Building of Informatics, Technology and Science (BITS)
Volume 6, No 3, Desember 2024 Page: 1910−1918
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.6162

Development of AI-Based Presentation Application using Deep Learning for Individuals with Disabilities

Carli Apriansyah Hutagalung[1,*], Adi Fitrianto[2], Gebran Akbar[1]

[1]Faculty Computer, Computer Science, MNC University, Jakarta, Indonesia
[2]Faculty Computer, Information System, MNC University, Jakarta, Indonesia
Email: [1,*]carli.apriansyah@mncu.ac.id, [2]adi.fitrianto@mncu.ac.id, [3]gebran.akbar@mncu.ac.id
Correspondence Author Email: carli.apriansyah@mncu.ac.id
Submitted: **30/10/2024**; Accepted: **25/12/2024**; Published**: 26/12/2024**

Abstract−This study addresses the challenge of controlling presentation devices for individuals with disabilities, particularly in noisy environments, where voice commands are often misinterpreted. To overcome this, we developed an AI-based application utilizing a hybrid deep learning architecture combining Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models with an attention mechanism to enhance voice command recognition. The primary objective of this research is to improve the accuracy and reliability of voice command recognition for essential commands like “next” and “back” in varying noise conditions. The model was trained using the Speech Commands Dataset and fine-tuned with noise-augmented data to simulate real-world scenarios. Experimental results show that the hybrid LSTM-GRU model achieved an accuracy of 96.5% in clean environments and 88.2% in noisy environments, significantly outperforming traditional models such as Hidden Markov Model (HMM) and Gaussian Mixture Model (GMM), which achieved 78.4% and 81.7% accuracy, respectively, in noisy conditions. The fine-tuned model demonstrated a precision of 90.1% and a recall of 89.5%, showcasing robust performance and practical applicability. This study contributes to the advancement of accessible technologies, enabling individuals with disabilities to interact more effectively with presentation tools despite environmental challenges. Further work will focus on enhancing noise resilience for broader real-world adoption.

Keywords: Speech Recognition; LSTM-GRU Model; AI Application for Disabilities.

1. INTRODUCTION

Presentations are a primary means of conveying information in various professional and educational settings. However, individuals with physical disabilities face significant challenges in delivering presentations, particularly in controlling devices directly. In Indonesia, the number of workers with disabilities has steadily increased, with approximately 720,000 individuals entering the workforce in 2022, most of whom are engaged in entrepreneurship [1][2]. Despite this progress, the proportion of disabled workers employed in formal labor or as employees remains relatively small, underscoring the need for inclusive technologies to support their professional activities [3][4][5].

AI-based applications, such as hand gesture detection and voice recognition, have shown potential for enabling hands-free control of devices. However, these approaches often face limitations in terms of accuracy and responsiveness, especially in scenarios with high input variability or noisy environments [6][7]. These limitations are largely due to reliance on traditional methods, such as Hidden Markov Models (HMM) and Gaussian Mixture Models (GMM), which, while effective, struggle to process audio signals with complex temporal dynamics and significant variability [8][9].

To address these challenges, this study proposes the development of an AI-based presentation application specifically designed for individuals with disabilities. The application employs a hybrid deep learning model that combines Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures. Hybrid models like LSTM-GRU have demonstrated superior performance in processing spatial-temporal data, offering greater accuracy and robustness compared to standalone models [10][11]. Additionally, the integration of an Attention Mechanism enhances the model’s ability to focus on the most relevant temporal features, improving adaptability and performance in noisy environments.

Previous research has applied similar hybrid models to tasks such as hydrological flow forecasting and speech emotion recognition, showcasing their resilience to data variability and noise [10]. However, their application to voice command recognition in noisy environments remains underexplored. This study addresses this gap by incorporating a fine-tuning process with noise-augmented data, enhancing the model’s robustness against background noisea critical factor in real-world conditions.

The primary objective of this research is to develop a robust and accurate voice command recognition system for presentation scenarios. The system is designed to recognize essential commands, such as "next" and "back," even in noisy environments. This study contributes to advancing inclusive technologies by providing individuals with disabilities a reliable tool to support their professional and educational activities. The integration of noise-augmented training data and hybrid modeling represents an innovative step toward improving real-world applicability and addressing the challenges of environmental variability.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 3, Desember 2024 Page: 1910−1918
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.6162

2. RESEARCH METHODOLOGY

The research methodology adopted in this study is systematically divided into six distinct stages, as illustrated in the flowchart. Each stage is designed to address specific objectives and challenges while ensuring a structured and logical progression toward the development and validation of the proposed solution (Figure 1).

