JOIN (Jurnal Online Informatika)
Volume 7 No. 1 | June 2022: 123-131
DOI: 10.15575/join.v7i1.882

p-ISSN: 2527-1682
e-ISSN: 2527-9165

Automatic Detection of Hijaiyah Letters Pronunciation using Convolutional Neural Network Algorithm

Yana Aditia Gerhana[1], Aaz Muhammad Hafidz Azis[2], Diena Rauda Ramdania[3], Wildan Budiawan Dzulfikar[4], Aldy Rialdy Atmadja[5], Deden Suparman[6], Ayu Puji Rahayu[7]

[1,2,3,4,5] Department of Informatics, UIN Sunan Gunung Djati Bandung
[6]Department of Biology, UIN Sunan Gunung Djati Bandung
[7]Faculty of Education Fujian Normal University Fuzhou, China

Article Info

Article history:

Received June 06, 2022
Revised June 29, 2022
Accepted June 16, 2022
Published June 30, 2022

Keywords:

CNN
CRISP-DM
Hijaiyah
MFCC
Speech recognition

ABSTRACT

Speech recognition technology is used in learning to read letters in the Qur'an. This study aims to implement the Convolutional Neural Network (CNN) algorithm in recognizing the results of introducing the pronunciation of the hijaiyah letters. The pronunciation sound is extracted using the Mel-frequency Cepstral Coefficients (MFCC) model and then classified using a deep learning model with the CNN algorithm. This system was developed using the CRISP-DM model. Based on the results of testing 616 voice data of 28 hijaiyah letters, the best value was obtained for accuracy of 62.45%, precision of 75%, recall of 50%, and f1-score of 58%.

Corresponding Author:

Yana Aditia Gerhana,
Department of Informatics
Universitas Islam Negeri Sunan Gunung Djati Bandung,
Jl. A.H. Nasution No. 105 Cibiru Kota Bandung, Indonesia
Email: yanagerhana@uinsgd.ac.id

1. INTRODUCTION

The implementation of artificial intelligence technology in education is increasingly widely used, one of which is in pronouncing Arabic or hijaiyah letters. Pronouncing the good hijaiyah letters in Islamic teachings is one of the requirements for reading the Qur'an [1-2]. *Speech recognition* is a technology that can detect acoustic signals or sound waves. Speech recognition aims to recognize the sound of words or sentences for further processing. Voice recognition is another term in speech recognition, training audio from a particular speaker [1][3-5]. Extraction methods can be used in speech recognition, one of which uses the Mel-frequency cepstral coefficients (MFCC) extraction model [6-9]. The MFCC extraction method is widely used in modeling audio recognition systems. The extraction results are then trained using the deep learning method [10-13]. Marlina et al.'s research describes the voice recognition system of makhraj hijaiyah letters using the MFCC method as an extractor and the SVM algorithm for children as learning media materials in learning the Koran [9]. Arshad et al. tested the MFCC extraction method and the MSE Classification algorithm for detecting the pronunciation of the makhraj letters of the Qur'an. They recognized the makhraj hijaiyah letters with the MFCC method with almost 100% accuracy [14].

Convolutional Neural Network (CNN) is one of the classification algorithms in deep learning, which has a good ability towards recognition systems [15-16]. CNN has several layers that function in the extraction process from images or audio [17-18]. CNN is also believed to be the best model for solving object recognition

e-ISSN 2527-1682
p-ISSN 2527-9165

and detection problems (8). Based on a computer vision competition held by Stanford University, three researchers from the University of Toronto revealed that the CNN algorithm reduced the error rate by 15.3% in object detection and object recognition [19]. CNN performs better in image classification than other classification algorithms such as SVM and K-NN [20]. CNNs are used for biometric systems such as signature verification. The CNN algorithm was trained on a sample of 20 users and achieved validation accuracy of 83% for Inception-v1 and 75% for Inception-v3[17]. Mursal Dawodi et al. tested the CNN algorithm and the MFCC extraction method to detect Afghan speech's pronunciation. As a result, the accuracy obtained is 99% [18].

