Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 04, April-2022

Geez Reading Level Classification by Audio Feature Extraction using CNN

Tadese Henok Seifu[1]
Software Engineering
School of Information Technology and Engineering
Tianjin University of Technology and Education,
Tianjin 300222, P.R.China

Assoc. Prof. Xiao Zheng[2]
Software Engineering
School of Information Technology and Engineering
Tianjin University of Technology and Education,
Tianjin 300222, P.R.China

Abstract:- **The Ge'ez language is an ancient Semitic language of Ethiopian. This language originates from the region encompassing Ethiopian and Eritrea Orthodox Tewahedo Church. The language is given as a course in the Orthodox Tewahedo Church spiritual school. Ge'ez has three types of reading these are Ge'ez, wurid, and kume. Each type of reading can characterize by different features and become distinguishable with its unique feature. The proposed system has five components: data acquisition, preprocessing, segmentation, feature extraction, and classification. In Audio signal processing, we collected a dataset from a spiritual scholar with audio files and different spiritual websites. Data preprocessing transform the raw data into a useful and efficient format. Segmentation is the stage used to split the audio signal before changing into the spectrogram with equal time intervals. In feature extraction, we propose to apply a Gabor filter on the input spectrogram image for texture feature extraction. Finally, the proposed model classifies the input spectrogram image using the convolutional neural network approach for grading into a specific class (Ge'ez, Wurid, and Kume). The proposed system is implemented using Python Anaconda and tested using a sample spectrogram image dataset.**

Keywords: Ge'ez reading Level, audio signal processing, Spectrogram, CNN.

1. INTRODUCTION

Ethiopia has language diversity such as Ge'ez, Amharic, Tigrigna, Guragegna, Afaan Oromo, Argobba, Harari, agewigna, etc. Among these languages, the Ge'ez language is the classical language of Ethiopia and is still used as the liturgical language of EOTC and the Beta Israel Jewish community of Ethiopia [1]. Nowadays, Ge'ez is used only as of the main liturgical language of the Ethiopian Orthodox Tewahedo Church, Eritrean Orthodox Tewahedo Church, the Ethiopian Catholic Church, Eritrean Catholic Church, and the Beta Israel Jewish community [2].

The advancement of technology gives valuable information to the community by providing the ideal solution to existing problems. Automatic music or audio categorization is one of the important tasks for Music Information Retrieval. With the development of the knowledge of Machine Learning, researchers have implemented different techniques for automatic audio classification and it involved audio analysis tasks like song identification, chord recognition, sound event detection, mood detection, and feature extraction [3].

Generally, Ethiopian Orthodox Tewahedo Church has three types of Ge'ez language, reading, and each type has its characteristics to distinguish each other. These are Ge'ez (ግዕዝ), Wurid (ውርድ), and Kume (ቁም) reading types [4]. The Ge'ez reading type is the first reading when the student learns it after finishing letters. This type of reading method is used to read all letters individually as number reading, but it reads with "Zema". The wurid reading type is the second reading type when the student learns it after finishing the Ge'ez reading type. In this type of reading each word is read in a calm and sad tone. The Kume reading type is the final stage of the Ge'ez reading type when the student learns it after finishing both Ge'ez and wurid reading types. This type of reading strategy is a well-read, alphabetical language that makes sense for both the reader and the listener [5].

Each type can be described with different features and the last goal of this study will be the classification of these reading types using a classifier and make them easily distinguishable by people.

2. GEEZ LANGUAGE

Ge'ez language is an ancient Semitic language of African nations that is the religious rite language of the Ethiopian Orthodox Church. It‟s the ascendant of the trendy Ethiopian languages like Amharic. The Ge'ez language has its script of over two hundred characters, used currently solely within the Orthodox Church. Ethiopian Orthodox Christian music is believed to have been established by Saint Yared within the sixth century and is sung in Ge'ez, the religious rite language [1].

Traditionally, boys learn to read Ge'ez, as Bible students. The Ge'ez language used in most modern-day church services derives from the Kingdom of Axum. The service is long, over three hours, delivered in monotonous Amharic and an older ecclesiastical language like Latin, called Ge'ez. This language, the classical form of an ancient Ethiopian language, has extensive Christian literature and is still used in Ethiopia as a liturgical language and many ancient kinds of literature were written in Ge'ez. The literature includes religious texts and secular writings. The ancient philosophy, tradition, history, and knowledge of Ethiopia were being written in Ge'ez [5]. The Ethiopian Orthodox Tewahedo Church has three types of Ge'ez language, and reading, and each type has its characteristics to distinguish from the others. These are Ge'ez (ግዕዝ), Wurid (ውርድ), and Kume (ቁም) reading types.

3. CONVOLUTIONAL NEURAL NETWORK

In fully connected layers, each unit (neuron) is connected to all of the units in the previous layer. On CNN, however, each

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 04, April-2022

unit is connected to a small number of units in the previous layer. In addition, all units are connected to the previous layer in the same way, with the same weights and structure. CNNs use convolution operation instead of general matrix multiplication in at least one of their layers.