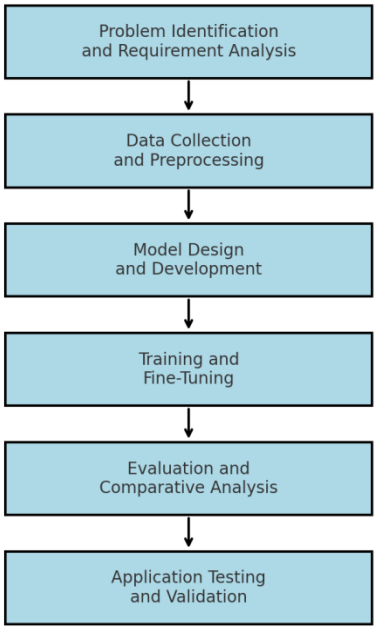

Figure 1. Flowchart of Research Methodology

Here is a general explanation describing the research methodology (Figure 1) adopted in this study systematically, ensuring a structured and logical progression toward the development and validation of the proposed solution, as outlined in the following subsections.

2.1 Data Collection

The data used in this study comes from the Speech Commands Dataset, widely used for speech recognition tasks. Specifically, the data for ‘right’ and ‘left’ were selected to represent the commands ‘next’ and ‘back.’ The dataset includes recordings of these words, mostly around 1 second long, though some are as short as 0.5 seconds. This variation will be normalized during preprocessing for consistency. The data shows significant variation, as seen in the spectrograms: ‘Right’ displays dominant energy in the mid-frequency range (26-42 dB), while ‘Left’ has a more varied energy distribution from -66.61 dB to -59 dB, spread across different frequencies, as shown in Figure 1.

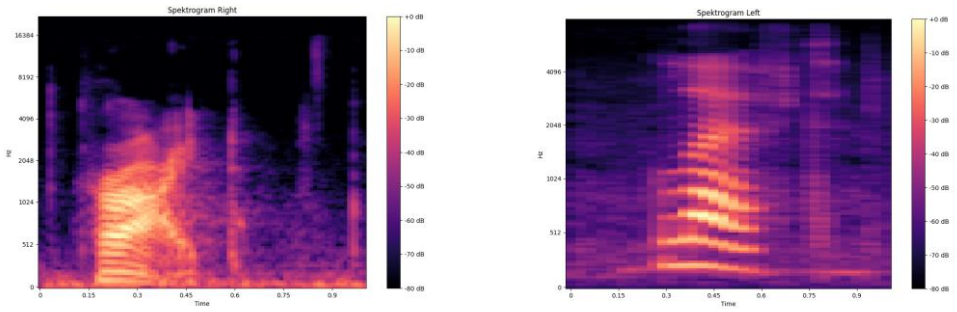

Figure 2. Spectrograms Right & Left

2.2 Preprocessing MFCC

In the data preprocessing stage, several important steps are taken to ensure that the voice data can be processed consistently and optimally by the machine learning model to be used.

a. Sequence Length Normalization
   The data used has varying sequence lengths, with the word ‘right’ having 87 timesteps and ‘left’ having 32 timesteps. To standardize the data for use in the LSTM-GRU model, normalization is performed by adding zero padding to sequences shorter than 87 timesteps and truncating sequences longer than 87 timesteps, ensuring that all inputs have the same length.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 3, Desember 2024 Page: 1910−1918
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.6162

b. Feature Extraction (MFCC)
Mel-Frequency Cepstral Coefficients (MFCC) were extracted from each audio recording to transform the raw audio data into a compact, informative representation suitable for input to the LSTM-GRU model. The extraction process involved the following steps:

1. Fourier Transform
Converts the audio signal from the time domain to the frequency domain using the Short-Time Fourier Transform (STFT):

$$\mathrm{X}(k) = \sum_{n=0}^{N-1} x(n) \,.\, e^{-j2\pi \mathrm{kn}/N} \quad (1)$$

where $\mathrm{X}(k)$ represents the magnitude of the *k*-th frequency bin, $\mathrm{x}(n)$ is the input signal, and *N* is the total number of samples [12].

2. Mel Filter Bank
Warps the spectrogram's frequency axis to the Mel scale and groups the power spectra into Mel-frequency bands [13]:

$$\mathrm{Mel}(f) = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right) \quad (2)$$

Here, *Mel(f)* converts the linear frequency *f* (in Hz) into the Mel scale.

3. Logarithm of Energy
Computes the log of the energy in each Mel-frequency band to compress the spectral amplitude range [14]:

$$E_m = log\left(\sum_k |X(k)|^2 \cdot H_m(k)\right) \quad (3)$$

where E_m is the log energy of the *m*-th Mel band, and $H_m(k)$ is the *m*-th Mel filter.

4. Discrete Cosine Transform (DCT)
Applies the DCT to produce MFCCs, concentrating the spectral energy into a few coefficients:

$$C_m = \sum_{n=1}^{K} log(E_n) \cdot \left(\frac{\pi \mathrm{m}(n-0.5)}{K}\right) \quad (4)$$

where C_m is the *m*-th MFCC, *K* is the number of Mel bands, and E_n is the log energy of the *n*-th Mel band [15].

