IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 6, December 2026, pp. 5678~5691
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i6.pp5678-5691

Influence of distorted speech features on smart wheelchair control

Fitri Utaminingrum[1], Eldad Vikorian[1], Anindya Zulva Larasati[1], Corina Karim[2]
[1]Department of Informatics Engineering, Faculty of Computer Science, Universitas Brawijaya, Malang, Indonesia
[2]Department of Mathematics, Faculty of Science, Technology, and Mathematics, Universitas Brawijaya, Malang, Indonesia

Article Info

Article history:

Received Sep 12, 2025
Revised Aug 15, 2026
Accepted Sep 19, 2026

Keywords:

Assistive technology
Deep learning
Feature extraction
Noisy speech
Smart wheelchair
Speech recognition

ABSTRACT

The growing number of individuals with severe physical disabilities highlights the need for more inclusive and intelligent mobility aids. While voice-controlled wheelchairs offer a hands-free solution for users with limited motor abilities, prior studies often failed to address the impact of environmental noise on voice recognition performance. This gap motivates the present study, which analytically compares four speech feature-extraction schemes: mel-frequency cepstral coefficients (MFCC), gammatone frequency cepstral coefficients (GFCC), bark frequency cepstral coefficients (BFCC), and linear predictive cepstral coefficients (LPCC), paired with three deep-learning models: transformer, residual network with 50-layer version 2 (ResNet50V2), and U-Net. The work is scoped to model training and comparative benchmarking rather than physical hardware deployment, with test accuracy serving as the evaluation metric for every pairing. Results show that the choice of feature extraction method greatly influences system accuracy, with the combination of GFCC and U-Net achieving the highest classification accuracy, reaching 98% on test data. This study contributes practical insights for designing robust and noise-resilient voice recognition systems in smart wheelchairs, with the broader aim of improving independence and quality of life for people with disabilities.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Fitri Utaminingrum
Department of Informatics Engineering, Faculty of Computer Science, Universitas Brawijaya
Veteran road, Malang 65145, Indonesia
Email: f3_ningrum@ub.ac.id

1. INTRODUCTION

According to the World Health Organization (WHO) [1], around 16% of the global population or approximately 1.3 billion people, live with some form of disability. This global issue is also evident in Indonesia, where nationally representative data indicate that approximately 14.8% of the population aged 15 years and above experience some degree of disability, highlighting disability inclusion as a critical national policy concern [2]. Disabilities are commonly classified into four main categories: physical, intellectual, mental, and sensory. Physical disabilities include conditions such as spinal diseases, cerebral palsy (CP), stroke, multiple sclerosis, progressive muscular dystrophy, poliomyelitis, arthritis, and limb amputation [3]. Wheelchairs are essential tools for independent movement and are relied upon by an estimated 80 million individuals living with physical disabilities globally [4]. However, many individuals with severe physical impairments, such as those who are paralyzed under the neck, are unable to operate traditional mobility aids. For these individuals, manual and even joystick-controlled wheelchairs are not viable options, as they lack the motor control required to use such devices [5]. This underscores the need for

more inclusive mobility technologies for individuals with severe motor impairments, beyond conventional wheelchair designs.

Artificial intelligence (AI) has increasingly supported innovations in healthcare and assistive technologies. A literature review by Ktistakis *et al.* [6] provides a comprehensive examination of AI applications aimed at improving healthcare services and enhancing the standard of living for individuals with disabilities and the elderly. Within smart wheelchair development, various hands-free control approaches have emerged, including eye-gaze tracking [7], head gesture control [8], [9], room plate recognition [10], and obstacle detection [11] offering alternatives for users with limited motor skills. However, challenges like environmental sensitivity, complexity, and high cost still hinder their widespread adoption.

Voice-controlled wheelchairs offer a promising alternative for users who cannot rely on visual or motor-based interfaces. By enabling hands-free control through speech, these systems enhance user independence, reduce caregiver burden, and improve quality of life [12]. Previous studies have explored voice-activated systems using Arduino-based voice modules [13], [14], long short-term memory (LSTM) [15], and convolutional neural network (CNN) architectures [16]. Sutikno *et al.* [17] further integrated CNN and LSTM models with mel-frequency cepstral coefficients (MFCC) for improved voice-based wheelchair navigation. However, most of these studies, including Sutikno *et al.* [17], did not explicitly evaluate system performance under varying noise conditions. Although their training data included environments with noise levels from 33.0 to 83.8 decibels, no specific analysis was conducted to assess robustness against environmental noise.

To address the limitations found in prior studies, this research proposes a comprehensive approach that integrates multiple speech feature extraction methods: MFCC, gammatone frequency cepstral coefficients (GFCC), bark frequency cepstral coefficients (BFCC), and linear predictive cepstral coefficients (LPCC) with advanced deep learning architectures, including residual network with 50-layer version 2 ResNet50V2, transformer, and U-Net. MFCC is widely adopted in speech recognition tasks [18], [19], while GFCC offers enhanced robustness in noisy environments due to its use of the gammatone filterbank, which emulates the human auditory system [20]. BFCC and LPCC, on the other hand, provide complementary spectral and linear predictive characteristics that further enrich speech signal representation [21], [22].

Regarding model architectures, ResNet50V2 is known for its strong capability in feature extraction [23]. U-Net excels in capturing temporal and contextual patterns in audio signals [24], and transformers have demonstrated high accuracy in large-scale speech recognition tasks [25]. Prior research has explored some of these components individually. For instance, Dua *et al.* [26] achieved an equal error rate (EER) of 1.78% by combining GFCC with a pretrained ResNet50. However, the system was only evaluated under controlled conditions and not tested in real-world noisy environments, limiting its practical applicability. Similarly, Wang *et al.* [27] reported that streamable transformer variants reduced word error rates by up to 26%, but their experiments were limited to clean datasets. Yang *et al.* [28] showed that a self-attention U-Net model could enhance robustness against adversarial speech attacks, yet its performance under real-world noise conditions remains untested.

Drawing on these findings, this research sets out with two goals: building a voice-controlled wheelchair system that holds up better against noise and systematically examining how different combinations of feature-extraction techniques and model architectures influence that performance. The resulting system is intended to recognize five Indonesian-language navigation commands: "*maju*" (forward), "*kanan*" (right), "*kiri*" (left), "*mundur*" (backward), and "*berhenti*" (stop), reliably even under noisy conditions, as part of a broader smart wheelchair application. Each feature-extraction method highlights different temporal and spectral characteristics of speech, while each model architecture brings its own strengths in capturing sequential, spatial, or contextual patterns. By testing these combinations comprehensively, the study aims to identify the configuration that offers the best balance of accuracy and responsiveness. The broader hope is that this work will improve the dependability and practicality of smart wheelchair systems in real-world settings and offer useful guidance for developing noise-robust voice recognition technology more generally.

2. MATERIALS AND METHOD

2.1. Proposed method

The proposed system initiates with the utilization of a speech navigation dataset comprising predefined voice commands such as forward, right, left, backward, and stop. This dataset serves as the primary input and is subsequently subjected to a series of processing stages to facilitate robust voice command recognition. The final output is a set of control signals that guide the wheelchair's movement according to the recognized commands, as illustrated in Figure 1.

