

Real-time recognition of Indonesian sign language SIBI using CNN-SVM model combination

Satriadi Putra Santika¹, Stefanus Benhard¹, Yulyani Arifin², Andry Chowanda³

¹Department of Computer Science, Binus Graduate Program-Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia

²Department of Computer Science, Binus Graduate Program-Doctor of Computer Science, Bina Nusantara University, Jakarta, Indonesia

³Department of Computer Science, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia

Article Info

Article history:

Received Dec 3, 2024

Revised Mar 24, 2025

Accepted Jul 2, 2025

Keywords:

CNN-SVM

Post-processing

Real-time recognition

SIBI

Sign language recognition

ABSTRACT

Real-time *Sistem Isyarat Bahasa Indonesia* (SIBI) sign language recognition plays a crucial role in improving accessibility for individuals with hearing and speech impairments. Despite advancements in SIBI recognition research, challenges remain in ensuring model stability and accuracy in real-time settings, particularly in handling gesture variations and classification inconsistencies. This study addresses these challenges by developing a convolutional neural network-support vector machine (CNN-SVM) combination model, integrating MediaPipe for hand coordinate extraction, CNN for feature extraction, and SVM for classification. To improve generalization and prevent overfitting, data augmentation is applied to expand the dataset. The model's performance is further enhanced through hyperparameter optimization (HPO) and post-processing techniques such as multi-window majority voting (MWMV) and SymSpell. Experimental results show that the CNN-SVM model trained on augmented data with HPO achieves 91% testing accuracy, outperforming both standalone CNN and SVM models. Furthermore, MWMV improves recognition stability, while SymSpell enhances spelling errors, ensuring more meaningful outputs. The system is integrated with OpenCV for real-time recognition, but current deployment remains limited to local execution. Future work will focus on developing lightweight models for web-based and mobile applications, making the system more accessible and scalable.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Satriadi Putra Santika

Department of Computer Science, Binus Graduate Program-Master of Computer Science

Bina Nusantara University

11530 Jakarta, Indonesia

Email: satriadi.santika@binus.ac.id

1. INTRODUCTION

Communication is a fundamental aspect of human life that enables social interaction, information exchange, and self-expression [1]. However, limitations in communication are a big challenge for individuals with hearing and speech impairments. These barriers create a gap that further complicates their participation in society. Based on the World Report on Hearing (WRH) published by the World Health Organization (WHO), people with hearing and speech impairments still experience limitations in communication, which impacts their accessibility in various aspects of life [2].

Sign language is a key solution in helping deaf and speech impaired people to communicate. However, the different sign language systems in different countries and the lack of public understanding of sign language are still issues in realizing inclusive communication. In Indonesia, there are two sign language

systems used, which are *Bahasa Isyarat Indonesia* (BISINDO) and *Sistem Isyarat Bahasa Indonesia* (SIBI) [3]. BISINDO translates a single word from spoken Indonesian according to its context, then followed by a gesture that describes the situation at that time [4]. In contrast to BISINDO, SIBI is more natural and versatile in translating spoken language to sign language [5]. SIBI maintained the grammatical structure of Indonesian, including prefixes and suffixes on each word. SIBI has been established as an official sign language system in the education curriculum for deaf and speech impaired students in special schools based on the Minister of Education and Culture Decree Number 0161/U/1997 [6].

Currently, society still faces a challenge in understanding sign language. This challenge emphasizes the need for a solution that can bridge communication between deaf and speech impaired people and society. One such approach is sign language recognition (SLR), a technology that converts sign language gestures into text or voice to enable more inclusive communication. Through the SLR system, individuals without sign language skills can more easily understand the messages conveyed by sign language users. In addition, this technology also has the potential to be an effective learning tool for society in learning sign language.

Research on SLR, especially SIBI, has been conducted with various approaches. We found that previous research on Indonesian sign language recognition explored various feature extraction methods and classification models. Feature extraction and classification in sign language recognition play an important role in improving accuracy and efficiency for interpreting signs [7]. Traditional machine learning-based models, such as support vector machine (SVM) and decision tree (DT) have been applied in SIBI alphabet classification. Nugraha *et al.* [8] implemented DT and C4.5, showing that DT achieved 77.82% accuracy, while C4.5 obtained 71.09% with the best performance at 250 nodes. Insani *et al.* [9] used leap motion control as a tool to perform feature extraction, resulting in 63 point coordinates (x, y, z) which were then fed into the SVM classification model and resulted in 96.2% accuracy. Furthermore, Veeraiah *et al.* [10] demonstrated the effectiveness of SVM in SLR, using MediaPipe for hand landmark feature extraction and SVM as the main classifier. The model was able to recognize letters and words in SIBI with 99.8% accuracy, making it one of the most stable methods in sign language classification.

Other than traditional machine learning methods, deep learning is also widely explored for SLR. Handayani *et al.* [11] implemented MediaPipe to perform feature extraction, resulting in 42 coordinate points (x, y) which were then input into a backpropagation based artificial neural network (ANN) model. This model achieved 86.26% accuracy with the optimal configuration of learning rate 0.1, momentum 0.1, and epoch 700. Bagaskoro *et al.* [12] proposed feed-forward neural network (FFNN) and multi layer perceptron (MLP) to classify SIBI alphabet. The dataset used consisted of 32,850 digital images of the SIBI alphabet, which were converted into numerical parameters. The results showed that MLP gave an accuracy of 85.46%, while FFNN only reached 81%. Moving away from the ANN model, Sihananto *et al.* [13] used convolutional neural network (CNN) to perform feature extraction which was then continued into a fully connected layer with SoftMax activation function as a classification model that achieved an accuracy of up to 93.29%. This research shows that CNN can recognize hand gestures well but still faces challenges in distinguishing letters that have similar hand shapes. Limantara and Trisianto [14] also adopted a similar approach, but before feeding into the CNN model for feature extraction, the images were processed using grayscale and Gaussian blur to improve the quality of the features resulting in 99% accuracy, suggesting that the combination of preprocessing and CNN can increase the effectiveness in recognizing sign language. Furthermore, Afkaar [15] tried to combine MediaPipe and CNN to perform feature extraction, which was then continued into a fully connected layer with SoftMax activation function as a classification model. The results showed that the accuracy reached 96.1% on the testing data.

Seeing the potential for further development in SLR, more complex models began to be adopted to overcome the challenges that still exist in SLR systems. For example, Suharjito *et al.* [6] developed a transfer learning-based model with inflated three-dimensional (3D) CNN, which allows the system to recognize sign language gestures with 97.5% accuracy. In addition, Handayani *et al.* [16] compared ResNet-50 and EfficientNet-B0 in SIBI alphabet classification, finding that EfficientNet-B0 with data augmentation achieved the highest accuracy of 98.3%. Furthermore, Suyitno *et al.* [17] implemented YOLOv8 to detect sign language gestures in real-time, using a dataset that includes 107 vocabulary and 7 affix classes. The system can recognize up to 100 words under optimal lighting conditions, but the detection accuracy drops to 58.02% under low lighting conditions.

