Accredited SINTA 2 Ranking
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online at: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 9 No. 2 (2025) 258 - 265 e-ISSN: 2580-0760

Hand Sign Recognition of Indonesian Sign Language System (SIBI) Using Inception V3 Image Embedding and Random Forest

Mayang Sari[1], Eko Rudiawan Jamzuri[2]*
[1,2]Department of Electrical Engineering, Politeknik Negeri Batam, Batam, Indonesia
[1]mayangsari06qq@gmail.com, [2]ekorudiawan@polibatam.ac.id

Abstract

This paper presents a sign language recognition system for the Indonesian Sign Language System (SIBI) using image embeddings combined with a Random Forest classifier. A dataset comprising 5280 images across 24 classes of SIBI alphabet symbols was utilized. Image features were extracted using the Inception V3 image embedding, and classification was performed using Random Forest. Model evaluation conducted through K-Fold cross-validation demonstrated that the proposed method achieved an accuracy of 85.40%, an F1 score of 85.20%, a precision of 85.30%, and a recall of 85.40%. Moreover, the total computation time required by the proposed method is 1152.85 seconds. While the performance indicates room for improvement, this study lays the groundwork for enhancing sign language recognition systems to support the preservation and broader adoption of SIBI in Indonesia.

Keywords: *hand sign recognition; SIBI; Inception V3; image embedding; random forest*

How to Cite: M. Sari and E. R. Jamzuri, “Hand Sign Recognition of Indonesian Sign Language System SIBI Using Inception V3 Image Embedding and Random Forest”, *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 9, no. 2, pp. 258 - 265, Mar. 2025.
DOI: https://doi.org/10.29207/resti.v9i2.6156

1. Introduction

The communication gap is a significant challenge for people with disabilities, particularly those who are deaf. Due to difficulties communicating, the deaf often has to utilize alternative methods to convey their messages. Sign language is one effective way to communicate, where symbols are visualized in the form of hand gestures to express the intent or message to be conveyed. Sign language is used by deaf individuals to communicate, with the hands as the primary tool in conveying information [1].

The sign languages used by the deaf community in Indonesia are Sistem Isyarat Bahasa Indonesia (SIBI) and Bahasa Isyarat Indonesia (BISINDO) [2]. SIBI is a sign language widely used in Indonesia, where each letter A to Z is visualized through specific hand gestures. These patterns can be performed using one hand by showing specific finger patterns [3]. In comparison, some letters in BISINDO require two hands to be demonstrated [4].

Sign language recognition is a topic that has attracted much attention from researchers, with the primary goal of translating hand gestures into text to facilitate communication. For example, research by [5] used OpenCV and MediaPipe libraries to recognize four hand patterns, where hand key points were identified using a Support Vector Machine (SVM). This research resulted in an F1 score, recall, and precision of 98.75% each. Another study by [6] focused on BISINDO recognition, which proposed a combination of YOLO as a hand detector and CNN as a hand pattern classifier. The method proposed by [6] resulted in an accuracy of 89%. Meanwhile, regarding SIBI recognition, [7] proposed the use of transfer learning with VGG16 and MobileNet architecture. The evaluation results show that MobileNet achieves the highest accuracy of 98%, although it requires a longer computation time than VGG16. A comparison of performance between machine learning and deep learning methods has also been studied in [8]. As a result, the deep learning method with Xception managed to achieve the highest F1 score, which is 99.57%, with a computation time of 1387 seconds.

Although some previous methods have recognized hand sign language successfully, there is still an opportunity to explore other method's performance. This research proposes image embedding combined with Random

Received: 31-10-2024 | Accepted: 14-03-2025 | Published Online: 21-03-2025

Forest to predict hand patterns in SIBI. In contrast to previous approaches that rely on training the CNN from scratch, the image embedding utilizes a pre-trained CNN model to result in feature vectors without retraining. The Random Forest is then used to classify these feature vectors into the appropriate classes. With this approach, it is expected that hand gesture recognition can be performed more efficiently in terms of computation time.

This paper is organized as follows: Section 2 discusses the materials and methods used in the research, including the image embedding technique and the Random Forest algorithm. Section 3 presents the evaluation results of the applied method, which provides a comprehensive performance analysis of the resulting model. The paper then concludes with conclusions and suggestions for further development in Section 4, which highlights the potential improvement in future research.

2. Research Methods

This section describes the dataset used in the research as well as the proposed method for predicting hand signs in SIBI. The research process starts with the collection of the dataset and is followed by the feature extraction. Once the features are extracted, the data is classified using the Random Forest algorithm. The final stage of the research involves evaluating the performance of the model with several classification metrics to assess its accuracy in recognizing SIBI.

