RRConvNet: Recursive-residual Network for Real-life Character Image Recognition

Tadele Mengiste[1], Birhanu Hailu Belay[1], Bezawork Tilahun[1], Tsiyon Worku[1] and Tesfa Tegegne[1,2]
[1]*Faculty of Computing, Bahir Dar Institute of Technology, Bahir Dar, Ethiopia*
[2]*ICT4D Research Center, Bahir Dar Institute of Technology, Bahir Dar, Ethiopia*

Keywords: Ethiopic Character Image, OCR, Pattern Recognition, Recursive-CNN, Skip-connection.

Abstract: Variations in fonts, styles, and ways to write a character have been the major bottlenecks in OCR research. Such problems are swiftly tackled through advancements in deep neural networks (DNNs). However, the number of network parameters and feature reusability are still the issues when applying Deep Convolutional Neural networks(DCNNs) for character image recognition. To address these challenges, in this paper, we propose an extensible and recursive-residual ConvNet architecture (RRConvNet) for real-life character image recognition. Unlike the standard DCCNs, RRConvNet incorporates two extensions: recursive-supervision and skip-connection. To enhance the recognition performance and reduce the number of parameters for extra convolutions, layers of up to three recursions are proposed. Feature maps are used after each recursion for reconstructing the target character. For all recursions of the reconstruction method, the reconstruction layers are the same. The second enhancement is to use a short skip-connection from the input to the reconstruction output layer to reuse the character features maps that are already learned from the prior layer. This skip-connection could be also used as an alternative path for gradients where the gradient is too small. With an overall character recognition accuracy of 98.2 percent, the proposed method achieves a state-of-the-art result on both publicly available and private test datasets.

1 INTRODUCTION

Nowadays, it is becoming increasingly important to have documents in a digital format for easily accessing information, efficient data storage, and retrieval. For example, if a manuscript was published 100 years ago, it is quite impossible to have text for this ancient manuscript in an editable document such as a word or text file. So, the only choice that remains is to type the entire text which is a very exhaustive process if the text is large. The solution to this problem is optical character recognition. The use and applications of Optical Character Recognition (OCR) systems have been developed and widely applied for the digitization of many documents written in various scripts (Elleuch et al., 2016).

Optical Character Recognition (OCR) is the process of extracting text from an image handwritten or machine-printed documents. A single page of the sample Ethiopic script is illustrated in Figure 1. OCR has been and is widely used for many scripts as a method of digitizing printed texts which can be electronically edited, searched, stored more compactly, displayed online, and also used to facilitate the human-to-machine and machine-to-machine communication such as machine translation, text-to-speech, key data, and text mining (Belay et al., 2019b).

Figure 1: Sample Ethiopic script image.

Optical character recognition is a strenuous field of research that requires great effort and researchers have been exploring different strategies for about the

Mengiste, T., Belay, B., Tilahun, B., Worku, T. and Tegegne, T.
RRConvNet: Recursive-residual Network for Real-life Character Image Recognition.
DOI: 10.5220/0011270400003277
In *Proceedings of the 3rd International Conference on Deep Learning Theory and Applications (DeLTA 2022)*, pages 110-116
ISBN: 978-989-758-584-5; ISSN: 2184-9277

past half a century. Recently, deep neural networks have drawn the observance of many researchers due to their competency in figuring out computer vision problems such as object detection, classification, and recognition undoubtedly well (Bai et al., 2014). CNN is one of the most prominent types of deep neural networks, it can learn and extract features from images. The CNN classifier can effectively recognize characters located in the image.

As Kim (Kim et al., 2016) presented in detail, using the standard deep CNN architecture (Bora et al., 2020), for character recognition substantially boosts the number of parameters and needs more data to prevent over-fitting. Important hyper-parameters such as the degree of parameter sharing, number of layers, units per layer, and the overall number of parameters must be selected manually through trial-and-error (Eigen et al., 2013). In very deep neural networks, the gradient becomes too small when we approach the earlier layers. Thus, we will not update the earlier layers since the gradient becomes zero (Tan and Lim, 2019; Dai and Heckel, 2019). In such standard network architectures, there is also low-level information shared between the input and output layers (He et al., 2016).

