DOI: http://dx.doi.org/10.21123/bsj.2021.18.2(Suppl.).0925

Arabic Speech Classification Method Based on Padding and Deep Learning Neural Network

Asroni Asroni[1*] *Ku Ruhana Ku-Mahamud*[2] *Cahya Damarjati*[1]
Hasan Basri Slamat[1]

[1] Universitas Muhammadiyah Yogyakarta, Indonesia.
[2] Universiti Utara Malaysia.
[*]Corresponding author: asroni@umy.ac.id, ruhana@uum.edu.my, cahya.damarjati@umy.ac.id, hasan.basri.2013@ft.umy.ac.id
[*]ORCID ID: https://orcid.org/0000-0001-9164-9128 , https://orcid.org/0000-0001-5451-0514 , https://orcid.org/0000-0003-4389-6321 , https://orcid.org/0000-0003-2496-0199

Received 28/3/2021, Accepted 12/4/2021, Published 20/6/2021

Abstract:

Deep learning convolution neural network has been widely used to recognize or classify voice. Various techniques have been used together with convolution neural network to prepare voice data before the training process in developing the classification model. However, not all model can produce good classification accuracy as there are many types of voice or speech. Classification of Arabic alphabet pronunciation is a one of the types of voice and accurate pronunciation is required in the learning of the Qur'an reading. Thus, the technique to process the pronunciation and training of the processed data requires specific approach. To overcome this issue, a method based on padding and deep learning convolution neural network is proposed to evaluate the pronunciation of the Arabic alphabet. Voice data from six school children are recorded and used to test the performance of the proposed method. The padding technique has been used to augment the voice data before feeding the data to the CNN structure to developed the classification model. In addition, three other feature extraction techniques have been introduced to enable the comparison of the proposed method which employs padding technique. The performance of the proposed method with padding technique is at par with the spectrogram but better than mel-spectrogram and mel-frequency cepstral coefficients. Results also show that the proposed method was able to distinguish the Arabic alphabets that are difficult to pronounce. The proposed method with padding technique may be extended to address other voice pronunciation ability other than the Arabic alphabets.

Key words: Arabic alphabet, COVID-19, Deep learning, Spectrogram, Speech classification.

Introduction:

Voice recognition is one of the areas of digital signal patterns that are used to recognize a specific word. One of the most popular machine learning techniques for pattern recognition and classification is the convolution neural network (CNN) (1-6). One of the most commonly used feature extraction techniques for voice recognition is the mel-frequency cepstral coefficients (MFCC) (7-10). Extracted features are fed into the CNN to produce the voice recognition model.

Taking advantage of the advanced in technology and online system, the performance of voice recognition is of utmost important especially when security is concern. Online teaching has also been very popular especially when face-to-face interaction is not possible. In the era of COVID-19 pandemic, education activities have to be implemented where students have to interact with teachers through online learning. However, the online learning may create obstacles for teachers to measure students' ability in pronunciation, especially, the Arabic alphabet in the case of Qur'an education. It is difficult for teachers to check the quality of the reading and switch to the video call channel, due to the waiting time of the long queue, and the sound quality influenced by the device and the Internet. Arabic alphabets have complex diversity, wealth, and morphology to represent text categorization and grouping (11). Arabic is a sematic language and one of the oldest languages in the world. It is the fifth language widely used today. Standard Arabic has 35 pharyngealized (*L*) basics which is rarely used, and six vocals, where three are long, and another three are short. Learning to read the Qur'an is unique as correct pronunciation is required. Thus, automated evaluation of the pronunciation can facilitate the Qur'an education and this can be done through a voice classification system. Research on pronunciation of Arabic reading conducted by Adhayani and Tresnawatihas focused on competency exams, initially conducted by teachers but later using an application. In the direct instructional learning mode, the teacher will recite Arabic alphabet and students will hopefully obtain or learn the correct pronunciation (12). However, the application that has been developed is only on learning the characters but not on learning the pronunciation.

This study has focused on the learning of Arabic alphabet pronunciation of students at the elementary level at Taman Pendidikan Al-Qur'an, a school specific for teaching and learning the Qur'an where it should be read, studied, and applied in daily life. Learning to read the Qur'an begins with learning the Arabic alphabet consisting of 28 letters as the initial knowledge that must be mastered properly to get to the next level of reading the Qur'an properly and correctly. Taman Pendidikan Al-Qur'an is part of a mosque organizational structure. As the whole world faced the COVID-19 pandemic at the end of 2019, all activities including learning the Qur'an were unable to be carried out in the normal way. Thus, face-to-face learning and competency tests must be carried out online. This creates difficulties, especially at the basic level of the learning process, namely the pronunciation of the Arabic alphabet which is one of the elements that will be tested. The learning process was limited to online meetings. However, the online learning mode has several restrictions on activities. For example, testing the competency of pronunciation and reading. This has been shown in a study conducted by Anwar where the finding showed that the application of direct instruction patterns with examples recited by teachers has the purpose of effectiveness and eases students to understand comprehensively, ranging from lip, tongue, and teeth position, and suppression of voice intonation (13).

In this study, a deep learning neural network algorithm to assess students' ability to pronounce Arabic alphabet is proposed. Related studies are presented in the next section followed by the proposed method for voice classification. Results and discussion are presented in the fourth section while the conclusion and future work are presented in the final section.

Related Studies:

In the dictionary of Indonesian, the word Arabic alphabet means "Arabic script system; the Arabic alphabet". The Arabic alphabet is 28 single letters or 30 if it includes the letters *lam-alif* (لا) and *hamzah* (ء) as stand-alone letters. The way the letters are pronounced varies depending on where they come out. As for the place of exit, there are five places, namely lips (intercession), throat (*halaq*), tongue (oral), oral cavity (*jauf*), and nasal cavity *(khaisyum)*. Arabic alphabet recognition interactive learning application for primary school students has been developed by Ramansyah and Madura (14). The application presents visualizations of shape of the alphabet accompanied by audio of the way they are pronounced. The goal is to help students in understanding the subject matter in an easier and quicker way. Thus, the learning process will be more exciting and enjoyable.

A similar application was created based on augmented reality markers using a mobile device (15**.** The application presents learning using a smartphone and sets the display light level comfortable for Arabic alphabet learning. The approach to sound classification has been carried out in several cases, including to detect the voices of stroke patients. The voices were used in the classification of stroke patients (16). Other research has examined the deep learning method as a tool for neural data analysis performing speech classification and cross-frequency willing in the human sensorimotor cortex results, aiming to predict syllables resulting from high gamma cortical surface electrical potential data set recorded from the human sensorimotor cortex (17). Further research has been conducted by Tamulevičius et al. (18) to classify speech emotions using fractal dimension-based features. The research has focused on effective feature sets, complex classification schemes, and multi-modal data acquisition. The experimental results disclosed a clear advantage of fractal dimension-based feature sets against acoustic feature sets. Average accuracy of 96.5% was obtained using a reduced feature set. The feature selection approach has obtained four-dimensional and eight-dimensional sets for Lithuanian and Germans' emotions.

