialized randomly. The dimensions of both states should be the same as the number of units in the LSTM cell which is 128 in our case. The final states of the encoder (i.e in this case h_{Tx} and c_{Tx}) are the relevant information of the whole input Amharic text-line image.

Table II
Convolutional layers of the proposed model adopted from the our previous paper [3] and their corresponding parameter values. K-size and F-map represent kernel size and the number of feature maps respectively.

Network Layers	K-Size	Stride	F-Maps
Convolution	3×3	1×1	64
Max-Pooling	2×2	1×1	-
Convolution	3×3	1×1	128
Max-Pooling	2×1	1×1	-
Convolution	3×3	1×1	256
Convolution	3×3	1×1	256
Max-Pooling	2×1	1×1	-
Convolution	3×3	1×1	256
BatchNormalization	-	-	-
Convolution	3×3	1×1	256
BatchNormalization	-	-	-
Max-Pooling	2×1	1×1	-
Convolution	3×3	1×1	512

Suppose that the input text-line image I consists of a sequence length Tx, then the encoder-CNN processes an input image I and transfers it into an intermediate-level feature map X, which can be thought of as a sequence of column vectors $X = (x_1, x_2, ..., x_{Tx})$. This sequence is then processed by two Bidirectional-LSTM layers and we get the combined state sequence of the final encoded feature map $H = (h_1, h_2, ..., h_{Tx})$, of the forward and backward hidden states.

Table III
The recurrent network layers an of the proposed encoder module with their corresponding parameter values.

Network Layers (Type)	Hidden Layer Size
BLSTM	128
BLSTM	128

B. Decoder

The decoder module generates the target character sequence present in the image given the current image summary and state vector. The proposed model uses a unidirectional LSTM network whose initial hidden state is randomly initialized as done for our encoder. The network parameters and their corresponding value are shown in Table IV. At each time-step t, the decoder computes a probability distribution over the possible characters and predicts the most probable character y_t, conditioned on its own previous predictions $(y_0, y_1, ..., y_{t-l})$ and a time-dependent context vector c_t, which contains information from the encoded features. Formally, for an output sequence length Ty, it defines a probability over the output