To address this, we have introduced a Recursive-residual Convent (RRConvNet) for real-life Ethiopic character image recognition. Sample Ethiopic script is shown in Figure 1. This method consists of two approaches that are used to ease the difficulty of training. First, all recursions are supervised. Feature maps after each recursion are used to reconstruct the target character. The reconstruction method (layers dedicated to reconstruction) is the same for all recursions. As each recursion leads to a different character prediction, all predictions resulting from different levels of recursions that deliver a more accurate final prediction are combined. A recursive neural network is a kind of deep neural network created after applying the same set of weights recursively to the structured inputs. Finally, a structured prediction over variable-size input structures or a scalar prediction on it is produced by traversing a given structure in topological order.

The second extension is to use a skip-connection from the input to the reconstruction layer. In the experiment, the input to the layers for output reconstruction have explicitly connected. This is especially effectual when input and output are tremendously correlated. It utilizes a very large context compared to previous character recognition methods with only a single recursive layer (Kim et al., 2016) and few parameters since adding another layer increase the number of parameters (Eigen et al., 2013). The skip-connection has two advantages. First, the network capacity to store the input signal during recursions is saved. Second, the exact copy of the input signal can be used during target prediction (Kim et al., 2016). The proposed method demonstrates state-of-the-art performance in common benchmarks.

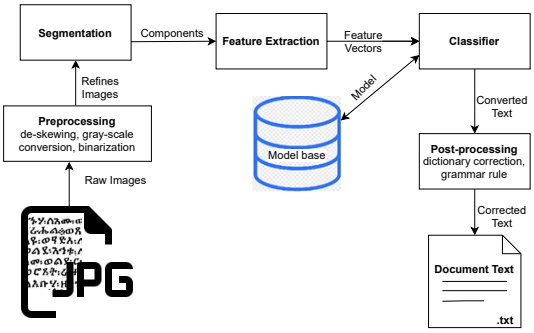

Figure 2: Overview of the generic OCR process: (Meshesha and Jawahar, 2007). The process of using an OCR system, in general, can be broken down into four key steps: The first phase is image prepossessing, which involves a wide range of imaging functions such as image rotation, binarization, and de-skewing to improve image quality. The second document analysis process defines the text recognition areas and provides data on the layout and formatting aspects of each page, as well as the document's overall structure. At the recognition stage, the actual texts are predicted. In the post-processing stage, the OCR errors are repaired and the model is updated.

The rest of the paper is organized as follows. Related works are reviewed in section 2. In section 3, the proposed system architecture and detail of datasets are presented. Section 4 presents experimental results and finally, conclusions are presented in section 5.

2 RELATED WORKS

Pre-processing, segmentation, feature extraction, and classification are the generic processes that character recognition entails. While each stage affects recognition accuracy, the feature extraction technique (Gondere et al., 2019) plays the most important influence. Layout analysis and text line extraction are the first steps in analyzing a document image. For each line, the text is divided into distinct character pictures. Finally, the classifier receives these character pictures and generates class labels. An overview of the OCR entire process is illustrated in 2. This generic OCR process is proposed by (Meshesha and Jawahar, 2007). Following such generic OCR process performs better for well-printed or well-written manuscripts.

For a long time, the document analysis community has been focused on automating reliable document

image recognition and information extraction methods (Younas et al., 2017). In contrast to Latin and Asian scripts, OCR research for low-resource scripts such as Ethiopic script is still lacking(Belay et al., 2019b; Assabie and Bigun, 2007; Cowell and Hussain, 2003). Various methodologies have been used to build OCR methods for a variety of scripts, with ground-breaking results. A CNN-based handwritten Bangala character recognition system has been proposed by (Rahman et al., 2015), which normalizes written character images before using CNN to classify individual characters, with a recognition accuracy of 85.36 percent on a dataset of 20000 characters.

Mars and Antoniadis (Mars and Antoniadis, 2016) presented a model for Optical Character Recognition (OCR) in the Telugu language, which includes three parts: a database of Telugu characters, a deep learning-based OCR algorithm, and an online client-server application for the developed algorithm. Their model is based on Convolutional Neural Networks (CNNs) algorithm reasonably to classify the characters. They have applied their OCR system to real data and the results were good. A hierarchical fuzzy convolutional neural network (HFCNN) (Chaudhuri and Ghosh, 2017) is used for the Czech language character recognition task. It takes full advantage of deep CNN towards modeling long-term information of data sequences located on the database that contains unconstrained handwritten text at a resolution of 300 dpi as PNG images with 256 gray levels.