The pipeline begins with audio pre-processing, applying standard signal-enhancement steps to improve the quality and consistency of the input waveform. Feature extraction follows, using four established

techniques in speech processing: MFCC, GFCC, BFCC, and LPCC, chosen for their ability to capture the spectral and cepstral properties that help distinguish voice commands across different acoustic conditions. Once the features are extracted, a normalization step brings them onto a comparable scale, which supports more stable training and faster convergence for the models. The normalized dataset is then divided into training, validation, and testing subsets to support supervised learning and subsequent performance evaluation.

For classification, three separate deep-learning architectures are used: transformer, ResNet50V2, and U-Net, each trained to perform multi-class classification by mapping input feature vectors onto their corresponding voice command labels. Model performance is then assessed at the evaluation stage using a confusion matrix, which quantifies classification accuracy and shows how errors are distributed across classes. Together, this end-to-end setup allows for a thorough analysis of how different feature-extraction techniques and model architectures affect voice command classification, particularly under noisy conditions. The models produced through this process are meant to be incorporated into a smart wheelchair system that can reliably recognize spoken navigation commands in real-world settings.

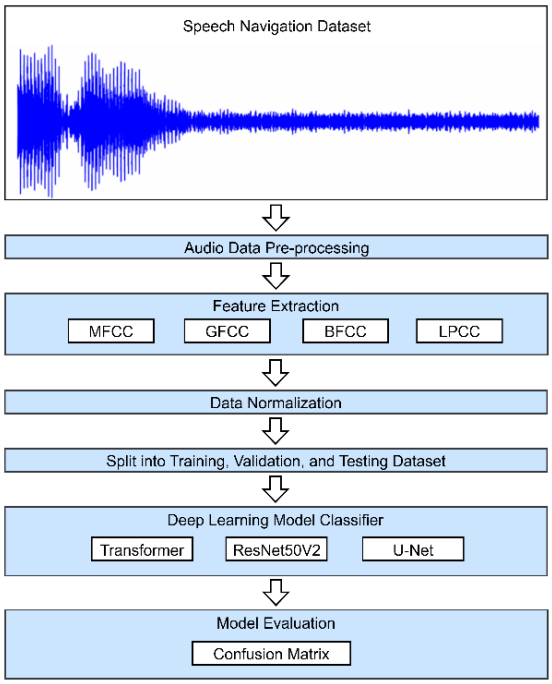

Figure 1. The system framework for voice-based navigation in a smart wheelchair

2.2. Dataset preparation

Audio samples were gathered by recording volunteers directly through a Logitech C270 webcam's built-in microphone. Contributors were students from the Faculty of Computer Science, Universitas Brawijaya, Indonesia and all clips were stored in .wav format for compatibility with standard audio-processing tools. Because the target users are local, every command was spoken in Indonesian to keep the dataset linguistically relevant to its intended application. Recordings were made under two conditions: a quiet environment (<50 dB) and a noisy environment (>50 dB), simulating real-world usage of the smart wheelchair. The dataset consists of 5,433 audio samples categorized into five navigation commands: forward, backward, right, left, and stop. Table 1 provides a detailed breakdown of the sample count for each class.

Table 1. Class-wise distribution

Class	Number of audio samples
Forward	1,108
Backward	1,106
Right	1,097
Left	1,116
Stop	1,006

Every recording went through pre-processing to standardize its length and clarity, including trimming silence and normalizing duration. The resulting dataset was then partitioned 70/20/10 into training, validation, and testing subsets while preserving class balance in each split. Figure 2 illustrates the waveform representations of sample audio from each navigation command category. The command forward is shown in Figure 2(a), backward in Figure 2(b), right in Figure 2(c), left in Figure 2(d), and stop in Figure 2(e).

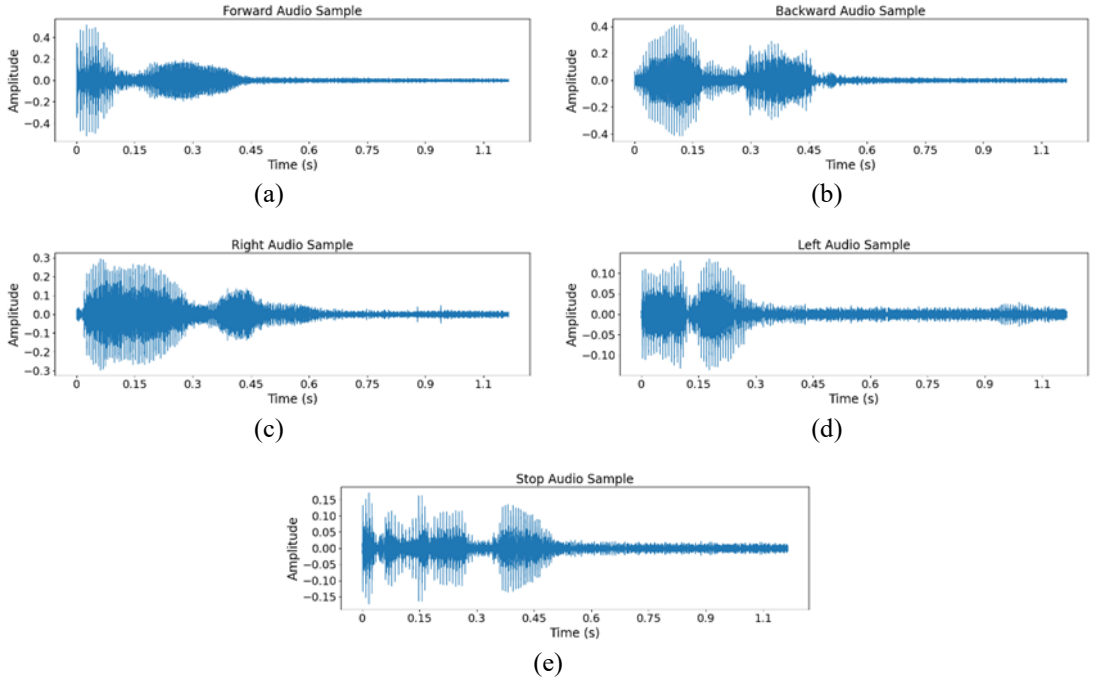

Figure 2. Visualization of the (a) forward dataset, (b) backward dataset, (c) right dataset, (d) left dataset, and (e) stop dataset

2.3. Feature extraction

2.3.1. Mel-frequency cepstral coefficient

MFCC is a widely adopted feature extraction method in automatic speech and speaker recognition [18]. Its design draws on how the human ear perceives pitch nonlinearly, rather than proportionally [19]: perception behaves roughly linearly under 1,000 Hz but shifts to a logarithmic pattern above it. MFCC reproduces this behavior through the mel scale, which stretches the low-frequency range while compressing higher frequencies. The MFCC extraction process consists of several steps, as illustrated in Figure 3. Pre-processing opens with pre-emphasis: a high-pass filter boosts the signal's high-frequency content so the overall spectrum is more balanced. The typical pre-emphasis filter is expressed in (1).

$$S'(n) = S(n) - \alpha \cdot S(n-1) \quad (1)$$

Where $S'(n)$ is the output signal, $S(n)$ is the current sample, $S(n-1)$ is the previous sample, and α is the pre-emphasis coefficient.