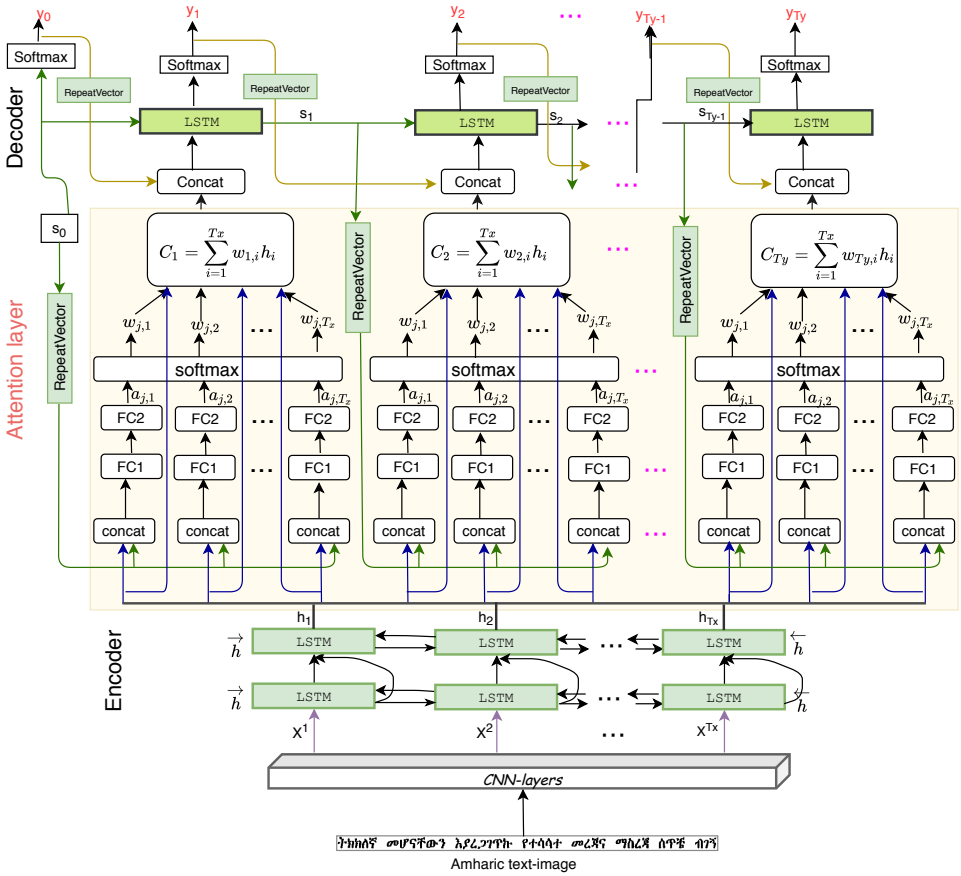

Figure 2. Attention-based encoder-decoder model for Amharic OCR. The encoder converts an input Amharic text-line image I into a sequence of constant feature vectors h. The decoder generates the output sequence $y = (y_0, y_1, ..., y_{Ty})$ one character at a time, where y_0 is the dummy sequence that is generated before the actual target character sequence started to generate. At each time step t, the decoder uses an attention mechanism to produce a context vector c_j based on the encoded feature vectors and a time-dependent decoder hidden state s_{j-1}. A concatenation of the context vector at t time step and the output y at t-1 time step serves as the decoder's next input. During concatenation, to have the same dimension, feature vectors are regenerated using, $RepeatVector$, a function in Keras API.

Table IV
THE DECODER RECURRENT NETWORK LAYERS AND THEIR PARAMETER VALUES. THE INPUT SIZE OF THE DECODER LSTM, AT EACH TIME-STEP t, IS A CONCATENATION OF THE CONTEXT VECTOR AT TIME t AND PREVIOUS PREDICTED OUTPUT AT TIME-STEP t-1.

Network Layers (Type)	Hidden Layer Size
LSTM	128
FC+Soft-Max	No. class = 281

sequence $Y = (y_1, y_2, ..., y_{Ty})$ by modeling each condition as $p(y_t|(y_0, y_1, ..., y_{T_{y-1}}, c_t)) = softmax(f(y_{t-1}, s_{t-1}, c_t))$ where f and s_{t-1} represents the current and previous LSTM hidden states respectively.

C. Attention Mechanism

At each step of decoding, the attention layer is introduced to focus on the most relevant part of the encoded feature representation. A typical attention model architecture is shown in Figure 3. For each particular input sequence, attention mechanism has the power to modify the context vector at

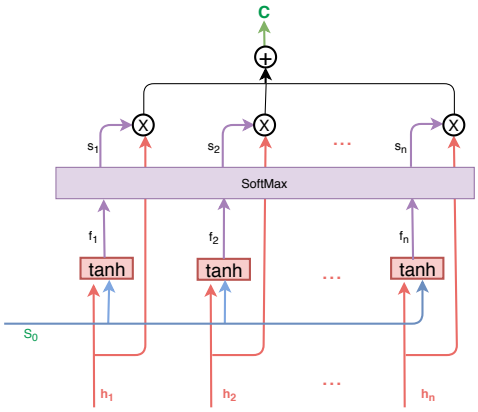

Figure 3. A typical attention model architecture. This network module computes the output c from the initial state s_0 and the part of the given sequence h=$h_1, h_1, ..., h_n$. The $tanh$ layer can be replaced by any other network that is capable to produce an output form the given s_0 and h_i, where the input length is equal to the output length which is n in this case.

each time step based on some similarity of the decoder hidden state s_{j-1} with the encoded features h of the context vector c_j where c_j is computed for each target character y_j as given in Equation (1),

$$c_j = \sum_{i=1}^{Tx} w_{j,i} h_i \tag{1}$$

where the $w_{j,i}$ is an attention weight which is computed by soft-maxing its corresponding alignment score $a_{j,i}$ using Equation (2),

$$w_{j,i} = \frac{exp(a_{j,i}))}{\sum_{k=1}^{Tx} exp(a_{jk})} \tag{2}$$

where $a_{j,i}$ is the alignment score of concatenated annotation between s_{j-1} and h_i at each time step t and it can be computed using Equation (3).

$$a_{j,i} = f(g(h_i, s_{j-1})), \text{for } i = 1, ..., T_x \tag{3}$$

The function g and f in Equation (3) are feed-forward neural networks, with tanh activation function, that are stacked consecutively. The intuition of the feed-forward networks is to let the model to learn the alignment weights together with the translation while training the whole model layers.