Although various approaches have been taken in SLR research, especially SIBI, there are still some limitations that need to be addressed to improve accuracy, stability, and efficiency in real-time scenarios. Previous research in SLR shows that most models still use a single model, such as SVM, ANN, or CNN. Several previous studies have proven the effectiveness of each model. Handayani *et al.* [11] and Sihananto *et al.* [13] show that CNN can recognize spatial patterns from hand gestures well. However, this model still has difficulty in distinguishing letters that have similar hand gestures. On the other hand, Insani *et al.* [9] and Veeraiah *et al.* [10] showed that SVM has high stability in the classification of SIBI letters and words. By

looking at the capabilities of each model, a combination approach of CNN as a feature extractor and SVM as the main classifier can provide a more optimal solution.

Furthermore, it is known that SLR systems suffer from fluctuating prediction results, mainly due to changes in lighting, camera angle, or variations in hand position, which can destabilize the system in real-time scenarios [18]. However, there has not been much previous research applying post-processing strategies to address this issue. Wahid *et al.* [19] showed that multi-window majority voting (MWMV) can improve stability in electromyography (EMG) signal-based classification. MWMV works by using information from various window sizes simultaneously so that the prediction becomes more stable and does not depend only on one frame. Additionally, errors in the transcription of sign language classification results are still a challenge that has not been studied much in previous research. Audah *et al.* [20] compared SymSpell and Damerau-Levenshtein Trie (DLTrie) in Indonesian spelling correction. The results showed that SymSpell is faster and more accurate with a runtime of only 0.39 ms per word compared to 44.15 ms per word for DLTrie. The research shows Symspell has the potential to improve the transcription quality of sign language classification results, with the aim of reducing spelling errors that arise due to inaccurate model predictions.

Besides classification and post-processing aspects, dataset selection is also an important factor in improving SLR systems. Afkaar [15] developed MediaPipe and CNN models for SLR and achieved high accuracy, but there is a tendency to experience overfitting due to the small number of datasets. In fact, this dataset has a complete alphabet class and enough variety to potentially be a strong baseline for this research. The overfitting problem will be addressed by applying data augmentation to improve model generalization. The application of augmentation allows the model to learn from a wider variety so that it can handle differences in lighting, hand orientation, as well as user variations which were previously limitations in Afkaar's research.

Based on the limitations found in previous studies, this research proposes MediaPipe and the combination of CNN with SVM as a solution to improve the accuracy and stability of real-time SLR systems. MediaPipe is used to extract hand point coordinates, which are then processed by CNN for feature extraction, and then classified using SVM to improve robustness to dataset variations. Furthermore, to ensure prediction stability in real-time scenarios, MWMV is applied as a post-processing strategy to reduce fluctuations in classification results between frames and improve overall system stability. In addition, SymSpell is also used to reduce errors in the transcription of the classified text so that the system is not only better at recognizing gestures, but also more stable and efficient for real-time applications. The system will be integrated with OpenCV to handle image processing straight from the camera. Through this approach, this research aims to improve classification accuracy, prediction stability, and transcription error so that this SLR system can be better and more efficiently applied in the real world.

2. METHOD

This research conducted several experiments with CNN, SVM, and CNN-SVM combination models using Afkaar's dataset [21] as the baseline dataset which can be accessed through this link <https://www.kaggle.com/datasets/mlanangafkaar/datasets-lemlitbang-sibi-alphabets>. The research stages are shown in Figure 1, which includes data preprocessing by performing data augmentation, extraction hand point coordinates with MediaPipe, feature extraction using CNN, and classification with SVM. The model is evaluated using accuracy metrics and training and loss plots, then MWMV and SymSpell post-processing are applied. The final stage is the integration of the real-time SLR with OpenCV.

We constructed eight experimental variants to evaluate the impact of data augmentation, HPO, and classifier selection on SIBI sign language recognition. The purpose of these experiments is to determine whether SVM outperforms SoftMax as a classifier and whether data augmentation improves generalization to find the most effective approach towards sign language recognition. The experimental setup and model variations are detailed in Table 1.

Table 1. Summarize of experimental setups

Experiment	Dataset	Feature extractor	Classifier	Hyperparameter optimization (HPO)
1	Baseline	CNN (Base model)	Softmax	No
2	Baseline	CNN (HPO)	Softmax	Yes
3	Baseline	-	SVM	Yes
4	Baseline	CNN (HPO)	SVM	Yes
5	Augmented	CNN (Base Model)	Softmax	No
6	Augmented	CNN (HPO)	Softmax	Yes
7	Augmented	-	SVM	Yes
8	Augmented	CNN (HPO)	SVM	Yes