Convolutional neural network has been shown to improve recognition and classification accuracy of images (1-4) and environment sound (5,6). This is largely influenced by advanced in computing technologies, emergence of large data sets, and new techniques for training deeper networks. CNN's ability is claimed to be the best model in solving object detection and object recognition problems. Research on CNN was able to conduct digital image recognition with accuracy that rivalled humans on certain datasets (19).

In LeCun et al., CNN has been used to deal with the 2D shapes and results have shown that CNN has eliminate the need for hand-crafted feature extractors (1). The study has pointed out that as learning algorithm becomes easier to understand, learning process will be of utmost important to recognition systems in improving the performance. Recognition of lower and upper cases English characters using a modified LeNet-5 CNN has been performed by (2). Special settings have been included in the architecture for the number of neurons in each layer and how several layers were connected. Error-corrected codes were implemented to enable the system to reject unwanted recognition results. Performance of the systems was better for upper case characters compared to the lowercase characters. Ren et al. have introduced a CNN architecture with a cascade learning for a deeper training which leads to a more accurate result (3). The work was to solve problem of image super resolution and favourable solution was obtained with small number of network parameters. Trimming was performed to reduce the network size and a function was also introduced to obtain images with sharper edges and better image quality. Benchmark datasets were used to evaluate the performance which showed that good accuracy and faster execution time compared to other existing deep super-resolution networks. A conventional approach for image-based face detection was proposed that combines correlation filter and CNN (4). This approach creates frames which relies on all pixels' activities within a predefined time window.

Like other deep learning models, the performance of CNN can be improved if the input is pre-processed to make it suitable for training. For example, study that involves the classification of sounds where audio clips can be converted into spectrogram images. Such conversion can provide a better classification accuracy and a less error rate. This has been shown by Boddapati et al. that with the use of CNN in environmental sound or acoustic scene classification, the performance of using the spectrogram image is much better in comparison with directly using audio files (20).

Mustaqeen, Sajjad and Kwon has proposed a method in which a short-time Fourier transform algorithm has been used to transfer speech into spectrogram before passing it to CNN for discriminative and salient feature extraction (21). The proposed method was able to produce better recognition accuracy compared to state-of-art speech emotion recognition methods when evaluated on three benchmark datasets. Furthermore, the research was able to proof that the computation cost and processing time have also been reduced. A two-dimensional deep CNN was used by Huang et al. to perform the classification of electrocardiogram arrhythmia data in the form of time domain signals (22). Short-time Fourier transform algorithm was again utilized to transform the signals into time-frequency spectrogram. Superior result was obtained from the proposed two-dimension CNN when compared a one-dimension CNN.

Padding is another technique to prepare data for training where it guarantees all inputs will have the same dimension. Semantic-based padding approach has been used in CNN to improve the performance in natural language processing and evaluation in sentiment analysis showed that classification accuracy is better compared to when padding is not used (23). In their study, two sentiment analysis datasets and seven different word embeddings were used to show the superiority of the proposed approach. However, their study do not focus on voice data.

Data augmentation is aimed to enhance the classification accuracy of environmental sound in CNN models (5, 6). Mushtag and Suinvestigated the performance of two deep CNN with max-pooling and without max-pooling functions on environmental sound data (5). The audio attribute extraction techniques to produce spectrogram images are mel-spectrogram, MFCC and log-mel. The best accuracy was attained by the deep CNN without max-pooling and using log-mel audio feature extraction. The models used in Mushtag, Su and Tran are deep CNN with seven and nine layers (6). Results showed that both models produced better results when augmented data were produced using mel-spectrogram extraction technique compared to traditional extraction techniques. The MFCC technique is widely used in speech recognition since they are powerful, effective, and easy to apply. The study of extra audio features by Tun has employed the MFCC to analyze the voice and extract the attributes (7). In speech signal random signals occurred naturally and had an independent carrier signal according to the time. To extract the sound signal feature, it is essential to analyze the speech audio signal because it is beneficial in classifying the signal based on the feature. In the study, feature extraction was analyzed on wav word files spoken using the MFCC technique and implemented with MATLAB programming.

In a study by Jin et al., MFCC has been used together with CNN to predict the sound quality of automotive transmission noise (8). The architecture of the general CNN has been modified where the softmax classification layer has been substitute for the linear transform prediction layer. The performance of MFCC is better than three conventional machine learning-based methods (multiple linear regression, back propagation neural network and support vector regression). Results of the experiments show that i) different sound qualities can be distinguished and ii) high correlation quality and low mean absolute error for the predicted value. Several types of MFCC techniques have been evaluated in an analysis of speech feature extraction by Ranjan and Thakur (10). Audio signal is divided into several frames making it possible for each frame to be analyzed and synthesized without loss of information. The metrics used for comparison of the different types of MFCC are the mean and standard deviation of the frames. The researchers concluded that the delta-delta MFCC provides the best classification accuracy based on the minimum standard deviation.

Nada et al. created speech recognition system using the hidden Markov model (24). The research aimed to solve the problem of how to appropriately read Arabic alphabet with the science of *tajwid mad thobi'i* with *harakat fathah*. The research resulted in the 54.6% accuracy of the system detecting Arabic alphabet. Similar research using the MFCC and Manhattan distance methods yielded 64.29% accuracy. The study involved five teachers as experts in taking data collection. In (25) deep learning together with MFCC were used for voice recognition resulted in a high degree of accuracy. Using hierarchical concepts, deep learning becomes an approach to problem-solving in computer learning systems that can learn a complex concept by combining more straightforward concepts. The process of feature extraction from human voices has been performed using MFCC in a study conducted by (9). In their work, spoken words are converted to digital signals by converting sound waves into a set of numbers, adapted to specific codes to identify the words. Borsky et al. has conducted a study on sound quality classification using Gaussian mixture model, support vector machine, random forest, and deep neural network as the classifiers (26). Three types of voice data were used in the experiment where the recording from 28 participants with normal vocal status. These participants were prompted to sustain vowels with modal and nonmodal voice qualities. Analysis of the results showed that MFCC and dynamic MFCC were able to classify breathy, strained voice and modal from two out of three types of input data.

In summary, many studies on voice recognition or classification have employed the MFCC, mel-spectrogram, spectrogram and padding techniques for the feature extraction process while for the classifiers, CNN, multi-layer perceptron and support vector machine are among the most common.