The overall steps followed in our attention-based network could be presented as follows:

1) Initialize the encoder hidden states: The initial encoder hidden states are a random small number and then the encoder produces hidden states of each element in the input sequence.
2) Initialize the decoder hidden states: Similar to the encoder hidden states, the decoder initial hidden states are initialized randomly.
3) Compute alignment scores: Alignment score is calculated between the previous decoder hidden state at t-1 and each of the encoder's hidden states at time step t. The alignment score function used in this paper is similar with the alignment score function in [15] which is called additive/concatenation. Therefore, the current encoder hidden states and the previous decoder hidden states are first concatenated and then passes through two consecutive feed-forward neural networks with $relu$ and $tanh$ activation function respectively. Alignment score could be computed using Equation (2).
4) Softmaxing the alignment scores: Each computed alignment score runs through a softmax layer.
5) Computing the Context Vector: The encoder hidden states and their respective soft-maxed alignment scores are multiplied and then summed up the results to form the context vector using Equation (1).
6) Decoding the output: The context vector is concatenated with the previous decoder output and fed into the decoder, to emit a character, at that time step along with the previous decoder hidden state. The process from step 3 to 6 are repeated themselves for each time step of the decoder Ty times.

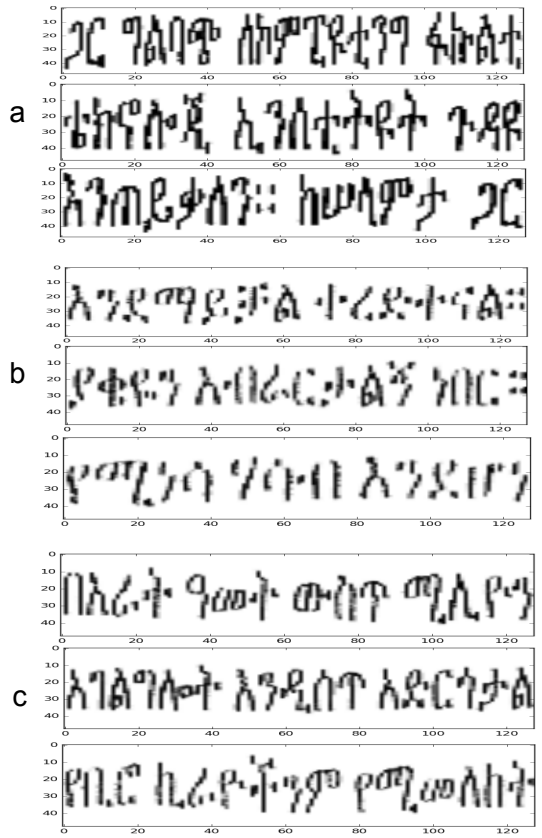

Figure 4. Sample text-line images from test sets of ADOCR database. The text-line images have a size of 48 by 128 pixels: (a) Printed text-line images written with the Power Geez font type. (b) Synthetically generated text-line images with the Power Geez font type. (c) Synthetically generated text-line images with the Visual Geez font type.

IV. Experimental Setup

In this section, the database used for training and evaluation, the training procedure, results obtained from our experiment are presented.

A. Database

There are few databases used in the various works on Amharic OCR reported in the literature. As reported in [45], the authors considered 5172 images of the most frequently used Amharic characters. A later work by Million et al [8] uses 76,800 character images with different font types and sizes which belong to 231 classes. Other researchers' work on Amharic OCR [46], [47], [19] reported that they used their private databases, but none of them were made their dataset publicly available. Therefore, the shortage of datasets has been continued as the main challenge and one of the limiting factors in developing reliable OCR systems for the Amharic script to date. To train and evaluate the performance of our Amharic OCR model, we use the same OCR database, employed in the work of Belay et al [1], [3], and which is freely available at http://www.dfki.uni-kl.de/~belay/.

