A New Hybrid CNN-SVM Model for Amharic Character Image Recognition

Muluken Zemed Tsegaye¹, Prof. Mogalla Shashi²

¹M.Tech. Department of Information Technology, Andhra University, Visakhapatnam-530003, AP, India
²Professor, Department of Computer Science, Andhra University, Visakhapatnam-530003, AP, India

Abstract: Optical Character Recognition is a method of converting scanned images of printed or handwritten documents into ASCII format or machine-encode text, making it easier to store, browse, retrieve and process electronic data online. In this work, we have prepared a printed Amharic characters dataset to train and test the proposed model. The dataset contains only 231 basic Amharic characters. The images in the dataset have been sized normalized to 32×32 pixels. Furthermore, we present the designed recognition system stages: image acquisition, pre-processing, segmentation, feature extraction, and classification stages. In this study, we present a hybrid model of two super classifiers: Convolutional Neural Network (CNN) and Support Vector Machine (SVM). In this hybrid model, CNN as a feature extractor automatically from the raw images, and then the extracted feature vectors are given as input to SVM for classification and recognition. Experiments have been conducted on own prepared dataset and benchmark Amharic Optical Character Recognition Database (ADOCR), synthetically generated with different degradation levels. From the experiment results, 99.84% test accuracy was achieved on the own prepared dataset and 95.59% test accuracy on the benchmark Amharic Optical Character Recognition Database (ADOCR). Thus, the proposed model outperforms previously existing works attempted by others to recognize printed Amharic characters on the same and different datasets. Finally, the performance of the CNN classifier and the proposed CNN-SVM model was compared in this research.

Keywords: Hybrid CNN-SVM model, Amharic Character Image, Convolution Neural Network (CNN), Amharic OCR, Support Vector Machine (SVM)

1) Introduction

Optical Character Recognition (OCR) has a relatively long history. It has been a long-held ambition to replicate human functions in machines and have them do simple activities like reading documents. Character recognition can be traced back to 1870, when C.R. Carey of Boston, Massachusetts invented the retina scanner, which was an image transmission system based on a mosaic of photocells[2,4]. OCR is a method of converting scanned images of printed or handwritten documents into ASCII format or machine-encode text, making it easier to store, browse, retrieve and process electronic data online. OCR has different applications [3, 8] in banking, institutional repositories, digital library, and health care industry, legal industries, document reader systems for the visually impaired, and much more.

Over the last few decades, many researchers have attempted to apply and implement OCR systems for both typed and handwritten documents in Latin and non-Latin scripts. As a result, OCR system are able to read documents written in a number of languages, including English, Japanese, Chinese, Hindu, Arabic and others. However, the OCR system for indigenous language like Amharic was relatively unexplored and a challenging task.

Amharic is one of indigenous language that has its own scripts and writing system, with nearly 100 million speakers worldwide [1]. It belongs to Afro-Asiatic Southern Semitic languages spoken in Ethiopia [1]. Until 2020, Amharic was the only official and working language of Ethiopia. Recently, four new working languages at the federal level added [11]. There are many historical documents written in this language available in churches, museums, libraries, and so on. These printed and written documents need to be converted to electronic format for preserving historical documents, to saving storage space, for simple searching and retrieval as users demand through internet. The visual similarities of shapes and the large number of characters coupled with a lack of well-organized dataset make Amharic characters recognition challenging [9, 10].

Over the last two decades, some researchers [6-8] employed tradition machine learning methods such as HMM, SVM and ANN. Recently, many works [12 -13], [15], [16] have been made for Amharic character recognition using convolutional neural network architecture (CNN) to improve recognition accuracy and performance. Berhiana [1], conducted a detailed research by employing deep learning by taken the advantage of deep learning techniques to recognize Amharic characters and text documents Moreover, He Prepared a synthetic generated Amharic Optical Character Recognition database (ADOCR) [18] and made it public to other researchers. Better recognition results was obtained.

Several works have been made based on CNN-SVM hybrid approach in many domains; handwritten digit recognition [18], Arabic handwritten recognition [19], offline handwritten Arabic character recognition [20], handwritten Arabic text recognition [21], handwritten Tamil character recognition [25], handwritten Chinese character recognition [27]. All authors were achieved good results by integrating the two super classifiers: CNN work as feature extractor and SVM work as classifier.

In feature extraction stage, the choice of features has a significant impact on the classification accuracy [32].there are two types of feature extraction methods: Handcrafted and Automatic (learned) feature extractions. Handcrafted features usually done manually, not robust and computational intensive due to high dimensions. On the other hand, Automatic (learned) feature extraction is the process of automatically extracting relevant features for classification using deep learning algorithm (here we have used CNN).

CNN has notable attributes such as hierarchical learning, automated feature extraction, multitasking, and weight sharing [40]. However, CNN requires a large training examples and vulnerable to over fitting problem. In contrast, SVM has a high generalization performance, which means it can classify correctly data that has never been seen before [10], [19-22]. Support Vector Machine (SVM)
is a supervised machine learning algorithm used for classification task based on statistical learning theory proposed by Vapnik [33] in 1998 and Cortes and Vapnik [34]. We focus our research on integrate the best characteristics of CNN and SVM classifiers. In this study, we have designed a new hybrid CNN-SVM model for recognizing Amharic characters by taking their notable attributes. In this model, a CNN trained using back-propagation to extract features automatically from the raw images, and then the extracted features are given as input to SVM to recognize unseen data.