Material and Method:

The proposed method consists of four stages as shown in Figure 1. In the first stage, voice data of six schoolchildren pronouncing the Arabic alphabet were recorded the school children are from Taman Pendidikan Al-Qur'an. The ages of the children are between five to twelve years old. Recording of each student pronouncing an Arabic alphabet letter was repeatedly performed for one minute, and this is performed for all 28 Arabic alphabets. Voice data with a length of one minute has several number of repetitions, i.e. as much as 60 times for the same pronunciation of a single alphabet. The voice is then cut into single pronunciation and labeled according to the alphabet, the children's name and its ordering. The process is repeated for all the 28 Arabic alphabets. The data for each child is then divided into training and test data in the ratio of 80:20.

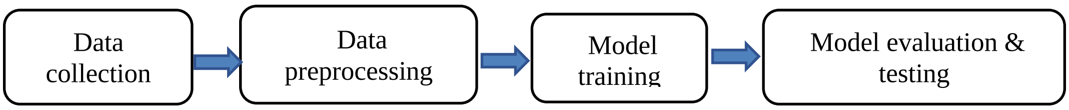


Figure 1. Research process

In the second stage, the collected voice data is pre-processed before it can be used for training to obtained the classification model. A filtering process using a trim function was performed on the voice data to eliminate sound/signals with a small intensity (soundless data) (16, 27). Thus, the frequency of the remaining voice data in the range of 512 Hz - 2048 Hz as this is frequency where the human speech that can be heard. After the filtering process, the data is then enhanced or augmented using the padding technique (28). Dimension setting is performed to the images so that data is suitable for the duration of the time and frequency.

The model development stage starts after the completion of the preprocessing stage, where CNN is used to train the augmented data in developing the classification model. This activity is performed in the third stage of the research process. The architecture of the CNN that has been used is adopted and adapted from (23). Figure 2 shows a one-dimensional 5-layer CNN architecture consisting of the input data, the CNN layers, two fully connected layers, softmax layer and the output. that will be processed in the several layers to generate a classification model. This is a 5-layer CNN consists of the convolution with filter variation, kernel, ReLu, padding and stride to produce compressed outer form. This is then proceed to the max pooling section, which is used to reduce the size and speed up calculations and make some output features more accurate. A fully connected layer is the part that will perform compression in the form of one layer up to the total number of 28 Arabic alphabets. Softmax is classify according to the typeface of Arabic alphabets and the output produces a classification result in the form of an Arabic alphabet.

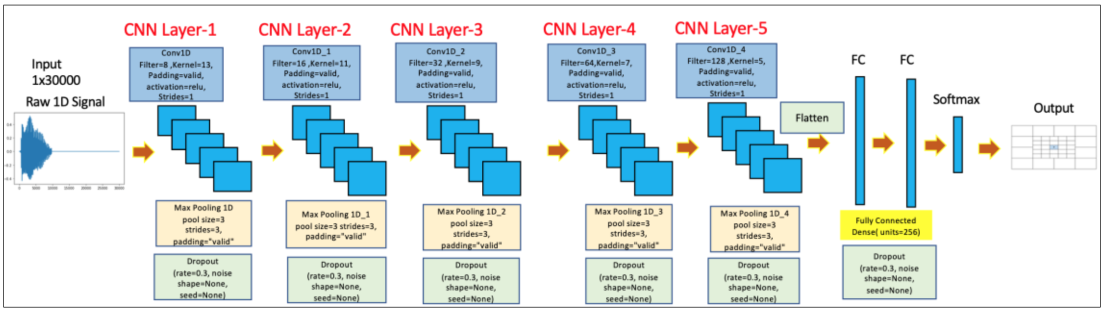


Figure 2. 1D CNN configuration with parameter adjustment

In the final stage of the research process, experiments were conducted to evaluate the performance of the classification model. A total of 3640 voice/sound data has been collected is as shown in Table 1. The ratio of the data for training (third stage) and testing (fourth stage) was 80:20 resulted in 2800 records for training and 840 records for testing. In both the training and testing data, the ratio between the male and female data is 50:50.

Table 1. Voice data

Name	Assign	Dataset Size
Zahra	Testing	840
Fani	Training	840
Bagas	Training	840
Nazma	Training	840
Gusti	Training	840
Nazwa	Training	840
Total		**3640**

In evaluating the performance of the model, the focus was classification accuracy of the pronunciation which relies on the proposed padding extraction technique in augmenting the voice data and the CNN architecture that comprises of five layers instead of three layers as implemented in (24). Experiments were also carried out on three other techniques, namely spectrogram, mel-spectrogram and MFCC.

The proposed method has been implemented as a system using the Phyton programming language. The spectrogram, mel-spectogram and MFCC techniques were also implemented in making the performance comparison of the voice recognition.

Results and Discussion:

In presenting the filtering process, the voice data of alphabet 'ب' is used as an example, as displayed in Figure 3. Trim function has been performed to remove unsound data (16, 27). The sound source to be tested is filtered with the aim that the sound can be exactly when the sound starts as shown in Figure 4.

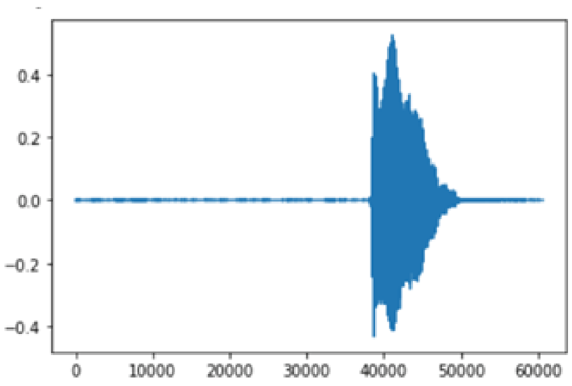


Figure 3. Wave of alphabet 'ب'

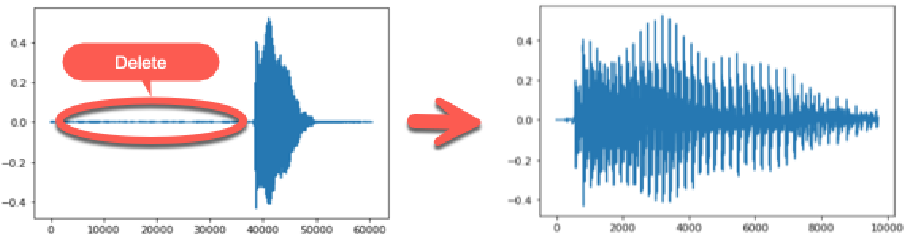


Figure 4. Filtering process of alphabet 'ب'

Results of the dimension setting are displayed in Table 2 where all techniques have the same dimension except for padding.