The ADOCR database consists both character level and text-line level images. In this paper we only use the text-line

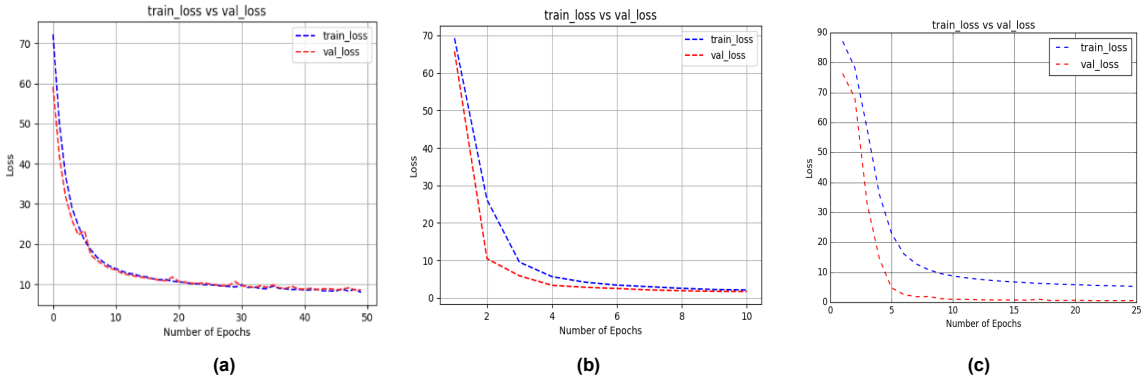

Figure 5. Training & validation losses of model training with different network settings: (**a**). CTC loss with an LSTM-CTC network settings [1]. (**b**) CTC loss with a CNN-LSTM-CTC settings [3]. (**c**) CE loss of the proposed model.

Table V
THE DETAILS OF THE ADOCR DATASET. PG AND VG DENOTE THE POWER GE'EZ AND VISUAL GE'EZ FONT TYPES RESPECTIVELY

	Printed	**Synthetic**	
Font type	**PG**	**PG**	**VG**
Number of samples	40,929	197,484	98,924
No. of test samples	2907	9245	6479
No. of training samples	38,022	188,239	92,445
No. of unique chars.	280	261	210

image database which consists a total of 337,337 Amharic text-line images with a size of 48×128 pixels, where 40,929 are printed text-line images written with power Geez font, 197,484, and 98,924 images are synthetically generated text-lines with Power Geez and Visual Geez fonts respectively. Sample text-line images from the test set and the details of text-line images in the ADOCR database are presented in Figuer 4 and Table V respectively.

B. Training procedure

The encoder and decoder of the RNNs have 128 hidden units each. The encoder RNN module consists of the forward and backward RNNs each having 128 hidden units that are stacked on top of CNN layers. Its decoder also has 128 hidden units. In the both cases, we use a multilayer perceptron network, two in the encoder part, and one in the decoder part, with a single soft-max hidden layer to compute the conditional probability of each target character. The whole architecture, depicted in Figure 2, computes a fully differentiable function, which parameters can be trained from end-to-end in a supervised manner.

The decoder is trained to generate the output based on the information gathered by the encoder by taking a prediction of randomly initialized hidden units as the first input to generate the first Amharic character. The input at each time step is predicted output which is usually called learning by injection technique. The loss is also calculated on the predicted outputs from each time step and the errors are backpropagated through time to update the parameters of the model. The optimized cost is the negative log-likelihood of the correct transcription and given as,

$$Loss_{(I,y)} = -\sum_{t=1}^{L} log(y_t|I) \tag{4}$$

where I is the image, $y = y_1, y_2, ..., y_{Ty}$ is the target character sequence and $p(y_t|I)$ is the probability of the decoder network output y_t given the image I.

LSTM network is employed for both the decoder and encoder module, where the encoder module uses two bidirectional-LSTM, and the decoder module uses a unidirectional-LSTM. The attention mechanism, in our model, computes the attention weight and outputs the context vector once it took all hidden sates of the encoder LSTM and the previous hidden state of the decoder-LSTM. We use a general attention strategy that usually attends to the entire input state space with all the encoder hidden states. It also manages and quantifying the interdependence between the input and output elements. Our attention module consists of two consecutively stacked feed forward neural network with a $relu$ and $tanh$ activation function respectively.

In the entire OCR model, the attention module is called till it reaches the maximum target sequence length, Ty times to give back the computed context vector which will be going to be used by the decoder-LSTM. At this time all y_t copies have the same weights by implementing layers with shareable weights. The networks are trained with an RMSProp optimizer [48], [49] and a batch size of 128 for 25 epochs. Since our model is trained to minimize categorical cross-entropy loss, each character in the ground truth was changed to one-hot encoding and then the decoder module generates the one-hot encoded equivalent of a character. In addition, considering research works done in the area of OCR, all the image from the ADOCR database are resized in to a size of 32×128 pixels.