2. METHOD

The sound extraction process uses MFCC, resulting in the values of specific parameters in the form of vectors. This research uses deep learning by training voice data to form a recognition system. Overall, the development method used in this study adopted CRISP-DM processes, starting from the business understanding stage, data understanding, data preparation, modeling, and evaluation [21][22]. Figure 1 describes the steps in the CRISP-DM methodology.

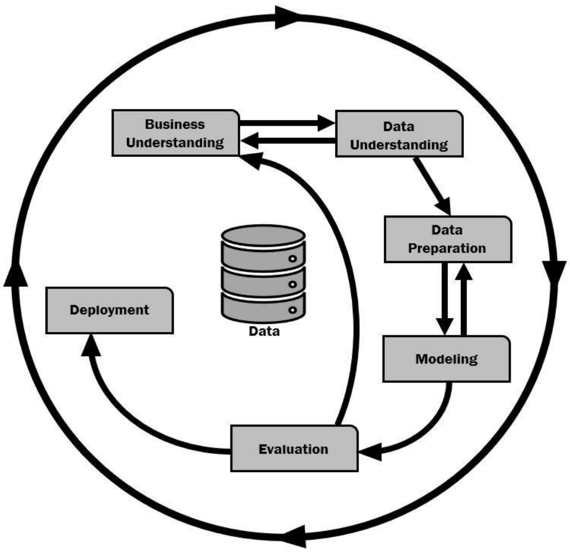

Figure 1. CRISP-DM Methodology [3]

Business understanding is the description stage of analyzing the system’s needs, objectives, and limitations that will be formed to achieve this research's goals. Data understanding is collecting initial data, describing, exploring, and verifying data quality. Data preparation consists of feature extraction using MFCC, normalization, and sharing data. Modeling is the stage of classifying the quality of the hijaiyah letter dataset with kasrah. The modeling stage in this study uses the CNN algorithm to find the best accuracy value based on three parameters, namely architectural model, batch size, and epoch. The evaluation contains test scenarios for both architectural models, confusion matrix, and testing the hijaiyah letters raised by the system. These three test scenarios are needed to find the maximum level of accuracy in the kasrah letter detection system. Deployment is the last stage where the results of research that have been carried out can be implemented directly.

3. RESULTS AND DISCUSSION

This section describes the results of implementing the automatic detection process for pronouncing the hijaiyah letters with kasrah. The scheme shown in Figure 2 is one of the sample data for letter i.

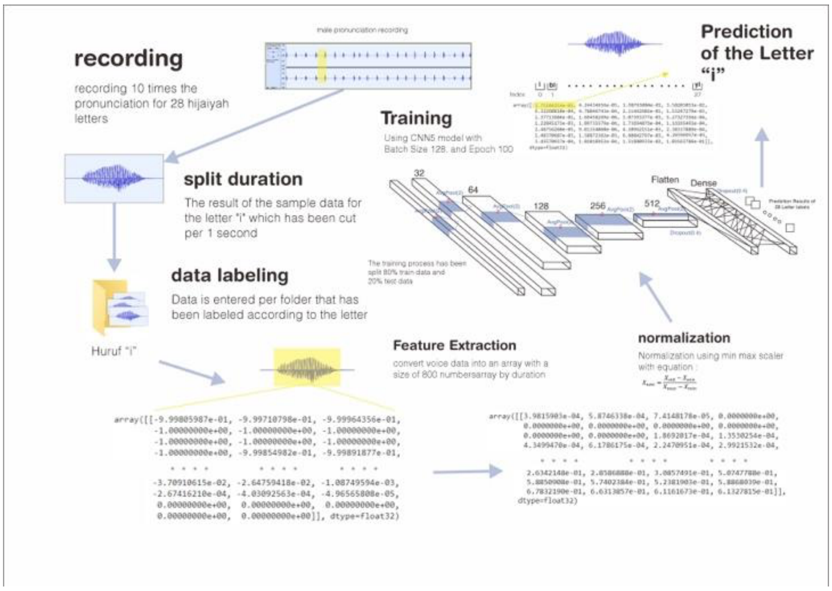

Figure 2. Visualization of the automatic detection of the pronunciation of the hijaiyah letters

Figure 2 describes the scenario from the initial stage to the final step of issuing predictions from letters. The stages in the system follow the stages of the CRISP-DM development method.

