Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (32-42)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.41067

JURNAL TEKNIK INFORMATIKA

*Homepage* : http://journal.uinjkt.ac.id/index.php/ti

# Syllable-Based Javanese Speech Recognition Using MFCC and CNNs: Noise Impact Evaluation

**Hermanto[1*]and Tjong Wan Sen[2]**

[1,2]Department of information Technology, Faculty of Computing
President University
[1,2]Jl. Ki Hajar Dewantara, Jababeka City, Cikarang Baru, Bekasi, Indonesia

**Article:**
Accepted: January 02, 2025
Revised: December 02, 2024
Issued: April 30, 2025

***Correspondence Address:**
hermanto@student.president.ac.id

## ABSTRACT

Javanese, a regional language in Indonesia spoken by over 100 million people, is classified as a low-resource language, presenting significant challenges in the development of effective speech recognition systems due to limited linguistic resources and data. Furthermore, the presence of noise is a significant factor that impacts the performance of speech recognition systems. This study aims to develop a speech recognition model for the Javanese language, focusing on a syllable-based approach using Mel Frequency Cepstral Coefficients (*MFCC*) for audio feature extraction and Convolutional Neural Networks (*CNNs*) methods for classification. Additionally, it will analyze how different types of colored noise: white gaussian, pink, and brown, when added to the audio, impact the model's accuracy. The results showed that the proposed method reached a peak accuracy of 81% when tested on the original audio (audio without any synthetic noise added). Moreover, in noisy audio, model accuracy improves as noise levels decrease. Interestingly, with brown noise at a 20 dB SNR, the model's accuracy slightly increases to 83%, representing a 2.47% improvement over the original audio. These results demonstrate that the proposed syllable-based method is a promising approach for real-world applications in Javanese speech recognition, and the slight accuracy improvement in noisy conditions suggests potential regularization effects.

**Keywords :** *javanese speech recognition;, MFCC; convolutional neural networks (CNNs); low resource language; color of noise.*

## 1. INTRODUCTION

Javanese is one of the regional languages in Indonesia used by the Javanese people for daily communication. The population of the Javanese people currently exceeds 100 million. The rapidly developing voice recognition technology should also be applicable to the Javanese language. To achieve this, more research needs to be conducted, however, the limited amount of research data sources might be one of the reasons for the lack of studies related to speech recognition in the Javanese language.

Speech recognition refers to the ability of computers or software to understand and interpret human speech. Here are some studies related to the topic of this research. Study [1] reviews Arabic Automatic Speech Recognition (ASR), highlighting HMM, GMM, CNNs, and RNN techniques. [2] develops an end-to-end ASR system for the Indonesian language using deep learning, specifically residual networks and Bi-GRU. Another study [3] combines CNNs with MFCC for ASR, using data from the Kaggle TensorFlow Speech Recognition Challenge. Additionally, [4] reviews deep learning in speech recognition, analyzing 174 papers from 2006 to 2018, noting the common use of public databases and MFCC for feature extraction

In terms of the Javanese language, several studies have been conducted related to speech recognition in the Javanese language. Study [5] enhanced gender recognition accuracy among Javanese speakers by using SVD and Deep Learning. In [6], a database was developed for emotion recognition in Javanese speech, achieving a 90% accuracy rate with Neural Networks. Study [7] created an Android application that translates Indonesian speech into Javanese script, with an accuracy of 82.46%. Additionally, [8] investigated speech recognition and synthesis for Javanese and other Indonesian languages using a cross-lingual approach. Lastly, [9] developed a speech recognition system for regional dialects, noting improved performance with Minang and Javanese dialects.

A crucial aspect of speech recognition is managing the challenge posed by the presence of noise. Several works have explored challenges and improvements in handling noisy environments for speech recognition. [10] focused on accurately detecting the start and end of speech signals amid background noise by comparing traditional algorithms with neural network-based methods. [11] introduced an improved MFCC technique called DDM-MFCC to enhance speech and speaker recognition accuracy in noisy conditions, tested specifically on Thai digits. Another study [12] found that MFCC features outperform RASTA-PLP when using a CNN-based approach for recognizing Bangla speech commands in noisy settings. Additionally, [13] demonstrated that using CNNs with log-Mel-spectrogram features effectively recognizes urban noises better than traditional methods, highlighting the strength of advanced neural networks in complex acoustic environments

In speech recognition technology feature extraction plays a role using MFCC which is a common approach for this purpose. MFCC has been applied in noisy speaker recognition [14], improving speech recognition with denoising [15], and analyzing noise effects on ALS and Parkinson's detection [16]. MFCC also helps improve children's speech recognition [17] and enhances remote speaker recognition with Polar error-correcting codes [18]. It's used for identifying fetal gender from heart sounds [19] and in deep learning for speaker identification [20]. Additionally, MFCC with feature warping improves speaker verification in noisy settings [21], and a new method using MFCC and parameter transfer addresses performance issues in speech emotion recognition [22].

CNNs has been widely applied in speech recognition. For example, in Speech Emotion Recognition (SER), a 3D CNN-based system was introduced in [23], while [24] proposed a 1D CNN model for feature extraction. Study [25] developed a SER system for the Algerian dialect using a new dataset and deep learning models like CNNs, LSTMs, and BLSTMs. Moreover, [26] enhanced a CNN algorithm with a Hidden Markov Model (HMM) to improve speech recognition accuracy, and [27] designed a robust multilingual speech recognition system for air traffic control. A combination of CNNs and sequence models was employed in [28] for Nepali speech recognition, while [29] utilized a deep CNNs model to recognize syllable audio from children, optimizing it to prevent overfitting.

