Journal of Intelligent System and Telecommunications

Journal homepage: https://journal.unesa.ac.id/index.php/jistel/index

A Comparative Analysis of CNN and SVM for Static Sign Language Recognition Using MediaPipe Landmarks

Akbar Kurnia Saleh[1*]
[1]Dept. of Electrical Engineering, Universitas Negeri Surabaya, Indonesia
*Correspondence: E-mail: 24051505008@mhs.unesa.ac.id

ARTICLE INFO

Article History:
Submitted/Received 11 June 2025
First Revised 30 June 2021
Accepted 30 June 2025
First Available Online 30 June 2025
Publication Date 30 June 2025

Keyword:
CNN, SVM, Mediapipe, SIBI, Handsign classification.

ABSTRACT

This study presents a direct comparative analysis of a Convolutional Neural Network (CNN) and a Support Vector Machine (SVM) for static Indonesian Sign Language (SIBI) alphabet recognition, utilizing landmarks extracted via MediaPipe. The primary contribution of this work is to provide a comprehensive performance benchmark, evaluating the trade-offs between a deep learning model (CNN) and a classical machine learning model (SVM) in terms of accuracy, computational efficiency, and robustness under a unified experimental framework. The evaluation, conducted using metrics such as accuracy, F1-score, balanced accuracy, and ROC AUC, reveals divergent performance profiles. The CNN model achieved perfect classification accuracy (1.00) across all metrics, with its learning curve demonstrating stable and effective generalization. In contrast, the SVM model achieved a respectable test accuracy of 80% and a ROC AUC score of 0.99, but exhibited some misclassifications for visually similar gestures. Notably, the SVM demonstrated significantly faster training times, completing its training in under 0.09 seconds, whereas the CNN required approximately 0.5 seconds per epoch. These findings empirically validate that while CNN offers superior accuracy, the SVM remains a highly relevant and efficient alternative for applications with constrained computational resources. This research provides a crucial reference for developers in selecting the appropriate architecture for real-time sign language recognition systems.

1. INTRODUCTION

For people who are deaf or find it difficult to speak, sign language is the basic means of communication. By means of a coordinated use of hand gestures, finger placements, and physical movements, it communicates meaning so facilitating visual and tactile interaction between speakers [5]. The creation of real-time sign language recognition systems has become absolutely essential in a society growing in inclusiveness. Such systems not only enable communication between the hearing-impaired and the larger society but also offer great possibilities in fields including education, public service delivery, and assistive technologies [3].

One of the leading-edge solutions in this field is using MediaPipe, a potent Google framework. MediaPipe is suited for gesture-based applications [8][13][17] since it is made to effectively and precisely detect and track 21 key points on human hands. Its capacity to process video data in real-time while preserving compatibility with a range of hardware platforms, including those with limited processing capacity [8][23][25] is among its main strengths.

Various methods have been developed based on landmark data generated by MediaPipe. One of the most prominent is the combination of MediaPipe with Convolutional Neural Network (CNN) [4][10]. CNN naturally excels at recognizing spatial patterns and has shown excellent performance in static sign language alphabet classification (A–Z), even with near-99% accuracy in some studies [21][23]. This approach is considered strong because it is able to extract spatial features from landmark coordinates effectively, resulting in precise and fast predictions [6][11][18],

However, the use of CNN directly on landmark data often results in high-dimensional features. This becomes an obstacle when the model is applied to edge devices or real-time systems with limited memory and computing power [1][7]. To overcome this challenge, Principal Component Analysis (PCA) is utilized as an effective technique for reducing data dimensionality. By identifying and retaining the most critical variations within the dataset, PCA simplifies the input structure, ensuring that the key characteristics of the data are maintained even in a more compact form [7][19]. The combination of CNN with MediaPipe shows great potential as an efficient solution as it brings together CNN's advantages in spatial feature extraction and in simplifying input complexity [19][22].

On the other hand, a Support Vector Machine (SVM)-based approach is also a viable alternative. Unlike CNN which belongs to the deep learning category, SVM is a classic supervised learning method that is effective on high-dimensional data at the scale of medium to small datasets. When combined with the MediaPipe extraction results feature, SVM is able to provide competitive results, especially in terms of efficiency and computing needs [19][15]. In a particular study, the combination of MediaPipe with PCA with SVM showed satisfactory performance for alphabet classification, even outperforming CNN in resource-constrained scenarios.

Although sequence-based models such as LSTM and GRU are known to excel at dynamic gesture recognition [2][20][24], this approach is less appropriate for static alphabets that do not require temporal information. In addition, sequence models such as LSTMs are generally more complex, require more parameters, and require longer training time [16][21].