Table 2. Image and Data Settings

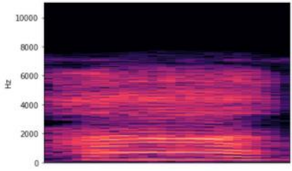
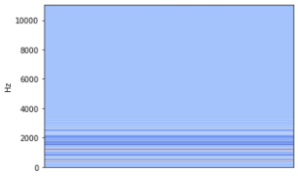
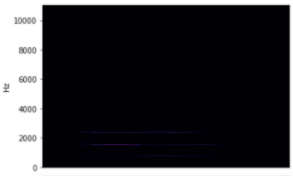
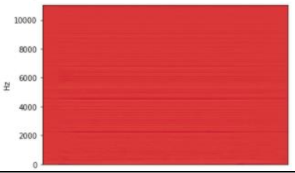
Technique	Size	Image
Spectrogram	750 x 40	
Padding	30000	
Mel-Spectrogram	750 x 40	
Mel-frequency cepstral coefficients	750 x 40	

Figure 5 displays the voice data for alphabet 'ب' that has gone through the dimension setting.

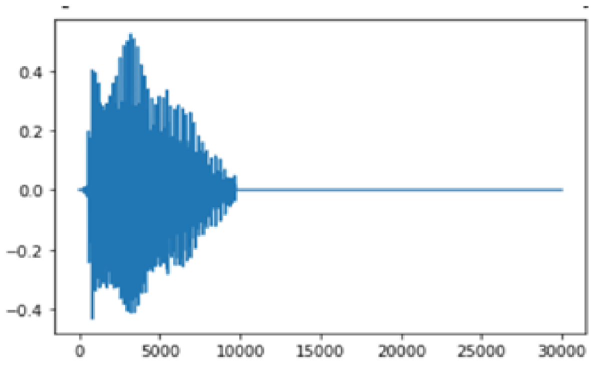


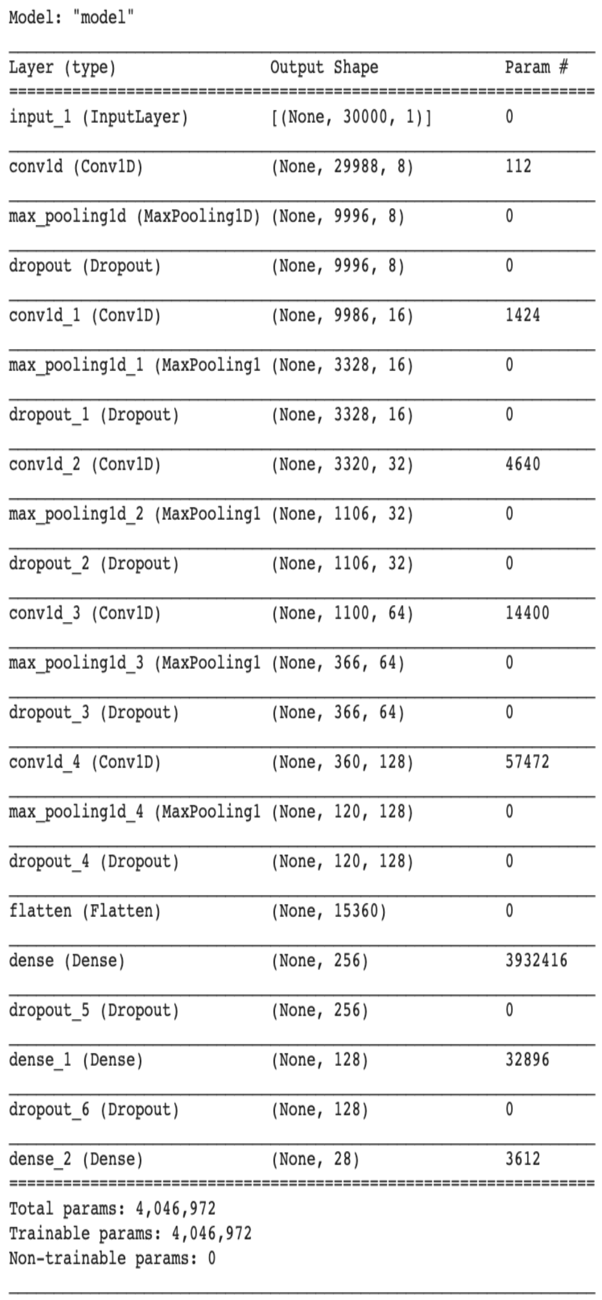
Figure 5. Dimension Adjustment of alphabet 'ب'

Table 3 shows a dimension arrangement of 2800 training data and 840 testing data combined with the 28 Arabic alphabet.

Table 3. Dimension Arrangement

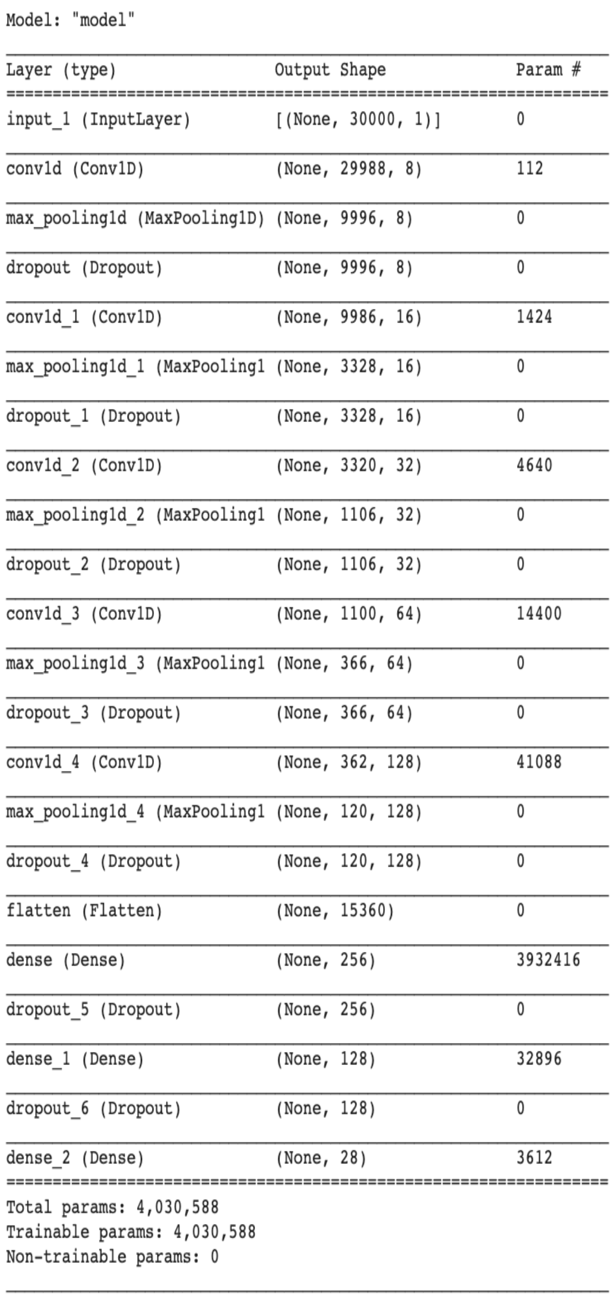
Spectrogram	**train_spectrograms: (2800, 30000)** **train_y: (2800, 28)** **test_spectrograms: (840, 30000)** **test_y: (840, 28)**
Padding	**train_X: (2800, 30000)** **train_y: (2800, 28)** **test_X: (840, 30000)** **test_y: (840, 28)**
Mel-Spectrogram	**train_mel_spectrograms: (2800, 30000** **train_y: (2800, 28)** **test_mel_spectrograms: (840, 30000)** **test_y: (840, 28)**
MFCC	**train_mfccs: (2800, 30000)** **train_y: (2800, 28)** **test_mfccs: (840, 30000)** **test_y: (840, 28)**