Based on the explanation above, it is known that research on Javanese language recognition is still very limited, and no research has yet been conducted on Javanese language recognition using a syllable-based approach. Additionally, noise is also an important factor that needs to be studied further in the development of speech recognition. To overcome this challenge, this study will apply a syllable-based approach to Javanese speech recognition using MFCC and CNNs, combined with noise analysis. MFCC and CNNs were chosen because, according to the literature review, they are capable of delivering good results.

Voice data will be collected from natural environments with various background interference conditions. However, it will be limited to Javanese speech with specific sentences, focusing on speech recognition through a syllable-based approach. Extract audio feature using MFCC and use CNNs model to recognize original voice and voices modified by synthetic noise: white gaussian noise, pink noise, and brown noise. This research is expected to demonstrate good detection accuracy from the developed model and explain the impact of various types of noise on the model's accuracy.

## 2. METHODS

### 2.1. Data Collection

The Dataset Compilation was conducted by collecting Javanese speech recordings using the WhatsApp audio recording feature, which can represent real-world conditions in everyday use. WhatsApp recordings may contain background noise, as they are usually recorded in natural environments. This noise can vary widely, from household sounds to outdoor sounds and public spaces. The acoustic conditions of the recording can vary based on the location and the device used to record. Factors such as echo, reverberation, and distance from the microphone can affect the audio quality. The volunteers spoke Javanese naturally as they normally would, thus reflecting real-world usage patterns.

The dataset will consist of recordings from adult men and women, each uttering a specific sentence "kowe uwis mangan sego tapi durung wareg" ("you have eaten rice but are not full"). This sentence was chosen because of its representative phonetic content and common usage in Javanese and it includes all vowel samples (a, i, u, e, o), making it well-suited for capturing a wide range of speech patterns. Additionally, its familiarity in everyday conversations ensures the model's practical relevance for real-world applications. The total number of volunteers was 37, each saying a predetermined sentence once. The audio recordings were in Ogg format with a sample rate of 22050.

### 2.2. Data Preprocessing

This study adopts a syllable-based approach. The following sections outline the key steps taken during the data preprocessing phase:

a. Each audio recording contains the Javanese sentence "kowe uwis mangan sego tapi durung wareg". Figure 1 shows the audio data.

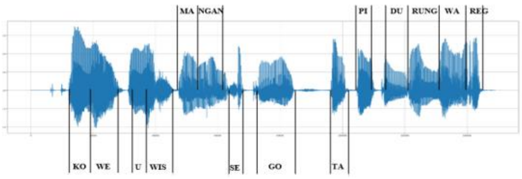


**Figure 1.** *Audio and the syllabel*

The start and end points of the frames for each syllable "ko," "we," "u," "wis," "ma," "ngan," "se," "go," "ta," "pi," "du," "rung," "wa," and "reg" will be identified. This analysis process is performed manually. The start and end frame data for each syllable will be collected with corresponding labels.

b. Once the syllable data has been collected, each syllable is divided into multiple segments, with the first segment starting from the initial sound position and having a length of 2048. The next segment will start 512 points from the starting point of the previous segment, and this process continues until the end of the sound is reached. With this method, the speech recognition process can be considered continuous, as it can detect continuous speech without requiring pauses or gaps between words. Figure 2 shows the method used in preprocessing.

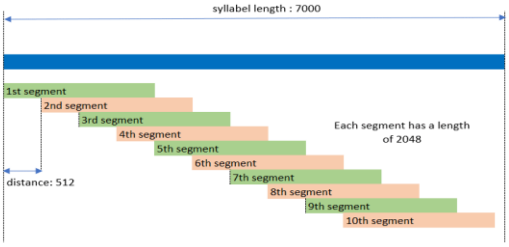


**Figure 2.** *Preprocessing of dataset*

The data preprocessing process produced a dataset of 3338 from 37 sentences, with each sentence containing 14 syllables. Figure 3 below show the distribution of the label.

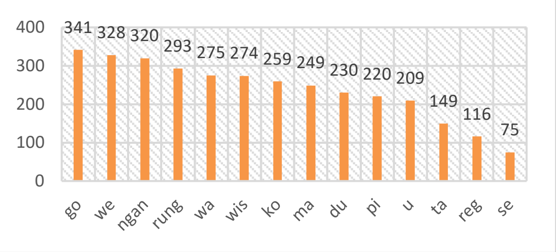


**Figure 3.** *Label/class distribution*

## 2.3. Noise Processing

After the process of divides each syllable into segments, it will be followed by the process of adding three types of synthetic noise to the dataset, namely white Gaussian noise, pink noise, and brown noise which were also studied in research [10].

a. Gaussian white noise is a type of white noise where the amplitude distribution follows a Gaussian (normal) distribution. This means the values of the noise at any point in time are drawn from a Gaussian probability density function.
b. The frequency spectrum of pink noise is linear on a logarithmic scale and maintains equal power per octave, resulting in a decrease in power as the frequency increases. This causes pink noise to be less intense at higher frequencies than white noise.
c. Brown noise is a type of noise where the power density decreases by 6.02 dB per octave as the frequency increases (following a frequency density proportional to $1/f^2$). This noise is sometimes also known as "red noise".

Each type of noise is added at six different Signal-to-Noise Ratio (SNR) levels: -5 dB, 0 dB, 5 dB, 10 dB, 15 dB, and 20 dB. The purpose of using these varying SNR levels is to simulate different levels of noise interference, ranging from highly noisy conditions (at -5 dB) to relatively clean conditions (at 20 dB), enabling a comprehensive analysis. With the addition of this noise, there will be 4 types of audios: 1 original audio and 3 audio that has been mixed with noise. Noise plays a critical role in this research as it directly influences the accuracy and robustness of the speech recognition system being developed. This study use “Colorednoise” as python library to generater the noise. Figure 4 below shows the example of original audio and noisy audio at each SNR level, from the top original audio, -5 dB noise until 20 dB.