3.1. Business understanding

This stage of business understanding focuses on describing the form of needs analysis, objectives, and the system's limitations to achieve this research's objectives.

In this study, a voice recognition system was formed, which aims to recognize the pronunciation of the hijaiyah letter with kasrah from the voice data of people who have been inputted beforehand by entering several variations of the hijaiyah letter sound for each person.

The development of the system to be built, of course, requires both functional and non-functional requirements.

a) Recording human speech via a condenser mic and storing it on a pre-prepared storage media
b) Perform preprocessing automatically using the system. The system can carry out the training process (training)
c) Perform feature extraction with MFCC and speech classification with the CNN model.
d) The system can carry out the evaluation process.
e) The system will issue the letter names according to the recognized voice signal.

In the planning process, system development uses google collab for code writing. The main libraries used are tensor flow and librosa.

3.2. Data understanding

The data used in the hijaiyah letter recognition system with kasrah is explained at this stage. The data is recorded from 15 female students from the At-Taufiqah Islamic Boarding School. The steps of forming this dataset start from recording, processing data recording, and data labeling.

a) Recording audio using a BM100 condenser mic. The recording process is done by reciting per letter and repeated ten times.

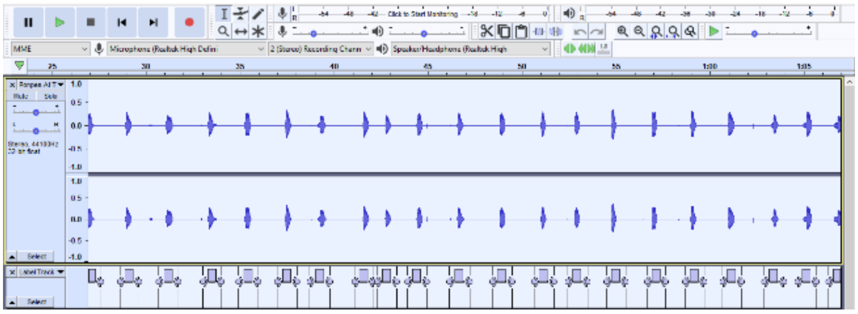

Figure 3. Recording of the initial sound data for the pronunciation of the hijaiyyah letters with kasrah

b) The recording results are processed by cutting each letter with a duration of 1 second.

c) The last stage is data labeling. The audio data from the cuts is entered into a folder that has been labeled. The number of letters as a result of labeling is 28 hijaiyah letters.

3.3 Data processing

Data processing includes all activities on the data before carrying out the modeling process using the CNN algorithm. Another term for data processing is preprocessing. There are three processes in building a data set before training: feature Extraction, data normalization, and data sharing.

a) Feature Extraction

b) The extraction method used is MFCC. The parameters in this method are set to as many as 800 parameters or features of each voice. The number of 800 parameters is poured into vector form.

c) Normalization of data

d) The normalization method in this study uses Min-Max Normalization: Min-Max normalization is a normalization method by performing a linear transformation of the original data to produce a balance of comparison values between the data before and after the process.

e) Data Sharing

f) Data labeling the hijaiyah letter label, then the data is divided into training and test data. *Training data* is the data used to train the model of a system. At the same time, the test data is used for the system testing process. The comparison of data separation is 80% for training data and 20% for test data.

3.4 Modeling

The modeling used uses the CNN algorithm to classify hijaiyah letters with kasrah. This modeling uses 1-dimensional Convolution, AveragePooling, Dropout, and Flatter Lauer layers.