Convolutional neural networks have a different architecture than other neural networks. In other neural networks, each layer is fully connected to all neurons in the previous layer. In convolutional neural networks, neurons in one layer are not connected to all neurons in the next layer, but only to a small region of it.

The basic layers of CNN are convolution layers, pool layers, and fully connected layers. Conv-Layer is a layer that gives the network name. This is the first layer to extract features from the input image. It performs an operation called convolution.

In the context of CNN, convolution is a linear operation involving multiplication according to the element between the input image and the filter. The main process at the level of Conv is convolution Operation. A filter is a small matrix used to detect patterns in an input image. Matrix values are started with random numbers using different methods.

The pooling layer simplifies the information obtained from the output tier of the convolution layer. The pooling layer receives the feature output from the convolution layer and prepares the condensed feature's output. It is used to gradually reduce the size of the input representation. Therefore, it reduces the number of parameters required, and the amount of computation needed and controls the over fittings. Max pooling is the most common method of pooling.

A fully connected layer is the only layer in the model where every neuron from the previous layer connects to every neuron from the next layer. It uses features from the output of the previous layer to classify the input image based on the training data.

4. PROPOSED NETWORK ARCHITECTURES

The proposed system architecture has the following stages, such as audio data acquisition, preprocessing, segmentation, audio-spectrogram conversation, feature extraction, and classification. The proposed system is shown in figure 1. In the audio data acquisition stage, we collected audio files covering all activities from the initial raw data to construct the last dataset.

In the audio data preprocessing phase, we normalize the audio into a comprehensible format. In the audio segmentation phase, we divide the audio signal into homogeneous segments. In the spectrogram phase, we visualized the given homogeneous segments of audio information into images. In the feature extraction phase, we find associations between different objects from the spectrogram images. Finally, the proposed model used CNN architecture has input layers, convolutional layers, and fully connected layers followed by SoftMax classifiers. Details about each stage are presented in the subsequent sections.

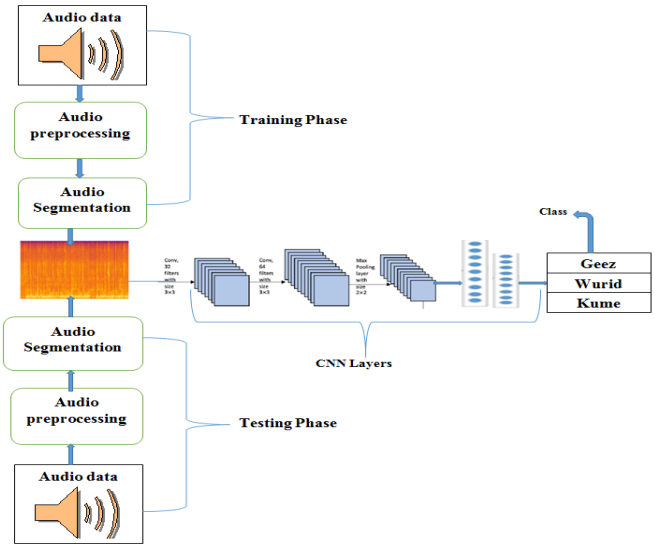

Figure 1: Proposed system architecture for Ge'ez Reading Type classification

4.1 Data Acquisition

Audio data acquisition is the primary task of the study because it is impossible to conduct the study without collecting audio data from different sources. Audio data acquisition is the process used for getting data ready for the classification model. It is a technique of acquiring required audio files from the spiritual school of EOTC experts as well as sources from different spiritual websites.

We try to use two forms of acquiring audio data. One way is recording the audio data from a spiritual school of EOTC experts and the second way is taking the annotated audio file from spiritual websites. 1701 audio datasets were collected from spiritual school experts as well as different websites. Collected datasets are to apply an existing classifier to fix-sized segments of an audio recording, yielding a sequence of class labels that characterize the whole signal. The sound length in this audio is about 20 seconds.

4.2 Data Preprocessing

The data are manually recorded or downloaded from different free source website initial transformation and filtering was done to make sure all the audio files have the same file extension format. Many of the websites allow downloading files in just Mp3 format which was then converted to .wav format. All audio files are converted into .wav format online using mp3cut.net convertor.

As the data is often taken from multiple sources that are normally not too reliable and that too in different formats, when working on a machine learning problem, more than half of the time is consumed in maintaining the data quality. The initial raw data may have diverse issues such as noises, distortion, or other irrelevant song signal detail that can hamper the performance of the model. Also, as the data is manually downloaded from different free source website initial transformation and filtering was done to make sure all the audio files have the same format.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 04, April-2022

Many of the websites allow downloading files in just Mp3 format which was then converted to .wav formats with a sampling rate of 44100Hz. It also focuses on audio file segmentation with the same amount of time interval that allows us to properly and correctly convert spectrogram images.

The audio data for this investigation was gathered from different sources and is commonly not reasonable for direct use in training the model preprocessing of audio data containing noise removal.