2.3 Data Splitting and Time Series Padding

The normalized and feature-extracted dataset is split into training, validation, and testing sets with proportions of 70%, 15%, and 15%, respectively. This split ensures that the model is trained and evaluated fairly using unseen data. For the LSTM/GRU model, input sequence lengths must be uniform. Therefore, zero padding is applied to shorter sequences to match the length of the longest sequence, ensuring consistent input shapes of (*batch_size*, *timesteps*, *features*) across the dataset.

2.4 Model Architecture

The model takes Mel-Frequency Cepstral Coefficients (MFCC) extracted from the audio recordings. The input shape is standardized to 87 timesteps with 13 MFCC coefficients per timestep to ensure consistent input dimensions for training.

a. Bidirectional LSTM Layer
A Bidirectional LSTM with 128 units captures dependencies in both forward and backward directions, enhancing pattern recognition, especially for context-dependent features, the operation of an LSTM cell can be described as [16][17][18][19][20]:

$$\begin{aligned} f_t &= \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \widetilde{C_t} &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\ C_t &= f_t \cdot C_{t-1} + i_t \cdot \widetilde{C_t} \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t \cdot \tanh(C_t) \end{aligned} \quad (5)$$

where:
f_t : forget gate.
i_t : input gate.
$\tilde{C}_t$: candidate cell state.
C_t : cell state.
o_t : output gate.
h_t : hidden state.
σ : denotes the sigmoid function, and *tanh* is the hyperbolic tangent function.

b. GRU Layer

Building of Informatics, Technology and Science (BITS)
Volume 6, No 3, Desember 2024 Page: 1910−1918
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.6162

Following the LSTM, a GRU layer with 64 units captures temporal dependencies efficiently. GRUs are more computationally efficient than LSTMs while maintaining similar functionality. GRU layer can be defined mathematically as [21][22][23]:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z)$$
$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r)$$
$$\tilde{h_t} = \tanh(W_h \cdot [r_t \cdot h_{t-1}, x_t] + b_h) \quad (6)$$
$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h_t}$$

where:
f_t : update gate.
r_t : reset gate.
$\tilde{h}_t$: candidate activation.
r_t : hidden state.

c. Attention Mechanism
To allow the model to focus on the most relevant parts of the input sequence, an attention mechanism is applied. This mechanism computes attention weights to highlight key temporal features [24]. The attention weights are computed as follows:

$$C_t = \sum_{t=1}^{T} \alpha_t \cdot h_t \quad (7)$$
$$\alpha_t = \text{softmax}(W_a \cdot h_t)$$

Where:
W_a : The weight matrix for attention.
h_t : The hidden state at time t.
α_t represents the attention weights. the context vector C_t is then used to focus on specific parts of the sequence that are most important for making predictions.

d. Dense Layer
The output from the attention mechanism is passed to a Dense layer with 32 units using a ReLU activation function to aggregate the extracted features. The operation of the Dense layer can be expressed as [25]:

$$y = \text{ReLU}(W_d \cdot c_t + b_d) \quad (8)$$

Where:
W_d : The weight matrix for the Dense layer.
b_d : The bias term.
$\text{ReLU}(x) = \max(0, x)$ is the Rectified Linear Unit activation function.

e. Output Layer
The final output layer consists of 2 neurons with SoftMax activation for classifying the ‘right’ and ‘left’ audio commands. A SoftMax activation function is used to generate a probability distribution over the classes [26]:

$$\hat{y}i = \frac{\exp(y_i)}{\sum j=1^2 \exp(y_j)} \quad (9)$$

where $\hat{y}_i$ is the predicted probability of class *i*, and y_i is the raw output score for class *i*.
This architecture is designed to balance complexity and efficiency, allowing the model to effectively process and classify voice commands in various environmental conditions, including those with background noise.

2.5 Training and Validation Process

The dataset was split into 70% for training, 15% for validation, and 15% for testing. Each sequence was zero-padded to ensure uniform input length. During the training phase, the model was optimized using the Adam optimizer and categorical cross-entropy loss. Early stopping and model checkpoint callbacks were used to save the best-performing model based on validation loss. Training is conducted from 5 to 30 epochs with a batch size of 32, and data augmentation was applied by adding background noise to simulate real-world conditions.