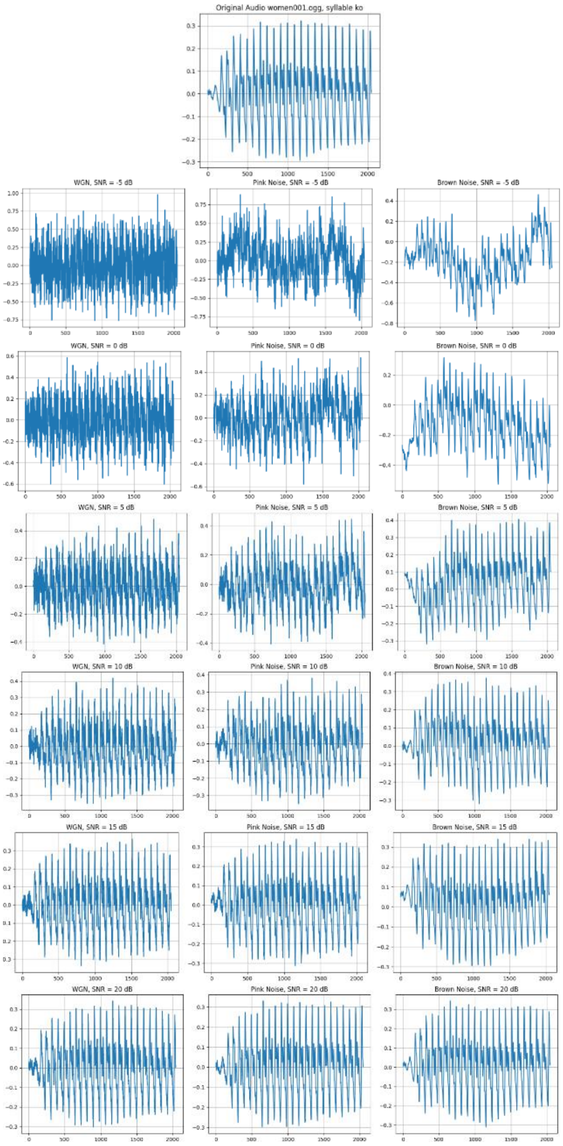


**Figure 4.** *Original audio and noisy audio*

### 2.4. MFCC Feature Extraction

MFCC will extract audio features from each syllable segment. MFCC are coefficients that collectively make up a Mel-frequency cepstrum, which is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency [19]. This is expected to effectively detect and distinguish the syllable structure in the Javanese language. The MFCC feature [5] extraction process is illustrated in Figure 5.

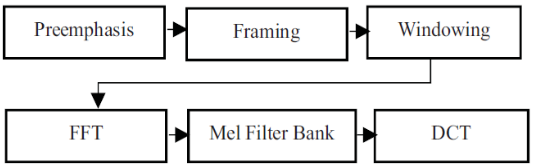


**Figure 5.** *MFCC feature extraction*

a. Pre-emphasis

The purpose of pre-emphasis is to boost the higher frequencies in the input speech signal, enhancing their magnitude within the frequency spectrum, as higher frequencies typically have lower magnitudes compared to the lower ones [3]. The pre-emphasis filter can be mathematically represented as follows:

$$y(t)=x(t)-\alpha x(t-1)$$

In this context, y represents the output speech signal, x is the input signal, and α is the coefficient, usually ranging between 0.95 and 0.97.

b. Framing

After pre-emphasis, the next step is framing, where the acoustic signal is divided into short segments of 20-40 milliseconds. This is important because frequency changes occur very quickly, and it wouldn't make sense to apply FFT to the entire speech signal at once [3].

c. Windowing

After breaking down the speech signal into sub-frames, the Hamming window can be mathematically expressed as follows [3]:

$$w_n = 0.54 - 0.46\, cos\left(\frac{2\pi n}{N-1}\right)$$

where N represents the length of the window.

d. FFT

The Fast Fourier Transform (FFT) is utilized to determine the frequency spectrum for each frame, and then the power spectrum is computed as follows:

$$p = \frac{|FFT(x_i)|^2}{2N}$$

where N refers to the number of FFT points, commonly 256 or 512, and xi denotes the ith sub-frame of the signal x.

e. Mel Filter Bank

The power spectrum is processed through a filter bank composed of triangular filters, which are spaced according to the Mel scale. This scale more accurately mimics the human ear's response compared to a linear frequency scale. Typically, a Mel filter bank consists of 40 triangular filters [30]. Figure 6 shows the response of the triangular Mel-scale filter bank [3].

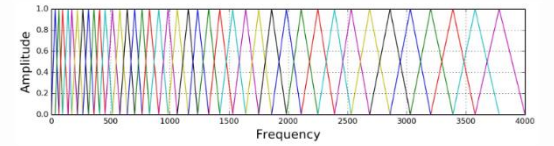


**Figure 6.** *The response of the triangular mel-scale filter bank*

f. DCT

The last process is the discrete cosine transform (DCT) that generates the MFCC coefficient. Typically, the first 12-13 coefficients are used, discarding the rest which usually contain less significant information. The DCT is computed by the below [30]:

$$x(k) = \sum_{n=0}^{N-1} x_{n^*}\, COS\left({}^{2\pi jnk}/_{N}\right), \qquad k = 1,2,3 \ldots N-1$$

This study use the MFCC library from Librosa, running on Python, to extract audio features from the dataset. The extraction process follows the default parameters provided by Librosa. Figure 7 shows an example of MFCC result.