The results of spectrogram, padding, mel-spectrogram and MFCC on the training data are displayed in Figures 6-9 respectively.



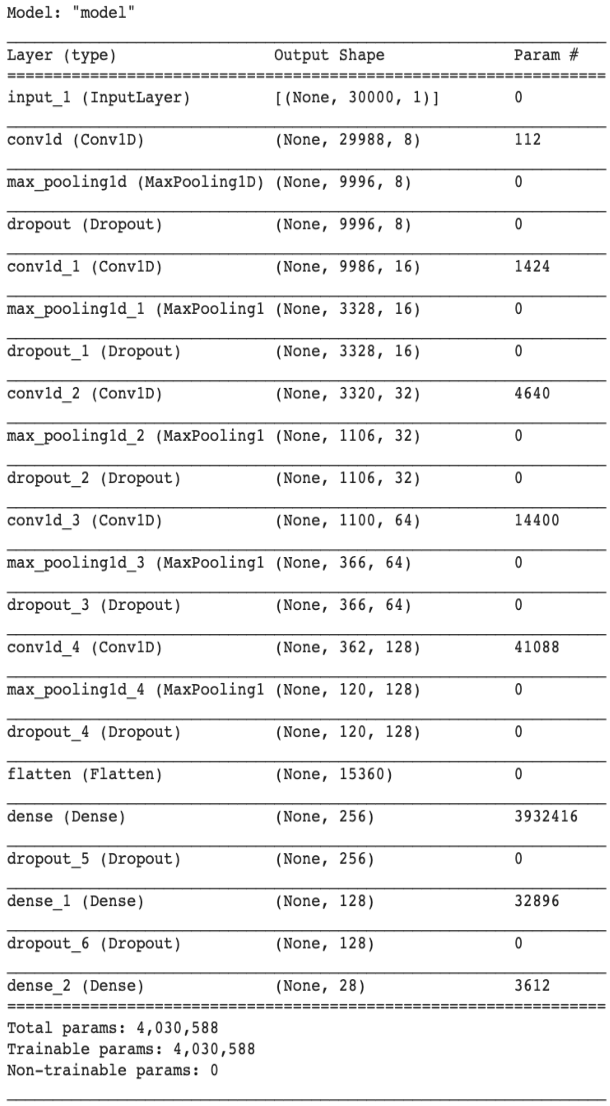
```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 30000, 1)]        0
_________________________________________________________________
conv1d (Conv1D)              (None, 29988, 8)          112
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 9996, 8)           0
_________________________________________________________________
dropout (Dropout)            (None, 9996, 8)           0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 9986, 16)          1424
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 3328, 16)          0
_________________________________________________________________
dropout_1 (Dropout)          (None, 3328, 16)          0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 3320, 32)          4640
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 1106, 32)          0
_________________________________________________________________
dropout_2 (Dropout)          (None, 1106, 32)          0
_________________________________________________________________
conv1d_3 (Conv1D)            (None, 1100, 64)          14400
_________________________________________________________________
max_pooling1d_3 (MaxPooling1 (None, 366, 64)           0
_________________________________________________________________
dropout_3 (Dropout)          (None, 366, 64)           0
_________________________________________________________________
conv1d_4 (Conv1D)            (None, 360, 128)          57472
_________________________________________________________________
max_pooling1d_4 (MaxPooling1 (None, 120, 128)          0
_________________________________________________________________
dropout_4 (Dropout)          (None, 120, 128)          0
_________________________________________________________________
flatten (Flatten)            (None, 15360)             0
_________________________________________________________________
dense (Dense)                (None, 256)               3932416
_________________________________________________________________
dropout_5 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               32896
_________________________________________________________________
dropout_6 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 28)                3612
=================================================================
Total params: 4,046,972
Trainable params: 4,046,972
Non-trainable params: 0
_________________________________________________________________
```