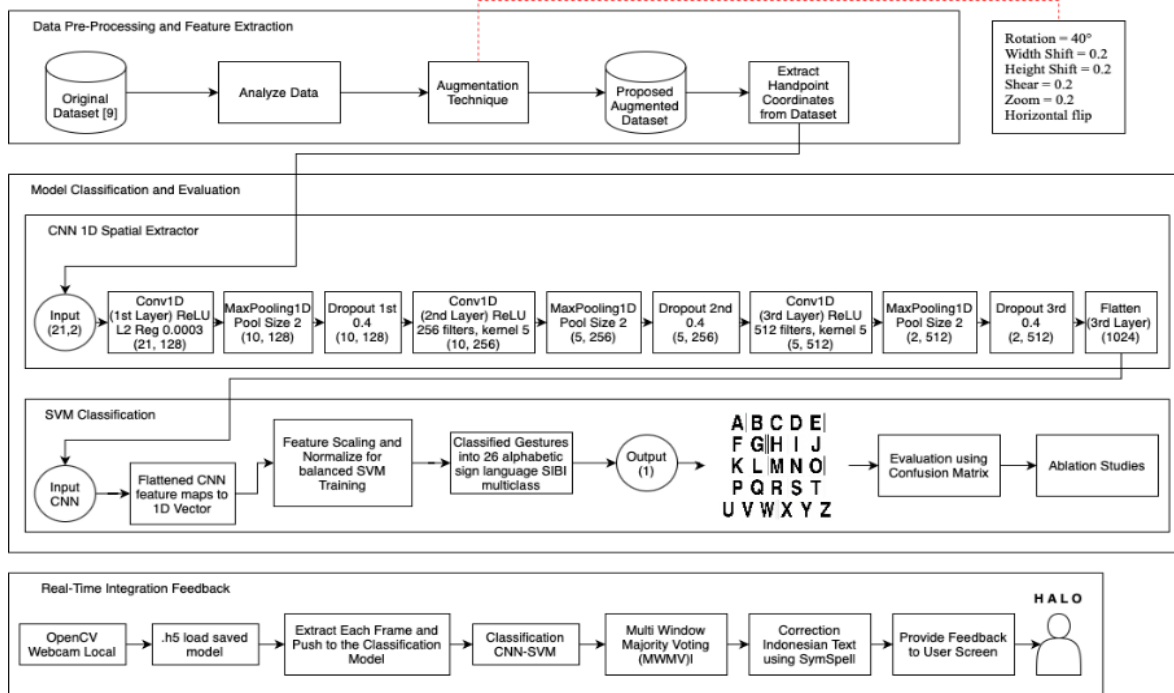


Figure 1. General process experiment

2.1. Data preprocessing

Data preprocessing is conducted by performing data augmentation. Data augmentation was performed to increase the amount and the model's ability to generalize in recognizing the data [22]. Various augmentation techniques were applied, such as rotation=40°, width shift=0.2, height shift=0.2, shear=0.2, zoom=0.2, and horizontal flip. This augmentation process helps the model adapt to position variations, making it more robust in handling diverse real-world conditions. Afterward, the dataset was displaced to ensure an optimal balance for model training and evaluation. The augmented training data is divided into training and validation sets at a ratio of 8:2, to ensure that the model learns effectively and is validated regularly to prevent overfitting. Then, the original validation data is combined with the original testing data to form a new testing set. In this study, the augmented data results will not be included in the testing set to maintain the integrity of the evaluation process and ensure that the model is evaluated using only the original data.

2.2. Extract hand point coordinates

We are using MediaPipe as it is a well-proven extraction method especially for hand gesture recognition in a real-time context. MediaPipe can detect 21 points on one hand by projecting the key positions of each finger, palm, and wrist, as shown in Figure 2. The choice of using 21 hand landmarks is based on its effectiveness in gesture-based human-computer interaction (HCI). In this study, we use MediaPipe to extract 42 coordinate points (x, y) from one hand. Extracting 42 coordinate points can increase precision in gesture recognition. The 42 coordinate points will be reshaped into the form (21, 2), where each hand point consists of x and y coordinates. The extracted (21,2) coordinate matrix serves as the input for the feature extraction stage, where a CNN 1D model processes the hand trajectory data for classification [23].

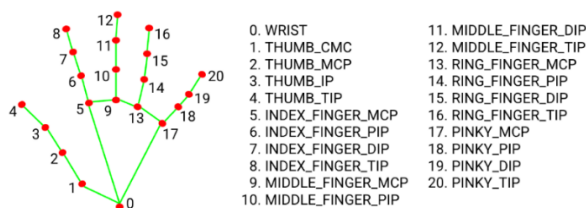


Figure 2. The 21 hand landmarks localized by MediaPipe

2.3. Feature extraction

At this stage, the results of extracting hand point coordinates with MediaPipe will be entered into the CNN1D model for feature extraction. CNN1D is used because of its ability to handle sequential data, such as hand gestures [24]. Feature extraction in this process aims to identify patterns and features that are important in recognizing each hand movement in sign language so that the model can more accurately distinguish the movements of each class [25].

There are two CNN1D architectures used in this research. The first architecture refers to the base model adapted from research conducted by Afkaar [15]. In the second architecture, we performed HPO through RandomSearchCV method with parameters filters=[32, 64, 128], kernel size=[3, 5], learning rate=[0.001, 0.0005, 0.0001], batch size=[64, 128], and epochs=[100, 150]. We limited the number of CNN1D layers to 3 layers and found that the best CNN1D model is with parameters filters=128, kernel size=5, learning rate=0.001, batch size=128, and epochs=150.

2.4. Classification

After the process of feature extraction, the classification of hand gestures is done by the SVM model [10], [26]. In this stage, we also use RandomSearchCV to perform HPO with kernel parameters=[linear, rbf, poly, sigmoid], gamma=[scale, auto], degree=[2, 3, 4, 5], coef=[0.1 - 1], and C=[0.001, 0.01, 0.1, 1, 10, 100]. We performed HPO twice, namely for the baseline dataset and the augmented dataset. On the baseline dataset, the best SVM model is obtained with kernel parameters=linear, gamma=scale, degree=4, coef=0.2, and C=10. Meanwhile, for the augmented dataset, the best SVM model is obtained with kernel parameters = linear, gamma=auto, degree=2, coef=0.8, and C=100.

2.5. Model evaluation

In the evaluation stage, the performance of the model will be examined in two ways: first, with the accuracy metric and second, by analyzing accuracy and loss function plot during the training process. The accuracy metric will show how well the model classifies the sign language into the right class [27]. Meanwhile, the plot can provide insight into how well the model learns and identify problems such as overfitting and underfitting [28].

2.6. Post-processing

In this experiment, post-processing techniques are used to improve the real-time SIBI SLR system to make the system more consistent and robust. There are two methods, which are MWMV and SymSpell. MWMV is a strategy that combines classification results from multiple overlapping data windows to improve gesture recognition accuracy [19]. Real-time systems often perform predictions in a short and continuous time which causes inconsistent prediction results. Therefore, in this study, each window is set within 2 seconds to collect multiple prediction results. The final prediction result displayed will be determined based on the prediction that appears most often or with the highest frequency. An illustration of the MWMV method can be seen in Figure 3.

Although the consistency of prediction has improved by using MWMV, some misclassification is still possible which will lead to errors in spelling. To address this, we added the SymSpell method. SymSpell is a method used for spelling correction that is well-known for its high performance and speed [20], making it suitable to be applied to real-time systems. If the resulting spelling is not correct, SymSpell will suggest the closest valid word according to the *Kamus Besar Bahasa Indonesia* (KBBI). An illustration of the SymSpell method can be seen in Figure 4.

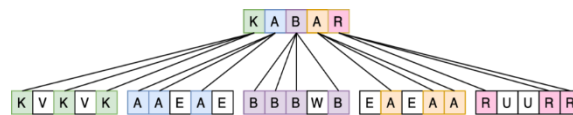


Figure 3. Illustration of MWMV method

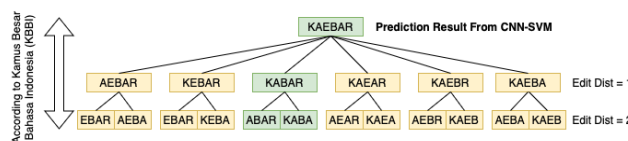


Figure 4. Illustration of SymSpell method

2.7. System integration with OpenCV

In this final stage, we integrated the SIBI SLR system with OpenCV for real-time gesture recognition. OpenCV is used to capture and process images so as to recognize hand gestures therefore allowing non-verbal interaction with computers [29], [30]. This will ensure that the user can interface with the system through a camera that will capture the hand gestures of the user and provide results in the form of live letter recognition. The integration steps are shown in Figure 5. The system started by capturing image per frame from videos, then it will be processed sequentially into CNN-SVM classification model to identified hand gestures. To enhance prediction stability, an MWMV and SymSpell technique has been used as a post-processing method, and the final processed text is displayed on the user interface and provide real-time feedback.