2.6 Fine Tuning & Evaluation

After initial training, the LSTM-GRU model underwent fine-tuning to improve its robustness in noisy environments. The fine-tuning process involved using augmented data, where noise was artificially added to simulate real-world conditions, such as background noise in public spaces or homes. By applying this noise to the original clean data, the model learned to distinguish between relevant voice commands (‘Right’ and ‘Left’) and background noise more effectively. Evaluation Metrics, To assess the performance of the model, several evaluation metrics were employed:

a. Accuracy: The proportion of correctly predicted commands out of all commands.
b. Precision: The ability of the model to correctly identify the positive class (‘Right’ or ‘Left’) when it makes a positive prediction.
c. Recall: The ratio of true positive predictions to the actual number of positive examples in the data.
d. AUC: Measures the model’s performance across various threshold levels.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 3, Desember 2024 Page: 1910−1918
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.6162

These metrics were applied during both the initial training and after fine-tuning to ensure the model's improvement in handling noisy data and to compare performance across different threshold values.

2.7 Comparison of Methods

In this study, we performed a comparative analysis using various models on the same dataset to ensure fairness in the evaluation. For traditional models such as HMM and GMM, the raw data without any fine-tuning was used. This choice was made because these models are not designed to take advantage of noise augmentation, or fine-tuning processes commonly used in deep learning.

For the deep learning models, including CNN, LSTM, GRU, and the LSTM-GRU Hybrid, as well as Google API's Speech Recognition model, the data underwent a fine-tuning process where noise was artificially added to better simulate real-world conditions. The fine-tuned data allowed these models to become more robust in noisy environments, thus reflecting their ability to generalize well to unseen, noisy data.

By comparing traditional models with deep learning models on the same dataset, both before and after fine-tuning, we can clearly observe the impact of noise augmentation on performance and understand why deep learning methods generally outperform traditional ones in challenging environments.

3. RESULT AND DISCUSSION

3.1 Model Performance Metrics

The LSTM-GRU model's performance was evaluated across epochs 5 to 30 using Accuracy, Precision, Recall, and AUC metrics to assess its ability to classify 'left' and 'right' audio commands. As shown in Table 1, the model improved steadily with more epochs. At epoch 5, it achieved 80.34% accuracy with high precision (94.40%) but relatively low recall (66.79%), indicating it minimized false positives but missed many true positives. By epoch 30, the model reached 89.59% accuracy, with a more balanced precision and recall, reflecting improved detection of both positive and negative instances effectively.

Table 1. Across Different Epochs

Epoch	Accuracy	Precision	Recall	AUC
5	0.80	0.94	0.67	0.93
10	0.84	0.92	0.76	0.94
15	0.88	0.85	0.92	0.95
20	0.88	0.88	0.89	0.95
25	0.88	0.88	0.90	0.96
30	0.90	0.88	0.93	0.96

Table 1 illustrates the performance metrics of the hybrid LSTM-GRU model across different training epochs. The model's performance improves steadily up to epoch 15, with notable increases in accuracy (88%) and recall (92%). Beyond epoch 15, performance metrics such as precision, recall, and AUC stabilize, indicating that the model achieves consistent performance at later stages of training. The highest accuracy (90%) is observed at epoch 30, coupled with balanced recall (93%) and AUC (96%), showcasing the model's reliability for noisy voice command recognition. These results suggest that fine-tuning with noise-augmented data enhances the model's robustness and real-world applicability.

The ROC curves for different epochs, displayed in Figure 2, further illustrate the model's performance improvement as training progresses. Initially, at epoch 5, the AUC was 0.933, indicating strong discriminative ability. The AUC value progressively increased to 0.961 by epoch 30, reflecting the model's enhanced capacity to distinguish between the 'left' and 'right' commands. This steady improvement is evident in the ROC curve, where the curve approaches the top-left corner of the graph, indicating fewer false positives and higher true positive rates.

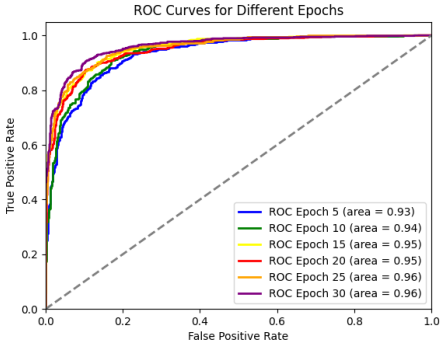

Figure 3. ROC Curves

Building of Informatics, Technology and Science (BITS)
Volume 6, No 3, Desember 2024 Page: 1910−1918
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.6162

The confusion matrices shown in Figure 4 for epochs 5, 10, 15, 20, 25, and 30, highlight the evolution of the model's classification ability across the epochs. At epoch 5, the model showed substantial misclassification between the two classes, with 260 false negatives for the 'right' class. As training continued, the number of misclassifications decreased significantly. By epoch 30, the number of false negatives for 'right' dropped to 55, reflecting the model's enhanced precision and recall as it correctly identified more instances in both classes.

