QUR’AN RECITATION CORRECTION SYSTEM USING DEEPSPEECH

Hajon Mahdy Mahmudin¹, Habibullah Akbar²
Faculty of Computer Science, Universitas Esa Unggul, West Jakarta, DKI Jakarta, Indonesia
¹hajonmahdy@student.esaunggul.ac.id ²habibullah.akbar@esaunggul.ac.id

ARTICLE INFO

Published: August 31st, 2023

Keywords: Al-Qur'an reading correction system, Deepspeech, MFCC features

This work is licensed under CC BY-SA 4.0

ABSTRACT

The purpose of this study was to compare the performance of the two types of models used in the task of classifying Quran verses based on audio similarity. The first model is Model B which uses MFCC features and the MaLSTM architecture, while the second model is Model C, which is Model B with additional delta features. The stages in this study consist of determining the dataset, determining the parameters, preprocessing, training, and testing. The dataset in this study was obtained from the local dataset https://sahabatibadah.com/fasih/. This study conducted data analysis based on 172,895 samples of Al-Qur'an recitation sounds from Juzz 30, which includes a total of 37 surahs with 564 verses. This sound data were taken from the recording on the Qara'a application and collected from 500 users of the application. In this study, 3 out of 500 users were used as training data to train speech recognition models, while one user was used as testing data. The training model used was DeepSpeech supported by TensorFlow. In the model training process, 30% of the samples were used as a validation set. Based on the results, Model B with the MFCC feature is the best model in the task of recognizing and classifying audio-based Quran verses. The use of the delta feature in Model B and Model C show a negative impact on model performance. The MFCC feature is more recommended in the recognition and classification of audio-based Qur’an verses, especially in the LSTM model architecture.

INTRODUCTION

The development of the digital world is currently accelerating. The era of industrial revolution has now entered industry 4.0. The era of the industrial revolution is characterized by information technology circulating increasingly diverse and increasingly sophisticated. Information technology is used for various activities, one of which is learning the Qur’an. The Qur'an is a book of divine guidance and guidance for mankind. The Qur'an contains the main religious text for Muslims which is word for word from Allah (Zakariah et al., 2017).

In reading the Qur'an, tahsin learning is needed, which is to improve and beautify the reading of the Qur'an so that the spoken reading is in accordance with the reading rules of the Prophet Muhammad (Arsyad & Rahman, 2022). Reading in accordance with the rules is reading by issuing letters that fulfill their makhraj properties and paying attention to the laws of tajweed reading (Arsyad & Rahman, 2022). The theory to learn tahsin is fardhu 'ain or obligatory, this has been stated in QS. Al-Muzammil verse 4.

Artificial intelligence technology is currently in vogue to be adopted by various industries. One of the utilization of artificial intelligence is speech to text or called automatic speech recognition (ASR). ASR is making machines to understand human speech. This can make the machine able to convert human speech into text (Nasib et al., 2018). The application of speech recognition in this research uses the Artificial Neural Network method. Neural Network has the
advantage of having a fast and accurate response. Neural Network does not require complex processing in feature matching (Siam, et al., 2021). The types of Artificial Neural Network used are Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM).

Recurrent Neural Network is a feed forward network that uses internal state to form a memory, which makes RNN not only capable of mapping inputs to outputs, but also mapping all previous inputs to each output (El-Moneim et al., 2019). Long Short Term Memory is one type of Neural Network that can learn sequence dependency in a prediction problem. LSTM contains three gates namely input gate, forget gate and output gate (Shashidhar et al., 2022). Researchers utilize these two methods for the correction of Qur’anic recitation. With this research, it is hoped that it can help to improve Al-Qur’an learning for Muslims in Indonesia. Where Muslims in Indonesia can read the Qur’an in accordance with its rules (tahsin) while applying tahfidz. Therefore, the author chose the "Quran Reading Correction System Using DeepSpeech".

