Fetulhak Abdurahman
Jimma University, afetulhak@yahoo.com, Jimma-Ethiopia

DOI	http://dx.doi.org/10.12739/NWSA.2019.14.2.1A0433
ORCID ID	0000-0002-5670-0319
CORRESPODING AUTHOR	Fetulhak Abdurahman

HANDWRITTEN AMHARIC CHARACTER RECOGNITION SYSTEM USING CONVOLUTIONAL NEURAL NETWORKS

ABSTRACT

Amharic language is an official language of the federal government of the Federal Democratic Republic of Ethiopia. Accordingly, there is a bulk of handwritten Amharic documents available in libraries, information centres, museums, and offices. Digitization of these documents enables to harness already available language technologies to local information needs and developments. Converting these documents will have a lot of advantages including (i) to preserve and transfer history of the country (ii) to save storage space (ii) proper handling of documents (iv) enhance retrieval of information through internet and other applications. Handwritten Amharic character recognition system becomes a challenging task due to inconsistency of a writer, variability in writing styles of different writers, relatively large number of characters of the script, high interclass similarity, structural complexity and degradation of documents due to different reasons. In order to recognize handwritten Amharic character a novel method based on deep neural networks is used which has recently shown exceptional performance in various pattern recognition and machine learning applications, but has not been endeavoured for Ethiopic script. The Convolutional neural network model is evaluated for its performance using our database that contains 132,500 datasets of handwritten Amharic characters. Common handwritten recognition systems using machine learning use a combination of both feature extractors and classifiers. Currently the use of deep learning techniques shows promising improvements for machine learning based classification tasks. Our proposed CNN model gives an accuracy of 91.83% on training data and 90.47% on validation data.

Keywords: Amharic, Handwritten, Character, Convolutional neural network, Recognition

1. INTRODUCTION

In recent years, there is much interest in the area of handwritten documents recognition. Compared with printed documents handwritten recognition task is very challenging. Different writers have different writing styles which changes the shape and size of the written characters. The similarities in distinct character shapes, the overlaps, and the cursive characters further complicate the problem. Recognition includes various fields of study such as, face recognition, object recognition, pattern recognition, character recognition, numerals recognition, etc. Handwritten Character Recognition System is an intelligent system able to classify handwritten Characters as human see.

There have been different methods that are used for offline handwritten document recognition [1]. In the conventional methods with features engineered manually and using different classification algorithms to classify the characters based on the extracted features. On the other hand, deep learning algorithms such as convolutional neural networks are able to do the feature extraction by themselves from the raw images of the handwritten document and classify characters on those features learned. Deep learning methods show better performance in different researchers work for handwritten recognition task. Among the deep learning methods convolutional neural networks, which is the one proposed for this research work, are the most commonly used algorithms. Due to large number of hidden layer interconnections the number of trainable parameters is large in deep neural networks. The deep neural network needs large amount of training data sets to avoid the problem of overfitting. Compared with other deep learning algorithms CNNs have comparatively less number of parameters and they are easier to train [2]. CNN is a special type of feed forward neural network extracting features directly from the input image or sound data and trained with back-propagation algorithm. CNN have the ability to learn from feature maps of the input data [3]. The advantage of CNN is that it automatically extracts using its convolution operation and other techniques such as max pooling capable of invariant to shift in certain degree and distortion in shape of the input data [4]. Another major advantage of CNN is the use of shared weight in convolutional layers which enables reduces number of trainable parameters and improves performance [5].

Recently, Convolutional Neural Network (CNN) becomes state of the art algorithm for recognition of handwritten characters' problems since it can learn visual patterns directly from the pixel images with minimal preprocessing [6]. A CNN based model for handwritten recognition task was tested on UNIPEN English character dataset and achieves a recognition rates of 93.7% and 90.2% for lowercase and uppercase characters, respectively [7 and 8]. Amharic language is an official language of the federal government of the Federal Democratic Republic of Ethiopia. The language has been used as official language of the county for several years. It belongs to Afro-Asiatic language family. It is the second most widely spoken Semitic language in the world, next to Arabic. It has its own alphabet which is significantly different from other alphabets such as Latin alphabet. Although the Ethiopic alphabet called Fidel has recently been standardized to have 435 characters. However, the most commonly Ethiopic script used by Amharic has 265 characters including 27 labialized characters (characters representing two sounds) ሏ ሟ ሯ ሷ ሿ ቊ ቌ ቧ ቷ ቿ ኋ ኗ ኟ ኲ ኵ ዟ ዧ ዷ ዿ ጓ ጧ ጯ ጿ ፏ ጐ ጕ ኧ and 34 base characters with six orders representing derived vocal sounds of the base character, 21 symbols for numerals and 8 punctuation marks. There is no capital and lower case distinction. When we see the features of Amharic characters they have the following basic characteristics:

Each symbol is written according to the sound it has when pronounced. Vowels are created by modifying the base characters in some form. The symbols are written in disconnected manner e.g ሀ, ሁ, ሂ, ሃ. The direction of writing the script is from left to right and top to bottom sequence. There is a proportional space between characters and words. There is no capital and lower case distinction as it is for the Latin characters. The Amharic language alphabet is conveniently written in a

tabular form of seven columns as shown in Table 1 where each column corresponds to vocal sounds in the order of ä, u, i, a, e, ə, and o [9].

ä order	u order	i order	a order	e order	ə order	o order
ሀ hä	ሁ hu	ሂ hi	ሃ ha	ሄ he	ህ hə	ሆ ho
ለ lä	ሉ lu	ሊ li	ላ la	ሌ le	ል lə	ሎ lo
ሐ hä	ሑ hu	ሒ hi	ሓ ha	ሔ he	ሕ hə	ሖ ho
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Figure 1. Part of the Amharic Alphabet

Several handwritten scriptures and documents written in this language are available on paper or on any other material. Converting the handwritten documents into digital forms helps us to process, share and store them in electronic form. The conventional way of converting the handwritten Amharic documents in to an electronic form, which is done by typing on the keyboard, is very time consuming, error prone and tedious. Due to the keyboard layout for Amharic characters which takes an average of two keystrokes to write one Amharic character, the conventional way of converting handwritten Amharic documents will be very difficult. This emphasizes the need for an automatic handwritten Amharic character recognition system which converts handwritten texts into machine-readable code that can be accepted by a computer for further processing.