The rest of the paper is organized as follows: section II describes reviews of the related works. CNN classifier, SVM classifier, hybrid CNN-SVM model, dataset preparation, parameter setting are presented in section III. Section IV describes experimental result and discussions. Conclusions are drawn in section V.

II) Literature Review

Many researches have been done intensively on character image recognition, with good results for a variety of Latin and non-Latin characters such as English [24], Tamil [25], Arabic [19-21], Chinese [22, 27], Bangla [23], and Devanagari [26]. However, the OCR for Amharic script was somewhat unexplored; it is an open research area and recognizing characters still a challenging task. Amharic uses writing system called Fidel (ፋ፩፲፴) [1, 5], which emerged from the Geez abugida ((ፏም፱፧) or Ethiopic, the liturgical language of the Ethiopian Orthodox church. In Amharic scripts, consists of about 317 different symbols of which 231 basic characters, seven special symbols (1^*7), 50 labialized, 20 numeric and 9 punctuation marks. Amharic script is written and read from left to right like English, from left to right in horizontal direction [1, 10].

Numerous researchers [6-10], have worked to address the problems in Amharic script recognition over the past two decades using traditional machine learning. In 1997, Worku [5] made the first study on Amharic OCR. He adopted a step-by-step segmentation algorithm and character recognition based on topological features. Dereje [6] conducted a research to recognize typewritten Amharic text in 1999. He had been implemented a mathematical morphology algorithm for salt-and-pepper noise, as well as a binary morphological filtering technique for subtractive and additive noise. Million M. and C.V. Jawahar [10], they employed a two-stage feature extraction principal component analysis and linear discriminant analysis obtain the most discriminating feature vector. The authors also noted that the use of SVM classifier is advantageous because of their efficiency and generalization capability.

Million M. [8] conducted a research on Amharic characters printed in various fonts and level of degradation to analysis the characteristics of characters. He noted that Amharic script recognition was difficult due to font variation, level of degradation, visual similarity of characters, and the large number of characters. Furthermore Yaregal [7] mentioned out that font variety is also a problem when developing an OCR system for Amharic scripts. In addition to this, he presented an HMM-based model for off-line recognition of handwritten Amharic words. However, all these researchers have been used traditional machine learning algorithm with limited private datasets to implement OCR for Amharic characters Furthermore, traditional machine learning algorithms need feature engineering, which is generally done manually and computationally intensive.

Recently, many works [12 –13, 15, 16] have been made for Amharic character recognition using convolutional neural network architecture (CNN) to improve recognition accuracy and performance. Berihanu et al. [13], proposed a simple convolutional neural network (CNN) based method for recognizing Amharic character images. The authors yield good result with an average recognition accuracy of 92.71%, on benchmark of synthetic Amharic Optical Character Recognition Database called ADOCR. Experiments were done on 80,000 Amharic character images generated with different level of degradation. In 2019, Mesay et al. [12] presented handwritten Amharic character recognition using a convolutional neural network. Moreover, Abeto [16] conducted a research for Character Recognition of Bilingual Amharic-English printed documents using convolutional neural network.

Berihanu et al. [15], they introduced Factored CNN architecture with two classifiers that recognize a character's row/consonant and column/vowel components. Before forking out at their final layers, the two classifiers share a common feature space. On a synthetically generated dataset, the method reaches state-of-the-art results. The proposed method achieves an overall character recognition accuracy of 94.97%. Experiments were done on 80,000 ADOCR database by removing 2006 distorted images from the database. They prepared a synthetic generated Amharic Optical Character Recognition database (ADOCR) [14] and made it public to other researchers.

Recently published work, Berihanu [1] conducted a detailed research by employing deep learning by taking the advantage of deep learning techniques such as sequence-to-sequence learning for Amharic text image recognition. He used CNN architectures, Attention mechanisms and RNNs for Amharic text image. Moreover, he develop a baseline Amharic database so as to address the issue of the dataset. To the best of our knowledge, no one has used hybrid CNN-SVM model for recognizing Amharic characters.

Several works have been made based on CNN-SVM hybrid approach in many domains; handwritten digit recognition [18], Arabic handwritten recognition [19], offline handwritten Arabic character recognition [31], handwritten Arabic text recognition [21], handwritten Tamil character recognition [25], handwritten Chinese character recognition [27]. All authors were achieved good results by integrating the two super classifiers: CNN work as feature extractor and SVM work as classifier. Furthermore, all works employed the same architecture, which consisted of two convolutional layers with alternate subsampling layers. The last subsampling layer is followed by two fully connected layers. The final fully connected layer is replaced by an SVM classifier with different parameters.

III) Materials and Methods

In this study, we present a hybrid model of two super classifiers: Convolutional Neural Network (CNN) and Support Vector Machine (SVM). Section 3.1 provides an overview of the proposed system. Section 3.2 presents the CNN classifier, section 3.3 describes the SVM classifier in detail, section 3.4 briefly describes the hybrid CNN-SVM model, section 3.5 describes dataset preparation and trained parameter setting, and mapping presents in section 3.6.
3.1 Overview of the Proposed System

The proposed system includes basic stages such as, image acquisition, preprocessing, segmentation, feature extraction, and classification. The final output of the system is an editable text file. The proposed system model is depicted in Fig. 1. The next subsection describes each stage in detail, as well as the techniques used on each stage.

![Figure 1. Model of proposed system for Amharic character image recognition](image)

A. Image Acquisition

To capture digital images, the first crucial stage in the OCR system is image acquisition. In fact, most researchers recommend scanning documents at a resolution of 300 dpi for optimal OCR accuracy [10]. We used a Xerox work centre 5845 scanner to collect images from printed documents at a resolution of 300dpi. These scanned images are fed into the pre-processing stages as an input for further processing.