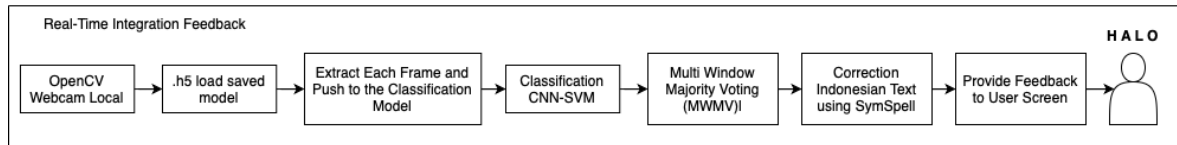


Figure 5. Real-time integration feedback flow

3. RESULT AND DISCUSSION

3.1. Data preprocessing

After augmentation, the amount of data in the dataset has increased significantly. The dataset that originally consisted of 1,092 images for training data, 220 images for validation data, and 26 images for testing data, is now expanded to 5,497 images in training data, 1,387 images in validation data, and 246 images for testing data. This adjustment ensures that each class has a more balanced number of samples so that the model can learn from a wider variety and improve its generalization ability. In addition, increasing the amount of testing data allows for a more representative evaluation of the model, reducing the risk of overfitting that arises from a too limited amount of data [31]. Figure 6 shows an example of data augmentation results.

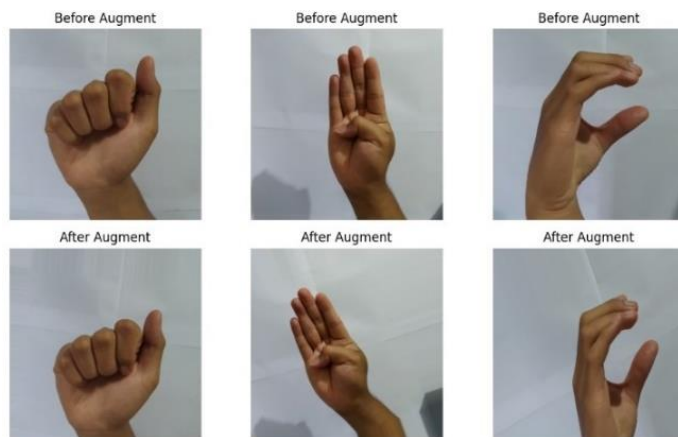


Figure 6. Example of data augmentation result

3.2. Model performance analysis

Evaluation of model performance in SLR is an important step to determine the effectiveness of the approach used in this study. Comparisons were made on various model configurations, including CNN base model [15], CNN with HPO, SVM as the classifier, and a combination of CNN (HPO)+SVM. Moreover, this experiment also evaluates the impact of data augmentation on model performance in training, validation, and testing data. Table 2 shows the comparison of training, validation, and testing accuracy for each model configuration, both on the baseline dataset (without augmentation) and the augmented dataset.

Table 1. Comparison of model accuracy for each experiment

No	Experiment	Training accuracy	Validation accuracy	Testing accuracy
1	Baseline dataset + CNN (Base model)	94.1%	88.8%	96.1%
2	Baseline dataset + CNN (HPO)	99.7%	96.1%	100%
3	Baseline dataset + SVM	100%	97%	100%
4	Baseline dataset + CNN (HPO)+SVM	100%	98%	100%
5	Augmented data + CNN (Base model)	96.8%	90.4%	77.5%
6	Augmented data + CNN (HPO)	95.1%	95.8%	90.1%
7	Augmented data + SVM	95%	87%	84%
8	Augmented data + CNN (HPO)+SVM	100%	96%	91%

Based on Table 2, accuracy tends to decrease after data augmentation. This is due to the increase in dataset size, which makes it more difficult for the model to effectively extract features from each image. However, despite the slight decrease in accuracy, the augmented dataset is still better than the baseline dataset because data augmentation helps improve the generalization of the model. With augmentation, the model is forced to learn from a wider range of variations, such as flipping, rotation, and scale, thus improving its ability to recognize sign language gestures under real-world conditions [32]. Althnian *et al.* [33] mentioned that small datasets can cause overfitting and hinder the model's ability to generalize to new data. Therefore, the use of data augmentation plays an important role in increasing the model's robustness to input variations. Additionally, although data augmentation helps improve generalization, its impact may vary depending on the complexity of the model, the quality of the dataset, and the preprocessing techniques used. In some cases, non-optimized models may struggle to adapt to the increased diversity in the augmented data, leading to slight fluctuations in accuracy. This highlights the importance of balancing dataset expansion with proper hyperparameter tuning, as excessive augmentation without optimization can increase computational complexity without significant performance gains.

In addition, the experimental results also show that applying HPO using RandomSearchCV can significantly improve the accuracy of the model on both the baseline and augmented datasets. This shows that HPO is effective in determining the best parameters for the model, as also discovered in the study of Erden *et al.* [34]. Models optimized with HPO showed consistent performance improvement, especially on more varied datasets after augmentation. The effectiveness of HPO is especially noticeable after data augmentation, where the complexity of the dataset increases. Without optimization, models may struggle to efficiently extract relevant patterns from more diverse datasets, leading to performance inconsistencies. However, HPO helps stabilize learning by selecting optimal hyperparameters, ensuring that the model remains robust even when trained on larger and more complex datasets.

Furthermore, when compared between the CNN, SVM, and CNN-SVM combination models, the results show that the CNN-SVM combination has better performance than the single model. On the baseline dataset, the difference in performance between models may not be very noticeable because the dataset size is too small so that all models tend to overfitting with very high accuracy reaching 100%. However, on the augmented dataset, there is an increase in accuracy from CNN (90.1%) and SVM (84%) to 91% with the CNN-SVM combination. This result further strengthens the theory put forward by Khairandish *et al.* [35] which states that the CNN-SVM combination can improve accuracy compared to using CNN and SVM separately. The improved performance of CNN-SVM on augmented data shows that this combination model is more robust in handling hand gesture variations, as CNN efficiently extracts spatial features, while SVM performs well in high-dimensional classification tasks. Moreover, the combination reduces misclassification errors, especially in cases where several sign gestures share similar visual characteristics, proving its effectiveness in real-world applications.

Moreover, if we look at the accuracy and loss plots during the training process, we can get an insight into the stability of the training and the ability of the model to generalize. The accuracy and loss plot comparison are shown in Figure 7. Specifically, Figure 7(a) presents the accuracy and loss plot for experiment 1, Figure 7(b) presents the accuracy and loss plot for experiments 2 and 4, Figure 7(c) presents the accuracy and loss plot for experiment 5, and Figure 7(d) presents the accuracy and loss plot for experiments 6 and 8.