DOI: https://doi.org/10.26740/jistel.v1n2.p225-238
e- ISSN 3090-613X

Despite the success of both CNN and SVM in various recognition tasks, there is a lack of direct, empirical comparisons within a single, controlled framework for static sign language recognition using MediaPipe landmarks. Previous studies often focus on a single methodology or do not simultaneously evaluate performance across accuracy, computational cost, and model complexity. This creates a critical gap for developers and researchers, who need clear benchmarks to decide whether a complex deep learning model is necessary or if a more lightweight classical model is sufficient, especially for deployment on resource-constrained devices like mobile phones or embedded systems. The problem, therefore, is the absence of a clear comparative baseline that weighs the trade-off between the high accuracy of CNNs and the computational efficiency of SVMs for this specific application.

One thing that has been lacking in previous studies is the absence of comprehensive comparative studies that thoroughly review the advantages and limitations of these approaches in a single experimental framework. Some studies focus on only one type of method or do not measure the performance of various aspects simultaneously such as accuracy, execution time, and implementation complexity [14][9][12].

To address this gap, this study provides an original contribution in the form of a direct comparative analysis between two primary approaches: a CNN architecture and an SVM classifier, both utilizing features extracted from MediaPipe. The strong point of this research lies in its rigorous and controlled experimental design.

Both approaches are tested on the same static SIBI alphabet (A–Z) dataset and evaluated not only on classification accuracy but also on a comprehensive set of metrics including computational efficiency (training time) and model simplicity. The findings from this research are intended to serve as a definitive reference to guide the selection of the most suitable architecture for building gesture-based communication systems, particularly for applications demanding real-time performance on devices with limited computational resources.

2. METHODS

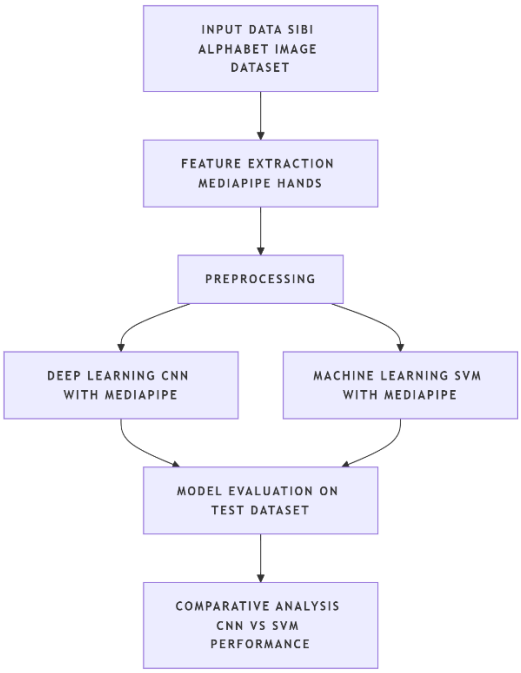

Figure 1. Overview of the Proposed Methodology

DOI: https://doi.org/10.26740/jistel.v1n2.p225-238
e- ISSN 3090-613X

This study's workflow in figure 1 is built upon a systematic pipeline, beginning with raw images from the SIBI Sign Language dataset. Instead of using the images directly, the methodology first leverages Google's MediaPipe framework to engineer features. This powerful tool analyzes each image to pinpoint 21 key hand landmarks, translating their three-dimensional (x,y,z) coordinates into a single 63-dimensional numerical vector. This vectorization step effectively converts complex visual data into a structured format that is highly optimized for machine learning algorithms.

Next, these feature vectors are meticulously prepared for training. The data is partitioned using a stratified 80:20 train-test split to ensure a balanced class representation, followed by Min-Max scaling to normalize all features to a uniform [0, 1] range. The prepared data is then simultaneously channeled into two distinct models for a head-to-head comparison: a deep learning-based Convolutional Neural Network (CNN) architected for this specific data structure, and a classical Support Vector Machine (SVM) equipped with a non-linear RBF kernel.

Finally, the performance of both trained models is rigorously benchmarked against the unseen test data. The evaluation employs a comprehensive suite of metrics including accuracy, F1 score, ROC AUC, and training time to provide a multi-faceted view of each model's capabilities. The results from this empirical testing form the basis for the final comparative analysis, aimed at delivering a clear verdict on the practical trade-offs between the accuracy-driven deep learning approach and the efficiency-focused classical method.

2.1. Material

2.1.1 Dataset

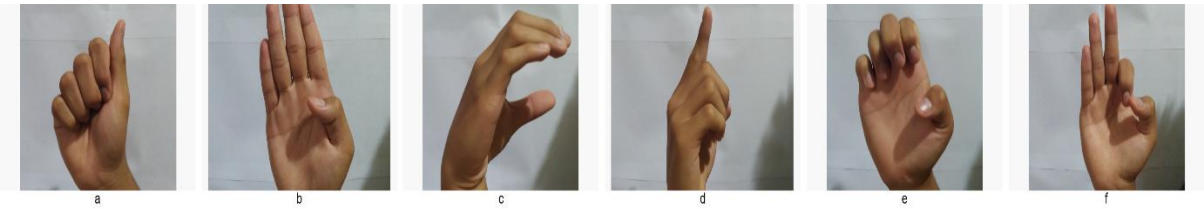

Figure 2. Visualization of the SIBI Dataset.