Next, the signal is segmented into overlapping frames, typically 20–40 ms in length with 20%–50% overlap, based on the assumption that speech signals are quasi-stationary over short durations [29]. Each frame is subsequently multiplied by a Hamming window, which smooths its edges and reduces spectral leakage. Once windowing is applied, each frame is converted into the frequency domain via the fast Fourier transform (FFT), which provides an efficient means of computing the discrete Fourier transform (DFT), as defined in (2).

$$X_k = \sum_{n=0}^{N-1} X_n \cdot e^{-j\frac{2\pi}{N}kn} \quad (2)$$

Where X_k is the DFT k-th output at, while N represents the total number of input samples. The index n corresponds to a specific point in the time domain, and X_n denotes the n-th sample of the input signal. The

term $e^{-j\frac{2\pi}{N}kn}$ represents the complex exponential function responsible for converting time-domain signals into their corresponding sinusoidal components in the frequency domain.

A bank of mel-scaled triangular filters is then applied to the spectrum, with each filter summing the energy that falls within its band. Taking the logarithm of these values approximates how loudness is perceived. The discrete cosine transform (DCT) compacts this information into a handful of coefficients, producing the final MFCC vector, a compact representation of the frame's short-term power spectrum that stays relatively stable across speakers and noise conditions, which explains its widespread use in speech recognition.

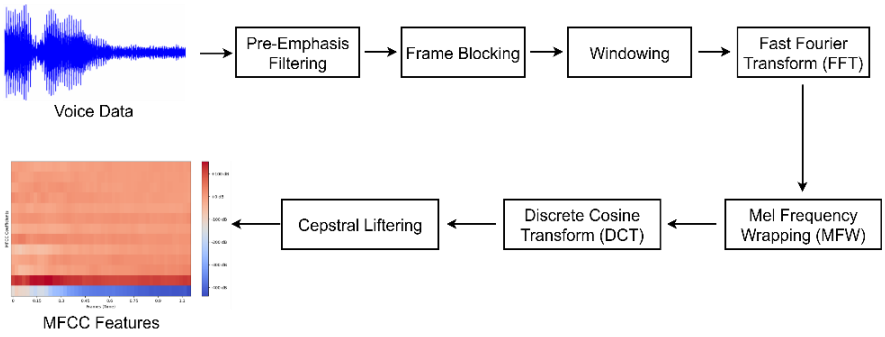

Figure 3. MFCC process

2.3.2. Gammatone frequency cepstral coefficients

GFCC are speech features derived using gammatone filterbanks that simulate the frequency selectivity of the human cochlea [30]. Like MFCC, GFCC includes preprocessing steps, such as pre-emphasis, framing, and windowing, but it differs in applying gammatone filters instead of mel filters, resulting in a more biologically plausible representation of auditory processing. At the heart of the GFCC method is a bank of gammatone filters through which the signal is passed, with each filter constructed to approximate the behavior of auditory nerve fibers. The impulse response of such a filter is given in (3).

$$g(t) = a \cdot t^{n-1} \cdot e^{-2\pi bt} \cdot cos(2\pi f_c t + \theta) \quad (3)$$

In (3), the equation models a filter's time-domain response to a signal, reflecting key aspects of auditory processing. The output $g(t)$ is shaped by several parameters: amplitude factor a controls gain, t^{n-1} defines the filter order n (typically set to 4), and $e^{-2\pi bt}$ d governs signal decay, with b related to the equivalent rectangular bandwidth (ERB). The oscillatory component $cos(2\pi f_c t + \theta)$ centers the response at frequency f_c, with θ as the initial phase. Together, these terms enable the gammatone filterbank to mimic cochlear behavior, supporting perceptually driven audio processing. Figure 4 illustrates the full GFCC feature extraction pipeline, including this filtering stage.

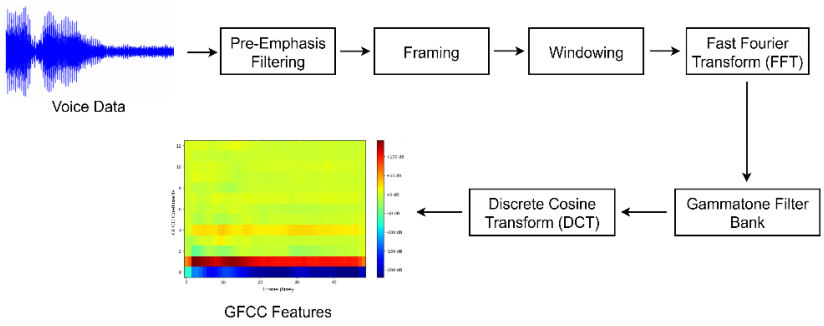

Figure 4. GFCC process

The GFCC pipeline transforms the signal via FFT, routes it through the gammatone filterbank. It then log-compresses and decorrelates the resulting energies using DCT. The resulting coefficients encode both spectral structure and perceptual cues, and tend to hold up better than MFCC when the recording

environment is noisy, a property that makes GFCC well suited to speech and speaker-recognition tasks under difficult acoustic conditions [31].

2.3.3. Bark frequency cepstral coefficients

BFCC is another spectral feature set applied to speech recognition, speaker identification, and audio classification tasks. Where MFCC relies on the mel scale, BFCC is built on the bark scale, a psychoacoustic scale that approximates how the basilar membrane in the inner ear processes different sound frequencies [21]. The overall BFCC computation process is illustrated in Figure 5.

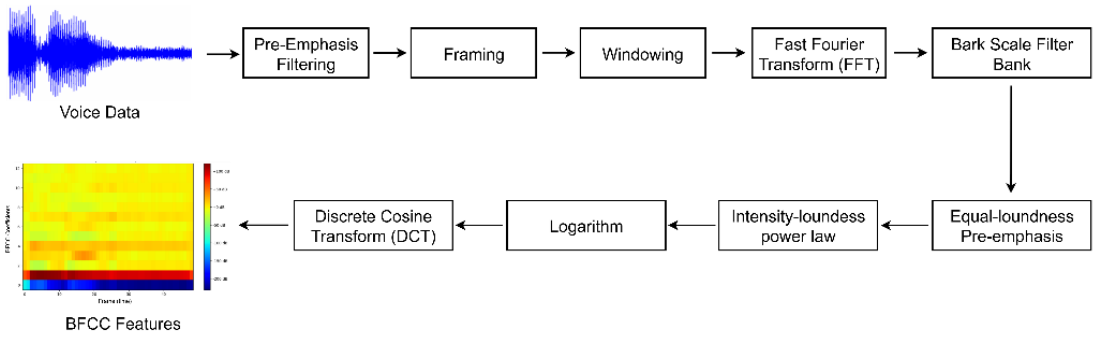

Figure 5. BFCC process

As with MFCC, extraction starts by framing and windowing (typically with a Hamming window) the signal, then converting it to the frequency domain via FFT before mapping frequencies onto the bark scale, typically using the approximation shown in (4).

$$Bark(f) = 13 \cdot \arctan(0.00076 \cdot f) + 3.5 \cdot arctan\left(\left(\frac{f}{7500}\right)^2\right) \quad (4)$$

A triangular filterbank shaped for the bark scale is then applied to the magnitude spectrum; structurally, it resembles the MFCC filterbank, but its filters are centered and spaced differently since they follow bark rather than mel. The center frequency f_{center} for each filter can be computed as presented in (5).

$$f_{center} = \frac{26.81}{1+\frac{1960}{f_{bark}+0.53}} - 0.53 \quad (5)$$

After filtering, the spectral energies from each channel are summed to produce a bark-scaled energy representation. These energies are then decorrelated by applying the DCT, yielding the cepstral coefficients, which are computed as shown in (6).

$$C_n = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} E_k \cdot cos\left(\frac{\pi n(2k+1)}{2N}\right) \quad (6)$$

Where E_k is the energy of the k-th filter and C_n is the n-th cepstral coefficient. The resulting BFCCs may undergo further transformations such as logarithmic compression or normalization to enhance robustness against variations in signal amplitude and environmental noise. Although BFCCs share similarities with MFCCs, such as filterbanks and DCT, the key difference lies in their perceptual modeling, with BFCCs offering a bark-scale representation that better mimics human hearing in certain contexts.