Amharic handwriting recognition is challenging due to mainly two reasons. First, it has huge number of symbols compared to that of the alphabet system. Second, most characters are very similar in shape. This is because of the minimal modification performed to get order of a character. For instance, ብ comes from በ, ሷ comes from ሰ, and ሁ comes from ሀ. Most core characters also show similarities to one another. One common feature is a mark of palatalization which sets off palatal ሸ from ሰ, ቸ from ተ, ጀ from ደ, ኝ from ነ and so on. The interclass variability is even minimal in case of handwritings where mostly these modifications are forgotten or placed at a wrong position. Lack of standard way of writing aggravates the problem by increasing the intra-class variability. Nowadays, it is becoming increasingly important to have information in digital format for increased efficiency in data storage, retrieval and to share information to users. Although a lot of work and research has been done for handwritten character recognition for other languages like English and Asian languages such as Japanese, Chinese and Korean, there is only a few research attempts at Amharic language.

A few works have been reported in scientific literature related to the recognition of Amharic printed and handwritten document recognition. In they develop a recognition system for Ethiopic script using direction field tensor mechanism [9]. They extract features manually and their spatial relationship. Since there is no standard database for Ethiopic text they use thirty pages scanned image from newspaper, books and clean printouts. The achieved performance for their system is 87%. They did not consider handwritten documents in their dataset. In they have used ANN approach for recognition of real life documents [10]. They collected their dataset from 'Addis Zemen' newspaper, Amharic Bible, 'Federal NegaritGazeta' newspaper and the fiction 'FikerEskeMekabir'. The generalized error rate is high and they achieve only 11.40% accuracy on

new test sets which is not satisfactory and the proposed system is trained with printed documents rather than handwritten. In they develop a system which uses a principal component and linear discriminant analysis followed by a decision directed acyclic graph based support vector machine based classifier [11]. Existing methods including those discussed above for Amharic document recognition systems, employ manually designed feature extractor and learned classifier and most of them use printed documents rather than handwritten. It is not easy to design an optimal feature extractor for a particular application. Hence, the performance of these algorithms is not satisfactory.

For languages other than Amharic there are a lot of works done with high performance rate of recognitions. In they propose shared-hidden-layer deep convolutional neural network (SHL-CNN) for image character recognition [12]. The SHL-CNN model developed in this paper used to extract common features existed in different languages at its hidden and input layers. In the final softmax layer the SHL-CNN model learns features only from characters of the destination language. The performance of the SHL-CNN model is verified on both English and Chinese image character datasets, showing that the SHL-CNN can reduce recognition errors by 16-30% relatively compared with models trained by characters of only one language using conventional CNN, and by 35.7% relatively compared with state-of-the-art methods. A modified LeNet-5 which is one of common CNN model with special settings of the number of neurons in each layer and the connecting way between some layers is proposed by for offline handwritten English character recognition [8]. They used the UNIPEN dataset in their experiments and they achieve an accuracy of 93.7% and 90.2% for uppercase and lowercase English characters respectively. Authors in proposed handwritten recognition method for Chinese character based on relaxation convolutional neural network (R-CNN) and alternately trained relaxation convolutional neural network [13]. The relaxation convolutional layer in their model, unlike the traditional convolution layer, does not require neurons within a feature map to share the same convolutional kernel, endowing the neural network with more expressive power. Authors in applied multi-pooling and data augmentation with non-linear transformation to a convolutional neural network (CNN) for multi-font printed Chinese character recognition (PCCR) [14]. They propose a multi-pooling layer on top of the final convolutional layer; this approach is found to be robust to spatial layout variations and deformations in multi-font printed Chinese characters. Outstanding recognition rate of 94.38% is achieved by combining the multi-pooling and data augmentation techniques and 99.74% by applying the multi-pooling and data augmentation techniques with non-linear transformation jointly. In the authors proposed new techniques which can improve online Chinese handwritten character recognition [15].

They include techniques such as deformation, non-linear normalization, imaginary strokes, path signature and 8-directional features. The contribution in this work is twofold. First the integration of the new techniques added with the deep CNN enhances the performance of the recognition system. Second, the developed deep convolutional neural networks with diversity in their domain knowledge are combined using a hybrid serial-parallel strategy. A promising accuracy of 97.20% and 96.87% are achieved using CASIA-OLHWDB1.0 and CASIA-OLHWDB1.1 dataset for Chinese character respectively. In an effective method to analyze the recognition rate of handwritten Chinese character based on softmax

regression score of a high performance convolutional network is studied [16]. In authors proposed a deeper architecture of CNN algorithm by using streamlined version of GoogLeNet [17]. They used the ICDAR 2013 offline Chinese character recognition system competition dataset. With incorporation of traditional directional feature maps the proposed GoogLeNet models achieve an accuracy of 96.35% and 96.74% as single and ensemble models respectively. A handwritten Hangul character recognition system using deep convolutional neural network by proposing several novel techniques to increase the performance and training speed of the networks is done by [18]. In Malayalam handwritten character recognition using the convolutional neural network is developed and in their work they discussed the CNN is better than the conventional handcrafted feature extractor based systems [19]. Deep learning based large scale handwritten Devanagari character recognition is proposed by with focus on the use of dropout and dataset increment approach to increase the test accuracy of their deep learning model [20]. A combination of four different pattern analysis techniques is used to develop a powerful and efficient system for handwritten Telgu character recognition system is proposed by [21]. Their system embodies convolutional neural networks, principal component analysis, support vector machines and multi-classifier systems.