4.3 Training

In this phase feature of the audio file is extracted after audio data is represented in visual form. The audio file is used for generating the spectrogram then this transformed image is directly fed to the CNN algorithm to learn features and with different layers, the spectrogram will be filtered out to be classified with its predefined classes.

4.4 Testing

In this phase, we will follow the same procedure as the training phase. We have to preprocess the image in the same manner as the training. If we follow any other way, it will lead us to incorrect classification since the network may be presented with inputs it cannot categorize. Similarly, feature learning is also done in the same manner as in the training by using the learning model constructed from the training. Input images that are different from the training datasets are used, which are called testing datasets.

5. EXPERIMENTS

5.1 Dataset

The process used for getting data ready for the classification model can be summarized in the following steps: collect data, preprocess data and transform data. We follow this process iterative with many loops to prepare the dataset required.

Step 1 is concerned with collecting available data needed to solve the problem. Audio data are collected from Ethiopian Orthodox Tewahedo church traditional school scholars and other spiritual sources.

Step 2 is about getting the collected data into a form that can be easy to work with. The formatting is about making the data selected in a format that is suitable for the work. The collected audio data are converted into .wav format. .wav format is selected because most of the collected audio data are in .wav format.

In step 3 we transform the data collected. This step is related to making the dataset suitable for the algorithm used and knowledge of the problem domain. Audio data must be segmented with equal size to have uniform time intervals. Finally, the segmented Audio files are changed into a visual representation form which is a spectrogram. The data which are fed from the convolutional network is the spectrogram image in the form of png or another image format. The number of data collected for each class is shown below.

TABLE1. COLLECTED AUDIO DATASETS TABLE STYLES

No.	Classes	Quantity	Audio Data Format	The Time Interval for each audio data
1	Ge'ez	120	.wav	20sec
2	Wurid	224	.wav	20sec
3	Kume	1357	.wav	20sec
Total				1701

5.2 Implementation

The experiment for this research work was done primarily in python using various python libraries. The most frequently used libraries in this research work are Pandas, NumPy, Librosa, glob, learn, and by dub. data exploration and extended library for visualizations were used while for data modeling, Conv2D used Keras with TensorFlow backend. Google Colab is a free online cloud-based Jupiter notebook environment that allows us to train our machine learning model on GPUs. It does not matter which computer you have, what its configuration is, and how ancient it might be [6].

The summary of the results was analyzed and generated using the matplotlib python library. The model is trained for 100 epochs, a batch size of 32, and a learning rate of 0.001 (1e-3). The data is segmented into training and testing datasets such that 80% percent of the data is assigned for training the model and 20% percent of the data is assigned for testing.

5.3 Test Results

Tests are conducted both on Convolutional Neural Network classifiers to decide the best performing classifier based on the criterion of classification accuracy. Experiments are conducted to assess the performance of the proposed model. To measure the performance of our model, we have used precision, recall, accuracy, and f1-score. In addition, we have also calculated the micro-average and macro-average for all the previously mentioned performance metrics.

5.4 *Evaluation of the Proposed Model*

To evaluate our proposed model, we have seen the result obtained from the other related architecture which is performed on image classification concerning our result and if it has better accuracy & performance well otherwise, we must apply different techniques to make our model more accurate and to have high performance [7]. Related models that are used to evaluate our model to compare with AlexNet, VGGNet, and GoogleNet.

We have trained and tested each model using our dataset. The comparison is done based on the following parameters.

- Accuracy: a model that records higher accuracy is better for the classification of Ge'ez reading type.
- Loss: a model that records a lower loss value is better for the classification of Ge'ez reading type.
- Size: a model with a smaller image size is better at processing the dataset due to the smaller number of parameters

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 04, April-2022

trying to learn. A model that has a smaller image size (or smaller number of parameters) is better since it is faster to train.

- Time: a model with the fastest time is preferred for classifying the testing data.

5.4.1 *Comparison with AlexNet Model*

The performance (accuracy and loss value) of the AlexNet model is shown in the figure below. It takes nearly an hour to train the model in python anaconda software and it is better to run in collab to execute within a few minutes even if it is connection-based. As clearly depicted in figure 2 below, AlexNet obtains 99.51% training and 85.79% testing accuracy of our data. The total time to train the model is more than an hour, and a few minutes in Colab takes on average 100 seconds per epoch.

```
Epoch 1/100
15/15 [==============================] - 11s 189ms/step - loss: 2.5897 - accuracy:
0.6691 - val_loss: 11.2113 - val_accuracy: 0.8053
Epoch 2/100
15/15 [==============================] - 2s 122ms/step - loss: 0.4351 - accuracy:
0.9088 - val_loss: 60.3874 - val_accuracy: 0.8053
Epoch 3/100
15/15 [==============================] - 2s 114ms/step - loss: 0.2636 - accuracy:
0.9250 - val_loss: 58.1353 - val_accuracy: 0.8053
Epoch 4/100
15/15 [==============================] - 2s 114ms/step - loss: 0.1610 - accuracy:
0.9496 - val_loss: 51.1535 - val_accuracy: 0.8053
.....
Epoch 98/100
15/15 [==============================] - 2s 117ms/step - loss: 0.0149 - accuracy:
0.9916 - val_loss: 6.2305 - val_accuracy: 0.8053
Epoch 99/100
15/15 [==============================] - 2s 116ms/step - loss: 0.0075 - accuracy:
0.9973 - val_loss: 6.7951 - val_accuracy: 0.8053
Epoch 100/100
15/15 [==============================] - 2s 117ms/step - loss: 0.0074 - accuracy:
0.9964 - val_loss: 2.2639 - val_accuracy: 0.8581
```