The SIBI Sign Language Alphabets datasets available on Kaggle are a collection of A–Z alphabet image imagery data in the Indonesian Sign Language System (SIBI). This dataset was developed by M. Lanang Afkaar and is intended to support research in computer vision-based sign language recognition.

This dataset consists of 26 classes, each representing the letters of the Latin alphabet (A to Z). Each class contains a number of hand images that form the letters according to SIBI standards. These images are taken in a variety of lighting conditions and backgrounds to reflect the real variations that may be encountered in real-world applications.

This study utilized only unprocessed photos from the dataset, without augmentation or pre-treatment. The objective is to evaluate the model's performance under data conditions that closely mimic real world scenarios. Each class in Figure 2 serves as a visual representation of each letter in model analysis and training.

2.2 Method

2.2.1 Datasets and Preprocessing

DOI: https://doi.org/10.26740/jistel.v1n2.p225-238
e- ISSN 3090-613X

The dataset used consists of SIBI alphabet images in the form of hand gestures, each representing the letters A to Z (without any other numbers or symbols). Each image is processed through MediaPipe Hands, a real-time framework from Google that can detect and extract 21 landmarks on the hand in the form of 3D coordinates (x, y, z). The results of this extraction are then used as feature representations for both classification methods.

The dataset is divided into training and testing subsets at an 80:20 ratio. The stratified split technique is utilized to maintain class balance, ensuring proportional distribution of letters across each group. All features are subjected to Min-Max Scaling within the interval [0, 1] before being assigned to the classification model.

2.2.2 CNN Architecture and Training with MediaPipe

This work presents a convolutional neural network (CNN) architecture especially designed to manage input structured as landmark vectors, derived by reformatting MediaPipe into tensor-compatible structures. The model's design extracts hierarchical elements from hand gesture data by means of a multi-stage processing approach:

- Two convolutional layers applied consecutively in the component of feature extraction each follow batch normalisation, ReLU activation, and max pooling for spatial downsampling. The model learns layered spatial relationships buried in the hand landmark coordinates by means of this sequence.

  The core of the convolutional layers is the convolution operation, which is defined as:

$$(f\ x\ g)(t) = \sum_{i=-\infty}^{\infty} f(i)\mathrm{g}(t-i) \tag{1}$$

  Where f is the input feature map and g is the kernel. This is followed by a Rectified Linear Unit (ReLU) activation function to introduce non-linearity, defined as:

$$ReLU(x) = \max(0, x) \tag{2}$$

- For classification, the architecture consists in two fully connected (dense) layers with an output layer comprising 26 units corresponding to the letters A through Z. After that, a flattening layer transforms the resulting multidimensional feature maps into a one-dimensional feature vector fit for classification. By means of a softmax function, the last layer transforms outputs into class probability. Optimized with the Adam method, the categorical cross-entropy loss function guides training. Metrics covering accuracy, F1 score, balanced accuracy, and ROC.

This architecture emphasizes systematic feature abstraction through its layered design, combining spatial pattern recognition in early stages with discriminative classification in subsequent layers, while maintaining computational efficiency through dimensionality reduction techniques.The model underwent training for 30 epochs utilizing a batch size of 32. To mitigate overfitting, early stopping and k-fold cross-validation approaches were employed, accompanied by the monitoring of validation metrics throughout the training process.

DOI: https://doi.org/10.26740/jistel.v1n2.p225-238
e- ISSN 3090-613X

2.2.3 SVM Architecture and Training with MediaPipe

This work makes use of a second method based on the Support Vector Machine (SVM) algorithm for classification problems. Using the same set of input features obtained from MediaPipe 21 hand landmarks taken across three spatial dimensions (x, y, z) this approach generates 63 numerical features overall. These properties are normalized to guarantee consistency in scale across dimensions before classification.

SVM does not include an inherent feature learning mechanism unlike the Convolutional Neural Network (CNN) method. Since SVM directly runs on pre-extracted features without depending on iterative learning of representations, this difference makes SVM more computationally efficient.

Radial Basis Function (RBF) kernel is used in construction of the SVM classifier this choice is appropriate for data with non-linear trends. The SVM algorithm works by finding an optimal hyperplane that maximizes the margin between classes. For a non-linear classification, the decision function is given by:

$$f(x) = sign\left(\sum_{i=1}^{N} f\alpha_i y_i K(x_i, x_j) + \mathrm{b}\right) \tag{3}$$

This study utilizes the Radial Basis Function (RBF) kernel, defined as:

$$K(x_i, x_j) = \exp(-\gamma \left\| x_i - x_j \right\|^2) \tag{4}$$

Where γ (gamma) is a hyperparameter that defines the influence of a single training example. Using a methodical grid search approach, the hyperparameters of the model—that of the regularization parameter C and the kernel coefficient gamma are tuned to find the best parameter combination.

Using a methodical grid search approach, the hyperparameters of the model—that of the regularization parameter C and the kernel coefficient gamma are tuned to find the best parameter combination.