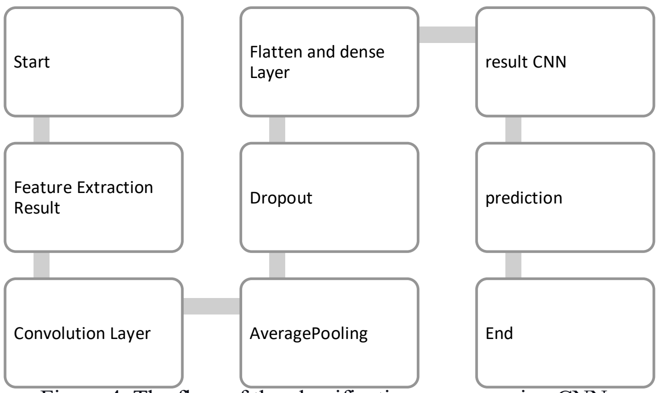

Figure 4. The flow of the classification process using CNN.

3.5 *Evaluation*

The evaluation aims to determine the system's performance because of development. Tests were carried out with three methods, architectural model testing, confusion matrix, and K-fold validation.

1. Architectural Model Testing

a) Tests using layer variations. The test consists of five variations of the CNN architecture. One-dimensional convolutions are then written as CNN1, CNN2 to CNN5. Multiple layers. Table 1 describes the layers of each architectural model.

Table 1. Variations of Architectural Models

CNN1	CNN2	CNN3	CNN4	CNN5
CONV1D	CONV1D	CONV1D	CONV1D	CONV1D
Average Pool	Average Pool	Average Pool	Average Pool	Average Pool
	CONV1D	CONV1D	CONV1D	CONV1D
	Average Pool	Average Pool	Average Pool	Average Pool
		CONV1D	CONV1D	CONV1D
		Average Pool	Average Pool	Average Pool
			CONV1D	CONV1D
			Average Pool	Average Pool
				CONV1D
				Average Pool
Flatten Layer	Flatten Layer	Flatten Layer	Flatten Layer	Flatten Layer
Dense (Relu)	Dense (Relu)	Dense (Relu)	Dense (Relu)	Dense (Relu)
Dense (SoftMax)	Dense (SoftMax)	Dense (SoftMax)	Dense (SoftMax)	Dense (SoftMax)

The results of the first test measurements are outlined in table 2. This first test led to the CNN5 architecture, which has the highest accuracy value of 57.70%.

Table 2. Architectural Model Accuracy Results

Arsitektur	Jumlah Layer CONV1D	Akurasi
CNN1	1 Layer	43, 18%
CNN2	2 Layer	52, 86%
CNN3	3 Layer	53, 79%
CNN4	4 Layer	54, 81%
CNN5	5 Layer	57, 70%

b) Testing using batch size variations: The second test is to test the CNN5 model using variations of batch size parameters consisting of 32, 64, 128, 256, and 512. This batch parameter test uses 100 epochs as the initial value. Table 3 describes the test results using batch size on CNN5.

Table 3. Results of Batch Variation Accuracy

Arsitektur	*Batch*	Akurasi
CNN5	32	57, 70%
	64	59,41%
	128	62,45%
	256	61,36%
	512	59,74%

This batch size is seen in how much data is divided from all datasets in training [19]. Based on the results of this test, it can be seen that the accuracy tends to increase when the number of batches added is increased. However, the accuracy reaches a steady state after the number of clusters is 128 and tends to decrease. These results explain that the training to achieve the maximum accuracy value requires 128 sample data distributed to each neural network layer.

c) The test uses a variation of the epoch parameter: the third test uses a variation of the epoch parameter as many as five variations consisting of 50,70,90, 100, and 120. This test uses the highest architectural model in the previous test. This model uses the CNN5 architectural model, and the Batch is set to 128z

 e-ISSN 2527-1682
p-ISSN 2527-9165

Table 4. Results of Epoch Variation Accuracy

Arsitektur	*Batch*	*Epoch*	Akurasi
CNN5	128	50	56,49%
		70	59.09%
		90	61,68%
		100	62,45%
		120	61,85%

Based on the test results, the higher the epoch value, the higher the accuracy value. The highest accuracy value obtained is 100. However, the accuracy value reaches a steady-state condition or becomes overfitting as it is added. Steady-state or over-fitting conditions occur because of too much training. The recognition is too rigid to the voice data, and the validation and loss results are unstable or fluctuate.