As compared to the handwritten automated character recognition system discussed above the Amharic character recognition system is the least studied subject both in the conventional handcrafted feature extractor based systems as well as deep learning based convolutional neural networks. In the proposed method, both the feature extraction and classification tasks are done through learning from labelled data. This method overcomes the problems faced by the existing methods. Visual recognition system using convolutional neural networks have shown a significant improvement in recent years [22]. Record-breaking results have been obtained using these methods. This has motivated the researcher to investigate the success of the CNN algorithms on this challenging problem. Visual recognition using convolutional neural networks enables us to train the complete system from end to end.

2. RESEARCH SIGNIFICANCE
The automatic handwritten Amharic character recognition system can be used in the following applications.
- To digitize handwritten documents for analysis and archival purposes
- Postal mail sorting according to destination address
- Bank cheque processing
- Bill processing
- Keying data to the computer (search data in a computer)
- Act as a pattern recognition challenge for students and researchers in Ethiopia

Introducing this novel paradigm to solve this challenging problem will have a significant impact in terms of technology transfer. The outcome of this research has a possibility to inspire other researchers to try different algorithms to solve these challenging problems which can improve the quality of our day to day life. The research will show the role of the modern artificial intelligence in solving challenging visual recognition tasks.

3. AMHARIC HANDWRITTEN CHARACTER RECOGNITION SYSTEM
### 3.1. General Architecture

In our proposed system we use convolutional neural networks for recognition of the handwritten Amharic characters. There are two major units in the system: the first one is preprocessing of collected data and segmentation of characters and the second one is extracting learnable features from character images and using those features for classification. Digitalization, noise removal, binarization, normalization and others belongs to the preprocessing step in the proposed system. In the segmentation step different segmentation methods are used such as line segmentation, word segmentation and character segmentation from the given scanned image of handwritten document to extract individual characters. The two essential components in recognition, feature extraction and classification, both are done in our CNN model.

- **Pre-processing Techniques:** Preprocessing is necessary for efficient recovery of the text information from scanned image. The processing algorithms employed on the scanned image depend on paper quality, resolution of the scanned image, the amount of skew in the image, the format and layout of the images and text and so on. For the proposed system digitization, noise removal, binarization, and normalization techniques are used at preprocessing stage.

- **Digitization:** For any offline handwritten recognition system the first task after collecting the handwritten document will be converting it in to electronic form. The data was collected from handwritings of persons who differ based on their age, education level and left handed or right handed individuals. The collected data on the prepared form was scanned using 332 dpi mobile phone scanner applications and prepare it for further pre-processing.

- **Noise Removal:** Sample scanned handwritten documents collected from different persons handwriting may contain noise that mainly arises due to, scanner quality, paper quality and quality issues when the person writes on white papers. The RGB image from the scanner is converted into gray image to reduce computation cost. In most character recognition tasks gray scale images are used to reduce the computational cost compared with RGB images. RGB images may have non uniform background or watermarks which makes it difficult to extract features without some kind of preprocessing. Figure 2 shows the original or raw image and the output of different preprocessing steps.

(a)   (b)
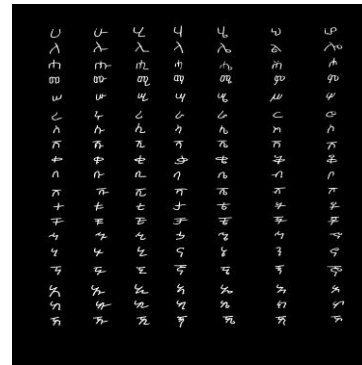
(c)   (d)

Figure 2. a) Original image b) Gray scale image
c) After noise removal d) Binary image

3.2. Architecture of Proposed System

An image with dimension of MxMxC where M is the height and width is given as input to the convolution layer of convolutional neural networks. The C stands for channel which 1 for gray scale and 3 for RGB images. A CNN consists of a number of layers (convolutional layers, pooling layers, fully connected layers). The convolutional layer will have K filters (kernels) of NxNxR where N is height and width of filter (kernels) and R is the channel. The convolution operation is done by convolving the filters (kernels) with the image giving feature maps as an output of the convolution operation. The size of the output feature map will be M−N+1. The next layer after the convolution layer is the max or average pooling operation which will decrease the size of the input feature maps based on the pooling size. Any number of fully connected layers will be added after the convolution and pooling layers of the network. The salient features from the input image are extracted during the convolution operation. Then, the convolutional layer applies its activation function to add non linearity to its output. We use RELU as an activation function to the output of every convolution layer. The ReLU [23 and 24] increases the nonlinear properties of the decision function and of the overall network without affecting the receptive fields of the convolution layer.

The pooling operation after each convolution layer takes a small window and subsamples it using either by taking the maximum value from the window or the average pixel values in the window producing a single output. The pooling layer makes the CNN capable to invariance to

distortion. In our proposed system we use a max pooling layer with 2x2 pixel window. A fully connected layer takes all neurons in the previous layer and connects it to every single neuron it has. Fully connected layers are not spatially located anymore (you can visualize them as one-dimensional), so there can be no convolutional layers after a fully connected layer. Figure 3 shown the proposed CNN architecture for Amharic handwritten character recognition that describe as follow: INPUT ---->CONV---->RELU------ >CONV --->RELU ----> MAX POOL ----> CONV ----> RELU ----> MAX POOL ----> CONV ----> RELU ---> MAX POOL ---> CONV ----> RELU ---> MAX POOL ---> FC ---> RELU ----> FC----- > Softmax.
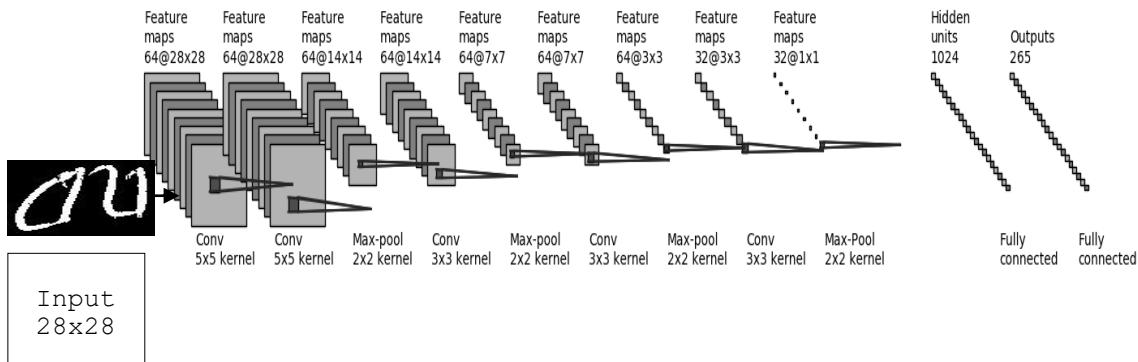