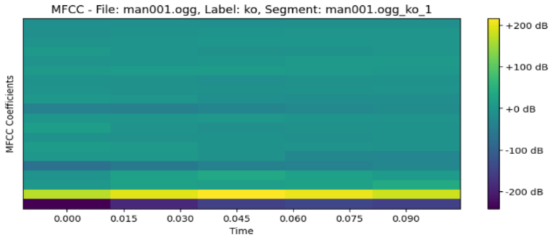
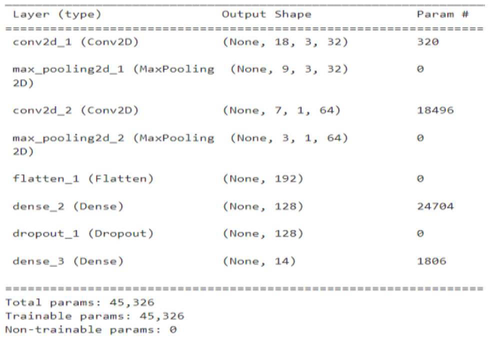


**Figure 7.** *Sample MFCC of label "Ko"*

### 2.5. Model Development

The model is built with a 2D convolutional layer featuring 32 filters and kernel size 3x3, followed by a max pooling layer with a 2x1 pool size. Then, the second Conv2D layer with 64 filters and kernel size 3x3, followed by a max pooling layer with a 2x1 pool size. The architecture also includes a flatten layer, a fully connected layer with 128 neurons, a dropout layer, and an output layer. Figure 8 shows the summary of the model.



```
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_1 (Conv2D)           (None, 18, 3, 32)         320

 max_pooling2d_1 (MaxPooling  (None, 9, 3, 32)         0
 2D)

 conv2d_2 (Conv2D)           (None, 7, 1, 64)          18496

 max_pooling2d_2 (MaxPooling  (None, 3, 1, 64)         0
 2D)

 flatten_1 (Flatten)         (None, 192)               0

 dense_2 (Dense)             (None, 128)               24704

 dropout_1 (Dropout)         (None, 128)               0

 dense_3 (Dense)             (None, 14)                1806

=================================================================
Total params: 45,326
Trainable params: 45,326
Non-trainable params: 0
```

**Figure 8.** *Model's summary*

After the audio feature extraction process, the data will be continued as input to the CNNs model. The data will be divided into 2-part, 1st 80% for training data and the 2nd 20% for test data. In the model training process, validation will be set at 20%.

### 2.6. Model Evaluation

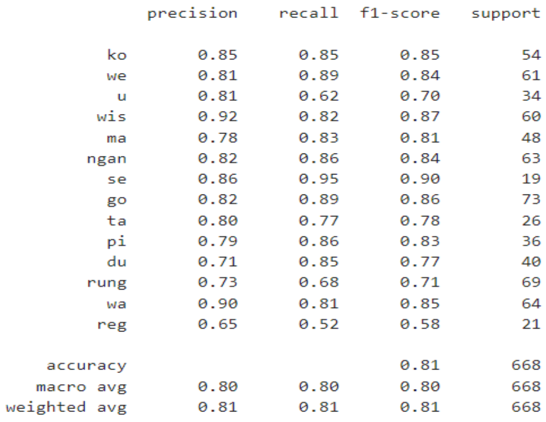
Once the model training was completed, the trained model was evaluated using both the confusion matrix and the classification report. The confusion matrix provides a detailed understanding of the model's classification performance, including true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). This matrix allows for the derivation of key performance metrics, such as accuracy, which represents the overall correctness of the model's predictions and is calculated as (TP + TN) / (TP + TN + FP + FN). Additionally, the classification report offers a comprehensive analysis of the model's precision, recall, F1-score, and support for each class, giving a deeper insight into the model's effectiveness across different evaluation criteria.

## 3. RESULTS AND DISCUSSION

We aim to study the performance of the model in recognizing speech from the original audio, specifically the original recordings from WhatsApp, as well as from a combination of original audio and noise. The impact of varying SNR noise levels on the model's performance will be evaluated. The model was trained over two different epochs, 50 and 200. First, training with 50 epochs provides insight into how the model performs with a shorter training period, potentially identifying patterns early. On the other hand, training with 200 epochs allows the model more time to learn complex patterns.

### 3.1. Original Audio

The results from the classification report, as shown in Figure 9, indicate that the overall accuracy (F1-score) of the model at 200 epochs is 81%, which is only slightly higher than the accuracy obtained from training with 50 epochs, which was 80%. Moreover, the confusion matrix is shown in Figure 10 below.

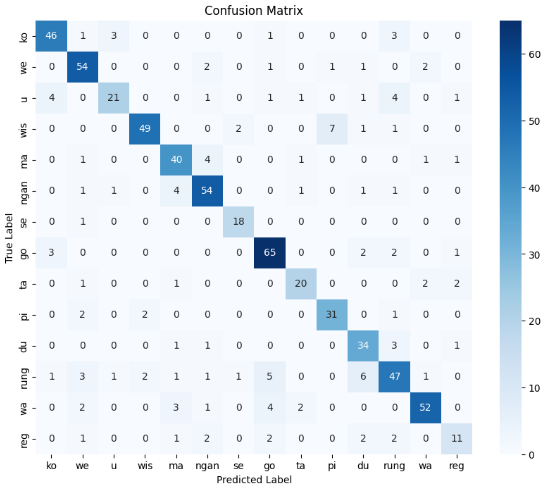


```
              precision    recall  f1-score   support

          ko       0.85      0.85      0.85        54
          we       0.81      0.89      0.84        61
           u       0.81      0.62      0.70        34
         wis       0.92      0.82      0.87        60
          ma       0.78      0.83      0.81        48
        ngan       0.82      0.86      0.84        63
          se       0.86      0.95      0.90        19
          go       0.82      0.89      0.86        73
          ta       0.80      0.77      0.78        26
          pi       0.79      0.86      0.83        36
          du       0.71      0.85      0.77        40
        rung       0.73      0.68      0.71        69
          wa       0.90      0.81      0.85        64
         reg       0.65      0.52      0.58        21

    accuracy                           0.81       668
   macro avg       0.80      0.80      0.80       668
weighted avg       0.81      0.81      0.81       668
```