2. K-Fold Validation Test

Using k-fold cross-validation is a test step by distributing data in the training process. This study tested k-fold cross-validation at each stage of training. The accuracy results that have been written in the previous section are the average results of 10 values in k-fold cross-validation. Based on previous tests, the test results are taken from the model with the highest accuracy. The model taken uses the CNN5 model batch size 128 and epoch 100. The test results are described in table 5.

Table 5. K-Fold Validation Results

Fold to -	Akurasi	Fold to -	Akurasi
1	59.90	6	61.85
2	61.36	7	66.72
3	63.12	8	59.74
4	64.77	9	63.47
5	64.28	10	59.25

The total accuracy of each fold averaged is obtained from the following equation.

$$average = \frac{JTotal\ accuracy\ of\ all\ folds}{Number\ of\ folds} X\ 100\% \ = \frac{6245}{10} 100\% = 62.45\ \%$$

3. Performance Testing Using the Confusion Matrix

The confusion matrix method is one of the tests using a matrix of prediction results that will be compared with the original input data class. The architectural model used is CNN5 with a batch value of 128 and an epoch of 100.

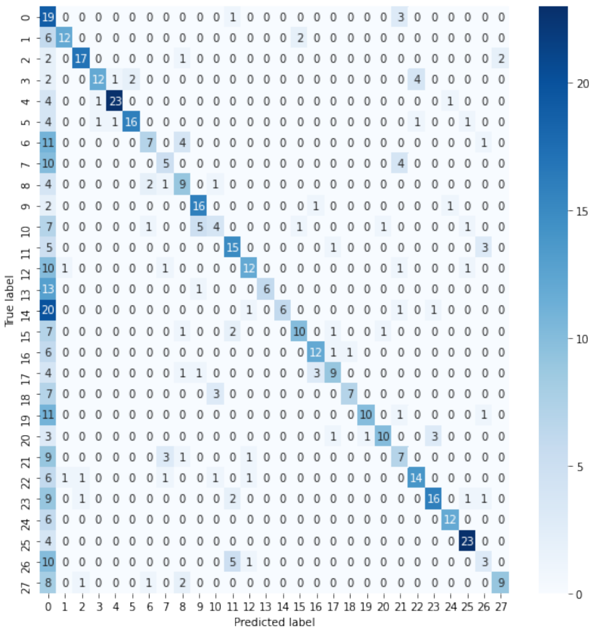

Figure 5. Confusion Matrix

Figure 5 was then averaged. The average results of all classes are 86% precision, 67% recall, and 76% for F1-score. While the accuracy obtained is 62.45%.

$$Precision_{(1)} = \frac{TTP_{all(0)}}{TTP_{all} + TFP_{(0)}} x100\% = \frac{12}{12+2} = \frac{12}{14} x100 = 86\ \%$$

$$Recall_{(1)} = \frac{TTP_{all(0)}}{TTP_{all} + TFN_{(0)}} X100 = \frac{12}{12+\ 8} = X100\% = 67\ \%$$

$$F1 - Score_{(1)} = 2\,X \frac{Presisi_{(1)}\ X\ Recall_{(1)}}{Presisi_{(1)} + Recall_{(1)}} x100\% \ = 2X \frac{0{,}86X0{,}67}{0{,}86 + 0{,}67} = 2X \frac{0{,}58}{1{,}53} x100\% = 76\%$$

The complete test results are described in Figure 6.

	precision	recall	f1-score	support
0	0.73	0.35	0.47	23
1	0.86	0.60	0.71	20
2	0.85	0.77	0.81	22
3	0.86	0.57	0.69	21
4	0.92	0.79	0.85	29
5	0.89	0.67	0.76	24
6	0.64	0.30	0.41	23
7	0.45	0.26	0.33	19
8	0.47	0.53	0.50	17
9	0.70	0.80	0.74	20
10	0.44	0.20	0.28	20
11	0.60	0.62	0.61	24
12	0.75	0.46	0.57	26
13	1.00	0.30	0.46	20
14	1.00	0.21	0.34	29
15	0.77	0.45	0.57	22
16	0.75	0.60	0.67	20
17	0.69	0.50	0.58	18
18	0.88	0.41	0.56	17
19	0.91	0.43	0.59	23
20	0.83	0.56	0.67	18
21	0.41	0.33	0.37	21
22	0.74	0.56	0.64	25
23	0.80	0.53	0.64	30
24	0.86	0.67	0.75	18
25	0.85	0.85	0.85	27
26	0.33	0.16	0.21	19
27	0.82	0.43	0.56	21
micro avg	0.74	0.50	0.60	616
macro avg	0.74	0.50	0.58	616
weighted avg	0.75	0.50	0.58	616
samples avg	0.50	0.50	0.50	616