Figure 6. Classification Model with Spectrogram Technique



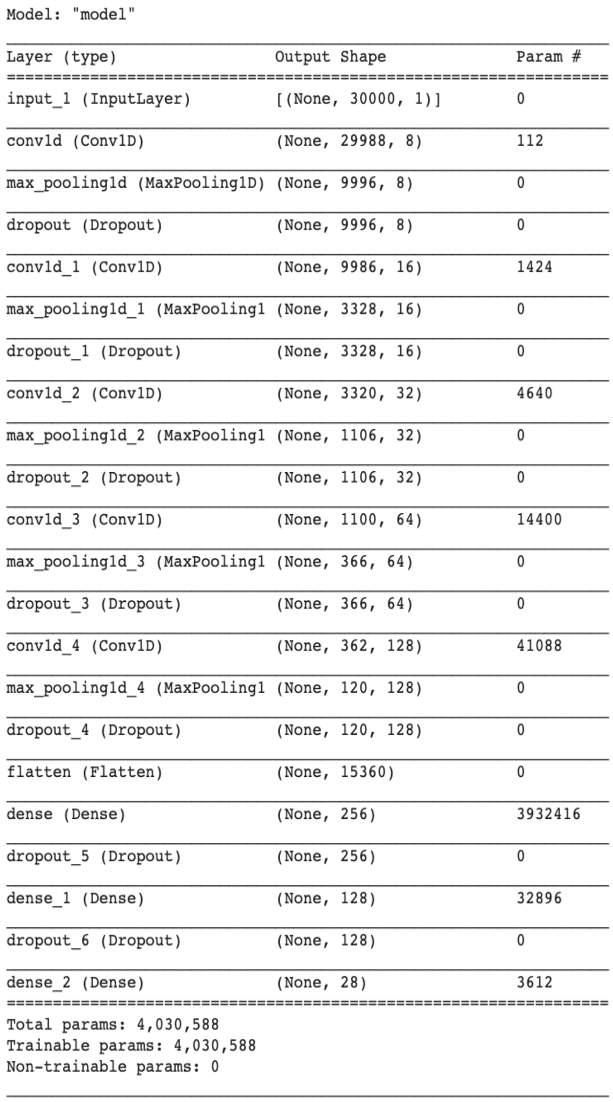
```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 30000, 1)]        0
_________________________________________________________________
conv1d (Conv1D)              (None, 29988, 8)          112
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 9996, 8)           0
_________________________________________________________________
dropout (Dropout)            (None, 9996, 8)           0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 9986, 16)          1424
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 3328, 16)          0
_________________________________________________________________
dropout_1 (Dropout)          (None, 3328, 16)          0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 3320, 32)          4640
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 1106, 32)          0
_________________________________________________________________
dropout_2 (Dropout)          (None, 1106, 32)          0
_________________________________________________________________
conv1d_3 (Conv1D)            (None, 1100, 64)          14400
_________________________________________________________________
max_pooling1d_3 (MaxPooling1 (None, 366, 64)           0
_________________________________________________________________
dropout_3 (Dropout)          (None, 366, 64)           0
_________________________________________________________________
conv1d_4 (Conv1D)            (None, 362, 128)          41088
_________________________________________________________________
max_pooling1d_4 (MaxPooling1 (None, 120, 128)          0
_________________________________________________________________
dropout_4 (Dropout)          (None, 120, 128)          0
_________________________________________________________________
flatten (Flatten)            (None, 15360)             0
_________________________________________________________________
dense (Dense)                (None, 256)               3932416
_________________________________________________________________
dropout_5 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               32896
_________________________________________________________________
dropout_6 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 28)                3612
=================================================================
Total params: 4,030,588
Trainable params: 4,030,588
Non-trainable params: 0
_________________________________________________________________
```

Figure 7. Classification Model with Padding Technique



```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 30000, 1)]        0
_________________________________________________________________
conv1d (Conv1D)              (None, 29988, 8)          112
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 9996, 8)           0
_________________________________________________________________
dropout (Dropout)            (None, 9996, 8)           0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 9986, 16)          1424
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 3328, 16)          0
_________________________________________________________________
dropout_1 (Dropout)          (None, 3328, 16)          0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 3320, 32)          4640
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 1106, 32)          0
_________________________________________________________________
dropout_2 (Dropout)          (None, 1106, 32)          0
_________________________________________________________________
conv1d_3 (Conv1D)            (None, 1100, 64)          14400
_________________________________________________________________
max_pooling1d_3 (MaxPooling1 (None, 366, 64)           0
_________________________________________________________________
dropout_3 (Dropout)          (None, 366, 64)           0
_________________________________________________________________
conv1d_4 (Conv1D)            (None, 362, 128)          41088
_________________________________________________________________
max_pooling1d_4 (MaxPooling1 (None, 120, 128)          0
_________________________________________________________________
dropout_4 (Dropout)          (None, 120, 128)          0
_________________________________________________________________
flatten (Flatten)            (None, 15360)             0
_________________________________________________________________
dense (Dense)                (None, 256)               3932416
_________________________________________________________________
dropout_5 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               32896
_________________________________________________________________
dropout_6 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 28)                3612
=================================================================
Total params: 4,030,588
Trainable params: 4,030,588
Non-trainable params: 0
_________________________________________________________________
```

Figure 8. Classification with mel-spectrogram technique



```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 30000, 1)]        0
_________________________________________________________________
conv1d (Conv1D)              (None, 29988, 8)          112
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 9996, 8)           0
_________________________________________________________________
dropout (Dropout)            (None, 9996, 8)           0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 9986, 16)          1424
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 3328, 16)          0
_________________________________________________________________
dropout_1 (Dropout)          (None, 3328, 16)          0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 3320, 32)          4640
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 1106, 32)          0
_________________________________________________________________
dropout_2 (Dropout)          (None, 1106, 32)          0
_________________________________________________________________
conv1d_3 (Conv1D)            (None, 1100, 64)          14400
_________________________________________________________________
max_pooling1d_3 (MaxPooling1 (None, 366, 64)           0
_________________________________________________________________
dropout_3 (Dropout)          (None, 366, 64)           0
_________________________________________________________________
conv1d_4 (Conv1D)            (None, 362, 128)          41088
_________________________________________________________________
max_pooling1d_4 (MaxPooling1 (None, 120, 128)          0
_________________________________________________________________
dropout_4 (Dropout)          (None, 120, 128)          0
_________________________________________________________________
flatten (Flatten)            (None, 15360)             0
_________________________________________________________________
dense (Dense)                (None, 256)               3932416
_________________________________________________________________
dropout_5 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               32896
_________________________________________________________________
dropout_6 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 28)                3612
=================================================================
Total params: 4,030,588
Trainable params: 4,030,588
Non-trainable params: 0
_________________________________________________________________
```

Figure 9. Classification model with MFCC technique

Table 4. Total Parameters (Neuron setting)

Spectrogram	Padding	Mel-Spectrogram	MFCC
4,046,972	4,030,588	4,030,588	4,030,588

In the training process, 500 epochs has been applied to obtain significant results to be used in testing the test data. Results for accuracy and validation accuracy from the training stage using tensorboard are shown in Table 5. The number of epoch affects the result and time taken to obtain the result.

Appropriate number of epoch is needed to obtain stable results. The highest accuracy among the four techniques has been obtained for the padding with validation accuracy of 0.9286 and validation loss of 0.5020. This, when associated with the previous parameter result, there is also a significant relationship with the corresponding value of neutron generated parameters.

Table 5. Voice classification accuracy and loss (epoch = 500)