Figure 2. Classification Accuracy of the training phase of the AlexNet model

The accuracy and loss of the AlexNet model with our dataset are shown in Figures 3 and 4 below. As clearly shown in the training loss and the accuracy curve shown in the figure below, the training accuracy was higher than loss accuracy throughout the curve it is shown when the number of epochs is increased, the accuracy of the model is high and the loss of the model also decreases.

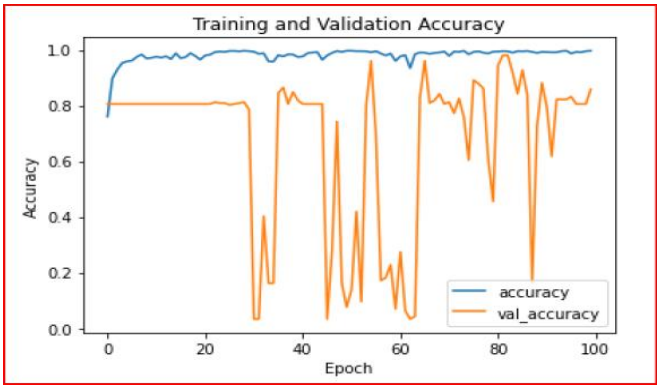

Figure 3. Training accuracy curve of AlexNet model

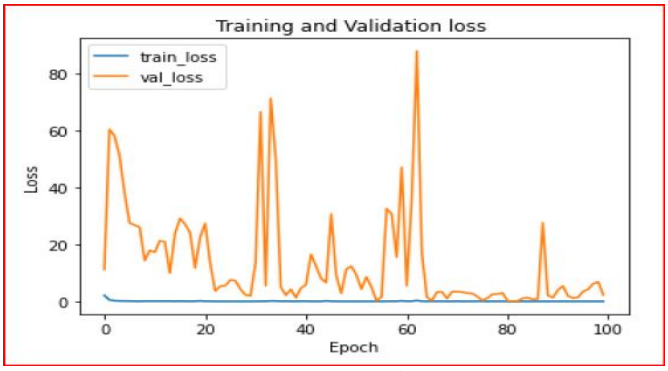

Figure 4. Training loss curve of AlexNet model

A basic sequential AlexNet model was designed by the architecture as shown in Figure 4 The first layer of the model is a convolutional layer using 32 filters and a kernel size of (7, 7). The input shape is specified to be (128,128,1) taking RGB images. The max-pooling layer is then used to reduce the spatial dimensions of the output layer with a stride parameter. The stride parameter is a 2-tuple of integers, specifying the step of the convolution along the X and Y-axis of the input volume.

Typically, the stride value is left to the default value which is (1, 1) but it is increased to (4,2) in this case to assure the size of the output volume is reduced. The pooling layer is followed by an activation layer of ReLU. Another set of Convolution, Pooling, and activation layers with enhanced filter criteria for convolution are used followed by another layer of convolution and activation layer.

The output of the activation layer is then fed to Flatten layers and the 0.5 dropout layer. The next layer is a fully connected dense layer input which is from the dropout layer. In the end, a dense layer and activation layer is used to interpret the features, the input features of the machine learning model are this output. The model was compiled using the Adam optimizer and categorical_crossentropy multiclass cross-entropy loss function. The summary statistics for the model are shown in Figure 5 below.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 32, 32, 96)	34944
batch_normalization (BatchNo	(None, 32, 32, 96)	384
activation (Activation)	(None, 32, 32, 96)	0
max_pooling2d (MaxPooling2D)	(None, 16, 16, 96)	0
conv2d_1 (Conv2D)	(None, 16, 16, 256)	614656
batch_normalization_1 (Batch	(None, 16, 16, 256)	1024
activation_1 (Activation)	(None, 16, 16, 256)	0
max_pooling2d_1 (MaxPooling2	(None, 7, 7, 256)	0
conv2d_2 (Conv2D)	(None, 7, 7, 384)	885120