Figure 3. The proposed CNN architecture for Amharic handwritten character recognition

The first and the second convolution layers have 64 filters of size 5x5, the next two conv layers have 64 filters with size 3x3 and the last conv layer has 32 filters with kernel size of 3x3. The second important layer in convolutional neural networks is the pooling layer. This layer simply acts as a downsampling filter. After the second conv layer a max-pooling with 2x2 pixel window is used. After this polling layer we have 3 conv---- > max pool layers with 3x3 kernel size for the convolution layer and 2x2 pixel window for the pooling layer. The size of the convolution filters is initialized based on experimental results. Experimentally select filter sizes for each conv layer by seeing the performance improvement of the network during training and validation. After each convolution layer the RELU activation function is used. The rectifier activation function is used to add non linearity to the network. Figure 4 shows the code snippet for the layers of the CNN for the proposed model.
The Flatten layer is use to convert the final feature map, which is an output from the last max pooling layer, into a one single 1D vector. This flattening step is needed so that you can make use of fully connected layers after some convolutional/maxpool layers.
In the end the features in two fully-connected (Dense) layers which is just artificial neural networks (ANN) classifier is used. In the last fully connected layer (Dense (265, activation="softmax")) the net outputs distribution of probability of each class. Dropout is a regularization method, where a proportion of nodes in the layer are randomly ignored (setting their weights to zero) for each training sample. This drops randomly a proportion of the network and forces the network to learn

features in a distributed way. This technique also improves generalization and reduces the problem of overfitting. A dropout of 0.5 at the last fully connected layer is used. The RMSprop optimizer to update filter weights and improve algorithm functionality is used. The default setting of this optimizer since it is recommended to use the default setting in the documentation is used. Batch size of 70 and run the network for 70 epochs is used. The batch size and the number of epochs are selected based on the best performance measurement values of the proposed CNN network during experimental set up. Python programming language with keras deep learning library with theano backend on CPU was used during the development of the convolutional neural network architecture. Image processing libraries designed for python such as numpy, opencv2, skyimage used for processing the input image for the conv net.

```
1  model = Sequential()
2
3  model.add(Convolution2D(64, (5,5),border_mode='same',input_shape=(input_shape)))
4  model.add(Activation('relu'))
5
6  model.add(Convolution2D(64, (5, 5),border_mode='same'))
7  model.add(Activation('relu'))
8  model.add(MaxPooling2D(pool_size=(2, 2)))
9
10 model.add(Convolution2D(64, (3, 3),border_mode='same'))
11 model.add(Activation('relu'))
12 model.add(MaxPooling2D(pool_size=(2, 2)))
13
14 model.add(Convolution2D(64, (3, 3),border_mode='same'))
15 model.add(Activation('relu'))
16 model.add(MaxPooling2D(pool_size=(2, 2)))
17
18 model.add(Convolution2D(32, (3, 3),border_mode='same'))
19 model.add(Activation('relu'))
20 model.add(MaxPooling2D(pool_size=(2, 2)))
21
22 model.add(Flatten())
23 model.add(Dense(1024))
24 model.add(Activation('relu'))
25 model.add(Dropout(0.5))
26 model.add(Dense(num_classes))
27 model.add(Activation('softmax'))
```

Figure 4. Code snippet for the proposed CNN model

3.3. Data Collection and Preparation

In research like character recognition for handwritten documents using deep neural networks, the primary task is collecting required data and preparing it for further processing. The Amharic character benchmark dataset is still not available for the research community in public and this was the major challenge during this research work. Since the research is dealing with pattern recognition, or more specifically character recognition, the data collected are of two types. The first being data for training the convolutional neural network (recognition

engine), the second will be for testing the performance of the CNN model. The Amharic Handwritten Character dataset used in our system is created by collecting the variety of handwritten Amharic characters from hand writings of different individuals from diverse fields. 500 dataset for each character were collected and having (500x265) 132500 datasets in total. The data was collected from 250 persons who write each character 2 times in a white paper. The selected individuals can write the Amharic alphabet randomly with different educational background and different age ranges. Each individual wrote each character on the forms shown in figure 5.