B. Preprocessing

Pre-processing comes after image acquisition, which aims to enhance image quality by removing unwanted features or noise in an image without losing any significant information. Most applications in character recognition systems used grey or binary images since processing colour image is computationally intensive [28-29]. In this work, pre-processing techniques such as grayscale conversion, binarization, noise removal, skew correction and morphological operation were employed as shown in table 1.

<table>
<thead>
<tr>
<th>Algorithm 1 : Pre-processing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input</strong>: RGB image(Irgb)</td>
</tr>
<tr>
<td><strong>Output</strong>: Cleaned image (Icleaned)</td>
</tr>
<tr>
<td>1 begin</td>
</tr>
<tr>
<td>2 Igray ← Convert to grayscale image(Irgb)</td>
</tr>
<tr>
<td>3 Dgray ← apply noise filter algorithm(Igray) Dgray= Denoised Image gray</td>
</tr>
<tr>
<td>4 Ibinary ← Convert to binary(Dgray)</td>
</tr>
<tr>
<td>5 Ideskew ← Apply skew detection and correction(Ibinary) Ibinary= Image Binary</td>
</tr>
<tr>
<td>6 Icleaned ← Apply morphology (Ideskew) Ideskew=Image deskew</td>
</tr>
<tr>
<td>7 Return Icleaned</td>
</tr>
<tr>
<td>8.end</td>
</tr>
</tbody>
</table>

In this work, denoised and binarized text images are given as the input to the skew detection and correction algorithm. First, find all the centre coordinates in the binary image to get the rotation matrix. Then compute the minimum rotated bounding box which contains the text region to obtain an angle value in the range [-90, 0], exclusive of 0. Based on the obtained angle and center coordinates of the image, we will find a 2x3 transformation matrix (M) using equation 1. Finally, using equation 2, rotate the binary image to get the rotated image (Irotated). A 2x3 transformation matrix (M) is given as follow [42]:

$$M = \begin{bmatrix} \alpha & \beta & (1 - \alpha) \cdot \text{center}.x - \beta \cdot \text{center}.y \\ -\beta & \alpha & \beta \cdot \text{center}.x + (1 - \alpha) \cdot \text{center}.y \end{bmatrix}$$  (1)
Where,

\[ \alpha = \text{scale} \cdot \cos \theta, \quad \beta = \text{scale} \cdot \sin \theta, \]

*center*. \( x \) and *center*. \( y \) are the coordinates along which the input image (here binary image) is rotated.

\[ I_{\text{rotated}}(x, y) = I_{\text{binary}} (M_{11}x + M_{12}y + M_{13} , M_{21}x + M_{22}y + M_{23}) \]  

(2)

C. Segmentation

Segmentation of images containing text documents has three levels of segmentation hierarchy, which are line segmentation, word segmentation and character segmentation. However, using all the three levels is not compulsory since it depends on the nature of applications [30]. In this stage, cleaned binary images are segmented into individual characters. In the proposed system, cleaned binary image is segmented into text line image using horizontal projection histogram and then individual characters segmented from text line image by using contour locating algorithm. The result of line segmentation and character segmentation presented in section IV.

D. Feature Extraction

After pre-processing and segmentation stages in OCR, feature extraction techniques are applied to get feature vectors from segmented character images that will be useful for classification. The aim of feature extraction is to capture the most relevant information from the original data and represent that information in a lower dimensionality space [31]. The choice of features has a significant impact on the classification accuracy [32]. There are two types of feature extractions: Handcrafted and Automatic (learned) feature extractions. Handcrafted feature extractions are features that are defined and extracted manually by the data scientist. On the other hand, learned (automatic) feature extraction is the process of automatically extracting relevant features for classification using deep learning algorithm. Convolutional neural network (CNN) is a good example of automatic feature extraction. The capacity to exploit spatial or temporal correlation in data is one of the appealing feature of CNN. It has notable attributes such as hierarchical learning, automated feature extraction, multitasking, and weight sharing. The popularity of CNN mainly derives from its ability to extract hierarchical feature [40]. Due to this reason we have used CNN for feature extraction in the proposed system.

E. Classification

Classification is the process of categorizing the feature vectors into target values or classes. After the feature extraction stage, the extracted feature vectors are fed into the classifier. Some popular classification algorithms have been employed for OCR system includes: Convolutional Neural Network, Artificial Neural Network, Support Vector Machine, K-NEAREST Neighbors, Gradient Boosting and other classification methods. In this study, SVM classifier was employed to recognize Amharic characters. CNN-Softmax classifier was also used to compare performance with the hybrid CNN-SVM model. The CNN classifier, SVM classifier, and hybrid CNN-SVM model are presented in the following sections.

3.2 CNN Classifier

Convolutional neural networks (CNNs), also called ConvNets, are hierarchical, multi-layer neural network with a deep supervised learning architecture. Convolutional neural networks were inspired by the human brain and works in a similar manner like humans identify objects visually [36]. CNN is comprised of several layers; including a convolutional layer, a non-linearity (activation) layer, a pooling layer, and a fully connected layer [41]. The first layer of CNN extracts simpler features such as lines, curves, etc. this output is passed to the next layer and detects more complex features (corners or combinational edges). When we move deeper into the network it detects more complex patterns such as faces, objects (here characters), etc. The detailed description of the convolutional neural network layers discussed below.