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 04, April-2022

batch_normalization_2 (Batch (None, 7, 7, 384) 1536
activation_2 (Activation) (None, 7, 7, 384) 0
conv2d_3 (Conv2D) (None, 7, 7, 384) 1327488
batch_normalization_3 (Batch (None, 7, 7, 384) 1536
activation_3 (Activation) (None, 7, 7, 384) 0
conv2d_4 (Conv2D) (None, 7, 7, 256) 884992
batch_normalization_4 (Batch (None, 7, 7, 256) 1024
activation_4 (Activation) (None, 7, 7, 256) 0
max_pooling2d_2 (MaxPooling2 (None, 3, 3, 256) 0
flatten (Flatten) (None, 2304) 0
dense (Dense) (None, 4096) 9441280
batch_normalization_5 (Batch (None, 4096) 16384
activation_5 (Activation) (None, 4096) 0
dropout (Dropout) (None, 4096) 0
dense_1 (Dense) (None, 4096) 16781312
batch_normalization_6 (Batch (None, 4096) 16384
activation_6 (Activation) (None, 4096) 0
dropout_1 (Dropout) (None, 4096) 0
dense_2 (Dense) (None, 3) 12291
activation_7 (Activation) (None, 3) 0
==========
Total params: 30,020,355
Trainable params: 30,001,219
Non-trainable params: 19,136

Figure 5. AlexNet Model Summary

5.4.2 Comparison with GoogelNet Model

The performance (accuracy and loss value) of the GoogleNet model is shown in the figure below. It takes nearly an hour to train the model in python anaconda software and it is better to run in collab to execute within a few minutes even if it is connection-based. As clearly depicted in figure 6 below, GoogleNet obtains 98.41% training and 83.79% testing accuracy of our data. The total time to train the model is more than an hour, and a few minutes in Colab takes on average 100 seconds per epoch.

training network
Epoch 1/100
15/15 [==========] - 35s 1s/step - loss: 3.8922 - accuracy: 0.4311 - val_loss: 19.2913 - val_accuracy: 0.8359
Epoch 2/100
15/15 [==========] - 14s 564ms/step - loss: 0.7391 - accuracy: 0.8521 - val_loss: 69.3127 - val_accuracy: 0.8359

15/15 [==========] - 13s 561ms/step - loss: 0.0930 - accuracy: 0.9807 - val_loss: 0.0230 - val_accuracy: 0.9844
Epoch 100/100
15/15 [==========] - 14s 563ms/step - loss: 0.0626 - accuracy: 0.9840 - val_loss: 0.4893 - val_accuracy: 0.9375

Figure 6. Classification Accuracy of training phase of GoogleNet model

The architecture has designed a simple sequential GoogleNet model as shown in Figure 5. The first layer of the model is a convolutional layer with 64 filters. The input shape is defined as having (128, 128, 3) RGB images taken. The max-pooling layer is then used with a stride parameter to decrease the spatial dimensions of the output layer. The stride parameter is a 2-tuple integer parameter, defining the convolution step along the input volume's X and Y-axis. The stride value is usually left to the default value of (1, 1) but is increased to (4, 2) in this case to ensure that the output volume size is decreased.

The pooling layer is accompanied by a layer of ReLU activation. Another layer of convolution, pooling, and activation layer with improved convolution filter criteria is used, followed by another layer of convolution and activation layer. Flatten layers and 0.5 dropout layers are then fed to the output of the activation layer. The next layer is a dense layer input that is completely connected and is from the dropout layer. In the end, to interpret the features, a dense layer and activation layer are used, the input features of the machine learning model are this output. The model was compiled using the multiclass cross-entropy loss function of categorical_crossentropy and the Adam optimizer. The summary statistics for the model is as shown in Figure 7

below.

Model: "model"

Layer (type) Output Shape Param # Connected to
==========
input_1 (InputLayer) [(None, 128, 128, 3) 0
batch_normalization_7 (BatchNor (None, 128, 128, 3) 12 input_1[0][0]
conv2d_5 (Conv2D) (None, 128, 128, 64) 4800 batch_normalization_7[0][0]
batch_normalization_8 (BatchNor (None, 128, 128, 64) 256 conv2d_5[0][0]

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 04, April-2022

activation_8 (Activation) (None, 128, 128, 64) 0
batch_normalization_8[0][0]

zero_padding2d (ZeroPadding2D) (None, 130, 130, 64) 0
activation_8[0][0]

max_pooling2d_3 (MaxPooling2D) (None, 64, 64, 64) 0
zero_padding2d[0][0]