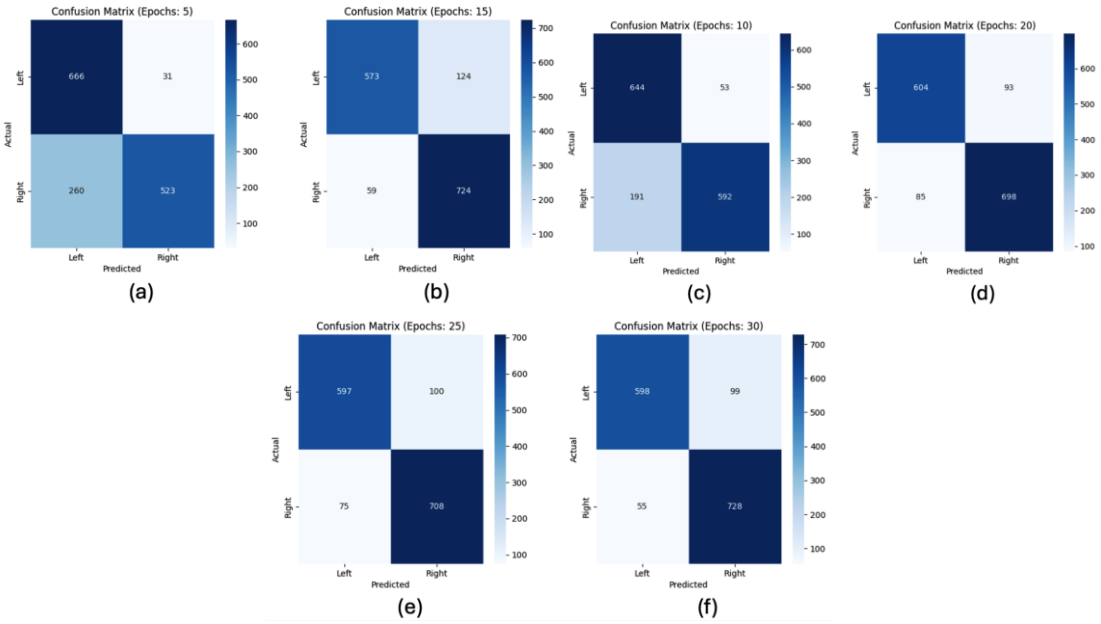

Figure 4. *Confusion Matrices*

Based on the evaluation metrics across epochs, epoch 30 emerges as the most optimal point for fine-tuning, with the highest accuracy (89.59%), balanced precision (88.03%), and recall (92.98%). The ROC AUC also peaks at 0.961, indicating that the model at this stage is highly effective in distinguishing between the two commands. This suggests that fine-tuning the model from epoch 30 would yield the best results, particularly in noisy or augmented data scenarios.

3.2 Fine-Tuning & Evaluation

To improve the model's generalization and robustness, fine-tuning was performed using noise-augmented data, simulating real-world environments with background noise. The aim was to enhance the model's ability to distinguish between 'right' and 'left' commands in noisy settings. Figure 4 compares spectrograms of the original (X) and noisy (Y) audio, where the noisy version shows added distortions, making classification more challenging.

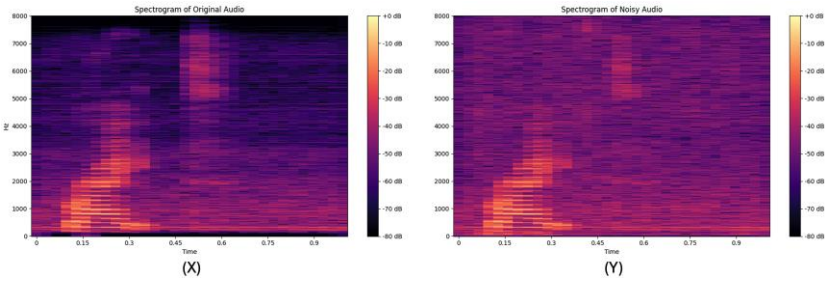

Figure 5. *Data Comparison*

The original model, trained for 30 epochs (as shown in Table 1), achieved an accuracy of 0.90 and an AUC of 0.96. To enhance performance in noisier environments, the model was fine-tuned using augmented data for an additional 30 epochs, while maintaining constant hyperparameters such as batch size and learning rate. The outcomes of this fine-tuning process are presented in Table 2.