Figure 6. Value of Precision, Recall, and f1-Score

4. Performance Testing using the confusion matrix

The test was performed ten times by entering an audio file with a Wav extension into the input field in the application. This test is intended to test the hijaiyah letter with kasrah, known as what letter by the system. The pronunciation of hijaiyah letters has many almost identical sounds but has different meanings. The following test results are shown in Table 6.

Table 6. Letter Test Results

Input	*Test output to -*									
	1	2	3	4	5	6	7	8	9	10
اِ	اِ	اِ	اِ	لِ	اِ	رِ	عِ	اِ	اِ	اِ
بِ	بِ	غِ	ضِ	بِ	بِ	اِ	ظِ	بِ	اِ	بِ
تِ	طِ	تِ	ثِ	تِ	تِ	سِ	تِ	طِ	تِ	طِ
ثِ	ثِ	اِ	ثِ	يِ	ثِ	سِ	ثِ	سِ	ثِ	اِ
جِ	جِ	ظِ	جِ	جِ	ضِ	اِ	جِ	زِ	خِ	جِ

Input	Test output to -									
	1	2	3	4	5	6	7	8	9	10
حِ	اِ	حِ	رِ	حِ	هِ	حِ	حِ	رِ	هِ	حِ
خِ	خِ	غِ	خِ	خِ	خِ	ظِ	خِ	اِ	خِ	غِ
دِ	دِ	ظِ	دِ	ذِ	دِ	دِ	اِ	ظِ	دِ	غِ
ذِ	ذِ	ذِ	ضِ	دِ	ذِ	ذِ	ظِ	دِ	اِ	ذِ
رِ	رِ	رِ	لِ	رِ	لِ	اِ	رِ	رِ	رِ	اِ
زِ	ضِ	زِ	زِ	جِ	زِ	اِ	زِ	زِ	ظِ	زِ
سِ	سِ	صِ	سِ	ش	سِ	سِ	ثِ	سِ	يِ	صِ
ش	ش	ش	اِ	ش	تِ	ثِ	ش	ش	صِ	سِ
صِ	صِ	ثِ	اِ	ش	ش	صِ	صِ	ثِ	اِ	ثِ
ضِ	ضِ	ضِ	ظِ	ضِ	ضِ	ضِ	ظِ	جِ	طِ	ظِ
طِ	اِ	طِ	ثِ	طِ	طِ	تِ	طِ	طِ	تِ	اِ
ظِ	ظِ	ظِ	اِ	ظِ	ظِ	ضِ	ظِ	ظِ	ظِ	طِ
عِ	اِ	عِ	مِ	عِ	قِ	غِ	اِ	عِ	عِ	نِ
غِ	غِ	غِ	جِ	ظِ	عِ	غِ	غِ	قِ	غِ	قِ
فِ	وِ	فِ	لِ	فِ	فِ	فِ	فِ	قِ	فِ	اِ
قِ	قِ	جِ	قِ	كِ	قِ	قِ	قِ	كِ	قِ	قِ
كِ	قِ	كِ	كِ	خِ	اِ	كِ	كِ	غِ	كِ	قِ
لِ	لِ	لِ	يِ	لِ	اِ	لِ	وِ	رِ	اِ	لِ
مِ	وِ	مِ	مِ	نِ	مِ	بِ	نِ	مِ	مِ	مِ
نِ	نِ	نِ	ظِ	نِ	مِ	نِ	نِ	دِ	نِ	نِ
وِ	وِ	يِ	وِ	اِ	لِ	وِ	وِ	حِ	وِ	وِ
هِ	هِ	حِ	هِ	هِ	حِ	هِ	هِ	اِ	اِ	هِ
يِ	يِ	لِ	يِ	يِ	اِ	يِ	حِ	يِ	يِ	يِ

The pronunciation of hijaiyah letters is sometimes confused between one letter and another. For example, people confuse sounding the letter i (اِ) with the sound 'i (عِ), the sound of hi (حِ) with Hi (هِ) or with Khi (خِ) , the sound of Si (سِ) with Syi (ش) or Shi (صِ), the sound of Di (دِ) with Dzi (ذِ), and many other examples.