2.1 Dataset Preparation

The data used in this research comes from a public dataset available on Kaggle: https://www.kaggle.com/datasets/alvinbintang/sibi-dataset. The dataset consists of 5280 images of 150×150 pixels, depicting SIBI hand sign patterns for the letters of the alphabet. However, the letters J and Z are not included in the dataset, as they cannot be represented as static images. Instead, they require the detection of multiple image frames in sequence. Thus, the dataset includes 24 classes, with each class consisting of 220 sample images. Figure 1 is an example of a hand pattern image in the dataset used in this study.

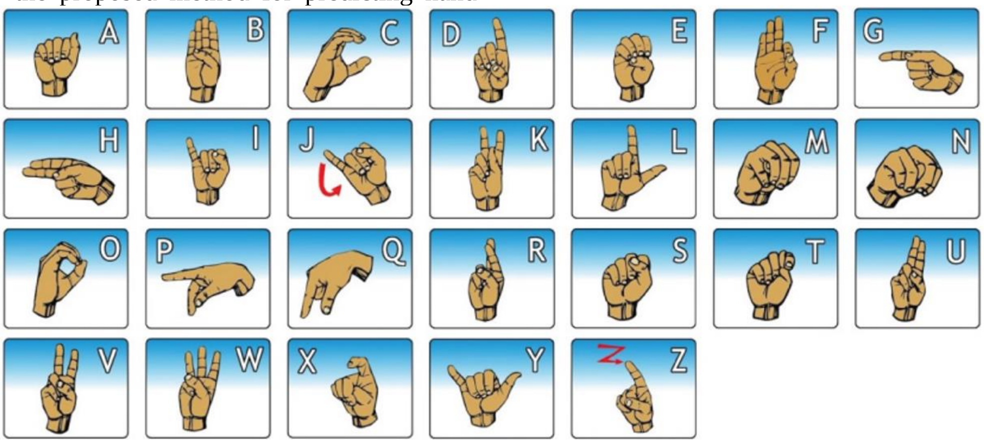


Figure 1. Sample illustration of SIBI hand gestures.

2.2 Feature Extraction with Inception V3 Image Embedding

Feature extraction is the process of transforming extensive image data into a smaller representation that is more informative and analyzable [9]. There are various techniques used for extracting features from hand patterns, for example, HOG-9ULBP [10], hand shape features [11], and finger features [12]. The feature extraction method used in this research is image embedding with Convolutional Neural Network (CNN) model. Image embedding is a method that converts images into vectors that represent essential features of the image [13]. Machine learning algorithms then use this vector to group, classify, or search for images based on content similarity. In this research, the feature extraction process is performed using the Inception V3 CNN architecture [14], a deep learning model designed to extract local and global features efficiently.

Inception V3 is a CNN architecture that is pre-trained using the ImageNet dataset and designed to work with color images. Figure 2 illustrates the architecture of Inception V3. The Inception module includes three different convolution sizes as well as one maximum pooling layer. The results of these convolution layers are combined and processed non-linearly, allowing the network to capture features at multiple scales and reducing the risk of overfitting. In traditional architectures, only one filter size is used in the convolution layer. However, with Inception V3, the network can use multiple filter sizes, such as 1×1, 3×3, and 5×5, by pooling in parallel. This model increases the network's ability to capture more information without significantly increasing computational complexity [15].

In the feature extraction stage, the initial image consisting of 150×150 pixels is resized to a fixed size of 299×299 pixels. After that, the image is fed into the Inception V3 model. After going through a series of convolution, pooling, and Inception module layers, the image dimensions are reduced to 8×8, while the number of channels increases to 2048. After passing through the average pooling layer, the final output is a 1×1×2048 feature representation. This feature is then used as input to the Random Forest algorithm as input data.