Table 2. Performance Comparison Before and After Fine-Tuning (FT)

Epoch	Accuracy	Precision	Recall	AUC
5	0.93	0.92	0.94	0.98
10	0.93	0.92	0.94	0.98
15	0.93	0.91	0.96	0.99

Building of Informatics, Technology and Science (BITS)
Volume 6, No 3, Desember 2024 Page: 1910−1918
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.6162

Epoch	Accuracy	Precision	Recall	AUC
20	0.93	0.92	0.94	0.98
25	**0.94**	**0.96**	**0.92**	**0.99**
30	0.93	0.95	0.91	0.98

After fine-tuning the model across multiple epochs, the best performance is achieved at Epoch 25. This epoch demonstrates the highest accuracy (93.82%), indicating the model's effectiveness in predicting the correct class, along with the best precision (95.90%), minimizing false positives. Additionally, the AUC score (0.98716) reflects the model's strong ability to distinguish between 'right' and 'left' commands. While Epoch 15 delivered the highest recall (95.80%), capturing more true positives, Epoch 25 offers the best balance across all key metrics, making it the optimal choice for the final model and suitable for real-world deployment where both accuracy and precision are crucial.

3.3 Comparative of Methods

This section presents a comparison of various models used for speech command recognition, including both traditional methods (HMM, GMM) and deep learning models (LSTM-GRU, CNN, etc.). Performance metrics such as accuracy, precision, and recall were used for evaluation.

The traditional models, HMM and GMM, demonstrate significantly lower performance compared to the deep learning models. HMM shows a particularly low precision (0.53), despite having a high recall (0.95), indicating many false positives. GMM performs moderately, with an accuracy of 0.68. On the other hand, deep learning models, such as CNN and LSTM-GRU, offer far superior performance. The CNN model achieves the highest accuracy at 97.42%, while the Google API model exhibits perfect precision (1.00), though it has a slightly lower recall (0.87).

Table 3. *Performance Comparison*

Model	Accuracy	Precision	Recall
HMM	0.52	0.53	0.95
GMM	0.68	0.66	0.80
Google API	0.96	**1.00**	0.87
LSTM-GRU	0.94	0.96	0.92
CNN	**0.97**	0.97	**0.97**
LSTM	0.89	0.89	0.88
GRU	0.88	0.88	0.88

The results clearly indicate that deep learning models significantly outperform traditional models in speech command recognition. The CNN model stands out with the highest accuracy, while the Google API shows exceptional precision. The balance between precision and recall in the LSTM-GRU model also makes it a strong candidate for real-time applications where both metrics are critical.

3.4 Real-World Application Performance

In this section, we evaluate the performance of the fine-tuned LSTM-GRU model under real-world conditions with model epoch 25. The dataset consists of 125 audio samples each for the "Right" and "Left" commands, recorded under minimal noise and real noise conditions. The model is tested at various threshold levels ranging from 50% to 95% to determine its robustness in noisy environments. In this section, we evaluate the performance of the fine-tuned LSTM-GRU model under real-world conditions. The dataset consists of 125 audio samples each for the "Right" and "Left" commands, recorded under minimal noise and real noise conditions. The model is tested at various threshold levels ranging from 50% to 95% to determine its robustness in noisy environments. The results of this evaluation are depicted in Figure 5 and Figure 6.

Figure 5 illustrates the scatter plot of predicted probabilities for the "Right" command under minimal noise and real noise conditions, showing the model's performance difference at a threshold of 50%. Similarly, Figure 6 presents a scatter plot for the "Left" command, comparing minimal noise and real noise conditions at the same threshold.

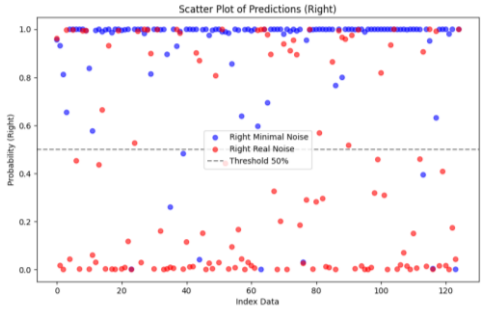

Figure 6. *Right Minimal Noise vs Noise*

Building of Informatics, Technology and Science (BITS)
Volume 6, No 3, Desember 2024 Page: 1910−1918
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.6162

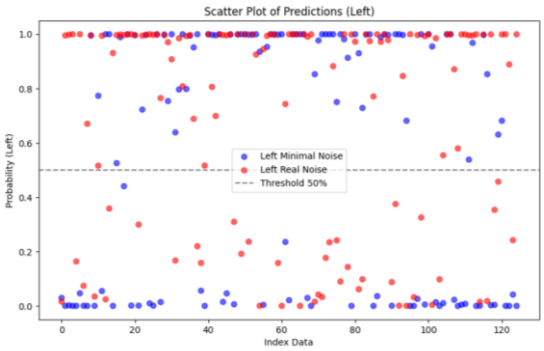

Figure 7. *Left Minimal Noise vs Noise*

Following the visual comparison, Table 4 provides a detailed analysis of accuracy across thresholds for both "Right" and "Left" commands, comparing minimal noise and real noise conditions at threshold levels from 50% to 95%.