Based on Figures 7(a) to 7(c), it can be seen that the training process is not stable, characterized by significant fluctuations. This pattern indicates that the model has not been able to achieve convergence well, and there are indications of overfitting in the experiment. This can be attributed to the baseline dataset being too small so that the model tends to memorize the training data without being able to recognize new patterns. In addition, the training accuracy and loss continues to decrease but the validation accuracy and loss remain fluctuating illustrates that the model has difficulty in generalizing the validation data.

In contrast to the previous results, Figure 7(d) shows that the training process is more stable with a smoother accuracy and loss plot that tends to converge. This indicates that the model is better able to

generalize the data and does not experience excessive fluctuations during the training process, which has been shown to improve model stability and maximize the training process. This finding is in line with research conducted by Rao [36], which shows that HPO plays an important role in determining the optimal parameters to improve model generalization in deep learning-based classification. The smoother convergence seen in Figure 7(d) suggests that HPO not only fine-tunes the hyperparameters but also helps in achieving a better balance between underfitting and overfitting. The ability of the model to generalize well without sharp fluctuations is crucial for real-time applications, where consistency and reliability in predictions are essential for user interaction and practical deployment.

From the analysis that has been presented, it can be concluded that the best model is found in experiment 8, which is a model that combines CNN (HPO) with SVM using augmented data. This model was chosen as the best model because it has the most stable training process and produces 91% testing accuracy on augmented data. After this, the best model will be used in the development of the real-time SIBI SLR system to ensure the system can run stably and accurately in real-world scenarios.

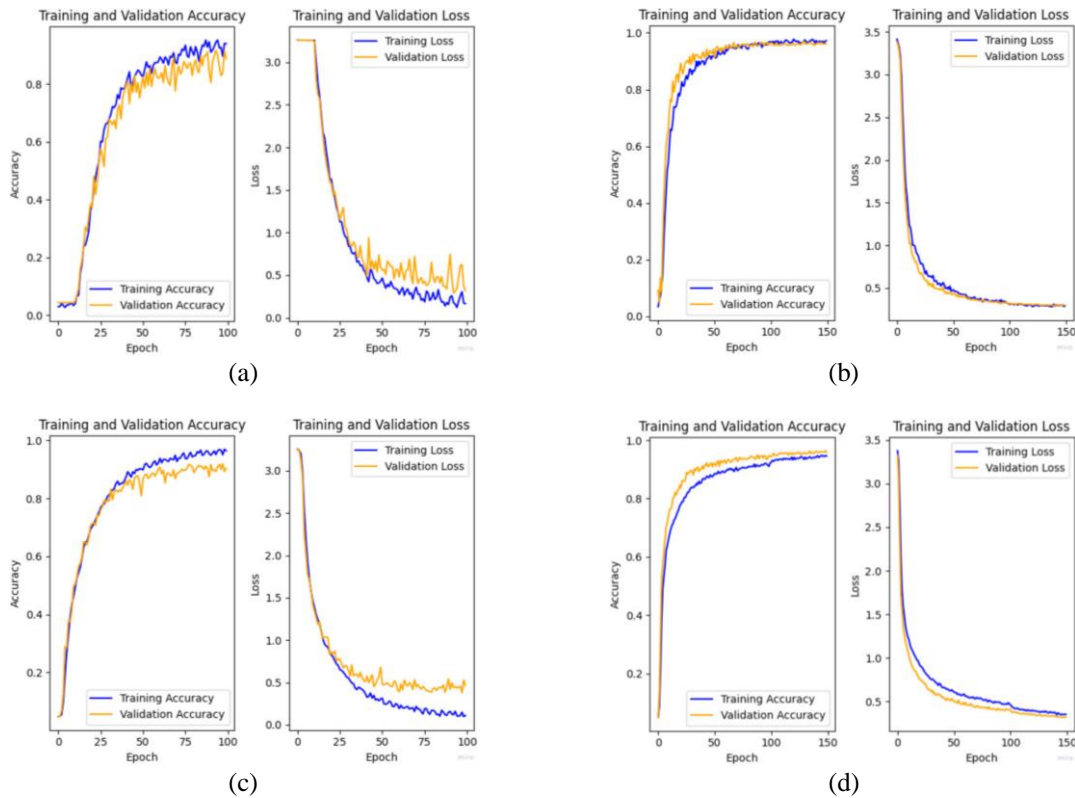


Figure 7. Plot accuracy and loss training and validation (a) experiment 1, (b) experiment 2 and 4, (c) experiment 5, and (d) experiment 6 and 8

3.3. Post-processing result

The stability of sign language recognition results is an important factor in real-time systems. The application of MWMV is done to overcome classification instability due to rapid frame changes [19]. Figure 8 shows the hand gesture recognition results. Specifically, Figure 8(a) shows the hand gesture recognition results without MWMV and Figure 8(b) shows the hand gesture recognition results with MWMV.

Based on Figure 8(a), without the use of MWMV, the recognition results are very unstable, with letters constantly changing in each frame. As a result, the recognized letters cannot be properly arranged into meaningful words or sentences. This is due to the similarity in the shape of some hand gestures in the SIBI alphabet which causes the model to often misrecognize letters in too fast frame changes.

In contrast to Figure 8(b), after applying MWMV, the recognition results become more stable and consistent. Through the MWMV method, the system can filter out the most frequent classification results within a 2s period so that the recognized letters are more appropriate and can be arranged into correct words, such as “HALO” in Figure 8(b). Thus, the use of MWMV not only improves the stability of the SLR system but also provides better user experience. This result is in line with the research conducted by Padfield *et al.*

[37] and Wahid *et al.* [19], which showed that MWMV is effective in correcting classification errors in real-world scenarios. The increased stability provided by MWMV is particularly important in real-time applications, where continuous motion and environmental factors can cause classification inconsistencies. By reducing recognition fluctuations, MWMV helps to minimize classification errors caused by temporary hand position changes or motion blur, thus ensuring that the system remains reliable under various conditions.

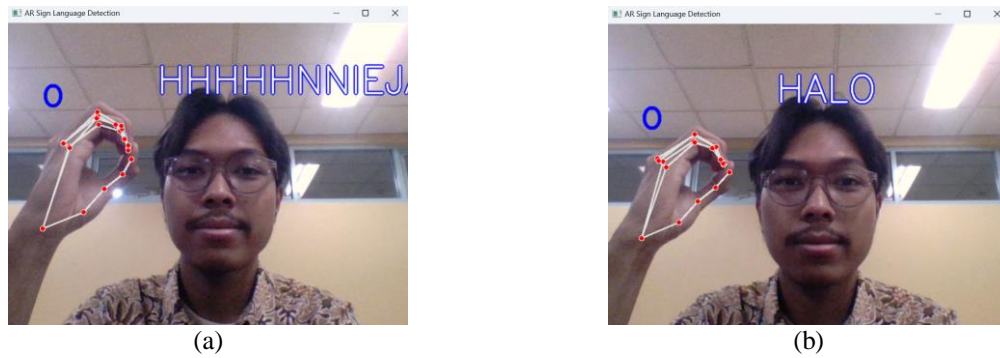


Figure 8. Recognition result (a) without MWMV and (b) with MWMV

Although MWMV has improved recognition stability, there are still spelling errors that require additional correction. Therefore, the SymSpell method is applied as an additional post-processing stage to correct spelling errors that occur during classification. Figure 9 shows the hand gesture recognition results. Specifically, Figure 9(a) shows the hand gesture recognition results without SymSpell and Figure 9(b) shows the hand gesture recognition results with SymSpell.