A. Convolutional Layer

Convolutional layer is the basic building block of CNN, which determines the output of related inputs in the receptive field. This output is obtained by convolving kernels over the height and width of the information data and computing the dot product between the inputs and filter values, tend to result in the building of a 2-dimensional activation map of that filter. The visualization of the 6x6 single channel sample input image convolved with 3x3 filter size with stride value 1 and padding zero and its 4x4 activation map are illustrated in Fig.2.
The main advantages of convolution operation in the same feature map, the weight sharing (parameter sharing) reduces the number of parameters [37, 39]. This weight sharing technique, improves the efficiency of CNN parameters efficient over fully connected networks [40].

Filter size (f), stride(s), and padding (p) are hyper parameters that determine the size of the output volume [37]. A stride value of one indicates that the sliding movement of filter. The output feature map has more information on the middle pixels and hence loses lots of information from corners. To overcome this the shrinking output, use padding before convolution by adding rows and columns zeros to the border of images. In this paper, we used a kernel size of 3, strides of 2 and the same padding.

Equation 3 and 4 describe the relationship between the size of feature map, the kernel size, padding, and the stride while calculating the output of the feature map. The \( n_h \times n_w \) input matrix convolved with \( f \times f \) filter size, \( s \times s \) stride size, and padding \( p \). output dimension of the feature map defined as:

\[
\begin{align*}
n_h^{[l]} &= \left[ \frac{n_h^{[l-1]} + 2p^{[l-1]} - f^l}{s^l} + 1 \right] \\
n_w^{[l]} &= \left[ \frac{n_w^{[l-1]} + 2p^{[l-1]} - f^l}{s^l} + 1 \right]
\end{align*}
\]  

Where the \([ ]\) symbol is the floor value, \((n_h, n_w)\) represent the dimension of the output feature map.

B. Activation (Non-linearity) Layer

Non-linearity is the layer that follows convolution. Non-linearity can be used to adjust the generated output. The most common types of nonlinearity, such as sigmoid function, tanh, rectified linear unit (ReLU), and leaky ReLU. However, because of its simplified definition and ability to create sparse connections, the Rectified Linear Unit (ReLU) has lately become increasingly popular [41]. In general, the ReLU activation function has several advantages, including its similarity to the biological neuron, ease of use, and the ability to perform faster training for large networks [36]. We used a rectified linear unit activation function. Mathematically, it is defined as:

\[
f(x) = \max(0, x)
\]  

C. Pooling Layer

A pooling layer (subsampling) is added between two convolutional layers [36]. Its primary function is to reduce the size of the feature-maps and the number of parameters in the network to control the overall computational complexity [37]. The pooling operation works by extracting only one output value from the input images tiled into non-overlapping sub-regions. The most common types of pooling layers are max-pooling and avg-pooling. In general, the max-pooling operation is popular in current applications since it takes the maximum values from each sub-region while preserving the maximum information. This results in faster convergence and better generalization [36]. A pooling size of 2 and strides of 2 were used in this work.

D. Fully Connected Layer

The fully connected layers work on the same principles as the regular neural network. In this layer, each node in a fully-connected layer is directly connected to every node in both the previous and next layers [38, 41]. It is a fully connected feed forward network that is mostly used as a classifier at the end of a network [36, 40]. The major drawback of a fully-connected layer is that it contains many parameters, which results in a large computational effort for training them [39]. Therefore, we need to reduce the number of nodes and connections. The removed nodes and connections can be satisfied by using the dropout technique [41]. In this paper, after tuning dropout hyper parameter we used dropout value 0.2.

As shown in Fig.3, CNN-Softmax model architecture has nine layers which contains six convolutional layers, one flatten layer and two fully connected layers. A max-pooling layer place between every two convolutional and ReLU layers. Dropout layer used after the last max-pooling layer and the first fully connected layer. The first dropout layer followed by flatten layer. After flatten layer the first fully connected layer with 256 neurons followed. The final fully connected layer has 231 neurons with softmax activation and used for classification. In this work, the last fully connected layer is replaced by SVM with a Radial Basis Function (RBF) and is presented in detail in section 3.4. Our CNN-Softmax model architecture can be represented as follows:

32x32 input image $\rightarrow$ **Block 1** $\rightarrow$ ((32conv$\rightarrow$ReLU) x2) $\rightarrow$ max-pool $\rightarrow$

**Block 2** $\rightarrow$ (((64conv$\rightarrow$ReLU) x2) $\rightarrow$ max-pool $\rightarrow$

**Block 3** $\rightarrow$ (((128conv$\rightarrow$ReLU) x2) $\rightarrow$ max-pool $\rightarrow$ Dropout $\rightarrow$

Flatten $\rightarrow$ [256FC$\rightarrow$ReLU] $\rightarrow$ Dropout $\rightarrow$ 231FC $\rightarrow$ **Output**
3.3 SVM Classifier

Support vector machine (SVM) is a supervised classification machine learning algorithm based on statistical learning theory proposed by Vapnik [33] in 1998 and Cortes and Vapnik [34]. The main objective of SVM is to find a decision boundary or hyperplane with maximum margin which optimally separate two classes as depicted in Fig.4. Where \( H_0 \) is the optimal hyperplane which maximize the margin, \( H_1 \) and \( H_2 \) are two dotted lines parallel to the hyperplane, the distance (margin = \( \frac{2}{|w|} \)) between \( H_1 \) and \( H_2 \) is the separating margin. The hyperplane with the maximum margin is the optimum hyperplane, where margin defined as the distance between the hyperplane and the nearest point for both classes. Support vectors are the samples or data points that are nearest to the decision boundary.