Figure 7 Classification Accuracy of the training phase of the GoogleNet model

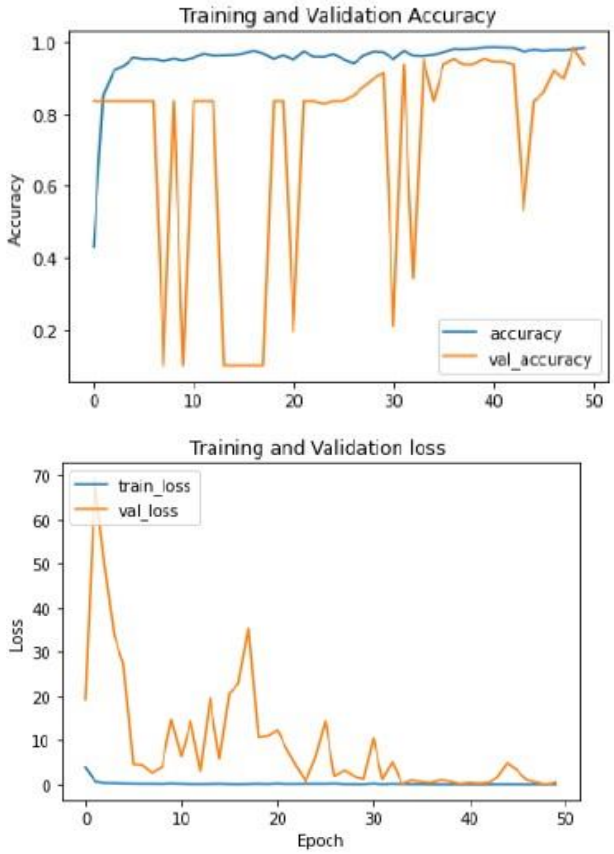

Figure 8 Training loss curve of GoogleNet model

The architecture has designed a simple sequential VGGNet model as shown in Figure 8. The first layer of the model is a convolutional layer with 8 filters. The input shape for RGB images is defined to be (128, 128, 64). The max-pooling layer is then used with a stride parameter to decrease the spatial dimensions of the output layer. The stride parameter is a 2-tuple integer parameter, defining the convolution step along the input volumes X and Y-axis.

The stride value is usually left to the default value of (1, 1) but is increased to (4, 2) in this case to ensure that the output volume size is decreased. The pooling layer is accompanied by a layer of ReLU activation. Another layer of convolution, pooling, and activation layer with improved convolution filter criteria is used, followed by another layer of convolution and activation layer. Flatten layers and 0.5 dropout layers are then fed to the output of the activation layer.

The next layer is a dense layer input that is fully connected and is from the dropout layer. In the end, to interpret the features, a dense layer and activation layer are used, the input features of the machine learning model are this output. The model was compiled using the multiclass cross-entropy loss function of the Adam optimizer and categorical cross-entropy.

The summary statistics for the model are shown in Figure 9 below.

Model: "sequential_1"

Layer (type) Output Shape Param #
conv2d_48 (Conv2D) (None, 128, 128, 64) 1792

activation_50 (Activation) (None, 128, 128, 64) 0

batch_normalization_49 (Batc (None, 128, 128, 64) 256

conv2d_49 (Conv2D) (None, 128, 128, 64) 36928

activation_51 (Activation) (None, 128, 128, 64) 0

batch_normalization_50 (Batc (None, 128, 128, 64) 256
flatten_2 (Flatten) (None, 8192) 0

dense_4 (Dense) (None, 4096) 33558528

activation_63 (Activation) (None, 4096) 0

batch_normalization_62 (Batc (None, 4096) 16384

dropout_7 (Dropout) (None, 4096) 0

dense_5 (Dense) (None, 4096) 16781312

activation_64 (Activation) (None, 4096) 0

batch_normalization_63 (Batc (None, 4096) 16384

dropout_8 (Dropout) (None, 4096) 0

dense_6 (Dense) (None, 3) 12291

activation_65 (Activation) (None, 3) 0
Total params: 65,116,483
Trainable params: 65,091,651
Non-trainable params: 24,832

Figure 9 VGGNet Model Summary

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 04, April-2022

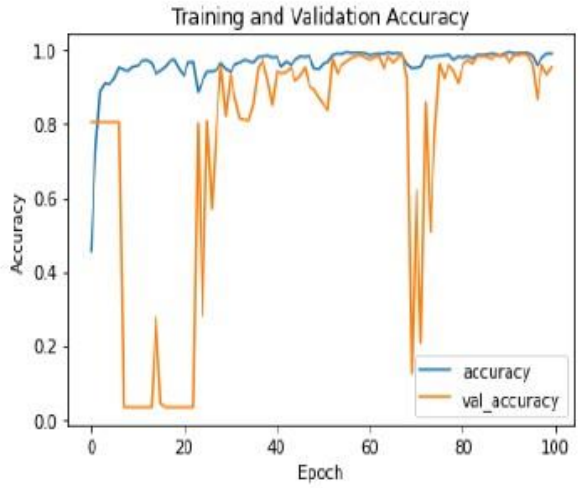

Figure 10 Training accuracy curve of VGGNet model

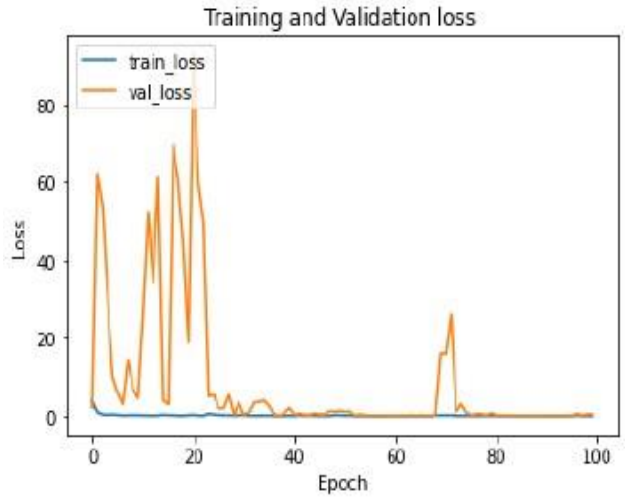

Figure 11 Training loss curve of VGGNet model

The accuracy and loss of the VGGNet model with our dataset are shown in Figures 10 and 11 above. As clearly shown in the training loss and the accuracy curve shown in the figure below, the training accuracy was higher than loss accuracy throughout the curve it is shown when the number of epochs is increased, the accuracy of the model is high and the loss of the model also decreases but it's getting better accuracy to compare to the other two models.