**Figure 9.** *Classification report at 200 epochs*



| | ko | we | u | wis | ma | ngan | se | go | ta | pi | du | rung | wa | reg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ko | 46 | 1 | 3 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 3 | 0 | 0 |
| we | 0 | 54 | 0 | 0 | 0 | 2 | 0 | 1 | 0 | 1 | 1 | 0 | 2 | 0 |
| u | 4 | 0 | 21 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 4 | 0 | 1 |
| wis | 0 | 0 | 0 | 49 | 0 | 0 | 2 | 0 | 0 | 7 | 1 | 1 | 0 | 0 |
| ma | 0 | 1 | 0 | 0 | 40 | 4 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| ngan | 0 | 1 | 1 | 0 | 4 | 54 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| se | 0 | 1 | 0 | 0 | 0 | 0 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| go | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 65 | 0 | 0 | 2 | 2 | 0 | 1 |
| ta | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 20 | 0 | 0 | 0 | 2 | 2 |
| pi | 0 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 31 | 0 | 1 | 0 | 0 |
| du | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 34 | 3 | 0 | 1 |
| rung | 1 | 3 | 1 | 2 | 1 | 1 | 1 | 5 | 0 | 0 | 6 | 47 | 1 | 0 |
| wa | 0 | 2 | 0 | 0 | 3 | 1 | 0 | 4 | 2 | 0 | 0 | 0 | 52 | 0 |
| reg | 0 | 1 | 0 | 0 | 1 | 2 | 0 | 2 | 0 | 0 | 2 | 2 | 0 | 11 |

**Figure 10.** *Confusion matrix at 200 epochs*

The "se" class is the best performing, with the highest F1-score of 0.90, supported by the highest recall of 0.95 and a strong precision of 0.86. Otherwise, the "reg" class is the worst performing, with the lowest F1-score of 0.58, a recall of 0.52, and a precision of 0.65, indicating significant issues in both recognizing and correctly identifying this class.

### 3.2. Noisy Audio

#### a. White gaussian noise

Summary of the model's accuracy on the mix original audio and additional white gaussian noise (WGN) as shown in Figure 11.

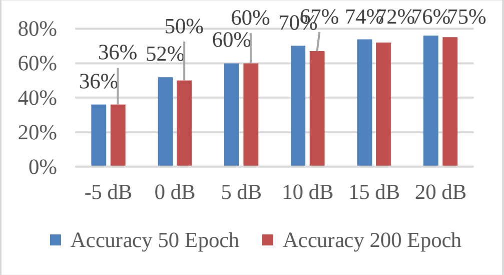


**Figure 11.** *Accuracy of WGN*

At lower SNR levels -5dB to 5 dB, both models (50 and 200 epochs) show relatively low accuracy, achieving 36% to 60% accuracy. When SNR level increases (i.e., less noise), the accuracy improves significantly for both models. The 50-epoch model reaches its highest accuracy of 76% at 20 dB, while the 200-epoch model reaches 75% at 20 dB. The model's accuracy is significantly affected by the SNR level, with better performance observed as the SNR level increases (indicating less noise).
Another aspect to be analyzed is class performance. Table 1 presents the performance of each class during the model's training process.

**Table 1.** Class performance of WGN audio

| No | Epoch | SNR Level (dB) | Best Performing Class | Worst Performing Class |
|---|---|---|---|---|
| 1 | 50 | -5 dB | 'wa' (TP: 34, F1: 0.50) | 'reg' (TP: 2, F1: 0.10) |
| 2 | 50 | 0 dB | 'wa' (TP: 40, F1: 0.66) | 'reg' (TP: 4, F1: 0.22) |
| 3 | 50 | 5 dB | 'ko' (TP: 42, F1: 0.80) | 'reg' (TP: 9, F1: 0.40) |
| 4 | 50 | 10 dB | 'se' (TP: 16, F1: 0.80) | 'reg' (TP: 6, F1: 0.40) |
| 5 | 50 | 15 dB | 'se' (TP: 24, F1: 0.83) | 'u' (TP: 14, F1: 0.52) |
| 6 | 50 | 20 dB | 'se' (TP: 16, F1: 0.89) | 'reg' (TP:11, F1: 0.54) |
| 7 | 200 | -5 dB | 'wa' (TP: 37, F1: 0.54) | 'reg' (TP: 1, F1: 0.06) |
| 8 | 200 | 0 dB | 'wa' (TP: 46, F1: 0.66) | 'u' (TP: 12, F1: 0.30) |
| 9 | 200 | 5 dB | 'se' (TP: 14, F1: 0.80) | 'reg' (TP: 7, F1: 0.41) |
| 10 | 200 | 10 dB | 'ta' (TP: 21, F1: 0.84) | 'u' (TP: 15, F1: 0.43) |
| 11 | 200 | 15 dB | 'wa' (TP: 51, F1: 0.80) | 'rung' (TP: 41, F1: 0.59) |
| 12 | 200 | 20 dB | 'wa' (TP: 56, F1: 0.84) | 'reg' (TP: 10, F1: 0.53) |

The "wa" class consistently performs well across different SNR levels and epochs, particularly in noisier conditions. Whereas, the "reg" class frequently underperforms, especially at lower SNR levels and higher epochs, making it the most challenging class for the model.