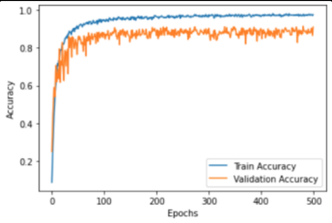
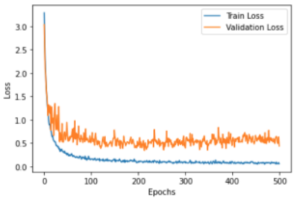
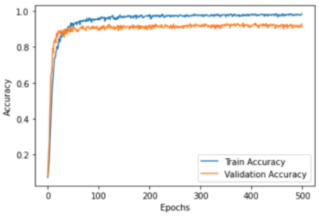
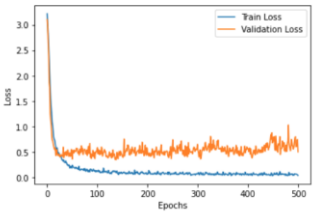
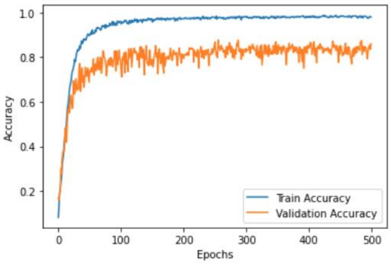
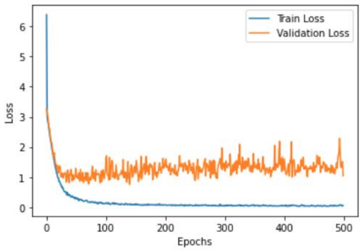
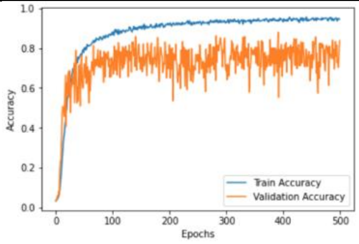
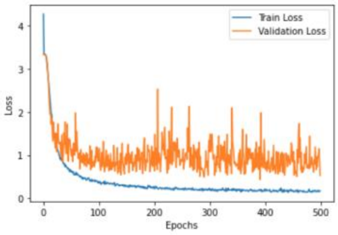
Technique	Accuracy	Loss
Spectrogram	 accuracy: 0.9701 val_accuracy: 0.9107	 loss: 0.0648 val_loss: 0.4353
Padding	 accuracy: 0.9829 val accuracy: 0.9286	 loss: 0.0451 val_loss: 0.5020
Mel-Spectrogram	 accuracy: 0.9815 val_accuracy: 0.8607	 loss: 0.0525 val_loss: 1.0442
MFCC	 accuracy: 0. 9506 val_accuracy: 0.8381	 loss: 0.1550 val_loss: 0.5291

Table 6 summarizes the results of the training stage where the best results are highlighted. The classification accuracies of the four techniques depends on the values of accuracy, loss and neuron. However, at this stage, only the spectrogram and padding techniques show superior results.

Table 6. Summary of classification results

Technique	Accuracy (%)	Loss (%)	Neuron
Spectrogram	accuracy: 0.9701 val_accuracy: 0.9107	loss: 0.0648 **val_loss: 0.4353**	**4,046,972**
Padding	accuracy: 0.9829 **val accuracy: 0.9286**	**loss: 0.0451** val_loss: 0.5020	4,030,588
Mel-Spectrogram	accuracy: 0.9815 val_accuracy: 0.8607	loss: 0.0525 val_loss: 1.0442	4,030,588
MFCC	accuracy: 0. 9506 val_accuracy: 0.8381	loss: 0.1550 val_loss: 0.5291	4,030,588

The model that has been developed during the training stage is tested on the test data. Table 9 displays the classification accuracy results of the four techniques which shows that padding and spectrogram techniques have obtained the highest accuracy. This accuracy level is influenced by several factors. During the training stage, padding has the highest validation accuracy (0.9286) and the lowest loss level (0.0451) while spectrogram has the lowest level for validation loss (0. 4353) and the highest level for the number of neuron (4046972).

Table 9. Classification Result for 28 Arabic alphabets

Technique	Classification Accuracy (%)
Spectrogram	92.86
Padding	92.86
Mel-spectrogram	82.14
MFCC	82.14

Analysis has also been performed to discover alphabets that are easy and difficult to pronounce. The probability of pronouncing several of these alphabets are shown in Table 7 which shows that alphabet ‘ب’ has the highest probability. This reflects that this is the easiest alphabet to pronounce by the children.

Table 7. Result of Correct Pronunciation

Alphabet	Probability
ا	2.3011212e-16
ب	**1.0000000e+00**
ت	5.9744099e-11
ث	6.7040449e-13
ج	7.4232634e-13
….	.....

There are several alphabets such as ‘س' and 'ث’, that are difficult to pronounce. This alphabet has similar sound. Results are displayed in Table 8 where from all four techniques that has been used, the alphabet "'ث’' has the highest probability of incorrect pronunciation.

Table 8. Result of Incorrect Pronunciation

Class	Probability
ا	5.05106174e-04
ب	1.58853392e-04
ت	2.33987768e-04
ث	**9.80831444e-01**
ج	3.66151624e-04
……	…….

Conclusion:

In this study, padding technique has been proposed in the classification of Arabic alphabet pronunciation and compare its performance with the spectrogram, mel-spectrogram and MFCC techniques. Padding and spectrogram technique have been shown to have to have superior performance than mel-spectrogram and MFCC.

Future work can be focused on the data that has been used which consists only of the voices of the children. Thus, testing the method on adult voices should be carried out. Furthermore, several of the Arabic alphabets that have almost similar pronunciation such as ‘ ض ‘ ،’ ص ‘ ،’ ز ‘ ،’ ح ‘ ،’ ث ط ‘ ،’ ’, which resulted in high probability of incorrect pronunciation. Thus, the proposed method should be enhanced to solve this issue.

Acknowledgment:

Funding for this research was from the Universitas Muhammadiyah Yogyakarta, Indonesia and work was conducted in collaboration with the Universiti Utara Malaysia, Malaysia.

Authors' declaration:

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are mine ours. Besides, the Figures and images, which are not mine ours, have been given the permission for re-publication attached with the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee in Universitas Muhammadiyah Yogyakarta.

References:

1. LeChun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998;86(11), 2278-2324.
2. Yuan A, Bai G, Jiao L, Liu Y. Offline handwritten English character recognition based on convolution neural network. Proceedings of the 10th IAPR International Workshop on Document Analysis Systems. 2012;125-129.
3. Ren H, El-Khamy, Lee J. CNF+CT: Context network fusion of cascade-trained convolution neural networks for image super-resolution. IEEE Transactions on Computational Imaging. 2019;6,447-462.
4. Li H, Shi L. Robust event-based object tracking combining correlation filter and CNN representation. Frontiers in Neurorobotics. 2019;13,82.
5. Mushtaq Z, Su SF. Environment sound classification using a regularized deep convolution neural network with data augmentation. Applied Acoustics. 2020;167,107389.
6. Mushtaq Z, Su SF, Tran Q. -V. Spectral images based environmental sound classification using CNN with meaningful data augmentation. Applied Acoustics. 2021;172,107581.
7. Tun PTZ. Audio feature extraction using mel frequency cepstral coefficients. International Journal

of Creative and Innovative Research in All Studies. 2020;2(12),95-98.