We divided this dataset in two groups: 20 percent for validation/testing and the remaining for training our system. Handwritten documents are then scanned with 332 dpi Samsung galaxy note 4 CamScanner mobile application and cropped for individual characters with a size of 28x28 pixels and pre-processed before feeding it to the CNN using semi-automated algorithm. The collected data was labelled based on the 265 classes for each character by using integer values {0 to 264} for each class of the character. The disadvantage of convolutional neural network is its rapacious appetite for labelled training data. Hence, real-world data collection and applying image pre-processing methods are needed to make the system more robust and efficient. A large handwritten character dataset has been collected in our work and labelled to train as well as to evaluate the system performance compared with other researcher works for Amharic character recognition system.
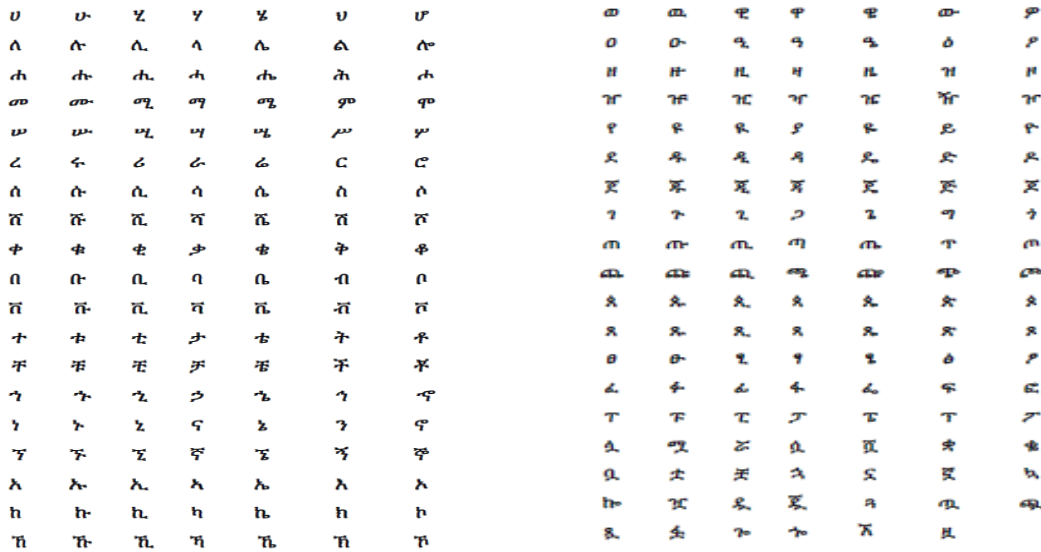
    (a) Form A   (b) Form B
Figure 5. Data collection forms for Amharic characters

3.4. Challenges in Amharic Character Recognition
The challenges in handwritten character recognition vary among different languages due to distinct shape, strokes and numbers of characters. In Amharic script there are characters which have similar structure which differs with each other with a little curve, line and strokes and this becomes even more challenging since such structural variations are forgotten during handwriting. Some examples of two different characters written similarly are shown in figure 6. The other

big challenge is unavailability of public datasets to use for such recognition systems which needs large amount of data to be trained.
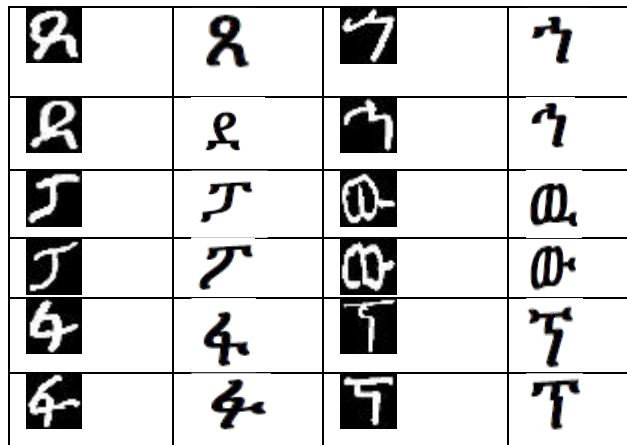

Figure 6. Different characters written similarly

4. FINDINGS AND DISCUSSION

In this section, the performance of the CNN architecture was investigated for training and validating Amharic character recognition system. ConvNets have a large set of hyper-parameters and finding the perfect configuration for your problem domain is a challenge. Different configurations of the proposed network were explored and attempted to optimize the parameters based on the validation and training set accuracy. The performance of our recognition system was evaluated by using the collected dataset discussed earlier. From the collected dataset 20% (26500) for validation and 80% (106000) for training the proposed convolutional neural network architecture discussed earlier was used. The training and validation datasets are evenly distributed over the underlying 265 classes. In experimental set up data augmentation was used to increase the number of training dataset to avoid overfitting. The performance of our proposed system was checked with and without data augmentation and training with data augmentation shows better performance result.

4.1. Data Augmentation

Data augmentation is increasing the training dataset artificially to enhance the accuracy of the model. It also helps to avoid overfitting problem due to less number of training datasets. The idea is to alter the training data with small transformations to reproduce the variations occurring when someone is writing a character, digit or punctuation marks. Data augmentation techniques change the array data of image pixels keeping the labels unchanged. Techniques such as grayscales, horizontal flips, vertical flips, random crops, color jitters, translations and rotations are used during transformation of our training data.
For the data augmentation, the following features are chosen:

- Randomly rotate some training images by 10 degrees
- Randomly zoom by 10% some training images
- Randomly shift images horizontally by 10% of the width
- Randomly shift images vertically by 10% of the height vertical_flip nor horizontal_flip is not applied since it could have lead to

misclassify symmetrical characters. The accuracy of the model is different when trained with and without data augmentation.

In all the experimental results illustrated below we use the data augmentation technique. For the data augmentation purpose I have used the keras Image Datagenerator object by giving the above parameters to transform the training dataset on the fly. On each epoch running the data generator function generates randomly transformed new training character images which are equal to the raw training images in size. The keras augmentation does not add more training data but actually it simply perturbs each and every dataset with some transformation in each epoch of training time. Thus, this will help the model to generalize and avoid overfitting problem.

4.2. Visualizing Losses and Accuracy

By changing different parameter values such as epoch size, batch size, optimizer selection, dropout and the layers of the network until the best fit model is found. In this section the results of experiments are presented. Based on experimental analysis, the CNN architecture discussed earlier is used for the proposed Amharic handwritten character recognition system. We perform different conv, max pool and FC layer configuration and observe the training and validation accuracy for a given CNN layer configuration to select the best depth of the proposed CNN. We selected the CNN layer configuration with the best performance result for our training and validation dataset. The accuracy of the CNN model with layer other than the proposed are less accurate as experimentally evaluated. Then we improve the performance of the selected CNN architecture by tuning other parameters of the network. The other network parameter we change it during our experimental analysis was the batch size selection. Table 1 illustrates an experiment conducted to observe the effect of batch size on the performance of the proposed CNN model with varying the batch size and using RMSprop optimizer with learning rate of 0.001. Using this experiment, a batch size of 70 selected which gives better accuracy. For the other experiments we use 70 for batch size and tune other parameters to get improved performance.