Based on Figure 9(a), before the application of SymSpell, the system recognized the word as “KAEBAR” which does not match the desired word. This error occurs due to unstable letter recognition in some frames so that the system produces an incorrect arrangement of letters. After the SymSpell method is applied, the system can correct the recognition result into a more appropriate word, namely “KABAR” as shown in Figure 9(b).

The success of this correction shows that the integration of SymSpell in the system can increase the model's resistance to spelling errors so that the recognition results become more accurate and easier to understand. This is in line with the research of Rivera-Acosta *et al.* [38] which shows that the application of spelling correction in sign language translation systems can increase the model's robustness to variations in input letters. By automatically correcting misclassified words, SymSpell ensures that recognition errors do not significantly impact the meaning of the output, making the system more practical for real-world use. This improvement is particularly important in real-time scenarios, where small variations in hand gestures or momentary recognition errors can lead to incomprehensible results, highlighting the important role post-processing plays in maintaining system reliability.

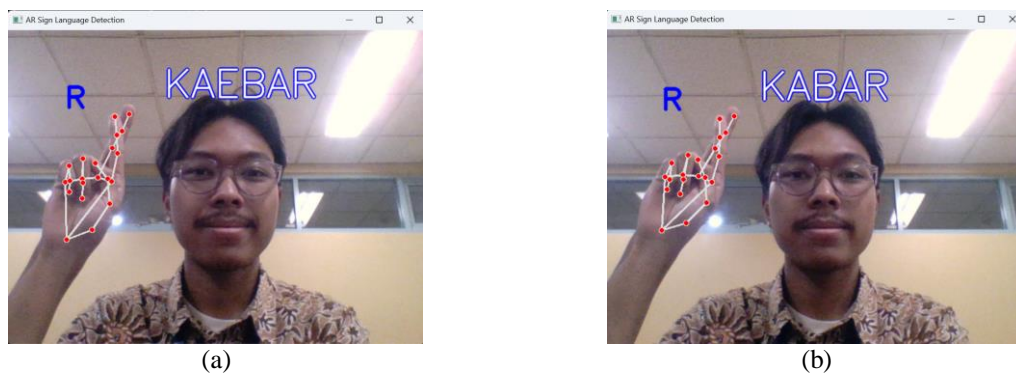


Figure 9. Recognition result (a) without SymSpell and (b) with SymSpell

By applying MWMV and SymSpell as post-processing methods, the system becomes more stable in composing meaningful words. This is especially important in real-time applications as errors in recognition can be corrected immediately, resulting in a more natural and user-friendly output. These two methods ensure that even if there are variations in hand gestures or misclassifications, the recognition results remain appropriate and form a meaningful word.

3.4. System integration

After all the systems have been successfully created, the last step is to integrate all the main components with OpenCV so that they can be recognized in real-time. This integration allows the real-time SIBI sign language recognition application to capture and recognize the user's hand movements directly. Figure 10 shows the appearance of the application that has been integrated. Through this integration, the application can recognize real-time hand gestures from camera (webcam) input, translate them into letters, and assemble them into a word, such as "APA KABAR". MWMV and SymSpell methods also makes sure the assembled letters to words are meaningful. This allows the application to not only recognize gestures accurately but also provide output that is more natural and easily understood by the user.

These findings have significant implications for the development of real-time assistive communication technologies, particularly for individuals with hearing and speech impairments. The successful integration of CNN-SVM with OpenCV, MWMV, and SymSpell demonstrates that a hybrid approach can improve accuracy, stability, and usability in real-world applications. Moreover, with the system now fully integrated and capable of recognizing sign language in real-time, it can serve as a practical tool for facilitating more inclusive communication, bridging the gap between sign language users and the public, and supporting educational environments where learning sign language is essential.



Figure 10. Result of system integration

4. CONCLUSION

SLR is essential in bridging communication gaps for individuals with hearing and speech impairments, enabling greater accessibility and inclusion in daily interactions. However, achieving stable and accurate real-time recognition remains a challenge, particularly in handling gesture variations and classification inconsistencies. This research successfully developed a real-time SIBI SLR system by integrating MediaPipe for hand point coordinate extraction, CNN for feature extraction, and SVM for classification. The study demonstrates that the CNN-SVM combination outperforms individual models, achieving 91% accuracy on the augmented dataset, which is higher than CNN (90.1%) and SVM (84%). The application of HPO using RandomSearchCV has also been shown to enhance model stability and generalization, ensuring robust performance under different conditions.

In addition, MWMV effectively reduces classification inconsistencies, making recognition results more stable, while SymSpell enhances spelling error, leading to more coherent and meaningful outputs. These findings highlight the importance of combining deep learning and machine learning techniques to enhance real-time sign language recognition. However, the current implementation is limited to local execution using OpenCV, restricting accessibility on web-based and mobile platforms. Future research should focus on developing lightweight models for broader accessibility, while maintaining an optimal balance between computational efficiency and classification accuracy.

Although independent models are often considered sufficient, this study shows that combination approaches, such as CNN-SVM, can improve accuracy and robustness, especially when integrated with post-processing techniques such as MWMV and SymSpell. These results reinforce the potential of integrating real-time sign language recognition into assistive communication technologies, contributing to a more

inclusive and accessible digital environment. Moreover, the combination of feature extraction, classification, and post-processing methods in this study provides a solid foundation for improving real-time sign language recognition, making interactions smoother and more effective for users.

FUNDING INFORMATION

The authors state that there is no funding involved in the conduct of this research.

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
Satriadi Putra Santika	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓
Stefanus Benhard	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓			✓
Yulyani Arifin				✓								✓		
Andry Chowanda				✓				✓				✓		

C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal analysis

I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review & Editing

Vi : Visualization

Su : Supervision

P : Project administration

Fu : Funding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

Informed consent was obtained from the author, Satriadi Putra Santika, for the use of his images in this study.

ETHICAL APPROVAL

This research involved only the authors as participants, who provided informed consent for all procedures. According to the policy of Bina Nusantara University, ethical approval was not required for this type of self-experimentation research.

DATA AVAILABILITY

The dataset used in this study was obtained from the public repository provided by Afkaar [21]. It is available at the following Kaggle link: <https://www.kaggle.com/datasets/mlanangafkaar/datasets-lemlitbang-sibi-alphabets> (accessed on 25 October 2024).