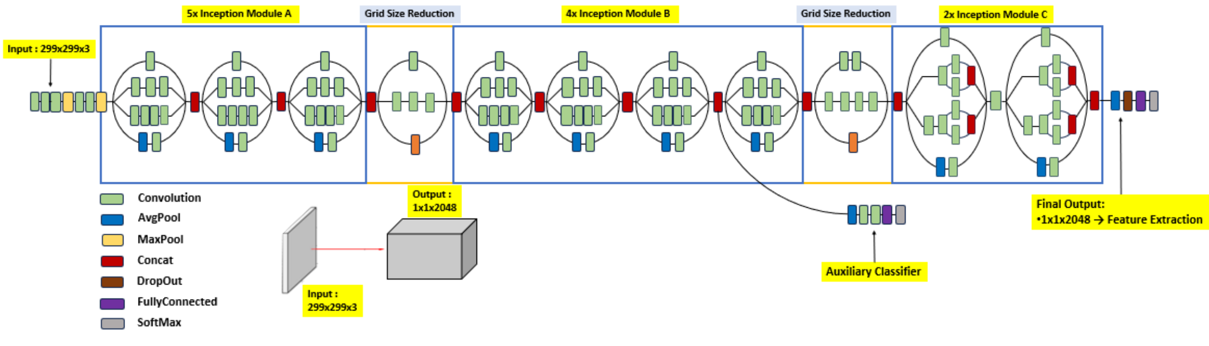


Figure 1. Inception v3 network architecture for image feature extraction

2.3 Classification with Random Forest

Random Forest is a classification technique consisting of a number of decision trees, where each tree is constructed using random samples drawn from an independent and identical, uniformly distributed distribution [16]. This technique helps overcome the risk of overfitting and reduces the correlation between decision trees, which often occurs in other ensemble methods. According to [17], random forest is a highly efficient classification method that can be widely applied to various types of datasets.

Figure 3 illustrates the principle of the Random Forest algorithm. The extracted features from the images are input into multiple decision trees. Random Forest is one type of ensemble learning approach that constructs numerous decision trees during training; each tree provides a prediction based on a random subset of the data and features.

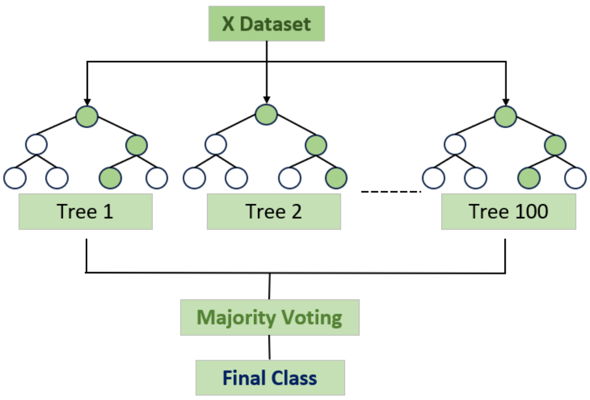


Figure 2. Random forest model structure with 100 trees for classifying SIBI hand gestures.

The final output is determined by aggregating the predictions from all the decision trees, typically using majority voting for classification tasks [18]. In building these decision trees, Random Forest utilizes the Gini Index, derived from the CART (Classification and Regression Trees) algorithm, to measure node impurity. The Gini Index is a commonly used metric in classification problems. It quantifies the probability of incorrectly classifying a randomly chosen element if it were labeled according to the distribution of labels in the node [19]. It helps the algorithm determine the optimal splits at each node to improve predictive accuracy.

Equation 1 provides the formula for calculating the Gini impurity, denoted as $Gini(T)$, where T represents the dataset or node being evaluated. Here, n is the total number of classes, and P_i is the probability of class-i, calculated by dividing the number of instances in the class i by the total instances in T. This impurity measure quantifies the likelihood of incorrectly classifying a randomly chosen instance according to the node's class distribution [20]. Gini impurity values range from 0, indicating a pure node (all instances in one class), to a maximum that depends on the class count. In binary classification, values range from 0 to 0.5, where 0.5 represents a perfectly balanced distribution between two classes.

$$Gini(T) = 1 - \Sigma_{i=1}^{n} (P_i)^2 \quad (1)$$

Equation 2 provides the formula for calculating the Gini split value, denoted as $Gini_{split}(T)$, which is used to evaluate the impurity of a dataset after a split in a decision tree. In this formula, T_1 and T_2 are the subsets resulting from the split of the dataset T, where N_1 and N_2 are the number of instances in each subset, respectively. The terms $Gini(T_1)$ and $Gini(T_2)$ represent the Gini impurity of each subset, measuring the impurity within them. Where $N = N_1 + N_2$ is the total number of instances in the dataset T.

$$Gini_{split}(T) = \frac{N_1}{N} \times Gini\ (T_1) + \frac{N_2}{N} \times Gini(T_2) \quad (2)$$

In the classification process using Random Forest, the 2048-dimensional feature vectors extracted from images are used as input. These features are classified using a Random Forest classifier composed of 10 decision trees. The model combines the predictions from these 100 trees to improve the accuracy of the final prediction through majority voting. Each tree is trained on a bootstrap sample of the dataset, introducing randomness by selecting a different subset of data for each tree. Additionally, at each node in a tree, a random subset of features is considered when determining the best split, which helps in reducing the correlation between trees and enhances generalization. In this model, the minimum number of samples required to split an internal node is set to 5. It means that a node will not be split further if it contains fewer than 5 samples, ensuring that only sizable subsets of data are used for further splitting. This parameter helps improve performance and reduces the risk of overfitting by preventing the creation of nodes that are too small to provide meaningful splits.