By averaging results over several data partitions, k-fold cross-valuation with k = 5 generates a strong estimate of generalization performance. This operation reduces possible bias connected to particular data splits. More research is done in order to have better understanding of the SVM model's performance and efficiency. These show how the accuracy of the model changes in response to different training set sizes, so reflecting its learning dynamics. Designed to evaluate the scalability and computational cost of the model during training, training time analysis relative to dataset size.

2.3 Evaluation and comparison

Using a dedicated test dataset, the performance of both classification models was evaluated over a whole range of criteria. Among these are accuracy, which gauges the general percentage of accurately labeled examples.

- F1 Score provides a balanced assessment of the classification performance of the model by representing the harmonic mean of recall and accuracy. Calculated as the average recall across all classes, balanced accuracy offers a fair evaluation especially in situations of class imbalance.
- By computing the area under the Receiver Operating Characteristic curve, ROC AUC Score measures the model's ability to differentiate between classes.

DOI: https://doi.org/10.26740/jistel.v1n2.p225-238
e- ISSN 3090-613X

- Confusion Matrix helps to identify particular misclassification patterns by providing a comprehensive picture of classification results per individual class in this case, every letter.
- To assess computational efficiency, training duration measured for the CNN model per epoch and as overall training time for the SVM model Graphically presented results from these assessments help to improve clarity and support for comparative study. Deeper interpretation and debate of the relative strengths and shortcomings of every model in the section on subsequent analysis build on these visualizations.

3. RESULTS AND DISCUSSION

3.1. Results

3.1.1 CNN Evaluation with MediaPipe

a. Confusion Matrix

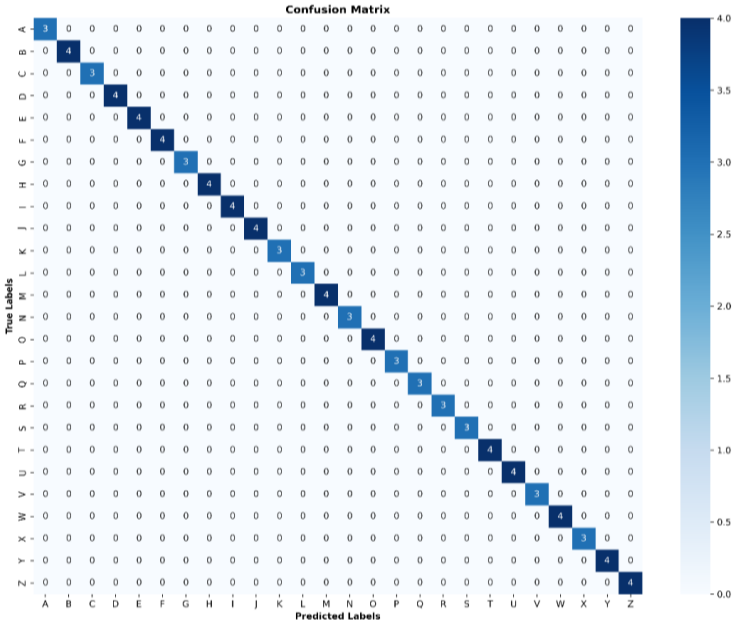

Figure 3. Confusion matrix

The confusion matrix for the model is illustrated in Figure 3, which evaluates the performance across the 26 classes of the SIBI (Indonesian Sign Language) alphabet. The matrix indicates that most predictions align with the diagonal, demonstrating a substantial quantity of accurate classifications for each letter from A to Z. It is important to highlight that nearly all classes were accurately predicted, exhibiting no significant misclassifications.

This indicates a strong performance across all categories, highlighting that the model has effectively learned to differentiate between the distinct hand shapes corresponding to each alphabet sign. Each class received a minimum of three correct predictions (indicated by the diagonal entries), with several classes achieving four correct classifications out of four samples.