The main goal is to find a parameter that minimize the risk, which is given by:

\[
R_{emp}(\alpha) = \frac{1}{2l} \sum_{i=1}^{l} |y_i - f(x_i, \alpha)|^2
\]

Where \( y_i \) is output, \( x_i \) is input and \( \alpha \) is parameters.

For linear separable SVM [9], Label data such as \( \{x_i, y_i\}, i = 1, 2, \ldots, l, y_i \in \{-1, 1\}, x_i \in \mathbb{R}^d \), SVM selects a hyperplane with maximum margin. The equations are given by:

\[
x_i \cdot w + b = \pm 1 \quad \text{for} \quad y_i = \pm 1
\]

(7)

(8)

The combined equation is given by:

\[
x_i \cdot w + b - 1 \geq 0 \quad \forall i
\]

(9)

In this paper, the data is nonlinear separable, SVM employs a kernel function (equation 10) for mapping the sample points \( (x_i, y_i) \) into a high-dimensional feature space and separate using nonlinear operator \( \Phi \). The optimal hyperplane is obtained by solving a quadratic programming problem dependent on regularisation parameters (equation 13). The last fully connected layer was replaced by SVM classifier with RBF kernel for Amharic character image classification.

The SVM kernel [19] is a function that takes low dimensional input space and transforms it to a higher. This transformation was performed by kernel functions such as linear, radial basis function (RBF), sigmoid and polynomial kernel types which are defined as:

Linear kernel: \( K(x_i, x_j) = x_i \times x_j \)

Polynomial kernel: \( K(x_i, x_j) = [(x_i \times x_j) + 1]^d \)

Sigmoid kernel: \( K(x_i, x_j) = \exp(\beta_0 x_i x_j + \beta_1) \)
Radial basis function (RBF) kernel: \( K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right) \)

\[ \gamma = \frac{1}{\sigma^2} \]

Where \( \sigma \), \( \beta_0 \), \( \beta_1 \) and \( \gamma \) are parameters to be determined empirically.

The kernel function \( k \), which given by:

\[ K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \] (10)

Where \( \Phi : R^d \rightarrow H \) used to map the data points in lower dimensions to a higher dimensional space.

The optimal separating hyper-plane (\( H_0 \)) is a margin classifier whose output is given by:

\[ f(x) = w^T \Phi(x) + b, \] (11)

Where, \( w \in R^n \), \( b \in R \) & \( \Phi(x) \)

In order to maximise the margin \( M(w, b) \), the following problem had to be solved:

\[ \min_{\frac{1}{2}} w^T w + C \sum_{i=1}^{l} \xi_i \] (12)

Subject to: \( w^T \Phi(x) + b \geq -1 - \xi_i, \xi_i \geq 0, i = 1, 2, \ldots, l \),

Where \( \xi_i \) a slack variable and \( C \) is a parameter that determines or tunes the trade-off between the maximum of the margin and the minimum classification error to avoid the over-fitting. By using the principle of Kuhn-Tucker [33] and applying Lagrangian multipliers \( \alpha_i \) we were led to maximize the dual form of Lagrangien L with respect to \( \alpha_i \) which has an upper bound of \( C \):

\[ L(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j K(x_i, x_j) \] (13)

Subject to \( \sum_{i=1}^{l} y_i \alpha_i = 0 \) with \( 0 \leq \alpha_i \leq C, y_i = 1, 2, \ldots, l \),

Where \( \alpha_i \) are the Lagrange multiplier, \( y_i \) represents the label for \( i \)th training sample, \( l \) is the number of training samples and \( C \) is real parameter which is varied through a wide range of values.

The optimal hyper-plane (\( H_0 \)) can be defined as:

\[ f(x) = \text{sgn} \left( \sum_{i=1}^{l} y_i \alpha_i K(x_i, x) + b \right), \] (14)

Where \( K(x_i, x) = \exp(-\gamma \|x_i - x\|^2) \) is the kernel function based on a RBF, and \( \text{sgn}(.) \) is the sign function.

According to Milgram et al. [35], the two popular multi-class SVM approaches are one versus one (ovo) and one versus all (ova). Multi-class SVM used divide and conquer method to decompose the multiclass problem into several binary class and build one SVM for each class. The one versus one approaches, which builds one SVM for each pair of class. Another approaches, one versus all method, which consists of building one SVM per class, trained to distinguish the samples in a single class from the samples in all remaining classes.

In this work, the one-versus-all method was employed for 231 classes. One versus all methods is computationally more efficient (only N classifiers are required) than one versus one approach, since the complexity of one versus one methods is \( O(N_{\text{classes}}) \).

3.4 Hybrid CNN-SVM Model

CNN has notable attributes such as hierarchal learning, automated feature extraction, multitasking, and weight sharing [40]. CNN is an extension model of the MLP because its theoretical learning method is the same as that of the MLP. MLP learning algorithm is based on Empirical Risk Minimization, which attempts to minimize errors in the training set. When the back-propagation algorithm finds the first separating hyperplane, whether it is the local or global minima, the training process is terminated, and the algorithm does not continue to improve the separating hyperplane solution. The generalisation ability of MLP is lower than that of SVM. On the other hand, the SVM classifier uses the Structural Risk Minimization principle to reduce generalization errors on unseen data with a fixed distribution for the training set [18].