2.4 Evaluation Method

The evaluation of the classification aims to assess the classifier's performance in accurately classifying SIBI hand patterns. We evaluated a dataset comprising 5280 images representing 24 classes of sign language alphabet symbols. Utilizing the K-Fold Cross-Validation method [21], we divided the dataset into K subsets (folds) of approximately equal size. The model is trained on $K-1$ folds and validated on the remaining fold. This process is repeated K times, with each fold serving as the validation set once. The final evaluation metrics are calculated by averaging the results from all K iterations. This technique reduces the bias associated with a single train-test split and provides a more robust estimate of the model's performance.

The classification results of the Random Forest are evaluated using a confusion matrix that visualizes the classifier performance by comparing the actual classes with the predicted classes for a set of test data. In the context of multi-class classification with 24 classes of SIBI hand gestures, the confusion matrix is a 24×24 grid. Each row of the matrix represents the instances in an actual class, while each column represents the instances in a predicted class. The diagonal elements indicate the number of correct predictions for each class, whereas the off-diagonal elements show where the model has misclassified instances, indicating confusion between specific classes [22]. For each class in the classification task, we define four key metrics to evaluate the model's performance: True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). The TP count represents the number of instances correctly predicted as belonging to a particular class; these instances are located along the diagonal of the confusion matrix. The FP metric refers to instances that are incorrectly predicted as belonging to the class but actually belong to other classes; this is calculated by summing the values in the corresponding column of the confusion matrix, excluding the TP. Conversely, the FN count includes instances that actually belong to the class but are incorrectly predicted as belonging to other classes, determined by summing the values in the corresponding row of the confusion matrix, excluding the TP. Lastly, the TN represents the number of instances correctly predicted as not belonging to the class, calculated by subtracting the sum of TP, FP, and FN for that class from the total number of instances. By analyzing this matrix, we can assess how well the model recognizes each SIBI hand gesture and identify patterns of misclassification. This detailed evaluation helps in understanding the model's strengths and weaknesses, guiding further improvements to enhance its accuracy in recognizing SIBI hand gestures.

In addition to using a confusion matrix, the performance of the classification model is evaluated using standard evaluation metrics for classification problems such as accuracy, precision, recall, and F1 score. Equation 3 describes the formula to calculate accuracy, which provides an overall indication of the model's performance. Accuracy measures the proportion of correct predictions made by the classifier out of all predictions [23]. However, accuracy alone may not be sufficient in multi-class classification tasks, where misclassifications between classes can vary.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \times 100\% \quad (3)$$

Equation 4 provides the formula for calculating precision, which measures the accuracy of the model's positive predictions. Precision is defined as the TP ratio to the total number of positive predictions made by the classifier, which includes both TP and FP. This metric indicates the proportion of instances that were correctly predicted as positive out of all instances predicted as positive by the model [24]. Precision measures how many of the instances that the model classified as positive are actually positive, reflecting the model's ability to avoid false positives.

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (4)$$

Equation 5 provides the formula for calculating recall, which indicates how effective the model is at identifying actual positive instances. The recall value is defined as the TP ratio to the total number of actual positive instances, which includes both TP and FN [25]. This metric measures how many of the actual positive instances were correctly identified by the classifier. A higher recall indicates that the classifier is effective at detecting positive instances and has fewer false negatives, reflecting its ability to capture as many relevant instances as possible.

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (5)$$

Equation 6 provides the formula for calculating the F1 score, an evaluation metric that combines precision and recall into a single value to measure the performance of a classification model, mainly when dealing with imbalanced datasets. The F1 score is calculated as the harmonic mean of precision and recall. By using the harmonic mean, the F1 score balances the trade-off between precision and recall, providing a more comprehensive measure of the model's performance [26]. This metric is especially valuable when the dataset has an uneven class distribution or when both FP and FN are essential to consider.