Table 1. Accuracy with varying batch size using RMSprop optimizer and 20 epochs

Batch size	Training acc.	Validation acc.	Training loss	Validation loss
60	86.7%	88.7%	0.45	0.4
65	87.2%	88.8%	0.42	0.4
70	87.4%	89.4%	0.42	0.38
75	86%	89%	0.46	0.4
80	86%	88%	0.46	0.42
86	86%	88.5%	0.45	0.41
90	86%	88.1%	0.47	0.42

The other parameter which has an effect on the performance of our proposed model was the optimizer selection. Optimizers are used for weight update of the network and they have their own behaviors. For our proposed model the RMSprop optimizer gives better results compared with stochastic gradient decent and Adam optimizer. Table 2 illustrates the effect of optimizers with different learning rates for the proposed model.

82

Table 2. Effect of optimizer selection on accuracy with batch size of 70 and 20 epochs

Optimizer	Training acc.	Validation acc.	Training loss	Validation loss
RMSprop	87.4%	89.4%	0.42	0.38
SGD with LR=0.001	67.8%	48%	1.9	1.16
SGD with LR=0.005	80.7%	86.3%	0.63	0.46
SGD with LR=0.01	82.8%	86.3%	0.57	0.47
SGD with LR=0.05	77.8%	84%	0.76	0.56
SGD with LR=0.1	33.5%	45.5%	2.64	2.1

Increasing the epoch size from 20 to 30 we get better results with 89.99% for training accuracy and 90.19% for validation accuracy. The training loss reduces from 3.80 to 0.32 and the validation loss reduces from 1.96 to 0.36. Figure 7 shows the graphs for accuracy and loss curves with 30 epochs.
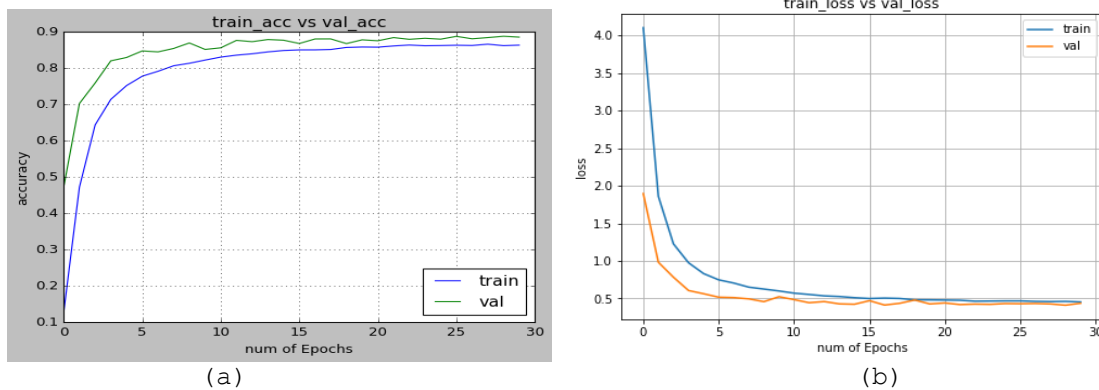

Figure 7. (a) Validation vs training accuracy (b) Validation vs training loss

The graphs in figure 8 show the accuracy and loss of training and validation with 40 epochs. The training accuracy is 90.40% and validation accuracy of 90.37%. The training loss decreases from 3.80 to 0.31 and validation loss decreases from 1.96 to 0.35
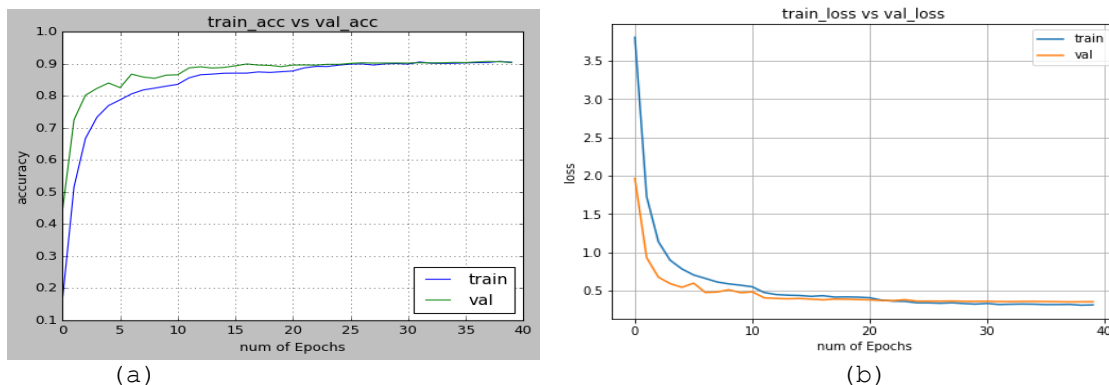

Figure 8. (a) Validation vs training accuracy (b) Validation vs training loss

The graphs in figure 9 show the accuracy and loss of training and validation respectively with 50 epochs. We achieved a training accuracy

of 90.74% and validation accuracy of 90% and the loss for training dataset decreases from 3.90 to 0.30 and the loss of validation decreases from 1.66 to 0.36.
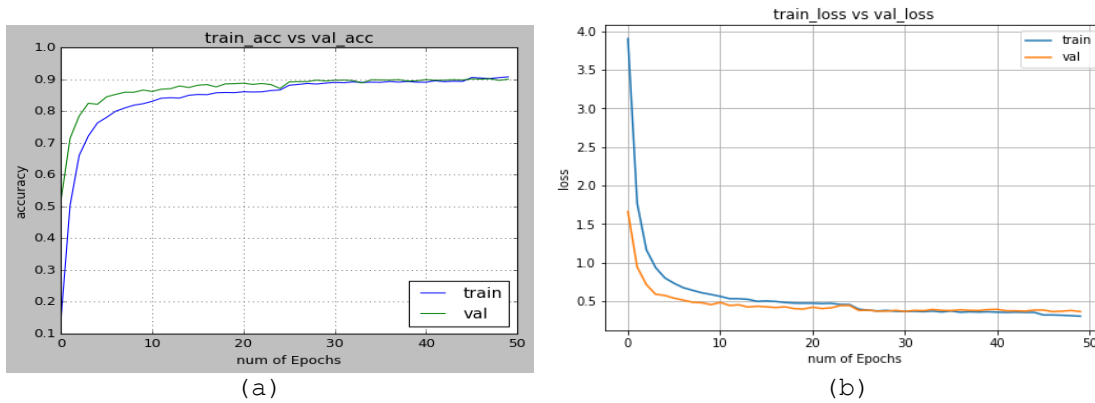