The proposed model is implemented with Keras Application Program Interface (API) [50] on a TensorFlow backend [51].

The learning loss of the proposed model and other Amharic OCR models that are trained using different network settings and loss functions are depicted in Figure 5. Compared to [1], the proposed model converges with a smaller number of epochs and even though it takes 15 more epochs to converge, it took a smaller time for training compared to [3]. The proposed model incorporates divers network layers and parameter settings in a single unified framework and trained in an end-to-end fashion which is not usually optimal during training [52]. Therefore, the proposed model could be assessed and training time may be further improved. The next section presents the experimental results recorded during the OCR model evaluation and sample text-line images with their corresponding prediction texts.

C. Results

The performance of the model is measure using the Character Error Rate (CER) [1], where CER is computed using Equation (5), and 1.17% and 5.21% on ADOCR test datasets that are synthetically generated with *Visual Geez* and *Power Geez* fonts respectively. We also carried out preliminary experiments on a printed Amharic text-line image dataset from the ADOCR test sets and achieved a promising result with a CER of 1.54%.

$$CER(P,T) = \left(\frac{1}{c} \left(\sum_{n \in P, m \in T} D(n,m) \right) \right) \times 100, \quad (5)$$

where c is the total number of target character labels in the ground truth, P and T are the predicted and ground-truth labels, and $D(n,m)$ is the edit distance [53] between sequences n and m.

Sample text-line images that are wrongly recognized during evaluation of our Amharic OCR model are depicted in Figure 6. Characters marked by colored-boxes are wrong predictions (it can be deletion, substitution or insertion errors), characters marked by blue-boxes, such as ብ and ር from the first and second text-line image respectively are sample deleted characters. Other characters, such as ን and ት, from second and fourth predicted texts that are marked with green-boxes are insertion errors, while character ዳ in the third predicted text which is marked by red-box is one of the substitution error recorded during experimentation.

The other type of errors that usually affect the recognition performance of our model is the missing of characters either on the ground-truth or on the text-line image it self. For example, the character ም, marked by violet-box, in the third text-line image's ground-truth is one of the character missed during the text-line image generation. Even though all visible characters in the text-line image are predicted correctly, since the CER is computed between the ground-truth texts and the predicted texts, such type of error are still considered as deletion errors. Beside wrongly predicted text-line images, the fifth text-line image in Figure 6, is sample correctly predicted text-line image from ADOCR test sets.

The recognition performance of our model is compared against the previous models' recognition performance on the test set of ADOCR database. The comparisons among the proposed approach and others' attempts made based on the ADOCR database [1] are listed in Table VI. The performance of the proposed model shows better results on the printed dataset while results on synthetic datasets are comparable. Most of the characters are missed at the beginning and/or end of the text-line images in the synthetic data during text-line image generation, while there are few miss-alignments in the ground-truth of the printed text-line images that reduce the recognition performance of our model. Using better annotation tools or careful manual annotation specifically if those missed characters during text-line image segmentation or synthetic data generation are annotated/aligned the recognition performance of the OCR model could be enhanced.