5.4.3 *Comparative Summary of evaluation metrics of model*

TABLE 2. SUMMARY OF COMPARISON MODELS

Models	Training accuracy	Testing accuracy	Numbers of Parameters
AlexNet	99.64	85.809	30,020,355
GoogleNet	98.43	98.340	2,176,975
VGGNet	99.40	95.380	65,116,483

classes	precision	recall	f1-score	support
geez	1.00	0.10	0.18	10
kume	0.85	1.00	0.92	244
wurid	1.00	0.31	0.47	49
accuracy			0.86	303
macro avg	0.95	0.47	0.52	303
weighted avg	0.88	0.86	0.82	303

Figure 12 Comparative Summary of AlexNet model

	precision	recall	f1-score	support
geez	0.00	0.00	0.00	8
kume	1.00	1.00	1.00	107
wurid	0.62	1.00	0.76	13
accuracy			0.94	128
macro avg	0.54	0.67	0.59	128
weighted avg	0.90	0.94	0.91	128

Figure 13 Comparative Summary of ResNet model

class	precision	recall	f1-score	support
geez	0.67	0.20	0.31	10
kume	0.97	0.99	0.98	244
wurid	0.90	0.94	0.92	49
accuracy			0.95	303
macro avg	0.85	0.71	0.74	303
weighted avg	0.95	0.95	0.95	303

Figure 14 Comparative Summary of VGGNet model

Micro average, Macro average, precision, recall, and F1 score of all the models can be compared to decide the best classification model out of three models designed as part of this research experiment. In the micro-average method, the individual true positives, false positives, and false negatives of the systems for different sets are summed and averaged to get the stats. As our research work is a case of multiclass label classification, the measurements for precision, recall, and F1-score for a particular will be different.

Figure 12, shows that the precision, recall, and F1 score of the AlexNet model are higher in comparison to other models. Thus, it can be concluded that the AlexNet model built using a set of multiple convolutions and max-pooling performs the task of both the feature extraction and classification accurately for Ge'ez Reading Level classification.

5.4.4 *Summary of comparison models*

The GoogleNet model, which has more than 2 million parameters, is much smaller than the AlexNet model, which has more than 30 million parameters, and the VGGNet model, which has more than 65 million parameters. Hence, GoogleNet learns 14x fewer parameters than AlexNet and 31x fewer parameters than VGGNet, while being more accurate. The large size of AlexNet makes it more efficient, especially for computational resources such as memory use.

5.4.5 *Comparision of confusion for each model*

A confusion matrix is a tabular visualization of statistical classification per class. Each row value represents the predicted class value, whereas the column value represents the actual value of the class. The diagonal entries are the number of correctly classified instances while all other values represent entries that are unclassified [33]. The confusion matrix for the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 04, April-2022

AlexNet, GoogleNet, and VGGNet models are shown in Figures 15, 16, and 17 showing stats better than other models classifying due to few records and missing the other ones.

```
confusion matrix
[[  1   9   0]
 [  0 244   0]
 [  0  34  15]]
```

Figure 15. Confusion Matrix of AlexNet Model

```
confusion_matrix
[[  0   0   8]
 [  0 107   0]
 [  0   0  13]]
```

Figure 16. Confusion Matrix of ResNet Model

```
confusion matrix
[[  2   6   2]
 [  0 241   3]
 [  1   2  46]]
```

Figure 17. Confusion Matrix of VGGNet Model

5.4.6 Summary

Generally, the data used for this study was collected from Ethiopian Orthodox Tewahedo church scholars, especially Nibab bete scholars. We also collected the data from Ethiopian Orthodox Tewahedo church websites. The total number of data taken for this study was 1701 with equal size splitting to 20 seconds. The model takes the converted data in image form. We used around 1701 images generated from the audio data and split those data as training and testing with the size of 80% and 20% respectively. To implement the coding part, we have used the python anaconda software with TensorFlow and Keras as backend and several libraries were imported. Specifically, Libros was used for audio files since we applied audio signal processing.

6. CONCLUSION

The proposed system has five components: data acquisition, preprocessing, segmentation, feature extraction, and classification. The classification model was trained on a 1701 dataset collected from different sources. We split the dataset into 80% for training and 20% for testing. The proposed system is implemented using Python Anaconda and tested using a sample spectrogram image dataset. The performance of the models was evaluated and compared with existing algorithms. Also, a comparative analysis was done by measuring the classification performance of the algorithm in terms of precision; accuracy, and F1 score as well as micro and macro average.