(a)   (b)

Figure 9. (a) Validation vs training accuracy (b) Validation vs training loss

With 60 epochs we achieved better accuracy of 91.43% for training and 90.39% for validation. The training loss decreases from 3.91 to 0.28 and the validation loss decreases from 1.79 to 0.37.
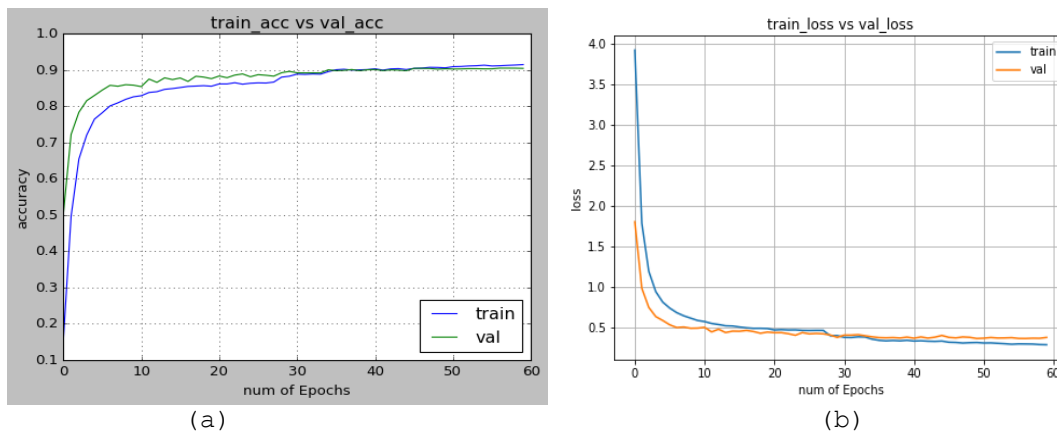

(a)   (b)

Figure 10. (a) Validation vs training accuracy (b) Validation vs training loss

Increasing the epoch size to 70 increases the training accuracy to 91.83% and the validation accuracy to 90.47%. We have get a little increase in the validation accuracy which means still the CNN model is improving its performance on test sets. The training loss decreases from 3.87 to 0.27 and the validation loss decreases from 1.87 to 0.36. Increasing the epoch size above 70 increases the processing time of the system. Due to this we take 70 as epoch size for our final model. Therefore, we are using 70 both for epoch and batch size respectively.
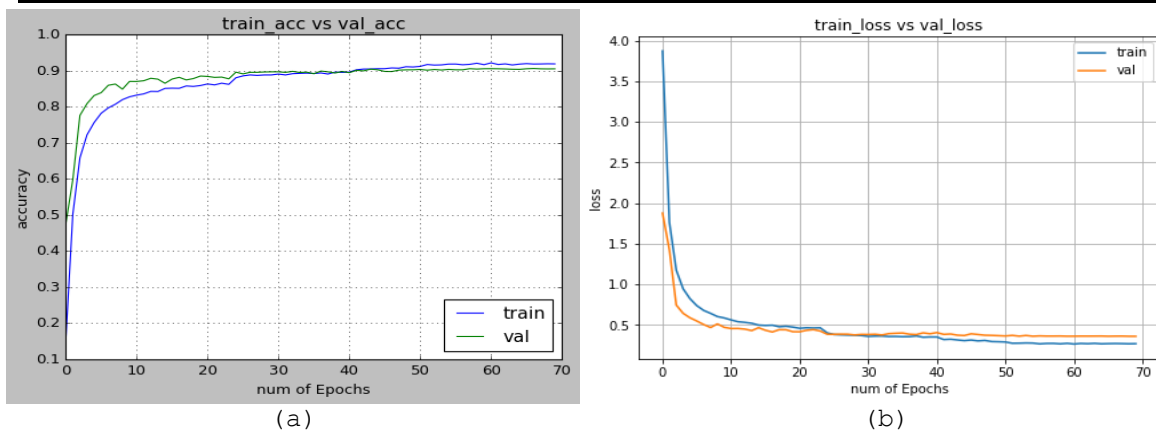
Figure 11. (a) Validation vs training accuracy (b) Validation vs training loss

5. CONCLUSION AND RECOMMENDATION

In this research we use convolutional neural networks for recognition of Amharic handwritten characters with 265 character classes. It is well known that convolutional neural networks are the current state of the art algorithm for classifying image data. We have collected large amount of Amharic characters from individual handwriting on prepared form for data collection. The collected handwritten document is scanned and pre-processed to get a 28 x 28 .jpg character images which are given as input to the convolutional neural network for classification. We presented a new public dataset for Amharic character dataset which is publicly available for any researcher and access it through contact address of the author. We developed a deep learning based Amharic handwritten character recognition system. To get the best fit model of CNN based architecture a lot of trial and error network configuration tuning mechanism has been used. Using the developed CNN model we have achieved better recognition accuracy compared with other research works based on conventional hand craft feature extraction based methods. We have achieved an accuracy of 91.83% on training dataset and 90.47% on validation dataset. Even though there are a lot of work for recognition of handwritten characters for English, Chinese, Arabic and some other Indian languages only little work is done for Amharic language. The previous research works did not use deep learning methods for Amharic character recognition. Due to lack of research works on the area there is a big challenge to get dataset for Amharic language. In this research we develop a dataset which can be used by other researchers in the future. By increasing the size of the collected dataset and proposing new deep learning algorithms we will increase the performance of the system. In the future, we want to extend this algorithm for recognition of Amharic words by creating new datasets.