In this study, we have designed a new hybrid CNN-SVM model for recognizing Amharic characters by taking their notable attributes of CNN and SVM classifiers. The architecture of proposed hybrid CNN-SVM model was designed by replacing the last fully connected layer of CNN-Softmax model with an SVM classifier. In this model, a CNN trained using back-propagation to extract features automatically from the raw images, and then the extracted features are given as input to SVM to recognize unseen data. The architecture of the new hybrid CNN-SVM model is depicted in Fig. 5. Firstly, the normalized and centered 32x32 pixel input images are given to the input layer, and the CNN-Softmax model is trained until the training process converges. After CNN-Softmax model is trained, the last fully connected layer replaced by the SVM with a Radial Basis Function (RBF). Then, the output of the first fully connected layer of the trained CNN-Softmax model is given as an input to the SVM classifier. A new model is created to get trained feature vectors from the output of the first dense layer (new model=Model(inputs=cnn-model.input, outputs=cnn-model.get_layer('dense_one'))). Using new model training and testing datasets are predicted. The predicted training dataset is used as the input feature vector to fit the SVM classifier. Once the SVM trained, it perform recognition task and make new decisions on images with such automatically extracted features.
3.5 Dataset Preparation and Parameter Setting

We have prepared printed Amharic database, we consider only 231 Amharic characters as shown in Fig. 6. For each class, 175 original character images were prepared. In addition to these original images, we used 45 augmented character images for each class to increase the dataset size. The augmented dataset has been applied by implementing rotation and adding some noise. The rotation degrees of -10, -5, 5, and 10 were used.

<table>
<thead>
<tr>
<th>Original</th>
<th>Augmented</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>40,425</td>
<td>8,085</td>
<td>2,310</td>
</tr>
<tr>
<td>50,820</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Total Prepared Amharic dataset
Parameter setting

The optimal value of kernel parameter gamma ($\gamma$) and penalty parameter ($C$) were determined by applying the fivefold cross validation method after the tests on the training printed Amharic database using randomized search cross validation. Early Stopping was also implemented to control the over-fitting of CNN model. Patience value 5 was set to stop training when validation loss had stopped improving.

Table 3. Trained parameter setting for CNN-SVM model

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>batch size</td>
<td>64</td>
</tr>
<tr>
<td>optimizer</td>
<td>Adam</td>
</tr>
<tr>
<td>learning rate</td>
<td>0.0001</td>
</tr>
<tr>
<td>dropout</td>
<td>0.02</td>
</tr>
<tr>
<td>Kernel parameter gamma ($\gamma$)</td>
<td>0.001</td>
</tr>
<tr>
<td>penalty parameter ($C$)</td>
<td>50</td>
</tr>
</tbody>
</table>

3.6 Mapping

In the proposed CNN-SVM model, SVM predicts the class label of characters ranging from 0-230 for 231 classes. The system maps each character class label to the corresponding hexadecimal code from a lookup table. The mapping or lookup table contains the class label with its corresponding hexadecimal code value. Finally, the system converts the hexadecimal value obtained from the lookup table into a byte. This converted byte is interpreted by a text editor or word processing software.

As shown in Table 4, the class label 21 corresponds to the hexadecimal code ‘E18889’. The system converts the hexadecimal code ‘E18889’ to a byte (b’xe1x88x98’). The text editor interprets this byte value, and the output is editable text format.

Table 4. Class label to Hexadecimal code lookup table

<table>
<thead>
<tr>
<th>Class label</th>
<th>Amharic characters</th>
<th>Unicode values</th>
<th>Hexadecimal codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>የ</td>
<td>U+1219</td>
<td>E18890</td>
</tr>
<tr>
<td>22</td>
<td>የ</td>
<td>U+121A</td>
<td>E1889A</td>
</tr>
<tr>
<td>23</td>
<td>የ</td>
<td>U+121B</td>
<td>E1889B</td>
</tr>
<tr>
<td>24</td>
<td>የ</td>
<td>U+121C</td>
<td>E1889C</td>
</tr>
<tr>
<td>25</td>
<td>የ</td>
<td>U+121D</td>
<td>E1889D</td>
</tr>
<tr>
<td>26</td>
<td>የ</td>
<td>U+121E</td>
<td>E1889E</td>
</tr>
</tbody>
</table>

IV) Experimental Results and Discussions

The experimental results are organized as follows: Section A shows the results of pre-processing stage. Section B presents the results of line and character segmentation. Section C analyse the results of CNN-Softmax model and CNN-SVM model. Finally, comparison result with other works for recognizing Amharic characters presented in section D.

A. Result of Preprocessing

The main objective of the pre-processing stage is to obtain skew free, denoised binary images, which are then fed into the next stage. Fig.7 shows the result of RGB image to grayscale image conversion.

(a)

(b)

Figure 7. The Result of grayscale conversion: (a) color image (b) grayscale image
In this work, we used the Otsu thresholding method to convert a denoised grayscale image to a binary image. It is recommended to use a noise-free image before using Otsu thresholding since images with noise and low-intensity illumination affect its thresholding performance. This is why we used denoised grayscale input using Gaussian blur as input to the Otsu thresholding method. As shown in Fig. 8(c), the foreground is represented as white color and the background as black color after applying inverted binary thresh with Otsu thresholding method on denoised grayscale image.

![Fig. 8](image)

Figure 8. The result of binarization: a denoised grayscale image (a) converted to a binary image (c) using the Otsu thresholding method, along with its corresponding histogram (b).

Skewed documents with text are typically detected and corrected after noise reduction and appropriate binarization. The result of skew detection and correction is depicted in Fig. 9.