b. Final Metric Evaluation

DOI: https://doi.org/10.26740/jistel.v1n2.p225-238
e- ISSN 3090-613X

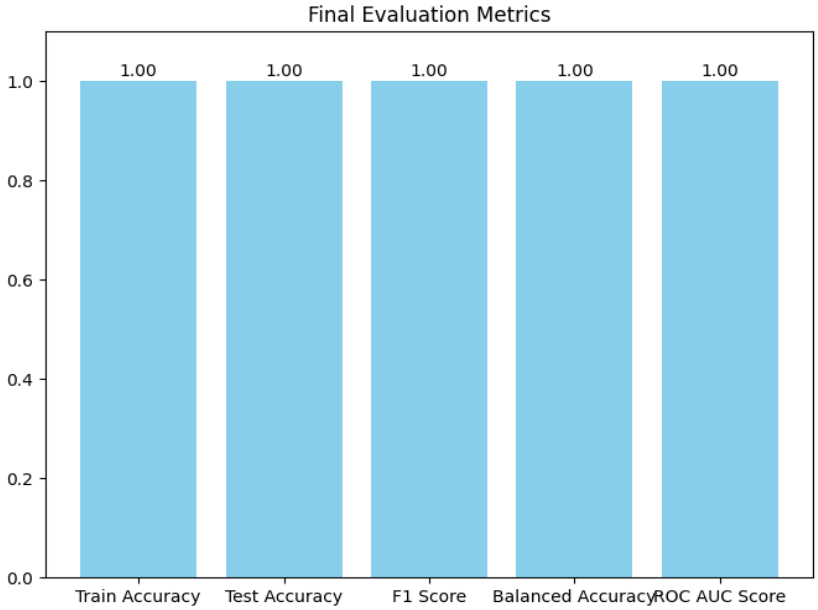

Figure 4. Evaluation matrix

The convolutional neural network (CNN) model's performance results in figure 4 for SIBI alphabet recognition. All recorded at 1.0, the model attained perfect scores across all key evaluation metrics—training accuracy, testing accuracy, F1 score, balanced accuracy, and the area under the ROC curve.

These findings confirm that, both during training and on fresh, unprocessed input data, the model can precisely identify every letter in the SIBI alphabet. Moreover, the high values in ROC AUC and balanced accuracy imply that the used dataset was well-balanced across classes and that the model preserved objectivity free from favoring particular classes. For sign language alphabet recognition in the framework of SIBI, the CNN-based method shows to be generally dependable and computationally effective.

c. Learning Curve

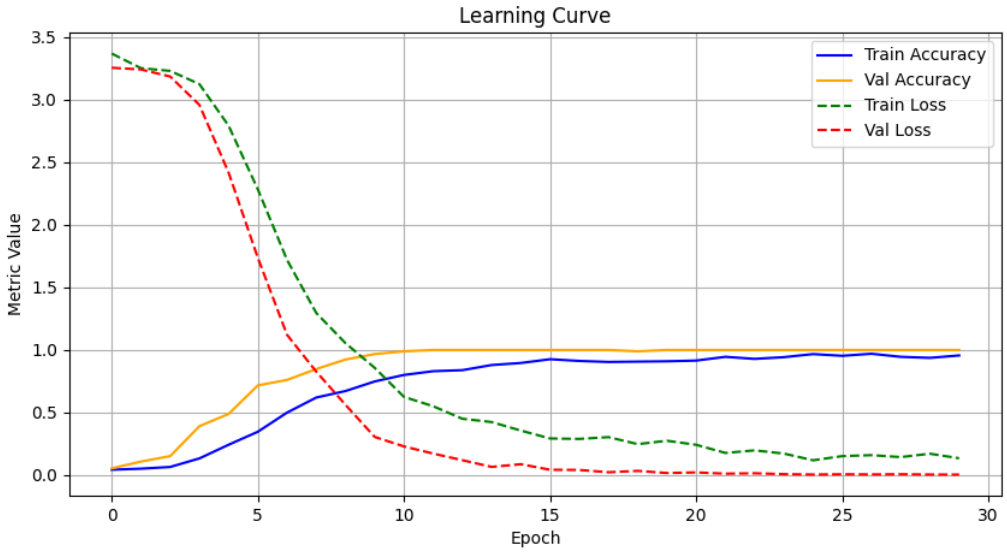

Figure 5. Learning curve graph

Figure 5 illustrates the CNN model's learning progression, specifically the trends in accuracy and loss over the span of 30 training epochs. From approximately the fifth epoch onward, both training and validation accuracy display a consistent upward trajectory, ultimately approaching

DOI: https://doi.org/10.26740/jistel.v1n2.p225-238
e- ISSN 3090-613X

near-perfect performance. This upward trend indicates the model's growing generalization capability to previously unseen data.

In parallel, the training and validation loss curves show a steady decline, nearing zero as training progresses. This pattern reflects the model's effective reduction of prediction error and indicates stable learning without signs of divergence. The close alignment between training and validation metrics further suggests that the model successfully avoids overfitting. Overall, the learning curve analysis demonstrates that the CNN architecture is capable of efficient training, robust generalization, and achieving a desirable balance between model bias and variance.