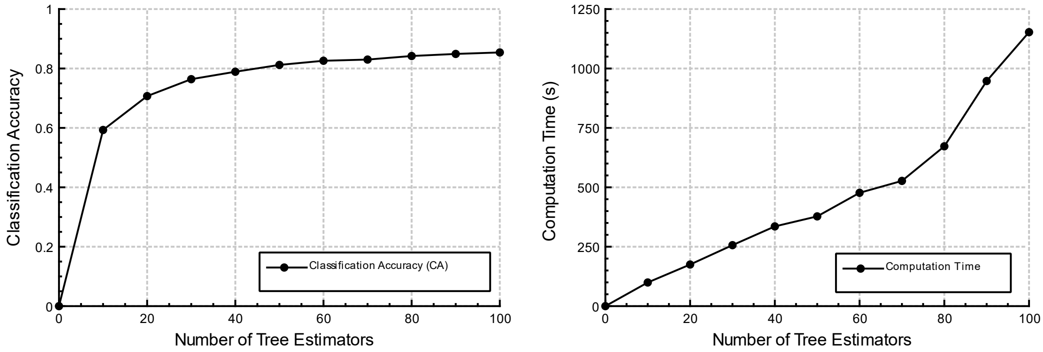
$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (6)$$

3. Results and Discussions

This section presents the performance evaluation of the classification model applied to the SIBI dataset. The analysis includes the model's ability to recognize each hand gesture, including gestures with the highest and lowest actual recognition rates based on the confusion matrix obtained using the Random Forest method. Furthermore, the test results are processed to obtain accuracy, F1 score, precision, and recall values.

3.1 SIBI Gesture Classification Results

In the first experiment, we evaluate the proposed method by looking at the performance of the classification accuracy (CA). In this experiment, we vary the number of tree estimators in the random forest to observe its correlation with CA. In addition, we also wanted to identify the relationship between the number of tree estimators and the computation time required for the proposed algorithm. The results of this experiment are shown in Figure .



(a) (b)

Figure 4. Relationship between a number of tree estimators with (a) classification accuracy, and (b) computation time.

Figure 4 (a) shows the correlation between the number of tree estimators and the CA. It can be seen that the more tree estimators in the random forest, the higher the CA. In this experiment, the random forest with a number of tree estimators above 50 achieved a CA above 80%. The highest value is obtained when using 100 tree estimators, which is CA = 85.40%.

Figure 4 (b) shows the effect of the number of tree estimators on the computation time, which includes the total training time and the total inference time of the random forest. From Figure 4 (b), it can be concluded that the more tree estimators there are, the longer the computation time is required. The most extended computation process occurs when the random forest uses 100 tree estimators, with a total computation time of 1152.85 seconds.

After testing and obtaining the best-performing algorithm, which is the random forest with 100 tree estimators, we then compared the results with the research conducted by [8]. The comparison results are shown in Table 1.

Based on the evaluation metrics of CA, precision, recall, and F1 score, our proposed method is still unable to surpass the transfer learning method Xception proposed by [8]. However, our research is superior in terms of computation time speed, which is 234.36 seconds faster than the method of [8], despite the lower computer specifications we use, which is a computer with AMD Athlon Gold 3150U 4CPUs@2.4GHz. This advantage is due to the absence of the training process in the CNN architecture part, so the computation process during training only occurs in the random forest.

Table 1. Comparison of the proposed method with state-of-the-art on SIBI hand gesture classification.

Method	CA (%)	Prec (%)	Recall (%)	F1 (%)	Computation Time (s)
[8]	99.57	99.57	99.57	99.57	1387.21
Proposed	85.40	85.30	85.40	85.20	1152.85

Table 2. Performance of random forest on SIBI hand gesture classification

Gestures	CA (%)	Prec (%)	Recall (%)	F1 (%)
A	98.60	80.80	85.90	83.30
B	99.00	86.10	90.50	88.20
C	99.60	92.30	97.70	94.90
D	99.60	93.10	98.60	95.80
E	98.80	82.90	88.20	85.50
F	98.50	83.20	80.90	82.00
G	99.20	93.70	87.70	90.60
H	99.70	93.20	99.10	96.00
I	97.70	79.60	58.60	67.50
K	98.20	76.90	81.80	79.30
L	99.00	90.80	85.50	88.10
M	98.10	78.60	75.00	76.70
N	98.00	75.70	76.40	76.00
O	99.80	97.70	98.60	98.20
P	99.40	94.80	91.40	93.10
Q	99.70	94.80	99.10	96.90
R	98.50	79.30	85.50	82.30
S	98.70	84.60	85.00	84.80
T	98.80	86.70	82.70	84.70
U	98.40	83.40	75.50	79.20
V	98.00	79.10	72.30	75.50
W	98.40	77.80	87.70	82.50
X	99.00	87.00	88.20	87.60
Y	98.60	83.00	84.10	83.50

Table 2 presents the evaluation results of the Random Forest model in classifying 24 hand gestures represented by SIBI. In general, the Random Forest algorithm used for classification showed good accuracy performance, with CA values above 97%. The gesture "O" has the best performance with a CA of 99.80%, an F1 score of 98.20%, a precision of 97.70%, and a recall of 98.60%. Moreover, based on the F1 score, the gesture "O" has the highest performance, which shows that the model is able to recognize this gesture very well. However, some gestures, such as "V" and "I," have much lower F1 scores, 75.50%, and 67.50%, respectively. This result shows that the model has difficulty recognizing these gestures. This condition is likely due to the similarity of the patterns or hand gestures that symbolize these gestures.