![Fig. 9](image)

Figure 9. The results of skew detection and correction: input denoised binary image (a), skew image detection (b) and deskewed image with -4.17° (c).

B. Result of Segmentation

The proposed line segmentation algorithm was tested on a variety of document images and achieved as expected. The algorithm extracts a list of text line images that have been segmented from a clear binary image. Fig. 10 shows a sample line segmentation result. The text line images were accurately segmented, as shown in the figure, showing that the employed segmentation algorithm performs well. Horizontal projection method is used to compute the sum of all foreground pixel is white along row for binary inverted image and construct the corresponding histogram. For line segmentation horizontal projection method was employed.
Fig. 11 shows the result of character segmentation after segmented text line image obtained. For character segmentation, we used a contour locating algorithm to extract segmented characters from text line images. The input to the contour detecting algorithm is a list of threshold text line images. Then find the contours to detect characters in the text line image. After finding contours, extract the coordinates for each character and then sort all the bounding boxes from left to right, top to bottom. Finally, iterate over the bounding box and extract characters from the text line binary image and save character images.

Figure 10. The result of line segmentation: Input binary image (a), histogram image (b), line separation (c), and segmented text line image (d)

Figure 11. The result of character segmentation: Input text line image (a), region of interest (b), and segmented character image (c)

C. Result of CNN-Softmax classifier and Hybrid CNN-SVM model

We trained the proposed model on Tesla T4GPU with a 13GB RAM and 13GB disk space on Google Colab. Experiments have been conducted on own prepared dataset and benchmark Amharic Optical Character Recognition Database (ADOCR) [14], synthetically generated with different degradation levels. As shown in Table 6, from the experiment results, 99.84% test accuracy was achieved on the own prepared Amharic dataset and a 95.59% test accuracy on the public ADOCR database. Thus, the proposed model outperforms previously existing works attempted by others [13, 15] to recognize printed Amharic characters on the same database. However, the recognition accuracy on Amharic optical character recognition (ADOCR) database was not achieved the same result as own prepared Amharic dataset. The reason is that the synthetic generated character images in ADOCR database using the OCRopus framework were degraded and unbalanced. The training accuracy and training loss results of the CNN-Softmax model at 100 epochs without early stopping are shown in Figs. 12 and 13, respectively. In the proposed hybrid model early stopping was employed. Patience value 5 was set to stop training when validation loss had stopped improving. The overall recognition accuracy is presented in Table 6.
Indeed, CNN extract features automatically. SVM has a high generalization performance, which mean that better in accuracy with robustness to predict class of unseen character images. By taking the advantage of the two classification, we proposed CNN-SVM model. The CNN-SVM model outperform the CNN-Softmax as presented in table 6.

Evaluate the performance of proposed in accuracy is not enough, we used other evaluation metrics which are confusion matrix and classification reports. We created 231 by 231- dimension confusion matrix using python. However, Amharic characters are large in number so it is difficult to plot the confusing matrix graphically. As shown in table 5, we obtained the overall recognition accuracy of 99.84% from confusion matrix for CNN-SVM model on own prepared Amharic database. Thus, the result of confusion matrix metrics the same result achieved as accuracy metrics. Fig. 15 and Fig. 16 show the confusion matrix and classification reports for 19 misclassified Amharic characters and have similar shapes out of 231 classes using the CNN-SVM model, respectively.

Table 5. Test accuracy of CNN-SVM model on our prepared Amharic dataset

<table>
<thead>
<tr>
<th>Proposed Model</th>
<th>Confusion Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN-SVM</td>
<td>Test Accuracy (%)</td>
</tr>
<tr>
<td></td>
<td>99.84</td>
</tr>
</tbody>
</table>

Table 6. Accuracy result of proposed CNN-SVM model on our prepared Amharic dataset

<table>
<thead>
<tr>
<th>Model</th>
<th>Training accuracy (%)</th>
<th>Training time(second)</th>
<th>Test Accuracy (%)</th>
<th>Testing time (second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN-Softmax</td>
<td>99.51</td>
<td>180</td>
<td>99.63</td>
<td>3</td>
</tr>
<tr>
<td>CNN-SVM</td>
<td>99.95</td>
<td>60</td>
<td><strong>99.84</strong></td>
<td>35</td>
</tr>
</tbody>
</table>
Fig. 14 shows the confusion matrix of CNN-Softmax for 19 misclassified Amharic characters that have similar shapes out of 231 classes in the test data. The character 1 times “አ” misclassified as “॰” 1 times “ና” misclassified as “ብ”, 3 times “ጓ” misclassified as “ጔ”, 2 times “ለ” misclassified as “ለ”, 4 times “ለ” misclassified as “ለ”, 1 times “ለ” misclassified as “ለ”, and 1 times “ለ” misclassified as “ስ”. 

Fig. 15 shows confusion matrix of CNN-SVM model for 19 misclassified Amharic characters and having similar shapes in the test data. The character 1 times “ስ” misclassified as “ስ”, 3 times “ስ” misclassified as “ስ”, 3 times “ስ” misclassified as “ስ”. The classification report of CNN-SVM model for 19 misclassified Amharic characters and having similar shapes in the test data is depicted in Fig. 16. All output parameters: precision, recall, and f1-score values are close to the classification accuracy. The output under support is the number of values whether they are predicted positive or negative in each category of the actual values. The experimental results show that more characters are misclassified using the CNN-Softmax classifier when compared with a hybrid model. This indicates that the proposed hybrid CNN-SVM Model more robust than CNN-Softmax model.
Figure 15. The result of CNN-SVM model confusion matrix for 19 misclassified Amharic characters and having the same shape.