d. Training Time

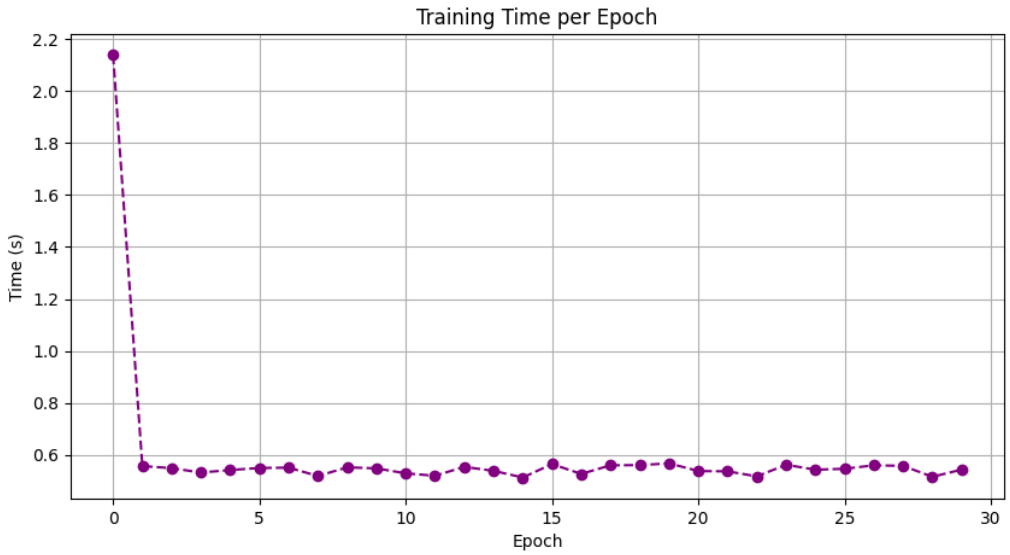

Figure 6. Training time graph

Figure 6 delineates the temporal characteristics of the CNN model's training process across successive epochs. As shown, the initial epoch necessitates approximately 2.1 seconds of computational time, a duration attributable to preliminary computational overheads associated with data pipeline initialization, memory resource allocation, and parameter configuration. Subsequent epochs exhibit a marked reduction in processing time, converging to a stable range between 0.5 and 0.6 seconds per epoch.

This temporal pattern demonstrates the model's operational efficiency and computational stability during the training phase. The observed reduction in temporal demands and consistent execution intervals across epochs indicate optimized architectural design and implementation, facilitating rapid iterative training cycles without substantial temporal variance. Such temporal predictability is critical for applications requiring extended training periods or deployment in time-sensitive operational contexts, where resource management and processing reliability are paramount. The empirical evidence derived from these temporal metrics validates the architecture's dual capability in maintaining classification precision while achieving computational effectiveness, reinforcing its suitability for both scalable training frameworks and real-time inference scenarios.

3.1.2 SVM Evaluation with MediaPipe

a. Confusion Matrix

DOI: https://doi.org/10.26740/jistel.v1n2.p225-238
e- ISSN 3090-613X

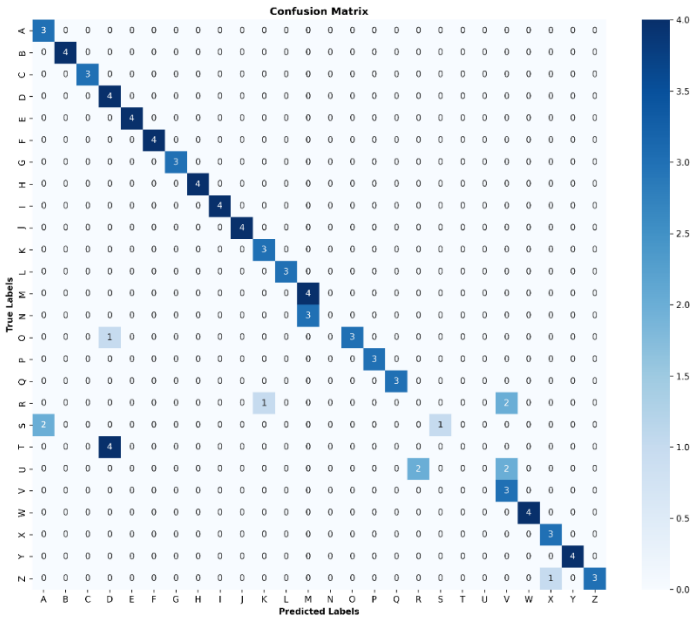

Figure 7. Confusion matrix

Figure 7 presents the confusion matrix, offering a detailed depiction of the model's classification accuracy across all individual letters in the SIBI alphabet. The prominence of high values along the diagonal axis of the matrix highlights the model's strong predictive accuracy, indicating that most letters were correctly classified. There are several off-diagonal instances, specifically for the letters 'R', 'U', 'V', and 'Z', that were misclassified on occasion.

The observed misclassifications indicate that specific hand gestures may exhibit visual similarities, or that additional data could be required to effectively distinguish between certain classes. The overall structure of the matrix demonstrates strong performance, exhibiting minimal misclassifications in relation to the number of classes. The model has successfully identified significant patterns and demonstrates the ability to accurately differentiate between a diverse range of sign language gestures.