In terms of precision and recall, low precision values occur for the gestures "N" and "V," which only reach 75.70% and 72.30%, respectively. This lowest precision in gesture "N" indicates that many predictions for these gestures are false positives. In addition, the lowest recall value occurred for the gesture "V," which indicates that the model predicts false negatives.

3.1 Confusion Matrix Results for SIBI Gesture Classification

Figure 5 is a confusion matrix that describes the model prediction results against the actual class. The left row represents the actual class, while the bottom column shows the predicted class. Each cell shows the amount of data classified, with the color helping to identify the concentration of predictions. The diagonal values from the top left to bottom right are the number of correct predictions in which the actual class is equal to the predicted class. On the other hand, off-diagonal values indicate misclassification. Symbol Σ in the bottom row and right column summarizes the total sum of the sample data.

Based on the confusion matrix results, the gesture "Q" achieved one of the highest recognition rates by the model. The classifier correctly predicted 217 out of the total samples for the gesture "Q." However, there were seven instances where the gesture "Q" was misclassified as another gesture, indicating minor weaknesses in recognizing its specific characteristics. Additionally, three samples of other gestures were incorrectly predicted as "Q," demonstrating that the model was relatively confident in identifying this gesture correctly with minimal false positives.

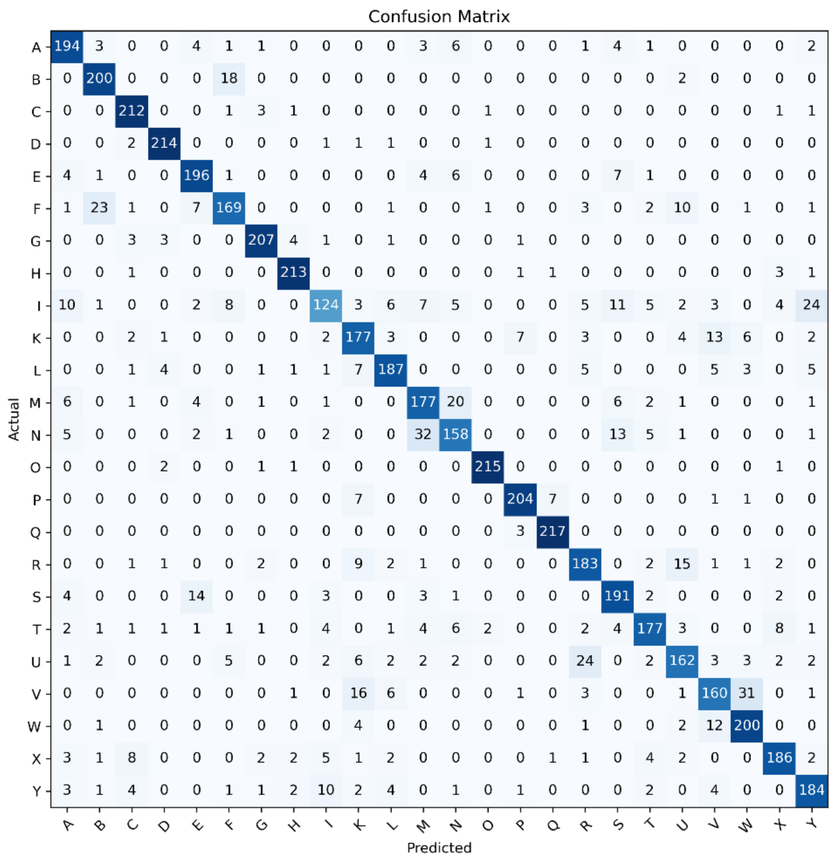


Figure 5. Confusion matrix of SIBI hand gesture classification results.

The gesture "I" exhibited a moderate identification rate in the classification model. Out of a total of 124 actual samples of the gesture "I," the model correctly predicted 124 instances, suggesting a strong recognition rate for this particular gesture. However, 11 instances of other gestures were misclassified as "I," leading to some false positives. Notably, the gesture "Y" was frequently misidentified as "I," indicating that the model struggled to distinguish between these similar visual gestures.