REFERENCES

[1] Sarkhel, R., Das, N., Saha, A.K., and Nasipuri, M., (2016). A Multi-objective Approach Towards Cost Effective Isolated handwritten Bangla character and Digit Recognition, Pattern Recognition, 58:172-189.

[2] Liang, Y., Wang, J., Zhou, S., Gong, Y., and Zheng, N., (2016). Incorporating Image Priors with Deep Convolutional Neural Networks for Image Superresolution, Neurocomputing, 194:340-347.

[3] Maitra, D.S., Bhattacharya, U., and Parui, S.K., (2015). CNN based common approach to handwritten character recognition of multiple scripts, Document Analysis and Recognition (ICDAR), 2015 13th International Conference on, pp:1021-1025.

[4] Shin, H.C., Roth, H.R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., and Summers, R.M., (2016). Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning, IEEE Transactions on Medical Imaging, 35, pp:1285-1298.

[5] Bai, J., Chen, Z., Feng, B., and Xu, B., (2014). Image character recognition using deep convolutional neural network learned from different languages, 2014 IEEE International Conference on Image Processing (ICIP), pp:2560-2564.

[6] Lecun, Y. and Bengio, Y., (1995). Pattern Recognition and Neural Networks, in Arbib, M.A. (Eds), The Handbook of Brain Theory and Neural Networks, MIT Press 1995.

[7] Guyon, I., Schomaker, L., Plamondon, R., Liberman, M., and Janet, S., (1994). Unipen Project of On-Line Data Exchange and Recognizer Benchmarks, in proc. of 12th International. Conference on Pattern Recognition (ICPR), vol:2, pp:29-33, IEEE.

[8] Yuan, A., Bai, G., Jiao, L., and Liu, Y., (2012). Offline Handwritten English Character Recognition Based On Convolutional Neural Network, in 10th IAPR International Workshop on Document Analysis Systems (DAS), pp. 125-129, doi: 10.1109/DAS.2012.61.

[9] Assabie, Y. and Bigun, J., (2006). Ethiopic Character Recognition Using Direction Field Tensor, in proc.int.conf. on pattern recognition, ICPR, pp:284-287, Hong Kong.

[10] Teshager, B.A., and Sethuraman, R., (2015). Artificial Neural Network Approach to the Development of OCR for Real Life Amharic Documents. International Journal of Science, Engineering and Technology Research (IJSETR), Volume:4, Issue:1, pp:2278-7798.

[11] Million, M. and Jawahar, C.V., (2007). Optical character recognition of Amharic documents. African Journal of Information & Communication Technology 3(2):14.

[12] Bai, J., Chen, Z., Feng, B., and Xu, B., (2014). Image Character Recognition Using Deep Convolutional Neural Network Learned from Different Languages. 2014 IEEE International Conference on Image Processing (ICIP), pp:2560-2564.

[13] Wu, C., Fan, W., He, Y., Sun, J., and Naoi, S., (2014). Handwritten Character Recognition by Alternately Trained Relaxation Convolutional Neural Network, Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on, pp:291-296.

[14] Zhong, Z., Jin, L., and Feng, Z., (2015). Multi-font Printed Chinese Character Recognition Using Multipooling Convolutional Neural Network. Document Analysis and Recognition (ICDAR), 2015 13th International Conference on, pp:96-100.

[15] Yang, W., Jin, L., Xie, Z., and Feng, Z., (2015). Improved Deep Convolutional Neural Network for Online Handwritten Chinese Character Recognition Using Domain-Specific Knowledge. Proc. Document Analysis and Recognition (ICDAR), 2015 13th International Conference on, pp:551-555.

[16] He, M., Zhang, S., Mao, H., and Jin, L., (2015). Recognition
     Confidence Analysis of Handwritten Chinese Character with CNN,
     Proc. Document Analysis and Recognition (ICDAR), 2015 13th
     International Conference on, pp:61-65.

[17] Zhong, Z., Jin, L., and Xie, Z., (2015). High performance offline
     handwritten Chinese character recognition using GoogLeNet and
     directional feature maps, Proc. Document Analysis and Recognition
     (ICDAR), 2015 13th International Conference on, pp. 846-850.

[18] Kim, I.J. and Xie, X., (2015). Handwritten Hangul Recognition Using
     Deep Convolutional Neural Networks, International Journal on
     Document Analysis and Recognition (IJDAR), 18, pp:1-3.

[19] Anil, R., Manjusha, K., Kumar, S.S., and Soman, K.P., (2015).
     Convolutional Neural Networks for the Recognition of Malayalam
     Characters, Proc. Document Analysis and Recognition (ICDAR), 2015
     13th International Conference on, pp:1041-1045.

[20] Acharya, S., Pant, A.K., and Gyawali, P.K., (2015). Deep Learning
     Based Large Scale Handwritten Devanagari Character Recognition,
     2015 9th International Conference on Software, Knowledge,
     Information Management and Applications (SKIMA), pp:1-6.

[21] Soman, S.T., Nandigam, A., and Chakravarthy, V.S., (2013). An
     Efficient Multiclassifier System Based on Convolutional Neural
     Network for Offline Handwritten Telugu Character Recognition, Proc.
     Communications (NCC), 2013 National Conference on, pp:1-5.

[22] Yann, L., et al., (1998). Gradient-based Learning Applied to
     Document Recognition. Proceedings of the IEEE 86.11, pp:2278-2324.

[23] El-Sawy, A., Loey, M., and El-Bakry, H., (2017). Arabic Handwritten
     Characters Recognition Using Convolutional Neural Network, WSEAS
     Transactions on Computer Research, vol:5, pp:11-19.

[24] Nair, V. and Hinton, G.E., (2010). Rectified Linear Units Improve
     Restricted Boltzmann Machines, Machine Learning -International
     Workshop Then Conference, pp:807-814.