Table 4. *Commands Multiple Thresholds*

Command	Threshold Prediction		
	50%	75%	95%
Right Minimal Noise	93	88	81
Right Medium Noise	36	33	22
Left Minimal Noise	40	46	54
Left Medium Noise	32	38	48

The results in Table 4 show that the model accuracy is significantly affected by noise. For the "Right" command, the accuracy drops from 93% in minimal noise to 36% in real noise at a threshold of 50%, with a consistent decrease as the threshold increases. Similarly, for the "Left" command, the accuracy decreases from 40% in minimal noise to 32% in real noise at the same threshold. These findings highlight the need for further fine-tuning to improve the model's robustness against real-world noise.

4. CONCLUSION

This study has successfully developed and fine-tuned an LSTM-GRU hybrid model to classify audio commands under varied noise conditions, specifically designed for individuals with disabilities. The model has shown significant accuracy improvements in clean environments, confirming the potential of deep learning techniques for enhancing voice command recognition. However, the results also highlighted a drop in performance as noise levels increased, revealing that while the model is effective in controlled conditions, further advancements are necessary for noisy environments. Comparing traditional models such as HMM and GMM to deep learning methods like LSTM-GRU, CNN, and Google's Speech Recognition API demonstrated that deep learning approaches, especially those incorporating noise augmentation during fine-tuning, outperform traditional models by a wide margin. The fine-tuning process proved critical in enhancing the model's robustness, with Epoch 25 emerging as the optimal point for deployment, balancing precision and recall while maintaining high accuracy in noisy conditions. In real-world testing, the model performed adequately under minimal noise but exhibited performance degradation with increased noise, indicating the need for more sophisticated noise-handling mechanisms. Future research should explore advanced noise augmentation techniques and alternative model architectures, such as Transformer-based approaches or attention-enhanced models, to further improve speech recognition systems for individuals with disabilities. This study underscores the importance of continuous fine-tuning and evaluation in building robust AI solutions for practical, real-world applications.

REFERENCES

[1] Agnes Z. Yonatan, 'Menilik Distribusi Sektor Pekerja Disabilitas Indonesia', Goodstats.
[2] Cindy Mutia Annur, 'Mayoritas Pekerja Disabilitas Di Indonesia Berstatus Wirausaha', Databoks.
[3] Freya Robinson, '5 Ways Ai Can Help Disabled People In The Workplace', Abilitynet.
[4] Hauke Timmermann, 'Using Ai To Support People With Disabilities In The Workplace', Dotmagazine.
[5] J. J. G. White, 'Artificial Intelligence And People With Disabilities: A Reflection On Human–Ai Partnerships', In *Humanity Driven Ai*, Cham: Springer International Publishing, 2022, Pp. 279–310. Doi: 10.1007/978-3-030-72188-6_14.
[6] X. Xu *Et Al.*, 'Training-Free Acoustic-Based Hand Gesture Tracking On Smart Speakers', *Applied Sciences*, Vol. 13, No. 21, P. 11954, Nov. 2023, Doi: 10.3390/App132111954.
[7] B. I. Alabdullah *Et Al.*, 'Smart Home Automation-Based Hand Gesture Recognition Using Feature Fusion And Recurrent Neural Network', *Sensors*, Vol. 23, No. 17, P. 7523, Aug. 2023, Doi: 10.3390/S23177523.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 3, Desember 2024 Page: 1910−1918
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.6162