From the background description above, the following problems can be formulated: (1) How to compare the performance of two types of models, namely Model B with MFCC features and MaLSTM architecture, and Model C which is Model B with additional delta features, in the task of classifying Quranic verses based on audio similarity? (2) Does the use of delta features in Model C have a significant effect on model performance in predicting the similarity between two audio verses? (3) How is the model performance evaluated on the test dataset and inference set using the Precision, Recall, F1-Score, Word Error Rate, and Accuracy metrics? (4) Does Model B with MFCC features outperform the other models in the Quranic verses classification task, and is MFCC more recommended than MFSC in the context of using LSTM as the model architecture? The research is expected to give more information on the digital correction system and how it is implemented on Qur’an recitation, and become a foundation for further relevant research.

**METHOD**

There are several stages in this research, namely the first stage of determining the dataset, then determining the parameters so that the dataset can be processed properly. Then preprocessing or data preparation is carried out before the training process. Furthermore, datasets that have gone through the preprocessing stage will be trained (training) and continued with the testing stage. The final result that will be obtained is from the testing result process.

The dataset in this study was obtained from the local dataset https://sahabatibadah.com/fasih/ Researchers used 172,895 voice samples of reading Al-Quran Juzz 30 with a different number of epochs in one training. The audio used is in Arabic according to the reading of the Koran which is voiced by the male and female genders. The dataset division for this research is 75% for training or training, 15% for dev or validation or development, and 10% for testing or testing.

The parameters used in this study are listed below.
Table 1. Research Parameters

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Parameter Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of input neurons</td>
<td>26 neurons</td>
</tr>
<tr>
<td>Number of hidden layers</td>
<td>5 hidden layers</td>
</tr>
<tr>
<td>Number of output neurons</td>
<td>1</td>
</tr>
<tr>
<td>Number of epochs</td>
<td>30 epochs</td>
</tr>
<tr>
<td>Total batch size</td>
<td>32</td>
</tr>
<tr>
<td>Number of timesteps</td>
<td>16</td>
</tr>
<tr>
<td>Number of learning rates</td>
<td>0.00001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Activation function</th>
<th>Output layer: Softmax</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hidden layer: RELU</td>
</tr>
</tbody>
</table>

After determining the dataset and parameters, pre-processing or data preparation is carried out. Pre-processing consists of several stages, namely:

1) Normalization: Performed by dividing the data in the speech signal by the maximum amplitude value to equalize the amplitude of the speech signal, as the speech signal recording process may have different intensities and consequently, different amplitude values.

2) Silence Removal: Performed to determine the silence of the area to be removed in the voice signal.

3) Pre-emphasis: It amplifies the higher frequencies in the input sound signal and increases the magnitude in the frequency spectrum as high frequencies tend to have small magnitudes compared to lower ones. To perform pre-emphasis using equation (3.1).

\[ y = x(t) - \alpha x(t - 1) \]  

\[ y = \text{output of the sound signal} \]
\[ x = \text{input signal} \]
\[ \alpha = \text{coefficient between 0.95 - 0.97} \]

4) Feature Extraction: The research performs feature extraction using MFCCs (Mel-Frequency Cepstral Coefficients) in the Tensorflow library. Feature Extraction plays an important role because it affects the performance improvement of speech recognition systems. Mel Cepstral Coefficients play a role in determining which one is more suitable as input for the RNN-LSTM model.

Datasets pre-processed were then used for the training process with DeepSpeech models that use Recurrent Neural Networks and Long Short Term Memory networks whose parameters have been determined. The training stage uses 70% training data and 15% dev data. The model is trained to digest speech spectrograms and generate English text transcriptions. One speech \( x \) and label \( y \) is sampled from the training set (equation 3.2).
\[ S = \{(x(1), y(1)), (x(2), y(2)), \ldots \} \] (3.2)

Every utterance, \( x(i) \) is a time series of length \( T(i) \) where each time slice is an audio feature vector, \( x(i) \) where \( t = 1, \ldots, T(i) \). The author uses MFCC as for feature extraction, so that \( x(i)_p \) denotes the \( p \)th MFCC feature in the audio frame at time \( t \). The purpose of DeepSpeech is to convert the input sequence \( x \) into a sequence of character probabilities for transcription \( y \) with \( y^t = P(ct | x) \) where for English \( ct \in \{a, b, c, \ldots, z, space, apostrophe, blank\} \).