2.3.4. Linear predictive cepstral coefficients

LPCCs are widely used in speech processing due to their ability to capture vocal tract characteristics in a compact and stable form [32]. They are derived from linear predictive coefficients (LPCs), which model the speech signal based on the source–filter theory. LPCs model the vocal tract well but are prone to numerical instability; converting them into the cepstral domain produces LPCCs, a more robust representation for downstream tasks such as speech and speaker recognition. Figure 6 depicts the full LPCC extraction pipeline, covering its main stages: pre-emphasis filtering, framing, windowing, linear predictive analysis, and cepstral transformation.

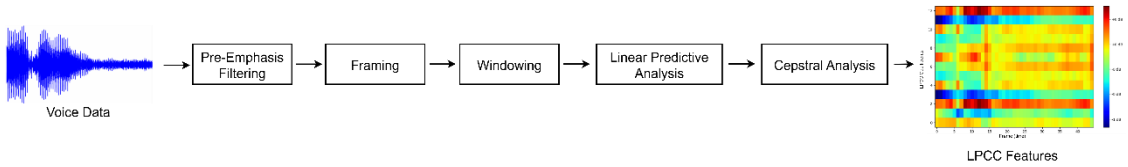

Figure 6. LPCC process

The process begins with a linear prediction model, in which the current audio sample is estimated based on a linear combination of its previous samples. This model is mathematically expressed as shown in (7).

$$s[n] = -\sum_{k=1}^{p} a_k \cdot s[n-k] \tag{7}$$

Where $s[n]$ is the current audio signal, a_k denotes the linear prediction coefficients, and p represents the prediction order. In (8) is then used to compute the prediction error signal, which captures the discrepancy between the actual signal and its predicted value.

$$e[n] = s[n] - \hat{s}[n] = s[n] + \sum_{k=1}^{p} a_k \cdot s[n-k] \tag{8}$$

This error signal captures the residual components not explained by the prediction model.

Subsequently, a cepstral transformation is applied to the LPCs to obtain the LPCCs. The zeroth-order cepstral coefficient c_0 is obtained by taking the logarithm of the signal's total energy R_0, while the remaining cepstral coefficients c_m for $m = 1, 2, \ldots, p$ are computed recursively, as described in (9).

$$c_0 = \log(R_0), c_m = a_m + \sum_{k=1}^{m-1} \frac{k}{m} \cdot c_k \cdot a_{m-k} \tag{9}$$

Where c_m represents the m-th cepstral coefficient, a_m are the LPCs, and c_k are previously computed cepstral coefficients. The total energy R_0 used to compute c_0 is obtained by summing the squared amplitude of the signal across all samples, as defined in (10).

$$R_0 = \sum_{n=0}^{N-1} s^2[n] \tag{10}$$

Where N refers to the total sample count of the audio signal. These transformations convert time-domain LPC representations into cepstral-domain features that are less sensitive to noise and more suitable for machine learning applications.

2.4. Deep learning architectures

2.4.1. Transformer

Transformer is a neural architecture built for sequence-to-sequence problems, using an encoder–decoder structure that can be computed in parallel. As described by Han *et al.* [33], it relies on self-attention, which allows the model to capture relationships across all elements in a sequence, regardless of their distance. The encoder builds a contextual representation of the input, and the decoder then generates output conditioned on that representation.

Central to this design is multi-head self-attention, which lets the model attend to several parts of the sequence at once and efficiently capture global context. Unlike recurrent networks, the transformer processes the whole sequence in parallel, which improves its scalability for tasks such as speech recognition and machine translation. Every encoder layer pairs a multi-head attention sub-layer with a feed-forward network, both wrapped in residual connections and normalization for training stability. Each decoder layer follows a similar pattern but adds a masked self-attention sub-layer and cross-attention over the encoder's output alongside its own feed-forward layer. The overall transformer architecture is depicted in Figure 7.

2.4.2. ResNet50V2

ResNet50V2 is an enhanced version of the original ResNet50 architecture, which was first introduced by Microsoft Research in 2015 [34]. ResNet50V2 begins with an initial convolutional layer followed by a series of residual blocks [35]. Each residual block contains several convolutional layers linked by shortcut connections that let information skips ahead directly rather than passing through every layer in sequence. This shortcut mechanism eases training by reducing the vanishing-gradient effect. Following the last residual block, a global average pooling layer is used to compress the spatial dimensions of the extracted features. The full architecture is depicted in Figure 8.

To cut down feature dimensionality, ResNet50V2 relies on global average pooling, which averages each feature map across its spatial extent. The condensed output then passes through a fully connected (dense) layer that produces the final classification result. Non-linearity is introduced through the rectified linear unit (ReLU) activations, while categorical cross-entropy serves as the loss function guiding training. Training itself is usually carried out with optimizers such as stochastic gradient descent (SGD) or adaptive moment estimation (Adam), which update the model's weights iteratively. Taken together, these elements allow ResNet50V2 to learn complex patterns effectively and achieve strong performance on tasks like audio signal classification.

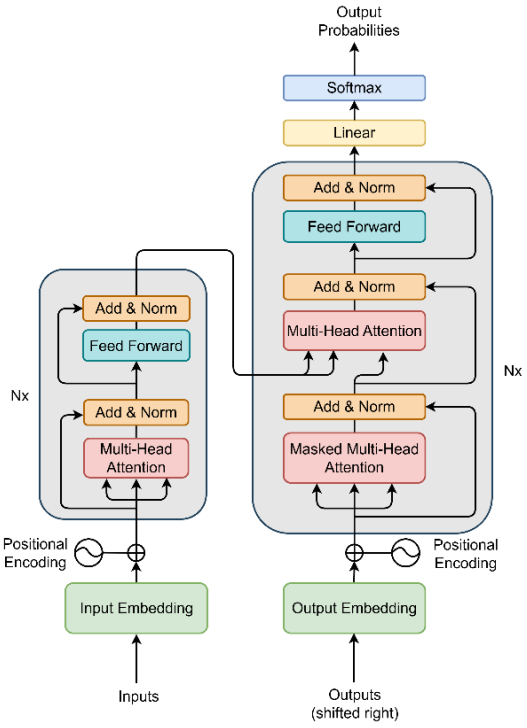

Figure 7. Transformer architecture

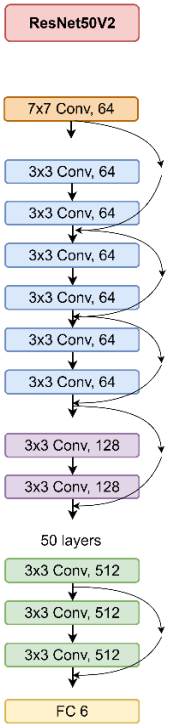

Figure 8. ResNet50V2 architecture

2.4.3. U-Net

U-Net originated in biomedical image segmentation but has since proven adaptable to a range of signal-processing tasks, audio included. Its hallmark is a symmetric, U-shaped structure composed of two main parts: a contracting path (encoder) and an expanding path (decoder) [36]. The encoder distills high-level context through repeated convolution and pooling, while the decoder rebuilds spatial resolution through upsampling. Skip connections tie matching encoder and decoder layers together, preserving fine spatial detail that would otherwise be lost during downsampling. Each stage of the encoder path typically pairs two 3 × 3 convolutions with ReLU activation, followed by 2 × 2 max-pooling, a sequence that shrinks spatial resolution while deepening the feature maps, pushing the network toward increasingly abstract representations. The structure is similar to that of traditional CNNs and is essential for building a compressed but informative representation of the input signal.