The evaluation result shows that the proposed system gives an effective classification for the Ge'ez reading Level. The system has learned a 99.43% training accuracy and 99.34% testing accuracy in Ge'ez reading Level classification. This research work explores different areas that can be further improved by Increasing the dataset to get better performance in future work.

REFERENCES

[1] Y. Abate, Morphological Analysis of Ge'ez Verbs Using Memory-Based Learning, ADDIS ABABA, 2014.
[2] N. Wakijera, Mahitote Tibebe Zelisane Geez, Addis Ababa: Maheber kidusan, 2015.
[3] Y. L. a. D. Ruinskiy, A Decision-Tree-Based Algorithm for Speech/Music, Israeli: Hindawi Publishing Corporation, 2009.
[4] Y. Sheferaw, "Metsehaf Geez Wesewasew Meriho Metsehafit, Bahirdar: st. Giorigis matemia bet," 2005.
[5] D. Kelebe, Geez Hiyaw Lisan bekelale zedie, Addis Ababa, 2001.
[6] R. V. M. D. N. T. N. TIAGO CARNEIRO, Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications, Brazil, 2017.
[7] S. Mohammad M. Soltani, "Evaluating the Performance of Convolutional Neural Network for Classifying Equipment on," in International Symposium on Automation and Robotic, Canada, 2017.
[8] M. P. a. R. S. Shanqing Gu, "Improve Image Classification Using Data Augmentation and Neural Networks," Dallas, 2019.
[9] K. A. S. S. A. N. S. H. V. K. S. Snigdha Chillara, "Music Genre Classification using Machine Learning Algorithms: A comparison," International Research Journal of Engineering and Technology (IRJET), vol. 06, no. 05, 2019.
[10] M. A. A. &. Z. A. Siddiqui, "Automatic Music Genres Classification using Machine Learning," International Journal of Advanced Computer Science and Applications, vol. 8, 2017.
[11] D. A. Siddhant C. Joshi, "MATLAB Based Feature Extraction Using Mel Frequency Cepstrum Coefficients for Automatic Speech Recognition," International Journal of Science, Engineering and Technology Research, vol. 3, no. 6, 2014.
[12] Y. Sheferaw, Metsehaf Geez Wesewasew Meriho Metsehafit, Bahirdar: st. Giorigis matemia bet, 2005.
[13] Y. L. a. D. Ruinskiy, A Decision-Tree-Based Algorithm for Speech/Music, vol. Volume 2009, Hindawi Publishing Corporation, 2009.
[14] R. C. D. Y. C. R. C. Mohammed Hameed Al-Darkazali, "Defining properties of speech spectrogram images to allow effective preprocessing before pattern recognition," in SPIE, University of Sussex, April 2013.
[15] C. N. S. J. &. A. L. Koerich, " Feature Selection in Automatic Music Genre Classification," in Tenth IEEE International Symposium on Multimedia, 2009.
[16] C. S. A. F. John Cast, Music Genre Classification, Autumn, 2014.
[17] Y.-H. P. Aziz Nasridinov, "A Study on Music Genre Recognition and Classification Techniques.," International Journal of Multimedia and Ubiquitous Engineering, vol. 9, no. 4, pp. 1-12, 2014.
[18] C. W. F. L. Boqing Zhu, Learning Environmental Sounds with Multi-scale Convolutional Neural Network, China, Mar 2018.
[19] J. A. N. R. Abdul Malik Badshah, "Speech Emotion Recognition from Spectrograms with Deep Convolutional Neural Network," in International Conference on Platform Technology and Service, 2011.
[20] A. Khamparia, D. Gupta, N. G. Nguyen, A. Khanna, B. Pandey, and P. Tiwari, "Sound Classification Using Convolutional Neural Network and Tensor Deep Stacking Network," 2019.
[21] D. D. Jasleen, "Feature Selection and Extraction of Audio," International Journal of Innovative Research in Science, Engineering and Technology, vol. 5, no. 3, pp. 1-8, 2016.
[22] B. E. &. Aydilek, " Music Emotion Recognition by Using Chroma Spectrogram and Deep Visual Features," International Journal of Computational Intelligence Systems, vol. 12, 2019.
[23] A. G. J. &. B. H. Iswanto, " Indonesian‟s Traditional Music Clustering Based on Audio Features," 2nd International Conference on Computer Science and Computational Intelligence, 2017.
[24] P. Karachi, "Automatic Music Genres Classification using Machine Learning," International Journal of Advanced Computer Science and Applications, vol. 8, no. 8, 2017.
[25] K. &. Mala, "CONTENT BASED AUDIO CLASSIFIER & FEATURE EXTRACTION USING ANN TECHNIQUES," International Journal of Innovative Research in Advanced Engineering, pp. 106-116, 2018.
[26] A. R. B. Badshah, Speech Emotion Recognition from Spectrograms with Deep Convolutional Neural Network, 2019.
[27] I. B. A. Mehmet Bilal Er*, "Music Emotion Recognition by Using Chroma Spectrogram and Deep Visual Features," International Journal of Computational Intelligence Systems, vol. 12, no. 2, 2019.