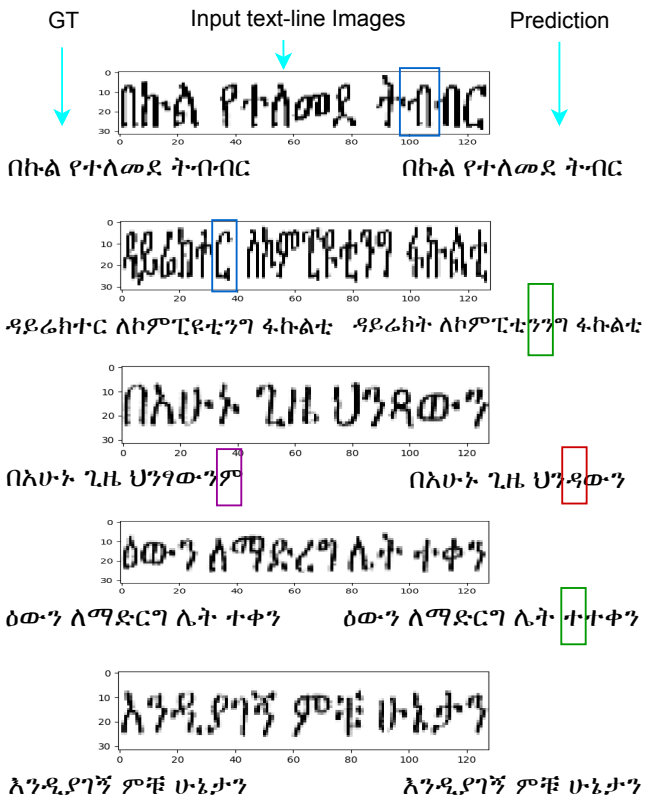

Figure 6. Sample predicted text-line images (*middle*), from test sets of ADOCR database [1], with their corresponding Ground-Truth (GT) texts (*left*) and model predictions (*right*).

V. Conclusions

This paper successfully approaches the Amharic script recognition with an attention-based sequence-to-sequence architecture. The proposed model integrates a convolutional feature extractor with a recurrent neural network to encode both the visual information, as well as the temporal context in the input image while a separate recurrent neural network is employed to decode the actual Amharic character sequence. Overall, we obtain results that are competitive with the state-of-the-art recognition performance on ADOCR datasets. As

Table VI
COMPARISON OF TEST RESULTS (CER IN %).

Methods	#train-set	#test-set	image-type	Font	CER
BLSTM-CTC [54] *	96,000	12 pages	printed	-	2.12%
BLSTM-CTC [1]	38,022	2,907	Printed	Power Ge'ez	8.54%
BLSTM-CTC [1]	188,239	9,245	Synthetic	Power Ge'ez	4.24%
BLSTM-CTC [1]	92,445	6,479	Synthetic	Visual Ge'ez	2.28%
CNN-BLSTM-CTC [3]	38,022	2,907	Printed	Power Ge'ez	1.56%
CNN-BLSTM-CTC[3]	188,239	9,245	Synthetic	Power Ge'ez	3.73%
CNN-BLSTM-CTC[3]	92,445	6,479	Synthetic	Visual Ge'ez	1.05%
Ours	38,022	2,907	Printed	Power Ge'ez	**1.54**%
Ours	188,239	9,245	Synthetic	Power Ge'ez	**5.21**%
Ours	92,445	6,479	Synthetic	Visual Ge'ez	**1.17**%

* Denotes methods tested on different datasets.

we observed the empirical results, the attention-based encoder-decoder model becomes poor when the sequence length increases. In most cases, the first characters are always correctly predicted while the rest errors have no patterns; thus it is hard to learn in the initial training stage for longer input sequences. Such character errors are not observed in the LSTM-CTC based networks. In addition, we have observed a coverage problem which leads to an over-translation or under-translation. Hence, to minimize errors in the longer input sequence and handle overall alignment information, the attention-based encode-decoder model should be further enhanced.

ACKNOWLEDGEMENTS

The first author was partially supported by DAAD scholarship (Funding program No. 57375975), Department of Computer Science, Technical University of Kaiserslautern, Germany and Bahir Dar institute of Technology Ethiopia. This research was carried out at the Augmented Vision lab at DFKI, Kaiserslautern, Germany.

REFERENCES

[1] B. Belay, T. Habtegebirial, M. Liwicki, G. Belay, and D. Stricker, "Amharic text image recognition: Database, algorithm, and analysis," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1268–1273, IEEE, 2019.

[2] G. T. Mekuria and G. T. Mekuria, "Amharic text document summarization using parser," *International Journal of Pure and Applied Mathematics*, vol. 118, no. 24, 2018.

[3] B. Belay, T. Habtegebrial, M. Meshesha, M. Liwicki, G. Belay, and D. Stricker, "Amharic ocr: An end-to-end learning," *Applied Sciences*, vol. 10, no. 3, p. 1117, 2020.

[4] Y. JEMANEH, "Will amharic be aus lingua franca." https://www.press.et/english/?p=2654#l, 2019.

[5] A. Teferra, "Amharic: Political and social effects on english loan words," *Multilingual Matters*, vol. 140, p. 164, 2008.