The decoder path reflects the encoder's structure in reverse, applying upsampling, commonly through transposed convolutions, followed by convolutional layers to progressively rebuild spatial resolution. At each level, the decoder receives concatenated feature maps from the encoder through skip connections. This integration of deep and shallow features allows for more accurate and detailed reconstructions, which is particularly beneficial for audio tasks requiring both temporal precision and structural preservation. Figure 9 provides an illustration of the U-Net architecture. Due to its flexibility, U-Net can be adapted for tasks, such as speech denoising, source separation, and speech enhancement [37].

2.5. Model performance evaluation

A confusion matrix is a tabular structure used to assess the performance of a classification model by comparing the actual class labels (represented as rows) with the predicted class labels (represented as columns) [38]. This matrix allows for a detailed evaluation of how many instances are correctly or

incorrectly predicted for each class, making it an essential diagnostic tool in classification tasks. In multi-class classification, the confusion matrix is structured as an $m \times m$ square matrix, where m denotes the number of classes. Diagonal elements represent true positives, correct predictions per class, while off-diagonal elements indicate misclassifications as false positives or false negatives [39].

As illustrated in Figure 10, in a five-class classification (A–E), each diagonal entry (e.g., TP_A) represents correctly predicted samples for a class, while off-diagonal entries (e.g., E_{AB}, and E_{AC}) indicate misclassifications. False negatives for class A are the sum of class A samples misclassified as others, while false positives are samples from other classes wrongly predicted as A. In general, diagonal values of an $m \times m$ confusion matrix shows correct classifications, and the total misclassifications equal $m^2 - m$, the sum of all off-diagonal elements.

From this matrix, several standard metrics can be derived: accuracy, precision, recall, and F1-score. Precision expresses what fraction of the model's positive predictions were actually correct, as shown in (11), while recall expresses what fraction of the true positive cases the model successfully identified, as illustrated in (12). F1-score presented in (13), combines the two through their harmonic mean, giving a single balanced measure that weighs both false positives and false negatives. Finally, accuracy captures the overall share of correct predictions the model makes across all classes combined, and is formally expressed in (14).

$$Precision = \frac{TP}{TP+FP} \tag{11}$$

$$Recall = \frac{TP}{TP+FN} \tag{12}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{13}$$

$$Accuracy = \frac{\sum_{i=1}^{m} TP_i}{\sum_{i=1}^{m} TP_i + \sum_{i=1}^{m} FP_i + \sum_{i=1}^{m} FN_i} \tag{14}$$

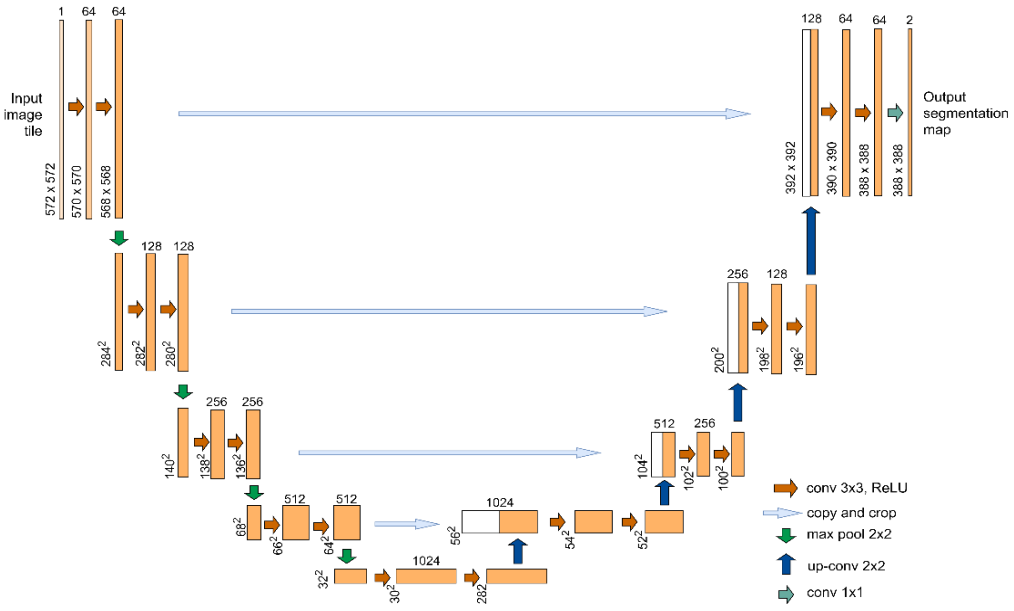

Figure 9. U-Net architecture

	Pred A	Pred B	Pred C	Pred D	Pred E
Actual A	TP_A	E_{AB}	E_{AC}	E_{AD}	E_{AE}
Actual B	E_{BA}	TP_B	E_{BC}	E_{BD}	E_{BE}
Actual C	E_{CA}	E_{CB}	TP_C	E_{CD}	E_{CE}
Actual D	E_{DA}	E_{DB}	E_{DC}	TP_D	E_{DE}
Actual E	E_{EA}	E_{EB}	E_{EC}	E_{ED}	TP_E

Figure 10. Multi-class confusion matrix

3. RESULTS AND DISCUSSION

Classification performance was initially evaluated using the confusion matrices in Figure 11, showing prediction outcomes across the five voice command classes (forward, backward, right, left, and stop) for each model-feature configuration. Figures 11(a)–11(d) show transformer results with MFCC, GFCC, BFCC, and LPCC features, respectively; Figures 11(e)–11(h) present ResNet50V2 results, and Figures 11(i)–11(l) show U-Net results. Diagonal entries represent correct predictions, while off-diagonal entries indicate misclassifications between command classes.