b. Final Metric Evaluation

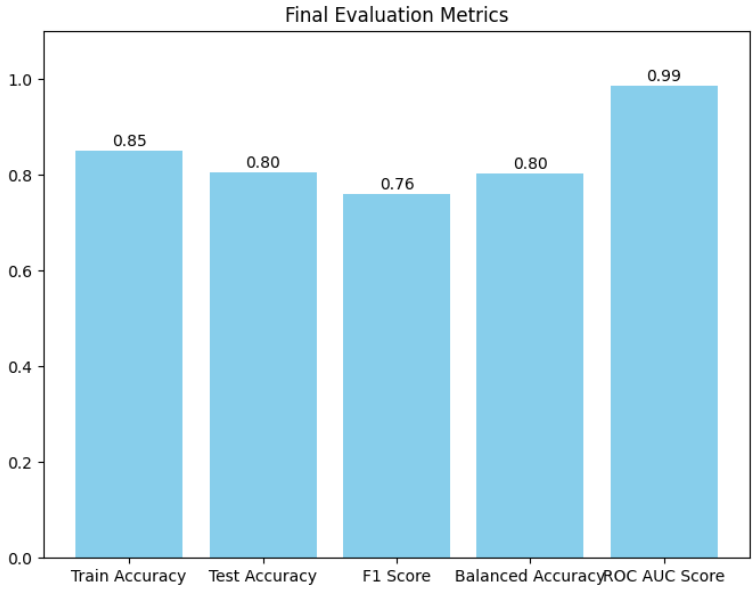

Figure 8. Final matrix evaluation

DOI: https://doi.org/10.26740/jistel.v1n2.p225-238
e- ISSN 3090-613X

Figure 8 presents the quantitative outcomes spanning multiple performance axes derived from the model's final evaluation phase. Empirical results reveal a training accuracy of 0.85 alongside a test accuracy of 0.80, signifying moderate generalizability to unseen data distributions while retaining a controlled degree of overfitting. The macro-averaged F1 score, recorded at 0.76, indicates acceptable harmonization of precision and recall metrics in the polychotomous classification context—a critical consideration in applications such as isolated gesture recognition.

Notably, the balanced accuracy metric mirrors the test accuracy at 0.80, demonstrating parity in predictive performance across heterogeneous classes despite potential marginal disparities in class representation. The most compelling evidence of discriminative capacity emerges from the ROC AUC metric, which attained a value of 0.99, approaching theoretical maximum separation between class probability distributions. These collective metrics substantiate the architecture's operational viability for SIBI alphabet classification tasks, with particular strength in inter-class differentiation. While the current performance thresholds satisfy baseline functional requirements, the discrepancy between the F1 score and accuracy metrics suggests opportunities for refinement in harmonizing precision and recall metrics, particularly for classes exhibiting lower feature saliency.

The convergence of these evaluation dimensions spanning discriminative power, generalization capacity, and class-agnostic performance provides empirical validation of the model's technical adequacy while delineating specific pathways for future optimization efforts. Such multidimensional validation proves essential when deploying classification systems in real-world human-computer interaction scenarios, where both reliability and equitable performance across all target classes constitute non-negotiable operational parameters.

c. Learning Curve and training time

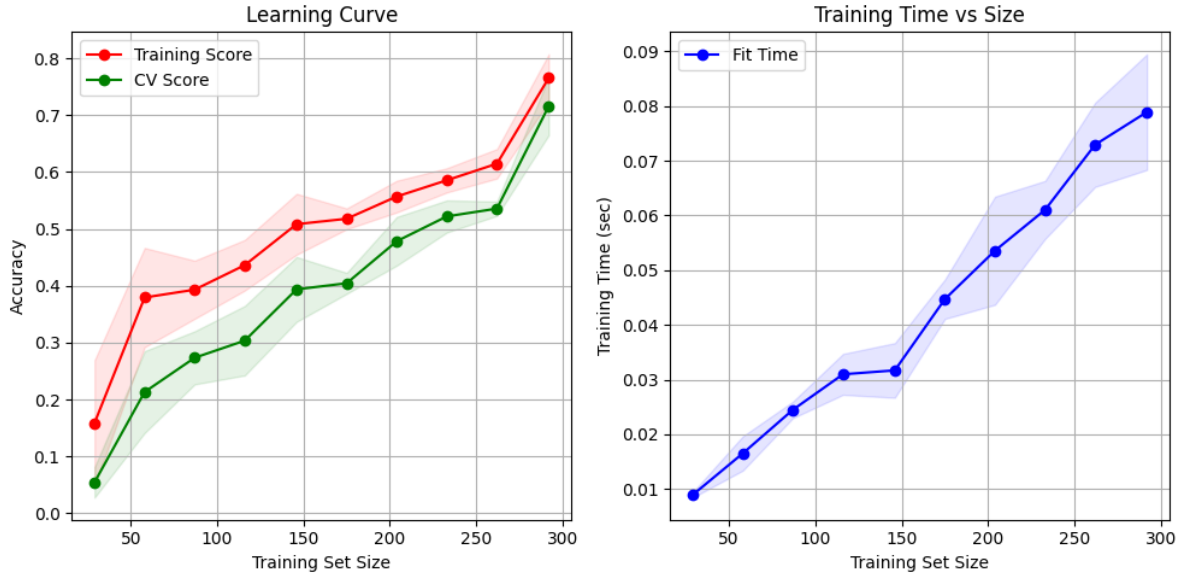

Figure 9. Learning Curve and training time

Figure 9 offers insights into the model's learning behavior and training efficiency relative to different training set sizes. In the left plot, the learning curve illustrates how both the training score and cross-validation (CV) score improve steadily as the training set size increases. This positive trend, accompanied by narrowing confidence intervals, suggests that the model benefits from more training data and is not overfitting. The CV score approaching the training score also indicates improved generalization.