The first work for Amharic script recognition is proposed by Alemu (Alemu, 1997). Alemu developed an algorithm based on the laser printouts of text with normal type style of WashRa font, 12-point font sizes, and reported 97.31% of character recognition accuracy. Later, Yaregal Assabie (Yaregal, 2002) explored various OCR development approaches to develop an OCR model for Amharic script and come up with a versatile algorithm that is independent of the Amharic characters' font size. The system correctly recognized 73.18% of the characters included in the training set.

Wondwossen (Mulugeta, 2004) developed an OCR model for a special type of handwritten Amharic text ("Yekum Tsifet") using a neural network approach. The results reported in this work are 95.96% for segmentation rate and 98.8% to 20.3% for recognition. An enhanced optical character recognition for real-life Amharic degraded documents have been developed by (Birhanu, 2008) using an Artificial Neural Network (ANN) approach for classifying the features generated. Accordingly, an average recognition rate of 96.87% for the test sets from the training sets and 11.40% recognition rate is observed for the new test sets. So far, there are very limited research efforts made for Ethiopic character recognition and there is no effective OCR application. The possible reasons mentioned are the use of a large number of characters in the writing, the existence of a large set of visually similar characters, variations in font, style, and writing materials, and the unavailability of standard dataset (Assabie and Bigun, 2011; Meshesha and Jawahar, 2007; Belay et al., 2019a).

The prior studies focused on creating a classifier that can handle character image sets with limited type and known fonts and has been focused on the standard deep neural network layers with a large number of parameters. In comparison to other character recognition methods, we developed an adaptive recursive CNN method that uses recursive convolution and a skip connection to reduce the number of parameters, save network capacity for storing inputs during recursions and enable feature reusability.

3 MATERIAL AND METHODS

A general OCR system includes the basic steps shown in Figure 2 (Meshesha and Jawahar, 2007) that start from preparing a dataset followed by training the model. In this section, the nature of the dataset used for training and model evaluation, the proposed algorithm, and the training schemes that we've followed throughout the research are presented.

3.1 Datasets

This section describes the datasets we utilize for our experiments. To train and test the proposed model, two different datasets are used. The first dataset is the ADOCR database [1], a public and freely available dataset, which contains about 77,994 Amharic character images. Out of 77,994 images, 7800 of 10% of them are used for the testing, and the remaining 70,194 for training purposes. The second dataset consists of 10470 character images that are collected from various private sources and it includes historical hand-written and printed characters with different and also unknown font types. Once we collect these character images, we manually labeled each of them. Then we consider randomly selected 10% of the dataset as a test set and the remaining character images for training purposes. Few sample character images taken from the database are shown in Figure 3.

[1] http://www.dfki.uni-kl.de/ belay/

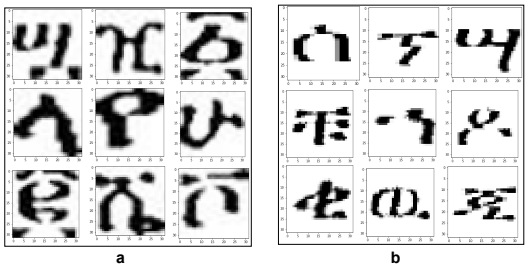

Figure 3: Sample segmented and binary Ethiopic character images. (a) Real-life Ethiopic character image. (b) Synthetically generated character images from ADOCR database[1].

3.2 Proposed Algorithm

The overall framework of the proposed approach is shown in Figure 4. CNNs have the main feature of sharing inner parameters across the network, which leads to architectural properties of scale, shift, and distortion invariance, making them a powerful tool for image feature extraction with few preprocessing steps (Goodfellow et al., 2016). Those properties mean that regardless of where and how a specific raw feature appears in the image, a suitable and well-trained CNN can capture that feature. After feature extraction has been practiced, images can be classified, segmented, or even reconstructed. CNNs are formulated as a feature extraction block and a classification block (Fig 4). The initial block receives a grid-like topology input and hierarchically extracts representative features followed by another block responsible for receiving the top hierarchical feature and providing a final matrix of prediction.

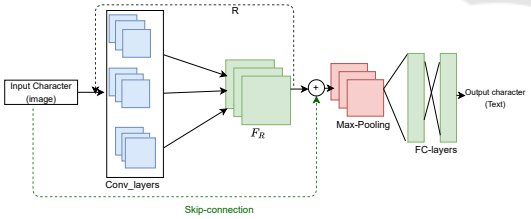

Figure 4: The proposed Recursive-ConvNet architecture showing the features extraction and character recognition zones together with main notations where R=3. The $\bigoplus$, denotes the concatenation operation as used in (Huang et al., 2017).