3.6 Deployment

This system was developed using google collab with the python programming language. Furthermore, the development results are deployed to a local web host with the address localhost //:5000.

4 CONCLUSION

Based on testing 616 voice data of 28 hijaiyah letters, the average value of accuracy, precision, recall, and f1-score obtained less than optimal results. These results show that the performance of CNN algorithm implementation in classifying voice data and MFCC in extracting voice data characteristics of hijaiyah pronunciation is influenced by several factors. The first factor that influences it is intonation. The next factor is pronouncing the letters (makhorijul letters) hijaiyah which is incorrect when recording the dataset or testing. Finally, the closeness of the consonant sounds of the letters causes the issued sounds to have similarities that are difficult to classify. Further research on feature extraction of voice data can use methods other than MFCC (ZCPA or LPC). At the same time, deep learning algorithms for classifying voice data can try using the Radial Basis Function Networks (RBFNs) algorithm.

5 REFERENCES

[1] Q. Nada, C. Ridhuandi, P. Santoso, and D. Apriyanto, "Speech Recognition dengan Hidden Markov Model untuk Pengenalan dan Pengucapan Huruf Hijaiyah," *J. Al-AZHAR Indones. SERI SAINS DAN Teknol.*, vol. 5, no. 1, p. 19, 2019, doi: 10.36722/sst.v5i1.319.

[2] S. Khairuddin *et al.*, "Classification of the Correct Quranic Letters Pronunciation of Male and Female Reciters," in *IOP Conference Series: Materials Science and Engineering*, 2017, doi: 10.1088/1757-899X/260/1/012004.

[3] A. T. Ali, H. S. Abdullah, and M. N. Fadhil, "Voice recognition system using machine learning techniques," *Materials Today: Proceedings*, 2021, pp. 1–7, doi: 10.1016/j.matpr.2021.04.075.

[4] S. Souli, R. Amami, and S. Ben Yahia, "A robust pathological voices recognition system based on DCNN