The model used in this research consists of 5 hidden layers. For input \( x \), hidden neuron in the layer \( l \) is denoted \( h(l) \), then \( h(0) \) is the input neuron. The first three layers are non-recurrent layers. Where for the first layer, at each time \( t \), the output depends on the frame MFCC \( x_t \) along with the frame context \( C \) on each side. We used \( C=9 \) for this study. The remaining layers operate on independent data for each time step. Thus, for each time \( t \), the first 3 layers are calculated by equation 3.3.

\[
h(l)_t = g(W(l)h(l-1)_t + b(l))
\] (3.3)

Where \( g(z) = \min\{\max\{0, z\}, 20\} \) is the ReLu activation function and \( W(l), b(l) \) are the weight matrix and bias parameters for layer \( l \). The fourth layer is the recurrent layer. This layer includes a set of hidden neurons with forward recurrence written in equation 3.4. It is known that \( h(f) \) must be calculated sequentially from \( t = 1 \) to \( t = T(i) \) for the \( i \)-th utterance.

\[
h(f)_t = g(W(4)h(3)_t + W(f)r(h(f))_{t-1} + b(4))
\] (3.4)

Then the fifth layer (non-recurrent) takes the forward unit as input whose calculation is written in equation 3.5.

\[
h(5) = g(W(5)h(f) + b(5))
\] (3.5)

Then the last one is the output layer. The output layer is a standardized logit that corresponds to the probability of the predicted character for each slice of time \( t \) and character \( k \) in the alphabet. The calculation is written in equation 3.6.

\[
h(6)_t, k = y^t, k = (W(6)h(5)_t)k + b(6)_k
\] (3.6)

\( b(6)_k \) denotes the \( k \)th bias and \( (W(6)h(5)_t)_k \) the \( k \)th element of the matrix product.

After going through the training process, a testing process is carried out using 15\% testing data and 15\% dev data. In the testing process, decoding is carried out for the labeling process of the LSTM output. After going through the decoding stage. The model will be evaluated using WER (Word Error Rate) and CER (Character Error Rate). WER is how accurately Deep Speech
can recognize a word, and is generally a measure of how well the language model (scorer) operates. Word Error Rate calculation uses equation (3.7).

\[
WER = \frac{X + Y + Z}{P} \times 100\%
\]  

(3.7)

X = number of incorrectly replaced words
Y = number of incorrectly deleted words
Z = number of incorrectly entered words
P = the number of all words in the test set.

Then the CER (Character Error Rate) is calculated. CER is how accurately DeepSpeech can recognize characters, and a measure of how well the acoustic model operates, along with the alphabet file. Character Error Rate calculation using equation 3.8

\[
CER = \frac{S + D + I}{S + D + C}
\]  

(3.8)

S = number of incorrectly replaced char
D = number of incorrectly deleted char
I = number of incorrectly entered char
C = number of all correct char

Training Server Specifications
1) Processor: AMD Ryzen 5 3600 6-core processor × 12
2) Operating system: Ubuntu 18.04
3) RAM: 8.4 GB
4) Storage: 250 GB SSD & 1 TB HDD
5) GPU: NVIDIA GeForce RTX 2060 with 6 GB VRAM
6) Software: Python 3.6, Tensorflow 1.15.4, CUDA 10.0, cuDNN 7.4

Test Scenario
In the training and testing scenarios, we evaluated the model using validation sets to find optimal parameters, such as network architecture, learning rate, and number of training iterations. We only used datasets with MFCC (Mel-Frequency Cepstral Coefficients) features in an attempt to improve the performance of the model.

After identifying the best parameters in the validation set, researchers proceeded to train the model using the entire training dataset, including the validation set. Next, the model was evaluated using two independent sets, namely the test set and the inference set.

In the test set, we used an appropriate evaluation metric to measure the performance of the model. In the context of data class imbalance, such as the number of pairs with label 0 being more dominant than pairs with label 1, we chose to use F1-score as the primary evaluation metric. F1-score provides a holistic picture of model performance by considering both Precision and Recall.
The use of F1-score is considered to be more accurate in measuring model performance on minority classes.

In addition to the F1-score, researchers also looked at the Precision value as an additional relevant metric. Precision plays an important role in avoiding false positive results, where the model misclassifies records that are actually false as true.