#### b. Pink noise

Summary of the model's accuracy on the mix original audio and additional pink noise (PN) as shown in Figure 12.

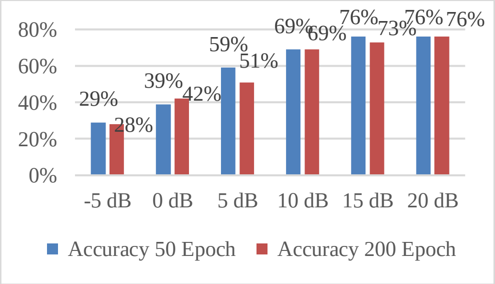


**Figure 12.** *Accuracy of PN*

The model shows improved accuracy as the SNR level increases, with performance peaking at 76% accuracy for the highest SNR levels (15 dB and 20 dB). At lower SNR levels (-5 dB and 0 dB), the model's accuracy drops considerably, especially for the 200-epoch model.

The class performance is shown in Table 2 below. The "se" and "we" classes consistently perform well across different SNR levels, especially at higher SNR levels and longer epochs. Otherwise, the "u" and "reg" classes often show the weakest performance, especially when the noise levels are high and the training runs for more epochs, suggesting they have more difficulty handling noisy environments.

**Table 2.** Class performance of PN audio

| No | Epoch | SNR Level (dB) | Best Performing Class | Worst Performing Class |
|---|---|---|---|---|
| 1 | 50 | -5 dB | ‘wa’ (TP: 26, F1: 0.40) | ‘rung’ (TP: 11, F1: 0.18) |
| 2 | 50 | 0 dB | ‘ko’ (TP: 27, F1: 0.54) | ‘u’ (TP: 7, F1: 0.17) |
| 3 | 50 | 5 dB | ‘ko’ (TP: 34, F1: 0.72) | ‘reg’ (TP: 6, F1: 0.37) |
| 4 | 50 | 10 dB | ‘se' (TP: 15, F1: 0.81) | ‘reg’ (TP: 7, F1: 0.40) |
| 5 | 50 | 15 dB | ‘se’ (TP: 17, F1: 0.89) | ‘u’ (TP: 19, F1: 0.54) |
| 6 | 50 | 20 dB | ‘we’ (TP: 53, F1: 0.83) | ‘reg’ (TP: 9, F1: 0.56) |
| 7 | 200 | -5 dB | ‘se' (TP: 9, F1: 0.53) | ‘rung’ (TP: 6, F1: 0.11) |
| 8 | 200 | 0 dB | ‘se' (TP: 11, F1: 0.71) | ‘u’ (TP: 6, F1: 0.15) |
| 9 | 200 | 5 dB | ‘wa’ (TP: 43, F1: 0.67) | ‘u’ (TP: 10, F1: 0.25) |
| 10 | 200 | 10 dB | ‘se' (TP: 16, F1: 0.89) | ‘u’ (TP: 13, F1: 0.39) |
| 11 | 200 | 15 dB | ‘se’ (TP: 18, F1: 0.92) | ‘reg’ (TP: 9, F1: 0.53) |
| 12 | 200 | 20 dB | ‘we’ (TP: 55, F1: 0.90) | ‘reg’ (TP: 10, F1: 0.57) |

c. Brown noise

Summary of the model's accuracy on the mix original audio and additional brown noise (BN) as shown in Figure 13. The model's accuracy generally improves as the SNR level increases, with the highest performance observed at 20 dB of 83% accuracy.

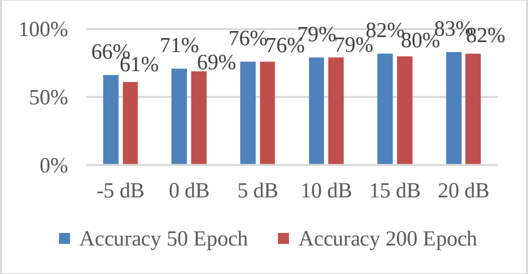


**Figure 13.** *Accuracy of BN*

Additionally, Table 3 highlights the performance of each class during the model's training process. The "se" class consistently ranks as the top performer, especially at lower SNR levels with both 50 and 200 epochs, demonstrating its resilience in noisy environments. The "reg" class consistently has difficulties, often ending up as the weakest performer, especially in situations with lower SNR levels and more noise.

**Table 3.** *Class performance of BN audio*

| No | Epoch | SNR Level (dB) | Best Performing Class | Worst Performing Class |
|---|---|---|---|---|
| 1 | 50 | -5 dB | ‘se' (TP: 17, F1: 0.85) | ‘reg’ (TP: 7, F1: 0.42) |
| 2 | 50 | 0 dB | ‘se’ (TP: 17, F1: 0.92) | ‘reg’ (TP: 8, F1: 0.50) |
| 3 | 50 | 5 dB | ‘wa’ (TP: 56, F1: 0.86) | ‘reg’ (TP: 11, F1: 0.55) |
| 4 | 50 | 10 dB | ‘se’ (TP: 19, F1: 0.93) | ‘reg’ (TP: 10, F1: 0.57) |
| 5 | 50 | 15 dB | ‘ko’ (TP: 51, F1: 0.90) | ‘reg’ (TP: 12, F1: 0.65) |
| 6 | 50 | 20 dB | ‘ko’ (TP: 51, F1: 0.90) | ‘reg’ (TP: 12, F1: 0.65) |
| 7 | 200 | -5 dB | ‘se' (TP: 14, F1: 0.76) | ‘u’ (TP: 12, F1: 0.35) |
| 8 | 200 | 0 dB | ‘se’ (TP: 16, F1: 0.86) | ‘u’ (TP: 12, F1: 0.45) |
| 9 | 200 | 5 dB | ‘se’ (TP: 16, F1: 0.89) | ‘rung’ (TP: 40, F1: 0.63) |
| 10 | 200 | 10 dB | ‘ta’ (TP: 21, F1: 0.88) | ‘reg’ (TP: 9, F1: 0.56) |
| 11 | 200 | 15 dB | ‘we’ (TP: 54, F1: 0.88) | ‘du’ (TP: 26, F1: 0.66) |
| 12 | 200 | 20 dB | ‘ngan’ (TP: 57, F1: 0.88) | ‘reg’ (TP: 10, F1: 0.65) |

## 3.3. Discussion

Figure 14 below shows a comparison of the model's accuracy for various audio conditions.