3.2.1 Recursive-ConvNet

Consider a character recognition problem in which the aim is to associate an input feature x, character image, with an output y, Unicode character, through the neural network function f(x). This neural network model is trained using *Adam* optimizer (Kingma and Ba, 2014) to minimize a loss function L over a character image dataset D. This network architecture is defined using a convolutional layer C, parameterized by weights W. To build a deep neural network, called Recursive-residual ConveNet, we use a single convolution layer, as proposed in (Kim et al., 2016), which is iteratively applied R times on successive steps. The convolution layer in the proposed model consists of a kernel size of 3×3, 32 filters and *same* padding. This network architecture, RecurssivConvNet, is then defined by the following recursive sequence:

$$\begin{cases} x_0 = 32 \times 32 \\ x_{t+1} = C(x_t) \text{ for t=0,1,..., R} \\ b = Add()[x_R, x_0], \\ b_{pool} = maxpooling(b) \\ y = FC(b_{pool}) \end{cases} \quad (1)$$

where x_0 denotes the input character image, x_{t+1} is feature maps after passing a convolution layer C. b is the concatenated value of the feature map from the x_R and input image that denotes the skip connection in the network, x_R is the feature map at the end of the whole iteration (R), and FC is the fully connected network layer.

In our proposed architecture, the RecursiveConvNet layer iteratively performs a convolution operation R times by receiving a character image as input where R=3 and adopted from (Kim et al., 2016). Followed by the input image tensor which is directly fed into the reconstruction net whenever it is used during the recursions. In this case, the skip-connection has two advantages. First, the network capacity to store the input features during recursions is saved. Second, the exact copy of the input features map of the character image can be used during target character recognition which is usually called feature reusability. In addition, this skip connection can be used as an alternative path during back-propagation where the gradient is too small; thus, a vanishing gradient might not be an issue during training our network. The concatenated features are then passed through a max-pooling layer. Before passing the features from the max-pooling layer to the recognition phase, a similar convolution operation is applied to it.

In this case, pooling has two effects. First, it diminishes the number of computations made by one iteration which significantly increases the speed of a forward pass in the network model training. Second, it allows the convolutional filter to take effect in larger regions of the initial character images. Finally, the feature maps from the max-pooling layer are flattened and fed into the Fully connected (FC) layer to compute the probability distribution over each class, where this probability is computed using, equation (2), soft-max activation function. The idea behind the

recursiveness, in this network architecture, is the enhancement of outputs quality by considering the previously simulated information.

$$f(z)_j = \frac{e^{z_j}}{\sum_{i=1}^{K} e^{z_i}} \quad (2)$$

where z is the input vector, e^{z_j} denotes the standard exponential function for input vector, K is number of classes, and e^{z_i} is the standard exponential function for output vector.

The recognition loss of the this network architecture is categorical cross-entropy loss function and can be computed as,

$$L = -\sum_{i}^{C} y_i log(\hat{y}_i) \quad (3)$$

where$\hat{y}_i$ is thei-th scalar value in the model output. y_i is the corresponding ground-truth value, C is the class of the sample.

The overall character recognition accuracy of the model is computed as a ratio of incorrectly recognized characters and the total number of characters in the test dataset.

4 EXPERIMENTAL RESULTS

Our methodology is enforced in Keras Application Program Interface(API) with a TensorFlow backend. In addition, we resized the images into a size of

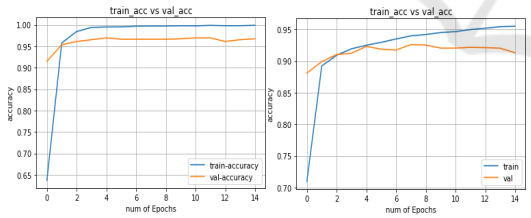

Figure 5: Learning curve. (a) Training and validation accuracy of real-life character images, (b) training and validation accuracy using the ADOCR dataset.

32×32 pixel. The architecture is trained with a batch size of 16 running for 15 epochs. To select suitable network parameters, different values of these parameters were considered and tuned during experimentation, and the results reported in this paper are obtained using an Adam optimizer employing a convolutional neural network with a feature map of 64, kernel-size of 3×3, and stride of 2. This CNN layer is called and convolves, over the input image, three times recursively. The input image tensor is concatenated, through skip connection, with the output tensor of the CNN layer and passes through a max-pooling layer that has a 2×2 kernel size followed by two fully connected layers having 512 and 1024 neurons respectively.