<table>
<thead>
<tr>
<th>Character</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>ኦ</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>44</td>
</tr>
<tr>
<td>ኧ</td>
<td>0.94</td>
<td>1.00</td>
<td>0.97</td>
<td>44</td>
</tr>
<tr>
<td>ከ</td>
<td>1.00</td>
<td>0.93</td>
<td>0.96</td>
<td>44</td>
</tr>
<tr>
<td>ኩ</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>44</td>
</tr>
<tr>
<td>ኪ</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>44</td>
</tr>
<tr>
<td>ካ</td>
<td>1.00</td>
<td>0.93</td>
<td>0.96</td>
<td>44</td>
</tr>
<tr>
<td>ኬ</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>44</td>
</tr>
<tr>
<td>ክ</td>
<td>0.98</td>
<td>0.93</td>
<td>0.95</td>
<td>44</td>
</tr>
<tr>
<td>ኮ</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>44</td>
</tr>
<tr>
<td>ኯ</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>44</td>
</tr>
<tr>
<td>ኲ</td>
<td>0.98</td>
<td>1.00</td>
<td>0.99</td>
<td>44</td>
</tr>
<tr>
<td>ኳ</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>44</td>
</tr>
<tr>
<td>ኴ</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>44</td>
</tr>
<tr>
<td>ኵ</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>44</td>
</tr>
</tbody>
</table>

Micro avg      0.99  0.99  0.99  836
Macro avg      0.99  0.99  0.99  836
Weighted avg   0.99  0.99  0.99  836

Figure 16: The result of CNN-SVM model classification for 19 misclassified Amharic characters and having the same shape.
D. Comparison

Our hybrid CNN-SVM model achieved an accuracy of 99.84% on own prepared Amharic dataset, which contains 50,820 character images. To evaluate the performance, the proposed model was applied on the benchmark of a synthetic generated ADOCR database [14], with a dataset size of 77,994 character images, and we obtained 95.59% recognition accuracy. Based on the comparison results, the proposed CNN-SVM model achieves higher recognition accuracy. The hybrid CNN–SVM model outperforms each individual classifier because the hybrid model taking the advantage of the CNN and SVM classifiers. The automatic learning features of CNN make the CNN-SVM model more robust than computed handcrafted features like ANN [7], SVM [8]. The experimental results shows that the proposed CNN–SVM model outperforms the other existing methods by achieving higher recognition accuracy. Table 7 shows the comparisons with various methods on different database related to this work.

Table 7. Comparison result of various methods on different datasets for Amharic OCR

<table>
<thead>
<tr>
<th>Authors</th>
<th>Database size</th>
<th>Document types</th>
<th>Method</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yaregal[7]</td>
<td>1044 character images</td>
<td>Printed</td>
<td>ANN</td>
<td>73.18</td>
</tr>
<tr>
<td>Million[8]</td>
<td>7680 character images</td>
<td>Synthetic</td>
<td>SVM</td>
<td>96.95</td>
</tr>
<tr>
<td></td>
<td>6240 character images</td>
<td>Printed</td>
<td>SVM</td>
<td>91.45</td>
</tr>
<tr>
<td>Berihanu et al.[13]</td>
<td>80,000 character images</td>
<td>Synthetic</td>
<td>CNN</td>
<td>92.7</td>
</tr>
<tr>
<td>Berihanu et al.[15]</td>
<td>77,994 character images</td>
<td>Synthetic</td>
<td>CNN</td>
<td>94.97</td>
</tr>
<tr>
<td>Proposed</td>
<td>77,994 character images</td>
<td>Synthetic</td>
<td>CNN-SVM</td>
<td><strong>95.59</strong></td>
</tr>
<tr>
<td>Proposed</td>
<td>40,425 character images</td>
<td>Printed</td>
<td>CNN-SVM</td>
<td><strong>99.84</strong></td>
</tr>
<tr>
<td></td>
<td>10,395 character images</td>
<td>Augmented</td>
<td>CNN-SVM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>= 50,820</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

V) Conclusion

This paper presents a hybrid architecture of CNN-SVM classifier for the recognition of printed Amharic characters. CNN and SVM are most popular and successful classifiers and widely used in many pattern recognition applications. In this hybrid model, CNN works as a trainable automatic feature extractor from the raw images and extracted features given as input to the SVM for classification task. Experimental results on the own prepared Amharic database and benchmark Amharic Optical Character Recognition Database (ADOCR) shows significant improvement of the proposed CNN-SVM model. It achieved an accuracy of 99.84% on own prepared Amharic dataset, which contains 50,820 character images and 95.59% recognition accuracy on ADOCR database. Both are the best results when compared with other recently research works. Overall, we conclude that the proposed CNN-SVM model is giving the start-of-the-art recognition results to recognize Amharic characters. Furthermore, we observed that CNN-SVM classifier more robust than CNN-Softmax to predict unseen character images.

Research on the hybrid CNN–SVM learning model is still at an open research. The performance of CNN-SVM model can be improved further by fine-tuning its structure and parameters. Our results show that the proposed hybrid model is a promising classification method for recognizing printed Amharic characters due to two properties: the first is that the relevant features can be extracted automatically by the hybrid model, whereas the success of most other traditional classifiers is largely dependent on the retrieval of good hand-crafted features, which is a time-consuming task. The second is that the hybrid model combines the merits of SVM and CNN classifiers, which are the most popular and successful classifiers in pattern recognition. Extending the proposed hybrid model to other applications like word/character recognition.
Reference