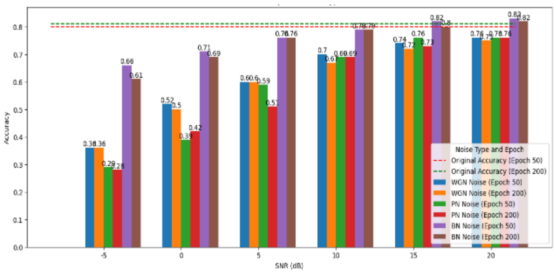


**Figure 14.** *Accuracy original vs noisy audio*

The model accuracy (F1-score) for the original audio is 80% at 50 epochs and 81% at 200 epochs. For audio with added noise, at SNR levels from -5 dB to 10 dB, the model accuracy for all types of noise does not exceed the accuracy of the original audio, even though the model accuracy gradually increases as the SNR value rises, indicating a reduction in noise in the audio. At 15 dB and 20 dB SNR, there is noise-added audio where the model accuracy is higher than the accuracy for the original audio, specifically with Brown Noise at 50 epochs, achieving 82% at 15 dB and 83% at 20 dB. At 20 dB with 200 epochs, the accuracy is 82%. For White Gaussian Noise and Pink Noise, no accuracy exceeds that of the original audio.

Based on the results above, there is a slight difference from the study [10], where in that study, each algorithm tested was able to achieve the best average accuracy in a white Gaussian noise environment. However, in this research, the voice recognition detection on audio with added brown noise produced the highest accuracy compared to audio with added white Gaussian noise or the original audio. Brown noise may interfere less with these critical features, allowing the model to better recognize speech patterns.

For the class performance, the “se” class consistently stands out with 45% accuracy as one of the top performers across different noise types, SNR levels, and training epochs, demonstrating resilience in a variety of conditions. The sounds in "se" (like the "s" sound) are quite common in many languages and may be more easily learned by the model during training.

On the other hand, the “reg” class often struggles, frequently having difficulty handling noise regardless of the SNR level or number of epochs, with 60% appearance as the worst-performing class. This syllable begins with the sound “r”, which seems to make it difficult for the model to identify its characteristics, as mentioned in [31]. To ensure this, further experiments need to be conducted with more Javanese words containing the syllable “se” and “reg”.

## CONCLUSION

This research has contributed to Javanese speech recognition studies, especially in addressing noise robustness. The experimental results have shown that the syllable-based approach in Javanese speech recognition using MFCC and CNNs achieved good accuracy of 81% when tested on original audio, sound that recorded from WhatsApp application without any synthetic noise added.

In terms of noise impact evaluation, it has a significant effect on performance. Among the three noise variants introduced at different SNR levels, the model's accuracy improves as the SNR increases or as the noise level decrease. Interestingly, when Brown Noise was added at 20 dB, the model's accuracy slightly improved to 83%, exceeding the accuracy obtained with the original audio and representing a 2.47% improvement. This suggests a possible regularizing effect that enhances the model's generalization. However, other types of noise and lower SNR levels generally caused a drop in accuracy, highlighting the model's sensitivity to noise interference.

This study focused on recognizing syllables, and with the impressive accuracy results, we see potential for developing a full speech recognition system for the Javanese language, particularly for mobile apps. For future work, expanding the dataset with a more complete set of Javanese syllables, utilizing advanced systems like large language models (LLMs), and exploring alternative feature extraction techniques (e.g., RASTA-PLP) could be beneficial. Additionally, employing models such as transformers, RNNs, or Transformer-based architectures, along with incorporating noise reduction techniques to clean the audio before processing, may enhance the model's ability to handle complex noise patterns and improve overall accuracy.

## REFERENCES

[1] A. Rahman, M. M. Kabir, M. F. Mridha, M. Alatiyyah, H. F. Alhasson and S. S. Alharbi, "Arabic Speech Recognition: Advancement and Challenges," *IEEE Access*, vol. 12, pp. 39689 -39716, 2024.

[2] M. I. F. Rifqi Adiwidjaja, "End-to-end indonesian speech recognition with convolutional and gated recurrent units," in *IOP Publishing*, Medan, Sumatera Utara, 2019.

[3] A. Mahmood and U. Köse, "Speech recognition based on convolutional neural networks and MFCC algorithm," Advances *in Artificial Intelligence Research (AAIR)*, vol. 1, no. 1, pp. 6 -12, 2021.

[4] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh and K. Shaalan, "Speech Recognition Using Deep Neural Networks: A Systematic Review," *IEEE Access*, vol. 7, pp. 19144 - 19165, 2019.

[5] K. Nugroho, E. Noersasongko, Purwanto, Muljono and H. A. Santoso, "Javanese Gender Speech Recognition Using Deep Learning And Singular Value Decomposition," in *IEEE*, Semarang, Indonesia, 2019.

[6] F. Arifin, A. S. Priambodo, A. Nasuha, A. Winursito and T. S. Gunawan, "Development of Javanese Speech Emotion Database (Java-SED)," Indonesian *Journal of Electrical Engineering and Informatics (IJEEI),* vol. 10, pp. 584 - 591, 2022.