[6] R. Meyer, "Amharic as lingua franca in ethiopia," *Lissan: Journal of African Languages and Linguistics*, vol. 20, no. 1/2, pp. 117–132, 2006.

[7] A. Wion, "The national archives and library of ethiopia," in *six years of Ethio-French cooperation (2001-2006)*, 2007.

[8] M. Meshesha and C. Jawahar, "Optical character recognition of amharic documents," *African Journal of Information & Communication Technology*, vol. 3, no. 2, 2007.

[9] B. Belay, T. Habtegebrial, M. Liwicki, G. Belay, and D. Stricker, "Factored convolutional neural network for amharic character image recognition," in *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 2906–2910, IEEE, 2019.

[10] T. Bloor, "The ethiopic writing system: a profile," *Journal of the Simplified Spelling Society*, vol. 19, no. 2, pp. 30–36, 1995.

[11] T. M. Breuel, A. Ul-Hasan, M. A. Al-Azawi, and F. Shafait, "High-performance ocr for printed english and fraktur using lstm networks," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pp. 683–687, IEEE, 2013.

[12] D. S. Maitra, U. Bhattacharya, and S. K. Parui, "Cnn based common approach to handwritten character recognition of multiple scripts," in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pp. 1021–1025, IEEE, 2015.

[13] M. Mondal, P. Mondal, N. Saha, and P. Chattopadhyay, "Automatic number plate recognition using cnn based self synthesized feature learning," in *Calcutta Conference (CALCON), 2017 IEEE*, pp. 378–381, IEEE, 2017.

[14] K. Cho, A. Courville, and Y. Bengio, "Describing multimedia content using attention-based encoder-decoder networks," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 1875–1886, 2015.

[15] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[16] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[17] A. Das, J. Li, G. Ye, R. Zhao, and Y. Gong, "Advancing acoustic-to-word ctc model with attention and mixed-units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 1880–1892, 2019.

[18] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[19] W. Alemu, "The application of ocr techniques to the amharic script," *An MSc thesis at Addis Ababa University Faculty of Informatics*, 1997.

[20] J. Cowell and F. Hussain, "Amharic character recognition using a fast signature based algorithm," in *Information Visualization, 2003. IV 2003. Proceedings. Seventh International Conference on*, pp. 384–389, IEEE, 2003.

[21] Y. Assabie and J. Bigun, "Hmm-based handwritten amharic word recognition with feature concatenation," in *2009 10th International Conference on Document Analysis and Recognition*, pp. 961–965, IEEE, 2009.

[22] B. Belay, T. Habtegebrial, and D. Stricker, "Amharic character image recognition," in *2018 IEEE 18th International Conference on Communication Technology (ICCT)*, pp. 1179–1182, IEEE, 2018.

[23] M. S. Gondere, L. Schmidt-Thieme, A. S. Boltena, and H. S. Jomaa, "Handwritten amharic character recognition using a convolutional neural network," *arXiv preprint arXiv:1909.12943*, 2019.

[24] A. Graves, M. Liwicki, H. Bunke, J. Schmidhuber, and S. Fernández, "Unconstrained on-line handwriting recognition with recurrent neural networks," in *Advances in neural information processing systems*, pp. 577–584, 2008.

[25] M. Liwicki, A. Graves, S. Fernàndez, H. Bunke, and J. Schmidhuber, "A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks," in *Proceedings of the 9th International Conference on Document Analysis and Recognition, ICDAR 2007*, 2007.

[26] V. Carbune, P. Gonnet, T. Deselaers, H. A. Rowley, A. Daryin, M. Calvo, L.-L. Wang, D. Keysers, S. Feuz, and P. Gervais, "Fast multi-language lstm-based online handwriting recognition," *International Journal on Document Analysis and Recognition (IJDAR)*, pp. 1–14, 2020.

[27] Y.-C. Wu, F. Yin, Z. Chen, and C.-L. Liu, "Handwritten chinese text recognition using separable multi-dimensional recurrent neural network," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1, pp. 79–84, IEEE, 2017.

[28] T. M. Breuel, "High performance text recognition using a hybrid convolutional-lstm implementation," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1, pp. 11–16, IEEE, 2017.

[29] N.-T. Ly, C.-T. Nguyen, K.-C. Nguyen, and M. Nakagawa, "Deep convolutional recurrent network for segmentation-free offline handwritten japanese text recognition," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 7, pp. 5–9, IEEE, 2017.