In addition to the F1-score evaluation metric, researchers also take into account the Word Error Rate (WER) and Accuracy in evaluating the performance of the Qur'an recitation recording speech recognition model. WER is used to measure the error rate in recognizing correctly spoken text. The lower the WER value, the better the model's performance in speech recognition and interpretation of al-Qur'an recitation text.

Meanwhile, Accuracy is also one of the evaluation metrics taken into account. This metric measures the extent to which the model provides correct results compared to the total assessed sample. Although accuracy is important, researchers realize that this metric can be less informative in the context of data class imbalance, such as in the Qur'an recitation recording speech recognition task.

RESULT AND DISCUSSION

Data Analysis

This research analyzes data based on 172,895 Quran recitation sound samples from Juzz 30, which includes a total of 37 surahs with 564 verses. This voice data was taken from recordings on the Qara'a application and collected from 500 users of the application. In this study, 3 out of 500 users were used as training data to train the speech recognition model, while one user was used as testing data.

The training model used in this research is DeepSpeech powered by TensorFlow. DeepSpeech is a deep learning-based machine learning method that is effective in recognizing voice patterns and converting them into text that can be understood by computers. TensorFlow as a machine learning framework supports and facilitates the training and evaluation of DeepSpeech models.

In the model training process, 30% of all samples are used as a validation set. The use of this validation set aims to prevent overfitting of the model, where the model is too complex and is able to memorize the training data well, but its performance decreases when faced with new data. By splitting the validation set, the model can be optimized and tested on data that has never been seen before, resulting in a more generalized model.

To prevent evaluation bias, researchers only use the test set after obtaining the best model from the training process. This approach involves separating the training data from the testing data, so that the model is evaluated on data it has never faced before. This maintains the integrity of the evaluation and prevents information leakage that could affect the final results. Thus, this study uses a careful and objective methodology to face the challenges in Quran recitation recognition through voice technology and makes a valuable contribution to the development of voice applications in the field of Quran learning.
In the initial stage, the researcher merged verse pairs by assigning label 1 to the same verse pair and label 0 to different verse pairs. However, it was recognized that the labeling resulted in an unbalanced class distribution, where the number of verse pairs with label 0 was more dominant than verse pairs with label 1. The researcher realized that training the model with unbalanced classes could improve the robustness of the model in discriminating the correct answers.

To overcome the problem of class imbalance, researchers implemented downsampling on the majority class, namely the class with label 0. The downsampling technique is performed by randomly selecting a number of samples from the majority class so that the amount of data in training and testing becomes balanced. Thus, the distribution of data between the two classes becomes more balanced and the model can better recognize matches between pairs of Qur'anic verses. To avoid the gap between the two classes, we downsampld the majority class by taking random samples from the data.

Hyperparameter Search

During the training process, we evaluated the model using the validation set to select five architectures that had shown satisfactory performance. The selected architectures can be seen in Figure 1. Figure 1 also shows whether the model uses additional layers as indicated and marked in the "Additional Layer" column. The researcher also recorded the values of learning rate, number of epochs, and threshold used in training, which are documented in Figure 2.

<table>
<thead>
<tr>
<th>Model</th>
<th>Type Model</th>
<th>LSTM Hidden Neuron</th>
<th>LSTM Layers</th>
<th>Additional Layer (Yes/No)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>Siamese Classifier</td>
<td>128</td>
<td>2</td>
<td>Yes</td>
</tr>
<tr>
<td>b</td>
<td>MaLSTM</td>
<td>64</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td>c</td>
<td>Siamese Classifier</td>
<td>256</td>
<td>3</td>
<td>Yes</td>
</tr>
<tr>
<td>d</td>
<td>MaLSTM</td>
<td>128</td>
<td>2</td>
<td>Yes</td>
</tr>
<tr>
<td>e</td>
<td>Siamese Classifier</td>
<td>128</td>
<td>2</td>
<td>No</td>
</tr>
</tbody>
</table>