Meanwhile, the gesture "F" showed significant classification difficulty. Although the model correctly identified 169 instances of "F," there were 23 instances where the model misclassified "F" as "B," indicating a challenge in differentiating between these two gestures. Similarly, the gesture "V" was frequently confused with "U," with 31 instances of "V" being misclassified as "U." This suggests that the model might require further refinement in feature extraction or training adjustments to better distinguish between gestures with similar shapes or movements.

These findings highlight areas where the model performs well and areas where further improvements could be made, such as refining feature selection, improving training data balance, or incorporating additional preprocessing techniques to enhance gesture differentiation.

4. Conclusions

Based on the conducted research, the Random Forest classification model achieved a classification accuracy of 85.40%, a precision of 85.30%, a recall of 85.40%, and an F1 score of 85.20% in recognizing SIBI hand sign symbols. These results indicate that the model has significant room for improvement, particularly in accurately classifying certain hand gestures. Specifically, the gesture "I" had the lowest classification accuracy, with only 124 out of 220 predictions being correct. In terms of computation efficiency, our proposed method is 234.36 seconds faster than state-of-the-art ones. This result shows that our proposed method has some advantages in computation efficiency. It is recommended that other classification algorithms be explored to enhance the model's performance. In addition, it is necessary to explore other image-embedding models that can extract hand gestures more efficiently.

References

[1] H. Halim and L. Lina, "Aplikasi Pengidentifikasi Bahasa Isyarat Berdasarkan Gerak Tubuh Secara Real Time Menggunakan YOLO," *Simtek : jurnal sistem informasi dan teknik komputer*, vol. 8, no. 2, Art. no. 2, Oct. 2023, doi: 10.51876/simtek.v8i2.215.

[2] I. Solihin, "Kesetaraan Sosial Dalam Bahasa Isyarat Sebagai Identitas Tuli Di Komunitas Rumah Setara Palembang," *Eastasouth Journal of Effective Community Services*, vol. 1, no. 03, Art. no. 03, Apr. 2023, doi: 10.58812/ejecs.v1i03.49.

[3] E. Rakun, S. Muzahidin, I. S. A. Darmana, and W. Setiaji, "SIBI (Sign System Indonesian Language) Text-to-3D Animation Translation Mobile Application," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13, no. 9, Art. no. 9, Dec. 2022, doi: 10.14569/IJACSA.2022.0130950.

[4] N. Palfreyman, "Social meanings of linguistic variation in BISINDO (Indonesian Sign Language)," *Asia-Pacific Language Variation*, vol. 6, no. 1, pp. 89–118, Jul. 2020, doi: 10.1075/aplv.00008.pal.

[5] A. R. Ardiansyah, A. H. Nur'azizan, and R. Fernandis, "Implementasi Deteksi Bahasa Isyarat Tangan Menggunakan OpenCV dan MediaPipe," *Seminar Nasional Teknologi & Sains*, vol. 3, no. 1, pp. 331–337, Jan. 2024.

[6] I. I. Arifah, F. N. Fajri, and G. Q. O. Pratamasunu, "Deteksi Tangan Otomatis Pada Video Percakapan Bahasa Isyarat Indonesia Menggunakan Metode YOLO Dan CNN," *Journal of Applied Informatics and Computing*, vol. 6, no. 2, Art. no. 2, Nov. 2022, doi: 10.30871/jaic.v6i2.4694.

[7] M. B. Subkhi, M. Y. Trinurais, R. K. A. Wibowo, and B. R. Prakosa, "Deteksi Bahasa Isyarat Berdasarkan SIBI (Sistem Bahasa Isyarat) menggunakan Transfer Learning," *Seminar Nasional Teknologi & Sains*, vol. 3, no. 1, pp. 361–369, Jan. 2024, doi: 10.29407/stains.v3i1.4347.

[8] M. F. Naufal and S. F. Kusuma, "Analisis Perbandingan Algoritma Machine Learning dan Deep Learning untuk Klasifikasi Citra Sistem Isyarat Bahasa Indonesia (SIBI)," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 10, no. 4, Art. no. 4, 2024, doi: 10.25126/jtiik.20241046823.

[9] Q. Dong, "Mobile Robot Vision Image Feature Recognition Method Based on Machine Vision," *EAI Endorsed Trans Energy Web*, vol. 10, Dec. 2023, doi: 10.4108/ew.3450.

[10] J. Li, C. Li, J. Han, Y. Shi, G. Bian, and S. Zhou, "Robust Hand Gesture Recognition Using HOG-9ULBP Features and SVM Model," *Electronics*, vol. 11, no. 7, Art. no. 7, Jan. 2022, doi: 10.3390/electronics11070988.