[7] A. Nursetyo and D. R. I. M. Setiadi, "LatAksLate: Javanese Script Translator based on Indonesian Speech Recognition using Sphinx-4 and Google API," in *IEEE*, Yogyakarta, 2018.

[8] S. Novitasari, A. Tjandra, S. Sakti and S. Nakamura, "Cross-Lingual Machine Speech Chain for Javanese, Sundanese, Balinese, and Bataks Speech Recognition and Synthesis," in *European Language Resources association*, Marseille, France, 2020.

[9] A. M. Warohma, P. Kurniasari, S. Dwijayanti, Irmawan and B. Y. Suprapto, "Identification of Regional Dialects Using Mel Frequency Cepstral Coefficients (MFCCs) and Neural Network," in *IEEE*, Semarang, Indonesia, 2018.

[10] T. Zhang, Y. Shao, Y. Wu, Y. Geng and L. Fan, "An overview of speech endpoint detection algorithms," *Applied Acoustics,* vol. 160, pp. 1 -15, 2020.

[11] A. Nosan and S. Sitjongsataporn, "Speech Recognition Approach using Descend-Delta-Mean and MFCC Algorithm," in *IEEE*, Pattaya, Thailand, 2019.

[12] Maruf, M. Raffael, Faruque, M. Omar, M. Salman, N. Nelima, M. Muhtasim and M. Pervez, "Effects of Noise on RASTA-PLP and MFCC based Bangla ASR Using CNN," in *IEEE*, Dhaka, Bangladesh., 2020.

[13] J. Cao, M. Cao, J. Wang, C. Yin, D. Wang and P.-P. Vidal, "Urban noise recognition with convolutional neural network," *Multimedia Tools and Applications*, 2018.

[14] F. Amelia and D. Gunawan, "DWT-MFCC Method for Speaker Recognition System with Noise," in *IEEE*, Sarawak, Malaysia, 2019.

[15] R. Hidayat, A. Bejo, S. Sumaryono and A. Winursito, "Denoising Speech for MFCC Feature Extraction Using Wavelet Transformation in Speech Recognition System," in *IEEE*, Bali, Indonesia, 2018.

[16] T. B. e. al, "Effect of Noise and Model Complexity on Detection of Amyotrophic Lateral Sclerosis and Parkinson’s Disease Using Pitch and MFCC," in *IEEE*, Toronto, Canada, 2021.

[17] H. M. S. Naing, Y. Miyanaga, R. Hidayat and B. Winduratna, "Filterbank Analysis of MFCC Feature Extraction in Robust Children Speech Recognition," in *IEEE*, Quezon City, Philippines, 2019.

[18] N. Wankhede and S. Wagh, "Enhancing Biometric Speaker Recognition Through MFCC Feature Extraction and Polar Codes for Remote Application," *IEEE Access*, vol. 11, pp. 133921-133930, 2023.

[19] M. M. Azmy, "Gender of Fetus Identification Using Modified Mel-Frequency Cepstral Coefficients Based on Fractional Discrete Cosine Transform," *IEEE Access*, vol. 12, pp. 48158-48164, 2024.

[20] M. Barhoush, A. Hallawa and A. Schmeink, "Speaker identification and localization using shuffled MFCC features and deep learning," *Int J Speech Technol*, vol. 26, p. 185–196, 2023.

[21] C. Jiang, L. Ba, X. Tang and D. Wen, "Speaker Verification Using IMNMF and MFCC with Feature Warping Under Noisy Environment," in *IEEE*, Xi'an, China, 2018.

[22] L. Lin and L. Tan, "Multi-distributed speech emotion recognition based on Mel frequency cepstogram and parameter transfer," *Chinese Journal of Electronics*, vol. 31, pp. 155-167, 2022.

[23] M. R. Falahzadeh, E. Z. Farsa, A. Harimi, A. Ahmadi and A. Abraham, "3D Convolutional Neural Network for Speech Emotion Recognition With Its Realization on Intel CPU and NVIDIA GPU," *IEEE Access*, vol. 10, pp. 112460-112471, 2022.

[24] Y. Li, C. Baidoo, T. Cai and G. A. Kusi, "Speech Emotion Recognition Using 1D CNN with No Attention," in *IEEE*, Phuket, Thailand, 2020.

[25] R. Y. Cherif, A. Moussaoui, N. Frahta and M. Berrimi, "Effective speech emotion recognition using deep learning approaches for Algerian dialect," in *IEEE*, Taif, Saudi Arabia, 2021.

[26] Z. Chen, J. Liu and Y. Zhang, "Research on an Improved CNN Speech Recognition System Based on Hidden Markov Model," in *IEEE*, Vientiane, Laos, 2020.

[27] Y. Lin, D. Guo, J. Zhang, Z. Chen and B. Yang, "A Unified Framework for Multilingual Speech Recognition in Air Traffic Control Systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, pp. 3608-3620, 2021.

[28] J. Banjara, K. R. Mishra, J. Rathi, K. Karki and S. Shakya, "Nepali Speech Recognition using CNN and Sequence Models," in *IEEE*, Hyderabad, India, 2020.

[29] S. Yang, M. Lee and H. Kim, "Deep Learning-based Syllable Recognition Framework for Korean Children," in *IEEE*, Jeju Island, Korea (south), 2021.

[30] Z. K. Abdul and A. K. Al-Talabani, "Mel Frequency Cepstral Coefficient and its Applications: A Review," *IEEE Access*, vol. 10, pp. 122136-122158, 2022.

[31] R. W. Schafer and L. R. Rabiner, Digital Processing of Speech Signals, Prentice-Hall, 1978.