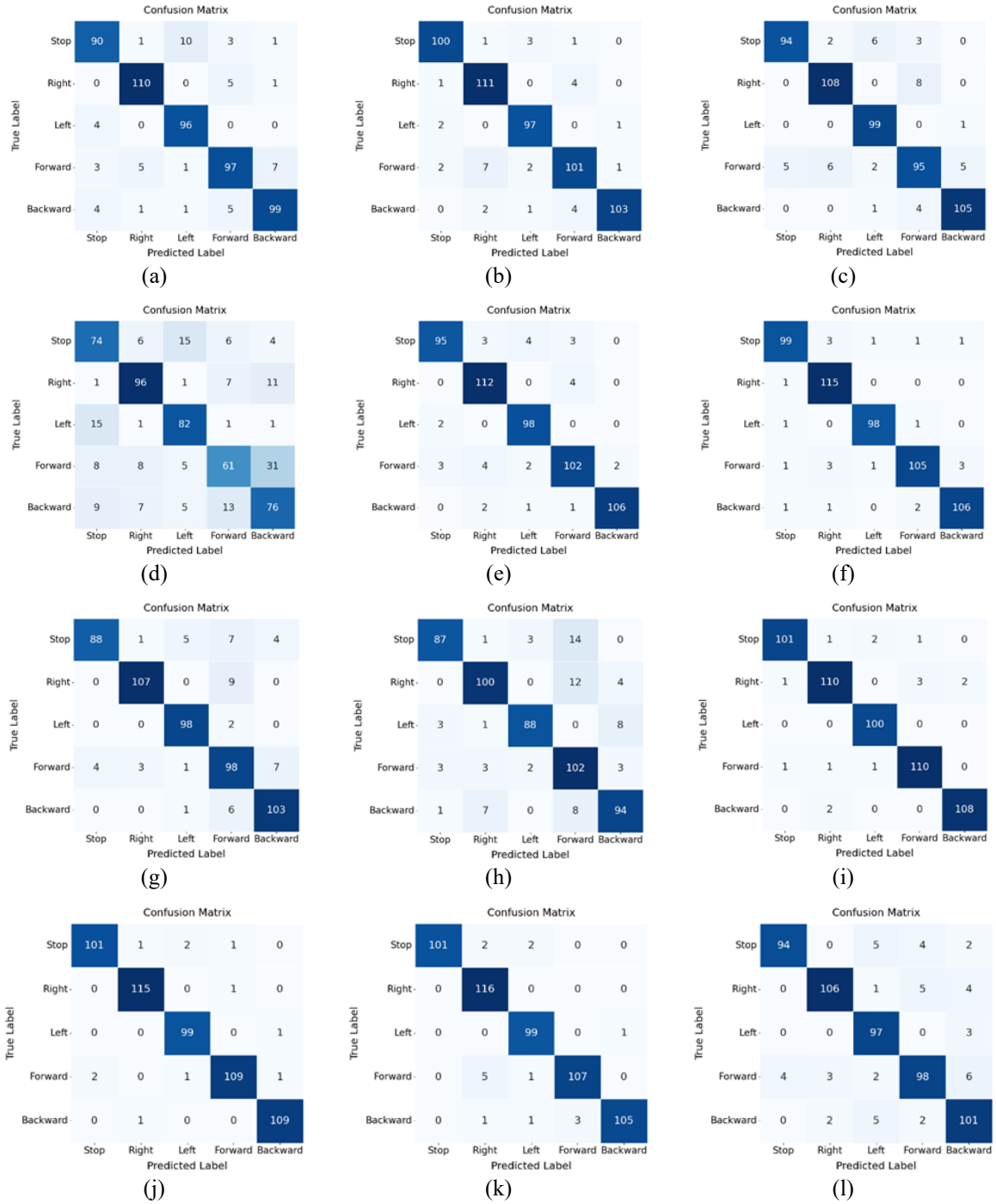

Figure 11. Confusion matrices for each classification result: (a) MFCC + transformer, (b) GFCC + transformer, (c) BFCC + transformer, (d) LPCC + transformer, (e) MFCC + ResNet50v2, (f) GFCC + ResNet50v2, (g) BFCC + ResNet50v2, (h) LPCC + ResNet50v2, (i) MFCC + U-Net, (j) GFCC + U-Net, (k) BFCC + U-Net, and (l) LPCC + U-Net

As shown in Figure 11, configurations such as GFCC + U-Net (Figure 11(j)), GFCC + ResNet50V2 (Figure 11(f)), and MFCC + U-Net (Figure 11(i)) exhibit strong diagonal dominance, indicating consistently high prediction performance across the five classes. In particular, GFCC + U-Net (Figure 11(j)) produces minimal misclassifications, suggesting that this configuration is robust in distinguishing between different voice commands. In contrast, LPCC + Transformer (Figure 11(d)) shows more frequent misclassifications, particularly for the *left* and *stop* classes, as indicated by the greater number of off-diagonal entries. This pattern suggests that the LPCC + Transformer configuration may be less reliable for real-time voice command control.

The visual observations from Figure 11(a)–11(l) are further supported by the quantitative results presented in Table 2, which reports the average accuracy, precision, recall, and F1-score for each model-feature configuration. These evaluation metrics were calculated from the corresponding confusion matrices using standard classification formulas. All metrics were obtained from a held-out test set consisting of 544 samples, representing 10% of the complete dataset, providing a consistent basis for comparing the classification performance of all configurations.

Table 2. Testing results

Feature extraction	Classifier	Accuracy	Precision	Recall	F1-score	Inference time (s)
MFCC	Transformer	0.92	0.90	0.91	0.90	0.94
	ResNet50V2	0.90	0.95	0.94	0.94	5.38
	U-Net	0.94	0.97	0.96	0.96	1.15
GFCC	Transformer	0.97	0.94	0.94	0.94	1.34
	ResNet50V2	0.94	0.96	0.96	0.96	3.55
	U-Net	0.96	0.98	0.98	0.98	1.18
BFCC	Transformer	0.92	0.92	0.92	0.92	1.33
	ResNet50V2	0.92	0.90	0.90	0.91	4.04
	U-Net	0.90	0.96	0.97	0.97	1.33
LPCC	Transformer	0.97	0.71	0.72	0.71	1.34
	ResNet50V2	0.86	0.88	0.86	0.87	4.60
	U-Net	0.85	0.93	0.91	0.91	1.27

GFCC combined with U-Net produced the strongest results overall, reaching 0.96 accuracy, 0.98 for both precision and recall, and an F1-score of 0.98, indicating highly accurate and consistent predictions. Other high-performing configurations include ResNet50V2 + GFCC and MFCC + U-Net, each reaching F1-scores above 0.95. Conversely, LPCC + transformer yielded the weakest results, with notably lower precision of 0.71 and an F1-score of 0.71, despite its high accuracy. This implies a tendency toward false positives and imbalanced prediction across classes. In addition to classification performance, inference time is a key consideration for applications that operate in real time. Among all configurations, transformer-based models showed the fastest inference time, averaging around 1.3 seconds per test batch, making them suitable for systems with limited processing capability. However, U-Net models achieved a favorable balance between speed and accuracy, with most configurations staying under 1.5 seconds while delivering superior F1-scores. In contrast, ResNet50V2-based classifiers, despite their strong accuracy, exhibited the longest inference times, ranging between 3.5 and 5.38 seconds, which may pose limitations in time-sensitive applications. Overall, these findings highlight how effective U-Net architectures are, especially when paired with GFCC or MFCC features. Such configurations consistently deliver high classification accuracy, balanced performance across classes, and processing times within an acceptable range, positioning them well for real-time, speech-driven control in assistive systems such as smart wheelchairs.