[11] N.-T. Do, S.-H. Kim, H.-J. Yang, and G.-S. Lee, "Robust Hand Shape Features for Dynamic Hand Gesture Recognition Using Multi-Level Feature LSTM," *Applied Sciences*, vol. 10, no. 18, Art. no. 18, Jan. 2020, doi: 10.3390/app10186293.

[12] Y. Li and P. Zhang, "Static hand gesture recognition based on hierarchical decision and classification of finger features," *Science Progress*, vol. 105, no. 1, p. 003685042210863, Jan. 2022, doi: 10.1177/00368504221086362.

[13] N. Gozzi *et al.*, "Image Embeddings Extracted from CNNs Outperform Other Transfer Learning Approaches in Classification of Chest Radiographs," *Diagnostics*, vol. 12, no. 9, Art. no. 9, Sep. 2022, doi: 10.3390/diagnostics12092084.

[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 2818–2826. doi: 10.1109/CVPR.2016.308.

[15] X. Fang *et al.*, "Ultrasound image intelligent diagnosis in community-acquired pneumonia of children using convolutional neural network-based transfer learning," *Front. Pediatr.*, vol. 10, Nov. 2022, doi: 10.3389/fped.2022.1063587.

[16] J. Loyola-Fuentes, L. Pietrasanta, M. Marengo, and F. Coletti, "Machine Learning Algorithms for Flow Pattern Classification in Pulsating Heat Pipes," *Energies*, vol. 15, no. 6, Art. no. 6, Jan. 2022, doi: 10.3390/en15061970.

[17] X. Fu, Y. Chen, J. Yan, Y. Chen, and F. Xu, "BGRF: A broad granular random forest algorithm," *Journal of Intelligent & Fuzzy Systems*, vol. 44, no. 5, pp. 8103–8117, Jan. 2023, doi: 10.3233/JIFS-223960.

[18] D. Angelakis, E. Ventouras, and P. Asvestas, "Brain Signals to Actions Using Machine Learning," *Engineering Proceedings*, vol. 50, no. 1, Art. no. 1, 2023, doi: 10.3390/engproc2023050007.

[19] H. Bassi, R. P. Yim, J. Vendrow, R. Koduluka, C. Zhu, and H. Lyu, "Learning to predict synchronization of coupled oscillators on randomly generated graphs," *Sci Rep*, vol. 12, no. 1, p. 15056, Sep. 2022, doi: 10.1038/s41598-022-18953-8.

[20] S. Akhtar *et al.*, "Short-Term Load Forecasting Models: A Review of Challenges, Progress, and the Road Ahead," *Energies*, vol. 16, no. 10, Art. no. 10, Jan. 2023, doi: 10.3390/en16104060.

[21] G. Alwafi and B. Fakieh, "A machine learning model to predict privacy fatigued users from social media personalized advertisements," *Sci Rep*, vol. 14, no. 1, p. 3685, Feb. 2024, doi: 10.1038/s41598-024-54078-w.

[22] M. D. Chaity and J. van Aardt, "Exploring the Limits of Species Identification via a Convolutional Neural Network in a Complex Forest Scene through Simulated Imaging

Spectroscopy," *Remote Sensing*, vol. 16, no. 3, Art. no. 3, Jan. 2024, doi: 10.3390/rs16030498.

[23] B. Düzgün, A. Çayır, U. Ünal, and H. Dağ, "Network intrusion detection system by learning jointly from tabular and text-based features," *Expert Systems*, vol. 41, no. 4, p. e13518, 2024, doi: 10.1111/exsy.13518.

[24] C.-S. Cheng, P.-W. Chen, Y.-C. Hsieh, and Y.-T. Wu, "Multivariate Process Control Chart Pattern Classification Using Multi-Channel Deep Convolutional Neural Networks," *Mathematics*, vol. 11, no. 15, Art. no. 15, Jan. 2023, doi: 10.3390/math11153291.

[25] W. Peng and J. Huang, "An Approach to Incorporating Implicit Knowledge in Object Detection Models," *Applied Sciences*, vol. 14, no. 5, Art. no. 5, Jan. 2024, doi: 10.3390/app14052124.

[26] S. Rhouas, A. E. Attaoui, and N. E. Hami, "Enhancing currency prediction in international e-commerce: Bayesian-optimized random forest approach using the Klarna dataset," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 3, Art. no. 3, Jun. 2024, doi: 10.11591/ijece.v14i3.pp3177-3186.