and scattering transform," *Appl. Acoust.*, vol. 177, June, pp. 1–7, 2021, doi: 10.1016/j.apacoust.2020.107854.
[5] M. T.LuetmerBA, C. H.Hunt, R. J.McDonald MD, B. J. B. MD, and D. F.KallmesMD, "Laterality Errors in Radiology Reports Generated With and Without Voice Recognition Software: Frequency and Clinical Significance," *J. Am. Coll. Radiol.*, vol. 10, no. 7, pp. 538–543, 2013, doi: /10.1016/j.jacr.2013.02.017.
[6] S. G. Koolagudi, D. Rastogi, and K. S. Rao, "Identification of Language using Mel-Frequency Cepstral Coefficients (MFCC)," in *ICMOC*, 2012, pp. 3391–3398, doi: 10.1016/j.proeng.2012.06.392.
[7] D. Taufik and N. Hanafiah, "AutoVAT: An Automated Visual Acuity Test Using Spoken Digit Recognition with Mel Frequency Cepstral Coefficients and Convolutional Neural Network," in *5th International Conference on Computer Science and Computational Intelligence 2020*, 2021, pp. 458–468, doi: 10.1016/j.procs.2021.01.029.
[8] G. Shen, Q. Nguyen, and J. Choi, "An Environmental Sound Source Classification System Based on Mel-Frequency Cepstral Coefficients and Gaussian Mixture Models," in *14th IFAC Symposium on Information Control Problems in Manufacturing*, 2012, pp. 1802–1807, doi: 10.3182/20120523-3-RO-2023.00251.
[9] L. Marlina *et al.*, "Makhraj recognition of Hijaiyah letter for children based on Mel-Frequency Cepstrum Coefficients (MFCC) and Support Vector Machines (SVM) method," in *2018 International Conference on Information and Communications Technology, ICOIACT 2018*, 2018, doi: 10.1109/ICOIACT.2018.8350684.
[10] Y. Wang and B. Lawlor, "Speaker recognition based on MFCC and BP neural networks," *2017 28th Irish Signals Syst. Conf. ISSC 2017*, pp. 0–3, 2017, doi: 10.1109/ISSC.2017.7983644.
[11] S. Tirronen, S. ReddyKadiri, and P. Alku, "The Effect of the MFCC Frame Length in Automatic Voice Pathology Detection," *J. Voice*, vol. 6, no. 3, pp. 297–440, 2022, doi: 10.1016/j.jvoice.2022.03.021.
[12] N. Sugan, N. S. S. Srinivas, L. S. Kumar, M. K. Nath, and A. Kanhe, "Speech emotion recognition using cepstral features extracted with novel triangular filter banks based on bark and ERB frequency scales," *Digit. Signal Process.*, vol. 104, pp. 1–22, 2020, doi: 10.1016/j.dsp.2020.102763.
[13] L. Abdel-Hamid, "Egyptian Arabic Speech Emotion Recognition using Prosodic, Spectral and Wavelet Features," *Speech Commun.*, vol. 122, pp. 19–20, 2020, doi: 10.1016/j.specom.2020.04.005.
[14] N. W. Arshad, S. N. Abdul Aziz, R. Hamid, R. Abdul Karim, F. Naim, and N. F. Zakaria, "Speech processing for makhraj recognition," pp. 323–327, 2011, doi: 10.1109/inecce.2011.5953900.
[15] S. Saha *et al.*, "Predicting motor outcome in preterm infants from very early brain diffusion MRI using a deep learning convolutional neural network (CNN) model," *Neuroimage*, vol. 215, pp. 1–35, 2020, doi: 10.1016/j.neuroimage.2020.116807.
[16] T. Masuda *et al.*, "Deep learning with convolutional neural network for estimation of the characterisation of coronary plaques: Validation using IB-IVUS," *Radiography*, vol. 28, no. 3, pp. 1–7, 2022, doi: /10.1016/j.radi.2021.07.024.
[17] Jahandada, S. M. Sam, K. Kamardin, N. N. A. Sjarif, and N. Mohamed, "Offline Signature Verification using Deep Learning Convolutional Neural Network (CNN) Architectures GoogLeNet Inception-v1 and Inception-v3," in *The Fifth Information Systems International Conference 2019*, 2019, pp. 475–483, doi: 10.1016/j.procs.2019.11.147.
[18] S. Ghimirea, T. Nguyen-Huy, R. C Deo, D. Casillas-Pérez, and S. Salcedo-Sanz, "Efficient daily solar radiation prediction with deep learning 4-phase convolutional neural network, dual stage stacked regression and support vector machine CNN-REGST hybrid model," *Sustain. Mater. Technol.*, vol. 32, 2022, doi: 10.1016/j.susmat.2022.e00429.
[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, 2017, doi: 10.1145/3065386.
[20] U. N. Wisesty, M. S. Mubarok, and A. Adiwijaya, "A classification of marked hijaiyah letters' pronunciation using hidden Markov model," in *AIP Conference Proceedings*, 2017, doi: 10.1063/1.4994439.
[21] Institute of Electrical and Electronics Engineers., "Extending CRISP-DM," pp. 0–4, 2009.
[22] C. Schröer, F. Kruse, and J. M. Gómez, "A Systematic Literature Review on Applying CRISP-DM Process Model," in *CENTERIS - International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies 2020*, 2020, pp. 526–534, doi: 10.1016/j.procs.2021.01.199.
[23] V. Plotnikova, M. Dumas, and F. P. Milani, "Applying the CRISP-DM data mining process in the financial services industry: Elicitation of adaptation requirements," *Data Knowl. Eng.*, vol. 139, 2022, doi: 10.1016/j.datak.2022.102013