4. CONCLUSION

This work examined how different speech feature-extraction methods and deep learning architectures shape the performance of voice-controlled smart wheelchairs operating in noisy conditions. Four extraction methods: MFCC, GFCC, BFCC, and LPCC were tested against three model architectures: transformer, ResNet50V2, and U-Net on a 544-sample test set (10% of the full dataset). Overall, the GFCC + U-Net pairing stood out as the top performer, with accuracy, precision, recall, and F1-score all reaching as high as 0.98, alongside a fast inference time of roughly 1.18 seconds, indicating strong potential for real-time application. Other strong performers included ResNet50V2 + GFCC and MFCC + U-Net, both achieving F1-scores above 0.95. In contrast, LPCC + transformer showed the weakest results, particularly for the left and stop commands, with an F1-score of 0.71, despite high accuracy, emphasizing that accuracy alone can be misleading in imbalanced multi-class tasks. While model architecture had some effect, U-Net

consistently balanced accuracy and speed, whereas ResNet50V2, although accurate, exhibited the slowest inference times ranging from 3.5 to 5.38 seconds, which may hinder real-time responsiveness. Overall, the results underscore the importance of using perceptually robust features such as GFCC with efficient architectures like U-Net to build accurate and responsive voice-based assistive technologies for users with severe physical disabilities.

FUNDING INFORMATION

The authors wish to express their appreciation for the funding received through the Basis Informasi Penelitian dan Pengabdian kepada Masyarakat (BIMA) 2025 research grant scheme, administered by Ministry of Higher Education, Science, and Technology Republic of Indonesia, under grant no. 00584/UN10.A0501/B/PT.01.03.2/2025. This financial support played a critical role in enabling the research to be carried out successfully.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Fitri Utaminingrum	✓	✓		✓	✓	✓	✓		✓	✓		✓	✓	✓
Eldad Vikorian	✓	✓			✓	✓		✓	✓		✓			
Anindya Zulva Larasati		✓		✓		✓			✓		✓			
Corina Karim			✓			✓				✓				

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, [FU], upon reasonable request.

REFERENCES

[1] World Health Organization, "Disability," *WHO.int*. 2022. Accessed: Jun. 27, 2025. [Online]. Available: https://www.who.int/health-topics/disability
[2] E. N. Arifin, A. Ananta, and C. Y. Hoon, "Quantifying disability using the Washington Group questionnaire: prevalence and patterns by age and sex in Indonesia," *Journal of Population Research*, vol. 42, no. 2, 2025, doi: 10.1007/s12546-025-09373-2.
[3] M. A. Said and M. M. Alhumaid, "Appraising the physical activity levels of saudis with physical disabilities: effects of disability type, mobility assistive devices, and demographic factors," *Healthcare*, vol. 12, no. 9, May 2024, doi: 10.3390/healthcare12090937.
[4] ISPO, ISWP, and WHO, *Wheelchair provision guidelines*. Geneva, Switzerland: World Health Organization, 2023.
[5] M. A. Al Rakib, S. Uddin, M. M. Rahman, S. Chakraborty, and F. I. Abbas, "Smart wheelchair with voice control for physically challenged people," *European Journal of Engineering and Technology Research*, vol. 6, no. 7, pp. 97–102, Dec. 2021, doi: 10.24018/ejeng.2021.6.7.2627.
[6] I. P. Ktistakis, G. Goodman, and A. Britzolaki, "Applications of AI in healthcare and assistive technologies," in *Advances in Assistive Technologies*, Cham, Switzerland: Springer, 2022, pp. 11–31, doi: 10.1007/978-3-030-87132-1_2.
[7] J. M. Araujo, G. Zhang, J. P. P. Hansen, and S. Puthusserypady, "Exploring eye-gaze wheelchair control," in *ACM Symposium on Eye Tracking Research and Applications*, Jun. 2020, pp. 1–8, doi: 10.1145/3379157.3388933.
[8] S. Chatzidimitriadis, S. M. Bafti, and K. Sirlantzis, "Non-intrusive head movement control for powered wheelchairs: a vision-based approach," *IEEE Access*, vol. 11, pp. 65663–65674, 2023, doi: 10.1109/ACCESS.2023.3275529.
[9] I. K. Somawirata and F. Utaminingrum, "Smart wheelchair controlled by head gesture based on vision," *Journal of Physics: Conference Series*, vol. 2497, no. 1, May 2023, doi: 10.1088/1742-6596/2497/1/012011.
[10] A. M. Alqadri, F. Utaminingrum, M. A. Fauzi, R. R. M. Putri, C. Karim, and F. Gapsari, "Attention module in YOLO-based object detection method for autonomous smart wheelchair room navigation system," in *2024 4th International Conference on Robotics, Automation and Artificial Intelligence*, 2024, pp. 312–316, doi: 10.1109/RAAI64504.2024.10949519.

[11] F. Utaminingrum, I. K. Somawirata, S. Mayena, A. Septiarini, and T. K. Shih, "Analysis of kernel performance in support vector machine using seven features extraction for obstacle detection," *International Journal of Control, Automation and Systems*, vol. 21, no. 1, pp. 281–291, Jan. 2023, doi: 10.1007/s12555-021-0702-z.

[12] S. K. Sahoo and B. B. Choudhury, "Voice-activated wheelchair: an affordable solution for individuals with physical disabilities," *Management Science Letters*, vol. 13, no. 3, pp. 175–192, 2023, doi: 10.5267/j.msl.2023.4.004.

[13] B. Sachdeva, Anshuman, N. Jha, and A. Raj, "Voice controlled wheelchair system," in *2024 2nd International Conference on Disruptive Technologies (ICDT)*, Mar. 2024, pp. 1448–1452, doi: 10.1109/ICDT61202.2024.10488988.

[14] A. I. Iskanderani *et al.*, "Voice controlled artificial intelligent smart wheelchair," in *2020 8th International Conference on Intelligent and Advanced Systems*, Jul. 2021, pp. 1–5, doi: 10.1109/ICIAS49414.2021.9642607.

[15] M. J. Haddad and D. A. Sanders, "Deep learning architecture to assist with steering a powered wheelchair," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 12, pp. 2987–2994, Dec. 2020, doi: 10.1109/TNSRE.2020.3031468.

[16] M. M. Abdulghani, W. L. Walters, and K. H. Abed, "Autonomous voice recognition wheelchair control system," in *2022 International Conference on Computational Science and Computational Intelligence (CSCI)*, Dec. 2022, pp. 226–230, doi: 10.1109/CSCI58124.2022.00043.

[17] Sutikno, K. Anam, and A. Saleh, "Voice controlled wheelchair for disabled patients based on CNN and LSTM," in *2020 4th International Conference on Informatics and Computational Sciences (ICICoS)*, Nov. 2020, pp. 1–5, doi: 10.1109/ICICoS51170.2020.9299007.

[18] 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, doi: 10.1109/ACCESS.2022.3223444.

[19] M. S. Sidhu, N. A. A. Latib, and K. K. Sidhu, "MFCC in audio signal processing for voice disorder: a review," *Multimedia Tools and Applications*, vol. 84, no. 10, pp. 8015–8035, Apr. 2024, doi: 10.1007/s11042-024-19253-1.

[20] H. Wang and C. Zhang, "The application of Gammatone frequency cepstral coefficients for forensic voice comparison under noisy conditions," *Australian Journal of Forensic Sciences*, vol. 52, no. 5, pp. 553–568, Sep. 2020, doi: 10.1080/00450618.2019.1584830.