**Figure 1. Model Architecture**

<table>
<thead>
<tr>
<th>Model</th>
<th>Jumlah Epoch</th>
<th>Learning Rate</th>
<th>Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>50</td>
<td>0.001</td>
<td>0.5</td>
</tr>
<tr>
<td>b</td>
<td>30</td>
<td>0.010</td>
<td>0.6</td>
</tr>
<tr>
<td>c</td>
<td>100</td>
<td>0.005</td>
<td>0.4</td>
</tr>
<tr>
<td>d</td>
<td>40</td>
<td>0.001</td>
<td>0.7</td>
</tr>
<tr>
<td>e</td>
<td>70</td>
<td>0.010</td>
<td>0.6</td>
</tr>
</tbody>
</table>

**Figure 2. Model Architecture**

We chose the number of epochs when the validation loss value no longer decreased or started to increase (overfitting) during the training process. The MaLSTM model has an additional hyperparameter, namely the threshold value, which serves to determine whether two input data are considered the same. Although the MaLSTM model has some drawbacks, such as requiring the process of finding the optimal threshold value, researchers acknowledge that this method provides the flexibility to determine the threshold value according to the characteristics of the data, model, and other hyperparameters. On the other hand, the Siamese-Classifier model does not require a
**Qur’an Recitation Correction System Using Deepspeech**

threshold determination process as it directly classifies whether two audio files read the same verse.

**Test Data Evaluation**

After the training process, the five models with the best hyperparameters were evaluated using test data. The results obtained are shown in Figure 3, indicating that Model B has the best performance in F1-score compared to the other four models.

![Figure 3. Test Data Model Performance](image)

Based on the above results, the researcher found that Model B, which is the original MaLSTM model, showed the best performance in predicting the similarity between two audio verses. Model B successfully achieved a high F1-score value, which indicates a balance between Recall and Precision, which is important to ensure the precision and accuracy of the model in classifying two verses as similar or different. In addition to its good performance in similarity classification between two audio verses, Model B also showed satisfactory results in terms of Word Error Rate (WER) and accuracy. With a WER value of 0.19, Model B is able to provide fairly accurate predictions with a relatively low error rate. In addition, the accuracy of Model B reaches 84%, indicating that it can correctly classify most of the samples.

**Comparison Test Results**

Researchers have compared the performance of models on the dataset using MFSC features. The results showed that the best model using MFSC features, Model B, outperformed all models with MFSC features. While it has been reported that the use of MFSC features can help improve model performance as they preserve the local features of the audio data, we suspect that this depends on the architecture used in the study. In this study, we used LSTM as a replacement model for the Convolutional Neural Network (CNN) used in other studies. LSTM is a model that processes data as a sequence, so it seems that MFCC features are more recommended in the context of this study.
Figure 4. Comparison between MFCC and MFSC

Figure 4 shows that Model B which uses MFCC as a feature has the highest F1-Score of 0.94, demonstrating the superiority of MFSC in capturing important characteristics of audio data to determine the similarity between verses.

<table>
<thead>
<tr>
<th>Model</th>
<th>Feature</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>MFCC</td>
<td>0.89</td>
</tr>
<tr>
<td>B</td>
<td>MFCC</td>
<td>0.94</td>
</tr>
<tr>
<td>C</td>
<td>MFCC</td>
<td>0.91</td>
</tr>
<tr>
<td>D</td>
<td>MFCC</td>
<td>0.87</td>
</tr>
<tr>
<td>E</td>
<td>MFCC</td>
<td>0.88</td>
</tr>
<tr>
<td>B</td>
<td>MFSC</td>
<td>0.92</td>
</tr>
</tbody>
</table>

Figure 5. Comparison After Adding Delta

From the evaluation results, it can be seen that the B+Delta model and the C+Delta model produce slightly higher Precision values compared to the original models (B and C) that use features without deltas. However, the Recall values for the original models are higher than the models that have added delta features. This result suggests that the original model is better at identifying true positive classes, whereas the model that uses delta features tends to have a tendency to further reduce false positives. On the other hand, the F1-Score for model B and model B+Delta both reached 0.91, indicating a good balance between Precision and Recall. However, the slight difference between the two models indicates that the addition of the delta feature does not provide a significant change in the performance of model B.