8. Jin S, Wamg X, Du L, He D. Evaluation and modeling of automotive transmission whine noise quality based on MFCC and CNN. Applied Acoustics. 2021;172,107562.
9. Almanfaluti IK, Sugiono JP. Identifikasi pola suara pada bahasa Jawa meggunakan mel frequency cepstral coefficients (MFCC). Jurnal Media Informatika Budidarma, 2020;4(1),22-26. https://doi.org/10.30865/mib.v4i1.1793
10. Ranjan R, Thakur A. Analysis of feature extraction techniques for speech recognition system. International Journal of Innovative Technology and Exploring Engineering. 2019;8(7C2),197-200.
11. El-Alami F, El Mahdaouy A, El Alaoui SO, En-Nahnahi N. A deep autoencoder-based representation for Arabic text categorization. Journal of Information and Communication Technology, 2020;19(3),381–398.
12. Adhayani A, Tresnawati D. Pengembangan sistem multimedia pembelajaran Iqro' menggunakan metode Luther. Jurnal Algoritma. 2015;12(1),264-270.
13. Anwar K. Pengenalan pengucapan huruf hijaiyah dengan mel-frequency cepstrum coefficients (MFCC) dan manhattan distance. [Masters thesis]:Universitas Islam Negeri Sultan Syarif Kasim, Indonesia. 2018.
14. Ramansyah W, Madura UT. Pengembangan multimedia pembelajaran interaktif dengan tema pengenalan huruf Arabic alphabet untuk peserta didik sekolah dasar. Jurnal Ilmiah Edutic. 2016;3(1),28-37.
15. Efendi R, Purwandari EP, Aziz MA. Aplikasi pengenalan huruf hujaiyah berbaris merker augmented reality pada platform android. Jurnal Pseudocode. 2015;2(2),124–134. https://doi.org/10.33369/pseudocode.2.2.124-134
16. Richardson A, Ari SB, Sinai M, Atsmon A, Conley ES, Gat Y, Segev G. Mobile applications for stroke: A survey and a speech classification approach. Proceedings of the 5th International Conference on Information and Communication Technologies for Ageing Well and e-Health. 2019;159–166.
17. Livezey JA, Bouchard KE, Chang EF. Deep learning as a tool for neural data analysis: Speech classification and cross-frequency coupling in human sensorimotor cortex. PLoS Computational Biology. 2019;15(9).
18. Tamulevičius G, Karbauskaitė R, Dzemyda G. Speech emotion classification using fractal dimension-based features. Nonlinear Analysis: Modelling and Control 2019;24(5),679–695.
19. Coates A, Lee H, Ng AY. An analysis of single layer networks in unsupervised feature learning. 2011
20. Boddapati V, Petef A, Rasmusson J, Lundberg L. Classifying environmental sounds using image recognition networks. Procedia Computer Science. 2017;112,2048–2056.
21. Mustaqeem M, Sajjad M, Kwon S. Clustering-based speech emotion recognition by incorporating learned features and deep BiLSTM. IEEE Access. 2020;8,79861-79875.
22. Huang J, Chen B, Yao B, He W. ECG arrhythmia classification using STFT-based spectrogram and convolutional neural network. IEEE Access. 2019;7,92871-92880.
23. Gimenez M, Palanca J, Botti V. Semantic-based padding in convolution neural networks for improving the performance in natural language processing. A case study in sentiment analysis. Neurocomputing. 2020;378, 315-323.
24. Nada Q, Ridhuandi C, Santoso P, Apriyanto D. Speech recognition dengan Hidden Markov Model untuk pengenalan dan pelafalan huruf Arabic alphabet. Jurnal Al-Azhar Indonesia Seri Sains dan Teknologi. 2019;5(1),19-26.
25. Nugroho K, Noersasongko E, Purwanto, Muljono, Santoso, HA. Javanese gender speech recognition using deep learning and singular value decomposition. Proceedings of the International Seminar on Application for Technology of Information and Communication. 2019;251–254.
26. Borsky M, Mehta DD, Van Stan JH, Gudnason J. Modal and nonmodal voice quality classification using acoustic and electroglottographic features. IEEE/ACM Transactions on Audio, Speech, and Language Processing. 2017;25(12),2281-2291.
27. Wu H, Yan W, Li P, Wen Z. Deep texture exemplar extraction based on trimmed T-CNN. IEEE Transactions on Multimedia. 2020.
28. Hashemi M. Enlarging smaller images before inputting into convolutional neural network: Zero-padding vs. interpolation. Journal of Big Data 2019;6(1),98. https://doi.org/10.1186/s40537-019-0263-7

طريقة تصنيف الكلام العربي على أساس الحشو والشبكة العصبية للتعلم العميق

آرسوني آرسوني[1] كوروهانا كو محمود[2] كحيا دمارجاتي[1] حسن بصري سلامات[1]

1 جامعة المحمدية يوجياكارتا ، إندونيسيا.
2 جامعة أوتارا ماليزيا.

الخلاصة:

تم استخدام الشبكة العصبية لالتفاف التعلم العميق على نطاق واسع للتعرف على الصوت أو تصنيفه. تم استخدام تقنيات مختلفة مع الشبكة العصبية الالتفافية لإعداد البيانات الصوتية قبل عملية التدريب في تطوير نموذج التصنيف. ومع ذلك ، لا يمكن لجميع النماذج إنتاج دقة تصنيف جيدة نظرًا لوجود العديد من أنواع الصوت أو الكلام. ان تصنيف الفاظ الأبجدية العربية هو أحد أنواع الصوت والنطق الدقيق المطلوب في تعلم قراءة القرآن. وبالتالي ، تتطلب تقنية معالجة النطق وتدريب البيانات المعالجة نهجًا محددًا. وللتغلب على هذه المشكلة ، تم اقتراح طريقة تعتمد على الحشو والشبكة العصبية لالتفاف التعلم العميق لتقييم نطق الأبجدية العربية. وقد تم تسجيل البيانات الصوتية لستة أطفال في المدارس واستخدامها لاختبار أداء الطريقة المقترحة. تم استخدام تقنية الحشو لزيادة البيانات الصوتية قبل تغذية البيانات إلى بنية CNNلتطوير نموذج التصنيف. بالإضافة إلى ذلك ، تم تقديم ثلاث تقنيات أخرى لاستخراج الميزات لتمكين مقارنة الطريقة المقترحة التي تستخدم تقنية الحشو. أداء الطريقة المقترحة مع تقنية الحشو هو على قدم المساواة مع الطيف ولكن أفضل من ميل الطيف ومعاملات cepstral التردد ميل. كما أظهرت النتائج أن الطريقة المقترحة كانت قادرة على تمييز الحروف الهجائية العربية التي يصعب نطقها. يمكن توسيع الطريقة المقترحة مع تقنية الحشو لمعالجة قدرة نطق الصوت الأخرى بخلاف الحروف الهجائية العربية.

الكلمات المفتاحية: الأبجدية العربية، كوفيد-19 ، التعلم العميق، المخطط الطيفي، تصنيف الكلام .