[21] R. Bhavani and K. Kathirvelu, "Advancements in speech recognition: exploring feature extraction techniques for enhanced accuracy and human-machine communication," in *2024 Fourth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies*, Jan. 2024, pp. 1–8, doi: 10.1109/ICAECT60202.2024.10468885.

[22] H. B. Bisheh, G. G. Amiri, M. Nekooei, and E. Darvishan, "Damage detection of bridges based on combining efficient cepstral coefficients," *Journal of Vibration and Control*, vol. 27, no. 19–20, pp. 2279–2290, Oct. 2021, doi: 10.1177/1077546320958348.

[23] N. Choudhary, A. Sharma, V. S. Rathore, and N. Tiwari, "Performance comparison of ResNet50V2 and VGG16 models for feature extraction in deep learning," in *Intelligent Sustainable Systems (WorldS4 2023)*, Singapore: Springer, 2024, pp. 223–229, doi: 10.1007/978-981-99-8031-4_21.

[24] S. Gul and M. S. Khan, "A survey of audio enhancement algorithms for music, speech, bioacoustics, biomedical, industrial, and environmental sounds by image U-Net," *IEEE Access*, vol. 11, pp. 144456–144483, 2023, doi: 10.1109/ACCESS.2023.3344813.

[25] L. Lu, C. Liu, J. Li, and Y. Gong, "Exploring transformers for large-scale speech recognition," in *Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2020, pp. 5041–5045, doi: 10.21437/Interspeech.2020-2638.

[26] M. Dua, S. Meena, Neelam, Amisha, and N. Chakravarty, "Audio deepfake detection using data augmented graph frequency cepstral coefficients," in *2023 International Conference on System, Computation, Automation and Networking (ICSCAN)*, Nov. 2023, pp. 1–6, doi: 10.1109/ICSCAN58655.2023.10395679.

[27] Y. Wang *et al.*, "Transformer in action: a comparative study of transformer-based acoustic models for large scale speech recognition applications," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2021, pp. 6778–6782, doi: 10.1109/ICASSP39728.2021.9414087.

[28] C.-H. Yang, J. Qi, P.-Y. Chen, X. Ma, and C.-H. Lee, "Characterizing speech adversarial examples using self-attention U-Net enhancement," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 3107–3111, doi: 10.1109/ICASSP40776.2020.9053288.

[29] S. Hawi, J. Alhozami, R. AlQahtani, D. AlSafran, M. Alqarni, and L. El Sahmarany, "Automatic Parkinson's disease detection based on the combination of long-term acoustic features and Mel frequency cepstral coefficients (MFCC)," *Biomedical Signal Processing and Control*, vol. 78, Sep. 2022, doi: 10.1016/j.bspc.2022.104013.

[30] Z. Mengxi and T. Zhiguo, "Research on failure identification of partial discharge ultrasonic signal based on GFCC," in *2020 IEEE Electrical Insulation Conference (EIC)*, Jun. 2020, pp. 412–416, doi: 10.1109/EIC47619.2020.9158683.

[31] A. Krobba, M. Debyeche, and S.-A. Selouani, "Mixture linear prediction Gammatone cepstral features for robust speaker verification under transmission channel noise," *Multimedia Tools and Applications*, vol. 79, no. 25–26, pp. 18679–18693, Jul. 2020, doi: 10.1007/s11042-020-08748-2.

[32] M. Ramashini, P. E. Abas, K. Mohanchandra, and L. C. D. Silva, "Robust cepstral feature for bird sound classification," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 2, pp. 1477–1487, Apr. 2022, doi: 10.11591/ijece.v12i2.pp1477-1487.

[33] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," in *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021, pp. 15908–15919.

[34] A. Jaiswal, Deepali, and N. Sachdeva, "Empirical analysis of traffic sign recognition using ResNet architectures," in *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, May 2023, pp. 280–285, doi: 10.1109/ICACITE57410.2023.10183247.

[35] S. Riyadi, F. A. Abidin, and N. Audita, "Comparison of ResNet50V2 and MobileNetV2 models in building architectural style classification," in *2024 International Conference on Intelligent Systems and Computer Vision (ISCV)*, May 2024, pp. 1–8, doi: 10.1109/ISCV60512.2024.10620099.

[36] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, "U-Net and its variants for medical image segmentation: a review of theory and applications," *IEEE Access,* vol. 9, pp. 82031–82057, 2021, doi: 10.1109/ACCESS.2021.3086020.

[37] H. Zunair and A. B. Hamza, "Sharp U-Net: depthwise convolutional network for biomedical image segmentation," *Computers in Biology and Medicine*, vol. 136, Sep. 2021, doi: 10.1016/j.compbiomed.2021.104699.

[38] M. Heydarian, T. E. Doyle, and R. Samavi, "MLCM: multi-label confusion matrix," *IEEE Access*, vol. 10, pp. 19083–19095, 2022, doi: 10.1109/ACCESS.2022.3151048.

[39] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, vol. 17, no. 1, pp. 168–192, Jan. 2021, doi: 10.1016/j.aci.2018.08.003.

BIOGRAPHIES OF AUTHORS



Fitri Utaminingrum earned her B.Sc. in Informatics Engineering from Institut Teknologi Nasional, Malang, Indonesia, in 2004, followed by an M.Sc. in Computer Science from Universitas Brawijaya, Malang, Indonesia, in 2006. She completed her doctorate in Computer Science and Electrical Engineering at Kumamoto University, Japan, in 2014. She is currently a professor at the Faculty of Computer Science, Universitas Brawijaya. Her research interests include image processing and computer vision. She has authored over 120 papers in international journals and conferences indexed by Scopus. Her professorship was officially conferred on June 26, 2023, when she delivered her inaugural scientific oration titled "Smart wheelchair with multi-feature assistance for persons with disabilities." She can be contacted at email: f3_ningrum@ub.ac.id.



Eldad Vikorian received a bachelor's degree in Computer Engineering from the Department of Informatics Engineering, Faculty of Computer Science, Universitas Brawijaya, Indonesia, in 2025. During his studies, he was engaged in research projects in the field of computer vision. He can be contacted at email: eldadrian05@gmail.com.



Anindya Zulva Larasati earned a B.Eng. in Computer Engineering from the Department of Informatics Engineering, Faculty of Computer Science, Universitas Brawijaya in 2025. She is currently pursuing a master's degree in Computer Science at the same university. Her research interests include computer vision and deep learning, which she has been actively involved in since her undergraduate studies. She can be contacted at email: anindyazulva09@gmail.com.



Corina Karim is a lecturer at the Department of Mathematics, Faculty of Science, Technology, and Mathematics, Universitas Brawijaya, Malang, Indonesia. She earned her Doctor of Philosophy degree from Kumamoto University, Japan. At Universitas Brawijaya, she leads the Laboratory of Analysis, focusing on real analysis, harmonic analysis, and nonlinear partial differential equations. She specializes in mathematical analysis and partial differential equations, particularly in the study of p-Laplacian systems and the regularity of weak solutions. Her research contributions include studies on Hölder regularity, local boundedness of weak solutions, and the existence of global weak solutions in evolutional p-Laplacian systems. Her work employs advanced mathematical techniques such as Campanato's direct approach and the intrinsic scaling method, significantly contributing to the understanding of nonlinear differential systems. She can be contacted at email: co_mathub@ub.ac.id.