REFERENCES




- [1] N. M. Sari and S. L. Kholia, "The function of communication in human life," *Journal of Social, Media, Communication, and Journalism*, vol. 2, no. 1, pp. 33–41, 2024, doi: 10.4324/9780429482427-3.
- [2] World Health Organization, *World Report On Hearing*. 2021. [Online]. Available: <https://www.who.int/publications/i/item/world-report-on-hearing>
- [3] T. Handhika, R. I. M. Zen, Murni, D. P. Lestari, and I. Sari, "Gesture recognition for Indonesian sign language (BISINDO)," *Journal of Physics: Conference Series*, vol. 1028, no. 1, 2018, doi: 10.1088/1742-6596/1028/1/012173.
- [4] N. Palfreyman, "Social meanings of linguistic variation in BISINDO (Indonesian Sign Language)," *Asia-Pacific Language Variation*, vol. 6, no. 1, pp. 89–118, 2020.
- [5] S. Amaliya, A. N. Handayani, M. I. Akbar, H. W. Herwanto, O. Fukuda, and W. C. Kurniawan, "Study on hand keypoint framework for sign language recognition," in *7th International Conference on Electrical, Electronics and Information Engineering: Technological Breakthrough for Greater New Life, ICEEIE 2021*, 2021, pp. 446–451, doi: 10.1109/ICEEIE52663.2021.9616851.

- [6] Suharjito, N. Thiracitta, and H. Gunawan, "SIBI sign language recognition using convolutional neural network combined with transfer learning and non-trainable parameters," in *5th International Conference on Computer Science and Computational Intelligence 2020*, 2021, vol. 179, pp. 72–80, doi: 10.1016/j.procs.2020.12.011.
- [7] J. Singh, H. Singh, and V. Goyal, "A comprehensive study on feature extraction and classification techniques for sign language recognition," *Proceedings of the 5th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2021*, pp. 959–964, 2021, doi: 10.1109/ICECA52323.2021.9676024.
- [8] A. Z. Nugraha, R. F. Salsabila, A. N. Handayani, and A. P. Wibawa, "Decision tree based algorithms for Indonesian Language Sign System (SIBI) recognition," *Applied Engineering and Technology*, vol. 3, no. 2, pp. 86–101, 2024.
- [9] C. N. Ansani, I. Nurtanio, and A. A. Ilham, "The effect of light on leap motion controller in the classification of sign language translator system," in *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2019, pp. 296–300, doi: 10.1109/ISRITI48646.2019.9034602.
- [10] D. Veeraiah, S. J. Basha, K. S. Deepthi, T. Sathvik, and P. Ganesh, "Enhancing communication for deaf and dumb individuals through sign language detection: a comprehensive dataset and SVM-based model approach," *Proceedings of the 3rd International Conference on Applied Artificial Intelligence and Computing, ICAAIC 2024*, pp. 947–954, 2024, doi: 10.1109/ICAAIC60222.2024.10575397.
- [11] A. N. Handayani, M. I. Akbar, H. Ar-Rosyid, M. Ilham, R. A. Asmara, and O. Fukuda, "Design of SIBI sign language recognition using artificial neural network backpropagation," in *2022 2nd International Conference on Intelligent Cybernetics Technology and Applications, ICICyTA 2022*, 2022, pp. 192–197, doi: 10.1109/ICICyTA57421.2022.10038205.
- [12] M. C. Bagaskoro, F. Prasajo, A. N. Handayani, E. Hitipeuw, A. P. Wibawa, and Y. W. Liang, "Hand image reading approach method to Indonesian Language Signing System (SIBI) using neural network and multi layer perceptron," *Science in Information Technology Letters*, vol. 4, no. 2, pp. 97–108, 2023, doi: 10.31763/sitech.v4i2.1362.
- [13] A. N. Sihananto, E. M. Safitri, Y. Maulana, F. Fakhruddin, and M. E. Yudistira, "Indonesian sign language image detection using convolutional neural network (CNN) method," *Inspiration: Jurnal Teknologi Informasi dan Komunikasi*, vol. 13, no. 1, pp. 13–21, 2023, doi: 10.35585/inspir.v13i1.37.
- [14] M. A. Limantara and D. Trisianto, "SIBI alphabet detection system based on convolutional neural network (CNN) method as learning media," *Internet of Things and Artificial Intelligence Journal*, vol. 4, no. 1, pp. 143–161, 2024, doi: 10.31763/iota.v4i1.716.
- [15] M. L. Afkaar, "CNN with Mediapipe for sign language recognition." *Kaggle*, 2020. [Online]. Available: <https://www.kaggle.com/code/mlanangafkaar/cnn-with-mediapipe-for-sign-language-recognition/notebook>
- [16] A. N. Handayani, T. Andriyanto, D. F. Azizah, M. Z. Wiryan, and H. A. Rosyid, "Comparison of ResNet-50 and EfficientNet-B0 method for classification of Indonesian Sign Language System (SIBI) using multi background dataset," in *2024 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)*, 2024, pp. 1–6, doi: 10.1109/CENIM64038.2024.10882836.
- [17] Y. K. Suyitno, I. W. Sudiarsa, I. N. B. Hartawan, and I. D. P. G. W. Putra, "Implementation of SIBI sign language realtime detection program (case studi at sekolah luar biasa negeri 1 tabanan)," *Journal of Computer Networks, Architecture and High Performance Computing*, vol. 6, no. 3, pp. 1431–1441, 2024, doi: 10.47709/cnahpc.v6i3.4405.
- [18] D. B. Adewole, A. Adesugba, O. Agbelusi, and O. V. Olatunde, "Sign language recognition using deep learning : advancements and challenges," *International Journal of Latest Technology in Engineering, Management & Applied Science (IJLTEMAS)*, vol. 13, no. 12, pp. 318–324, 2024, doi: 10.51583/IJLTEMAS.
- [19] M. F. Wahid, R. Tafreshi, and R. Langari, "A multi-window majority voting strategy to improve hand gesture recognition accuracies using electromyography signal," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 2, pp. 427–436, 2020, doi: 10.1109/TNSRE.2019.2961706.
- [20] H. A. Audah, A. Yuliawati, and I. Alfina, "A comparison between SymSpell and a combination of Damerau-Levenshtein distance with the trie data structure," in *2023 10th International Conference on Advanced Informatics: Concept, Theory and Application, ICAICTA 2023*, 2023, pp. 1–6, doi: 10.1109/ICAICTA59291.2023.10390399.
- [21] M. L. Afkaar, "Datasets SIBI Sign Language Alphabets," 2020. [Online]. Available: <https://www.kaggle.com/datasets/mlanangafkaar/datasets-lemilitbang-sibi-alphabets>
- [22] J. Zhang, "Classification and Comparison of data augmentation techniques," in *Transactions on Computer Science and Intelligent Systems Research*, 2024, vol. 6, pp. 180–187.
- [23] Indriani, M. Harris, and A. S. Agoes, "Applying Hand gesture recognition for user guide application using MediaPipe," in *Proceedings of the 2nd International Seminar of Science and Applied Technology (ISSAT 2021)*, 2021, vol. 207, pp. 101–108, doi: 10.2991/aer.k.211106.017.
- [24] C. L. Fan, "Advancements in image recognition for urban land use: multi-scale CNN extraction," in *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 2023, pp. 261–267, doi: 10.1109/EECSI59885.2023.10295839.
- [25] P. T. Nguyen, V. Q. Huynh, T. N. Phan, and T. Van Huynh, "The fusion of feature extraction applications and blurring techniques for classifying irish sign language," in *IFMBE Proceedings*, 2024, vol. 95, pp. 404–417, doi: 10.1007/978-3-031-44630-6_33.
- [26] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.
- [27] L. T. Woods and Z. A. Rana, "Constraints on Optimising encoder-only transformers for modelling sign language with human pose estimation keypoint data," *Journal of Imaging*, vol. 9, no. 11, pp. 1–28, 2023, doi: 10.3390/jimaging9110238.
- [28] G. Simon and C. Aliferis, *Artificial Intelligence and Machine Learning in Health Care and Medical Sciences*. Springer, 2024.
- [29] R. Gupta and A. Singh, "Hand gesture recognition using OpenCV," in *2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2023, pp. 145–148.
- [30] C. D. Patil, A. Sonare, A. Husain, A. Jha, and A. Phirke, "Controlled hand gestures using Python and OpenCV," *International Journal for Research in Applied Science and Engineering Technology*, vol. 11, no. 5, pp. 2973–2977, 2023, doi: 10.22214/ijraset.2023.52285.
- [31] X. Kong, H. Wu, and H. Hu, "Deep learning-based test data augmentation technology," in *Proceedings - 2023 16th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, CISP-BMEI 2023*, 2023, pp. 1–6, doi: 10.1109/CISP-BMEI60920.2023.10373379.
- [32] D. Guo, W. Zhou, M. Wang, and H. Li, "Sign language recognition based on adaptive HMMS with data augmentation," in *International Conference on Image Processing, ICIP*, 2016, pp. 2876–2880, doi: 10.1109/ICIP.2016.7532885.
- [33] A. Althian *et al.*, "Impact of dataset size on classification performance: an empirical evaluation in the medical domain," *Applied Sciences*, vol. 11, no. 2, pp. 1–18, 2021, doi: 10.3390/app11020796.
- [34] C. Erden, H. I. Demir, and A. H. K ok am, "Enhancing machine learning model performance with hyper parameter optimization: a comparative study," *arXiv preprint arXiv:2302.11406*, 2023.