Since we have two different datasets, we conducted two experiments and learning curve of each experiment is shown in Figure 5. In the first experiment, the model was trained and evaluated with 28×28 synthetic character image from the ADOCR dataset having 231 unique characters. The second experiment was conducted using real-life character images that consist of 319 unique Ethiopic characters and digits written with different and also unknown fonts.

Our network is trained using the different batch sizes, epochs and network setups. The best test result is recorded with in batch size of 16 running for 15 epochs. The character recognition accuracy of our model is calculated as a ratio of correctly recognized characters and the total number of characters in the test set and then multiply by 100 (see equation (4)).

$$A = \frac{\#\text{correctly recognized characters}}{\#\text{ characters in the test set}} \times 100 \quad (4)$$

where A denotes accuracy.

As illustrated in Figure 6, some characters are incorrectly recognized due to similarity (e.g ⓫ as ⓪) while the others are incorrectly recognized (e.g ⓥ as ❦) even with no similarity between them. While the others are miss-recognized due to the quality of the image. Some of these deformed character image are show in Figure 7.

4.1 Performance Comparison

Based on the results recorded during experimentation, 98.2 % of real-life character images and 94.75% of test images from the ADOCR database are correctly predicted respectively. Compared to the real-life character images, Character images in the ADOCR dataset are highly degraded and even there are some deformed character images as illustrated in figure 7. Due to this the character recognition accuracy on the ADOCR test dataset is much lower than that of the recognition accuracy on the real-life test sets. As it is observed in Table 1, we have achieved better performance compared with works done on Amharic OCR using 231 classes. Others' work is presented in Table 1, here, it is not to compare the performance directly since they used different datasets and experimental settings. However, it's simply to indicate the progress of OCR for the Amharic script.

Table 1: Performance of prior works and proposed method.

Authors	#Dataset	Type	Accuracy
(Yaregal, 2002)	1010	handwritten	73.18%
(Meshesha and Jawahar, 2007)	76800	printed	90.37%
(Belay et al., 2018)	80,000	synthetic	92.71%
(Belay et al., 2019a)*	77,994	Synthetic	93.45%
Ours*	77,994	Synthetic	94.75%
Ours	10470	real-life	98.2%

* Denotes methods tested on similar datasets.

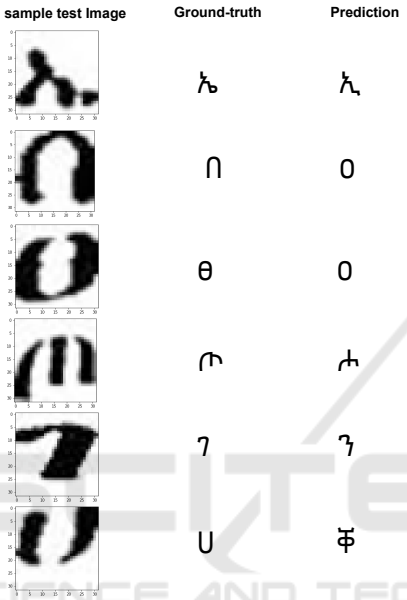

Figure 6: A typical diagram that shows the sample miss-recognized historical characters images.

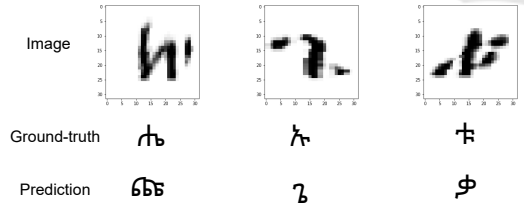

Figure 7: Deformed characters, from the ADOCR test set that are wrongly recognized.

5 CONCLUSIONS

In this paper, we have introduced an adaptable Recursive-CNN-based method, for real-life Ethiopic character image recognition. This method is a lightweight architecture with few network parameters and limited network capacity to store input features that can be easily adapted to other image-based pattern recognition problems. The proposed method is designed based on the existing VGGnet-like CNN architecture with two extensions; the recursive convolution where a single Convolutional layer recursively is convolved through the input character image and one pooling layer followed by two fully connected layers. Second, is the Skip-connection where the input feature maps are concatenated with the output feature maps of a character image. We evaluated our RRConvNet model on a publicly available dataset which has 77994 sample Amharic character images and 10470 real-life Ethiopic character images. Experiments on these datasets have shown that our model achieves high character recognition accuracy. Our proposed method minimizes the number of network parameters and allows feature reusability. As part of future work, the proposed network architecture can be extended to text-line level image recognition tasks, as a feature extractor, by integrating with the recurrent neural networks.