[30] R. Ghosh, C. Vamshi, and P. Kumar, "Rnn based online handwritten word recognition in devanagari and bengali scripts using horizontal zoning," *Pattern Recognition*, vol. 92, pp. 203–218, 2019.

[31] A. Yuan, G. Bai, L. Jiao, and Y. Liu, "Offline handwritten english character recognition based on convolutional neural network," in *2012 10th IAPR International Workshop on Document Analysis Systems*, pp. 125–129, IEEE, 2012.

[32] P. Shivakumara, D. Tang, M. Asadzadehkaljahi, T. Lu, U. Pal, and M. H. Anisi, "Cnn-rnn based method for license plate recognition," *CAAI Transactions on Intelligence Technology*, vol. 3, no. 3, pp. 169–175, 2018.

[33] R. Messina and J. Louradour, "Segmentation-free handwritten chinese text recognition with lstm-rnn," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 171–175, IEEE, 2015.

[34] J.-C. B. J.-M. O. Vinh Loc Cu, Trac Nguyen, "A robust watermarking approach for security issue of binary documents using fully convolutional networks," *International Journal on Document Analysis and Recognition (IJDAR)*, 2020.

[35] G. Renton, Y. Soullard, C. Chatelain, S. Adam, C. Kermorvant, and T. Paquet, "Fully convolutional network with dilated convolutions for handwritten text line segmentation," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 21, no. 3, pp. 177–186, 2018.

[36] K. Manjusha, M. A. Kumar, and K. Soman, "Integrating scattering feature maps with convolutional neural networks for malayalam handwritten character recognition," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 21, no. 3, pp. 187–198, 2018.

[37] P. Doetsch, A. Zeyer, and H. Ney, "Bidirectional decoder networks for attention-based end-to-end offline handwriting recognition," in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pp. 361–366, IEEE, 2016.

[38] T. Bluche, "Joint line segmentation and transcription for end-to-end handwritten paragraph recognition," in *Advances in Neural Information Processing Systems*, pp. 838–846, 2016.

[39] J. Poulos and R. Valle, "Character-based handwritten text transcription with attention networks," *arXiv preprint arXiv:1712.04046*, 2017.

[40] A. Chowdhury and L. Vig, "An efficient end-to-end neural model for handwritten text recognition," *arXiv preprint arXiv:1807.07965*, 2018.

[41] C.-Y. Lee and S. Osindero, "Recursive recurrent nets with attention modeling for ocr in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2231–2239, 2016.

[42] J. Zhang, J. Du, and L. Dai, "Track, attend, and parse (tap): An end-to-end framework for online handwritten mathematical expression recognition," *IEEE Transactions on Multimedia*, vol. 21, no. 1, pp. 221–233, 2018.

[43] X. Chen, T. Wang, Y. Zhu, L. Jin, and C. Luo, "Adaptive embedding gate for attention-based scene text recognition," *Neurocomputing*, vol. 381, pp. 261–271, 2020.

[44] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

[45] D. Teferi, "Optical character recognition of typewritten amharic text," Master's thesis, School of Information studies for Africa, Addis Ababa, 1999.

[46] Y. Assabie, "Optical character recognition of amharic text: an integrated approach," Master's thesis, Addis Ababa University, Addis Ababa, 2002.

[47] B. Y. Reta, D. Rana, and G. V. Bhalerao, "Amharic handwritten character recognition using combined features and support vector machine," in *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 265–270, IEEE, 2018.

[48] Y. Bengio and M. CA, "Rmsprop and equilibrated adaptive learning rates for nonconvex optimization," *corr abs/1502.04390*, 2015.

[49] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

[50] F. Chollet, "Introduction to keras," *March 9th*, 2018.

[51] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pp. 265–283, 2016.

[52] T. Glasmachers, "Limits of end-to-end learning," *arXiv preprint arXiv:1704.08305*, 2017.

[53] G. Navarro, "A guided tour to approximate string matching," *ACM computing surveys (CSUR)*, vol. 33, no. 1, pp. 31–88, 2001.

[54] D. Addis, C.-M. Liu, and V.-D. Ta, "Printed ethiopic script recognition by using lstm networks," in *2018 International Conference on System Science and Engineering (ICSSE)*, pp. 1–6, IEEE, 2018.