- [35] M. O. Khairandish, M. Sharma, V. Jain, J. M. Chatterjee, and N. Z. Jhanji, "A hybrid CNN-SVM threshold segmentation approach for tumor detection and classification of MRI brain images," *IRBM*, vol. 43, no. 4, pp. 290–299, 2022, doi: 10.1016/j.irbm.2021.06.003.
- [36] N. S. V. Rao, "Study of overfitting by machine learning methods using generalization equations," *2023 26th International Conference on Information Fusion*, no. 1, 2023, doi: 10.23919/FUSION52260.2023.10224198.
- [37] N. Padfield, J. Ren, C. Qing, P. Murray, H. Zhao, and J. Zheng, "Multi-segment majority voting decision fusion for MI EEG brain-computer interfacing," *Cognitive Computation*, vol. 13, no. 6, pp. 1484–1495, 2021, doi: 10.1007/s12559-021-09953-3.
- [38] M. Rivera-Acosta, J. M. Ruiz-Varela, S. Ortega-Cisneros, J. Rivera, R. Parra-Michel, and P. Mejia-Alvarez, "Spelling correction real-time american sign language alphabet translation system based on yolo network and LSTM," *Electronics*, vol. 10, no. 9, 2021, doi: 10.3390/electronics10091035.

BIOGRAPHIES OF AUTHORS






Satriadi Putra Santika, S.Stat    is a Master student of computer science at Bina Nusantara University. He completed his bachelor's degree in statistics at Brawijaya University, graduating Summa Cum Laude. His research interests cover a wide range of areas, including statistical modeling, machine learning, deep learning, and time series analysis. He focuses on applying advanced statistical methods and algorithms to solve complex problems in data analysis and prediction. He is passionate about applying advanced algorithms to solve real-world problems, especially in fields like technology. Always eager to learn, he actively participates in research and collaborative projects to enhance his skills and make an impact in the data science field. He can be contacted at email: satriadi.santika@binus.ac.id.






Stefanus Benhard, S.Kom. PMEC, CLSSWB    is a passionate and innovative professional with a strong capability in computer science applications and multidisciplinary studies. He graduated with honors from Petra Christian University, where he earned his bachelor's degree, and then began his career by contributing to digital transformation in several industries, including Astra International - Daihatsu and Bernofarm Pharmaceutical Company, where he played a key role in triggering their research in digitalization area and efforts. As of today, He currently pursuing Master's to Doctoral's degree in computer science at BINUS University, by focuses on advanced research in areas such as affective computation, cognitive computation, IoT, and immersive technology. He is committed to addressing multidisciplinary challenges, particularly in psychology and its intersection with technology, and been actively seeking research partners to collaborate on advancing social science research and helping humanity live their best lives. He can be contacted at email: stefanus.benhard@binus.ac.id.



Dr. Ir. Yulyani Arifin, S.Kom, MM.    currently serves as Deputy Head of Doctor of computer science, BINUS University. He completed his Doctoral Education in 2021 at the Doctoral of computer science BINUS University, as well as pursuing a Master of management specialization in information systems in 2005 and a Bachelor of management informatics in 1998. In 2022, he completed his professional engineer degree. Before focusing on academics, he had 11 years of professional experience in the ERP field. He is also active in the ACM-SIGCHI Chapter Indonesia organization and a member of the ISACA international organization Since 1998, he has been actively teaching until now. His research interests include human computer interaction, natural language processing, text processing, multimedia and game virtual reality, augmented reality and mixed reality, as well as IT risk management and software engineering. She can be contacted at email: yulyaniarifin@binus.ac.id.



Ir. Andry Chowanda, S.Kom., MM, Ph.D., MBCS, CCP, CME, IPM, SMIEEE    earned his Bachelor's degree in computer science from Bina Nusantara University (Indonesia, 2009), a Master's in business management from BINUS Business School (Indonesia, 2011), and a Ph.D. in computer science from Nottingham University (England, 2017). He is now a Computer Science Lecturer at Bina Nusantara University. His research is in agent architecture and machine (and deep) learning. His work is mainly on how to model an agent that can sense and perceive the environment based on the perceived data and build a social relationship with the user over time. In addition, he is also interested in serious game and gamification design. He can be contacted at email: achowanda@binus.edu.