ACKNOWLEDGEMENTS

This research work was partially supported by the ICT4D research center (annually research grant), Bahir Dar Institute of Technology, Bahir Dar University

REFERENCES

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

Assabie, Y. and Bigun, J. (2007). A neural network approach for multifont and size-independent recognition of ethiopic characters. *Advances in Pattern Recognition*, pages 129–137.

Assabie, Y. and Bigun, J. (2011). Offline handwritten amharic word recognition. *Pattern Recognition Letters*, 32(8):1089–1099.

Bai, J., Chen, Z., Feng, B., and Xu, B. (2014). Image character recognition using deep convolutional neural network learned from different languages. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 2560–2564. IEEE.

Belay, B., Habtegebrial, T., Liwicki, M., Belay, G., and Stricker, D. (2019a). Factored convolutional neural network for amharic character image recognition. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2906–2910. IEEE.

Belay, B., Habtegebrial, T., and Stricker, D. (2018). Amharic character image recognition. In *2018 IEEE 18th International Conference on Communication Technology (ICCT)*, pages 1179–1182. IEEE.

Belay, B. H., Habtegebirial, T., Liwicki, M., Belay, G., and Stricker, D. (2019b). Amharic text image recognition: Database, algorithm, and analysis. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1268–1273. IEEE.

Birhanu, A. T. (2008). *Amharic Character Recognition System for Printed Real-Life Documents*. PhD thesis, Addis Ababa University.

Bora, M. B., Daimary, D., Amitab, K., and Kandar, D. (2020). Handwritten character recognition from images using cnn-ecoc. *Procedia Computer Science*, 167:2403–2409.

Chaudhuri, A. and Ghosh, S. K. (2017). Optical character recognition system for czech language using hierarchical deep learning networks. In *Proceedings of the Computational Methods in Systems and Software*, pages 114–125. Springer.

Cowell, J. and Hussain, F. (2003). Amharic character recognition using a fast signature based algorithm. In *Proceedings on Seventh International Conference on Information Visualization, 2003. IV 2003.*, pages 384–389. IEEE.

Dai, Z. and Heckel, R. (2019). Channel normalization in convolutional neural network avoids vanishing gradients. *arXiv preprint arXiv:1907.09539*.

Eigen, D., Rolfe, J., Fergus, R., and LeCun, Y. (2013). Understanding deep architectures using a recursive convolutional network. *arXiv preprint arXiv:1312.1847*.

Elleuch, M., Tagougui, N., and Kherallah, M. (2016). A novel architecture of cnn based on svm classifier for recognising arabic handwritten script. *International Journal of Intelligent Systems Technologies and Applications*, 15(4):323–340.

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

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

Kim, J., Lee, J. K., and Lee, K. M. (2016). Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1637–1645.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Mars, A. and Antoniadis, G. (2016). Arabic online handwriting recognition using neural network. *International Journal of Artificial Intelligence and Applications (IJAIA)*, 7(5).

Meshesha, M. and Jawahar, C. (2007). Optical character recognition of amharic documents. *African Journal of Information & Communication Technology*, 3(2).

Mulugeta, W. (2004). Ocr for special type of handwritten amharic text. *Yekum Tsifet"), Neural Network Approach*.

Rahman, M. M., Akhand, M., Islam, S., Shill, P. C., and Rahman, M. H. (2015). Bangla handwritten character recognition using convolutional neural network. *International Journal of Image, Graphics and Signal Processing*, 7(8):42.

Tan, H. H. and Lim, K. H. (2019). Vanishing gradient mitigation with deep learning neural network optimization. In *2019 7th international conference on smart computing & communications (ICSCC)*, pages 1–4. IEEE.

Yaregal, A. (2002). Optical character recognition of amharic text: an integrated approach. *School of Information Studies for Africal. Addis Ababa University. Addis Ababa*.

Younas, J., Afzal, M. Z., Malik, M. I., Shafait, F., Lukowicz, P., and Ahmed, S. (2017). D-star: A generic method for stamp segmentation from document images. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 248–253. IEEE.