Model Size Evaluation

Based on the evaluation results in Figure 6, we can conclude that Model B, which is the original MaLSTM model, showed the best performance in classifying the verses in the inference data set. Model B achieved high Precision, Recall, and F1-Score values of 0.92, 0.93, and 0.93, respectively, demonstrating its ability to identify similar verses accurately and consistently. Model B+Δ, which utilizes the delta feature in model B, also performed well with an F1-Score of 0.89, although slightly lower than Model B.
Qur'an Recitation Correction System Using DeepSpeech

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
<th>Word Error Rate</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>0.92</td>
<td>0.93</td>
<td>0.93</td>
<td>0.07</td>
<td>93.33%</td>
</tr>
<tr>
<td>B+Δ</td>
<td>0.90</td>
<td>0.89</td>
<td>0.89</td>
<td>0.11</td>
<td>88.89%</td>
</tr>
<tr>
<td>C</td>
<td>0.88</td>
<td>0.87</td>
<td>0.87</td>
<td>0.13</td>
<td>87.14%</td>
</tr>
<tr>
<td>C+Δ</td>
<td>0.86</td>
<td>0.85</td>
<td>0.85</td>
<td>0.15</td>
<td>85.71%</td>
</tr>
</tbody>
</table>

**Figure 6. Model Performance**

Additional evaluation on Word Error Rate and Accuracy shows that Model B has the lowest Word Error Rate (0.07) and the highest Accuracy (93.33%), underscoring the superiority of this model in producing accurate and efficient predictions.

**CONCLUSION**

Model B with MFCC features is the best model in the task of recognizing and classifying audio-based Quranic verses. This model shows better performance compared to other models, by achieving the highest F1-Score of 0.93.

The use of delta features in Model B and Model C shows a negative impact on model performance. Model B+Δ and C+Δ experienced a decrease in F1-Score value to 0.89 and 0.85 respectively.

MFCC features are more recommended in the recognition and classification of audio-based Quranic verses, especially when used in the LSTM model architecture as in this study.

Although Model B has a high F1-Score value, it is important to note that the Precision and Recall values in this model have minimal differences, so it is expected to avoid many false positives and false negatives.

The results of this study provide a deeper understanding of the use of features and model architecture in the recognition and classification of audio-based Quranic verses. Recommendations from this research can help in the development of more accurate and efficient Quran recognition and understanding applications.

This research has an important contribution in the development of audio-based Quran verse recognition and classification technology. The results can be used in various contexts, such as Quran learning, recitation, and academic research.

Thus, this research makes a significant contribution in improving the understanding and application of the Quran in various fields as well as providing important guidance in feature selection and model architecture for audio-based Quranic verse recognition and classification tasks.

Based on the results of this study, there are several suggestions for further research that can increase the contribution and applicability in the recognition and classification of audio-based Quranic verses:

1) Dataset Expansion: This research can be expanded by collecting more data from various reciters and various voice recording qualities. With a larger and more representative dataset, the model can better recognize variations in the pronunciation of Quranic verses.
2) Feature Exploration: In addition to MFCC and MFSC features, future research can explore other features that have the potential to improve model performance, such as MFCC-delta and MFCC-delta-delta. Exploration of new features can provide further insight into the audio characteristics of Quranic verses.

3) Alternative Model Approaches: In addition to LSTM, alternative model approaches such as Convolutional Neural Network (CNN) or Transformer can be tested to see how they perform in Quranic verse recognition and classification tasks. Comparing different model approaches can open up opportunities to improve accuracy and efficiency in data processing.

4) Handling Speech Variations: Variation in the pronunciation of Quranic verses by the reciter is an interesting challenge in this task. Future research could consider advanced speech processing techniques or transfer learning to address these variations and improve the accuracy of the model.

5) Evaluation on Multilingual Data: This research has so far been conducted on Arabic data. Further evaluation on multilingual data, including translations of Quranic verses into other languages, may provide insight into how the model behaves in a multilingual environment.

6) Application to Practical Applications: The results of this research can be applied in the development of voice-based applications and technologies related to Quranic verses. Automatic recognition and classification of Quranic verses can be useful in various contexts such as learning, worship guidance, and automatic translation.

By adopting these suggestions, future research can make a broader contribution in strengthening the recognition and classification of audio-based Quranic verses and expanding the benefits of its application in various aspects of life.

REFERENCE


**Qur'an Recitation Correction System Using Deepspeech**