[8] Z. Lv, F. Poiesi, Q. Dong, J. Lloret, And H. Song, 'Deep Learning For Intelligent Human–Computer Interaction', *Applied Sciences*, Vol. 12, No. 22, P. 11457, Nov. 2022, Doi: 10.3390/App122211457.
[9] I. Elmagrouni, A. Ettaoufik, S. Aouad, And A. Maizate, 'A Deep Learning Framework For Hand Gesture Recognition And Multimodal Interface Control', *Revue D'intelligence Artificielle*, Vol. 37, No. 4, Pp. 881–887, Aug. 2023, Doi: 10.18280/Ria.370407.
[10] N. Zafar, I. U. Haq, J.-R. Chughtai, And O. Shafiq, 'Applying Hybrid Lstm-Gru Model Based On Heterogeneous Data Sources For Traffic Speed Prediction In Urban Areas', *Sensors*, Vol. 22, No. 9, P. 3348, Apr. 2022, Doi: 10.3390/S22093348.
[11] H. C. Kilinc, S. Apak, F. Ozkan, M. E. Ergin, And A. Yurtsever, 'Multimodal Fusion Of Optimized Gru–Lstm With Self-Attention Layer For Hydrological Time Series Forecasting', *Water Resources Management*, Aug. 2024, Doi: 10.1007/S11269-024-03943-4.
[12] E. Salah, K. Amine, K. Redouane, And K. Fares, 'A Fourier Transform Based Audio Watermarking Algorithm', *Applied Acoustics*, Vol. 172, P. 107652, Jan. 2021, Doi: 10.1016/J.Apacoust.2020.107652.
[13] N. Peng *Et Al.*, 'Environment Sound Classification Based On Visual Multi-Feature Fusion And Gru-Aws', *Ieee Access*, Vol. 8, Pp. 191100–191114, 2020, Doi: 10.1109/Access.2020.3032226.
[14] F. Wang And X. Shen, 'Research On Speech Emotion Recognition Based On Teager Energy Operator Coefficients And Inverted Mfcc Feature Fusion', *Electronics (Basel)*, Vol. 12, No. 17, P. 3599, Aug. 2023, Doi: 10.3390/Electronics12173599.
[15] Q. Li *Et Al.*, 'Msp-Mfcc: Energy-Efficient Mfcc Feature Extraction Method With Mixed-Signal Processing Architecture For Wearable Speech Recognition Applications', *Ieee Access*, Vol. 8, Pp. 48720–48730, 2020, Doi: 10.1109/Access.2020.2979799.
[16] G. Liu And J. Guo, 'Bidirectional Lstm With Attention Mechanism And Convolutional Layer For Text Classification', *Neurocomputing*, Vol. 337, Pp. 325–338, Apr. 2019, Doi: 10.1016/J.Neucom.2019.01.078.
[17] R. L. Abduljabbar, H. Dia, And P.-W. Tsai, 'Unidirectional And Bidirectional Lstm Models For Short-Term Traffic Prediction', *J Adv Transp*, Vol. 2021, Pp. 1–16, Mar. 2021, Doi: 10.1155/2021/5589075.
[18] Y. Imrana, Y. Xiang, L. Ali, And Z. Abdul-Rauf, 'A Bidirectional Lstm Deep Learning Approach For Intrusion Detection', *Expert Syst Appl*, Vol. 185, P. 115524, Dec. 2021, Doi: 10.1016/J.Eswa.2021.115524.
[19] M. Fazil, S. Khan, B. M. Albahlal, R. M. Alotaibi, T. Siddiqui, And M. A. Shah, 'Attentional Multi-Channel Convolution With Bidirectional Lstm Cell Toward Hate Speech Prediction', *Ieee Access*, Vol. 11, Pp. 16801–16811, 2023, Doi: 10.1109/Access.2023.3246388.
[20] J. Jorge, A. Gimenez, J. A. Silvestre-Cerda, J. Civera, A. Sanchis, And A. Juan, 'Live Streaming Speech Recognition Using Deep Bidirectional Lstm Acoustic Models And Interpolated Language Models', *Ieee/Acm Trans Audio Speech Lang Process*, Vol. 30, Pp. 148–161, 2022, Doi: 10.1109/Taslp.2021.3133216.
[21] A. Shewalkar, D. Nyavanandi, And S. A. Ludwig, 'Performance Evaluation Of Deep Neural Networks Applied To Speech Recognition: Rnn, Lstm And Gru', *Journal Of Artificial Intelligence And Soft Computing Research*, Vol. 9, No. 4, Pp. 235–245, Oct. 2019, Doi: 10.2478/Jaiscr-2019-0006.
[22] Y. Dai, H. Rong, Y. Wu, C. Yang, And Y. Xu, 'Stall Flutter Prediction Based On Multi-Layer Gru Neural Network', *Chinese Journal Of Aeronautics*, Vol. 36, No. 1, Pp. 75–90, Jan. 2023, Doi: 10.1016/J.Cja.2022.07.011.
[23] S. Mahjoub, L. Chrifi-Alaoui, B. Marhic, And L. Delahoche, 'Predicting Energy Consumption Using Lstm, Multi-Layer Gru And Drop-Gru Neural Networks', *Sensors*, Vol. 22, No. 11, P. 4062, May 2022, Doi: 10.3390/S22114062.
[24] Z. Niu, G. Zhong, And H. Yu, 'A Review On The Attention Mechanism Of Deep Learning', *Neurocomputing*, Vol. 452, Pp. 48–62, Sep. 2021, Doi: 10.1016/J.Neucom.2021.03.091.
[25] A. M. Javid, S. Das, M. Skoglund, And S. Chatterjee, 'A Relu Dense Layer To Improve The Performance Of Neural Networks', In *Icassp 2021 - 2021 Ieee International Conference On Acoustics, Speech And Signal Processing (Icassp)*, Ieee, Jun. 2021, Pp. 2810–2814. Doi: 10.1109/Icassp39728.2021.9414269.
[26] X. Liang, X. Wang, Z. Lei, S. Liao, And S. Z. Li, 'Soft-Margin Softmax For Deep Classification', 2017, Pp. 413–421. Doi: 10.1007/978-3-319-70096-0_43.