In the right plot, the graph of training time versus dataset size shows a nearly linear increase in training time as the number of training samples grows. Despite this rise, the fit time remains

DOI: https://doi.org/10.26740/jistel.v1n2.p225-238
e- ISSN 3090-613X

low, not exceeding 0.09 seconds even at the maximum set size. This demonstrates the scalability and computational efficiency of the model, affirming its feasibility for use in larger datasets or real-time applications.

3.1.3 Comparative Analysis of Methods

Table 1. Comparison of cnn with mediapipe and svm with mediapipe

Aspects	CNN with MediaPipe	SVM with MediaPipe
Test Accuracy	100%	80%
F1 Score	1.00	0.76
Skor ROC AUC	1.00	0.99
Noise resistance	Very high	Keep
Training Time	~0.5 seconds/time	~0.01–0.09 seconds total
Model Complexity	High (deep learning)	Low (classic machine learning)
Data Dependency	Efficient, even on small data	Sensitive to small data
Scalability	Excellent	Limited to data complexity

Table 1 In general, CNN with MediaPipe excels in accuracy and robustness of complex data, but has higher computing overhead. SVM with MediaPipe is lighter and faster, but not as precise as CNN especially in multi-class classifications such as the SIBI alphabet.

4. CONCLUSION

The CNN model with MediaPipe excels significantly in terms of accuracy and generalization, making it particularly suitable for real-time sign language letter classification applications, especially if accuracy is a top priority. The uniqueness of this approach is its ability to make optimal use of the geometric features of MediaPipe in convolutional networks.

In contrast, SVM models with MediaPipe have advantages in terms of training speed and model interpretability, suitable for lightweight deployments in devices with limited resources. This approach shows that classical techniques can still compete when combined with modern feature extraction.

These two methods complement each other and are worthy of further research for publication, especially in the context of the development of interactive systems based on sign language.

5. ACKNOWLEDGMENT

This section aims to express gratitude to all people/parties who helped with the research. Acknowledgements may also be written to anyone who provided intellectual contributions, technical assistance (including in writing and editing), or special equipment or materials.

6. AUTHORS' NOTE

The authors declare that there is no conflict of interest regarding the publication of this article. The authors confirmed that the paper was free of plagiarism.

7. AUTHORS' CONTRIBUTION/ROLE

Akbar Kurnia Saleh: Data collection, analysis, programming and technical experiment, and writing article.

DOI: https://doi.org/10.26740/jistel.v1n2.p225-238
e- ISSN 3090-613X

8. REFERENCES

[1] M. S. Abdallah, G. H. Samaan, A. R. Wadie, F. Makhmudov, and Y. I. Cho, "Light-Weight Deep Learning Techniques with Advanced Processing for Real-Time Hand Gesture Recognition," *Sensors*, vol. 23, no. 1, Jan. 2023, doi: 10.3390/s23010002.

[2] A. Ahmad Ilham and I. Nurtanio, "Dynamic Sign Language Recognition Using Mediapipe Library and Modified LSTM Method," vol. 13, no. 6, 2023.

[3] M. Al-Hammadi *et al.*, "Deep learning-based approach for sign language gesture recognition with efficient hand gesture representation," *IEEE Access*, vol. 8, pp. 192527–192542, 2020, doi: 10.1109/ACCESS.2020.3032140.

[4] M. F. Azzaky Rizki, L. Anifah, and A. Aye Mon, "SURFACE DETECTION FOR QUADRUPED ROBOT USING YOLO-V3 TINY," 2024. [Online]. Available: http://dx.doi.org/10.xxxxx/jistel.vXiX

[5] M. A. Bencherif *et al.*, "Arabic Sign Language Recognition System Using 2D Hands and Body Skeleton Data," *IEEE Access*, vol. 9, pp. 59612–59627, 2021, doi: 10.1109/ACCESS.2021.3069714.

[6] Z. Cui, Z. Chen, Z. Li, and Z. Wang, "Spatial-Temporal Graph Transformer with Sign Mesh Regression for Skinned-Based Sign Language Production," *IEEE Access*, vol. 10, pp. 127530–127539, 2022, doi: 10.1109/ACCESS.2022.3227042.

[7] Z. Dozdor, Z. Kalafatic, Z. Ban, and T. Hrkac, "TY-Net: Transforming YOLO for Hand Gesture Recognition," *IEEE Access*, vol. 11, pp. 140382–140394, 2023, doi: 10.1109/ACCESS.2023.3341702.

[8] A. Gupta, N. Chawla, R. Jain, N. Thakur, and A. Devi, "Gesture-Based Touchless Operations: Leveraging MediaPipe and OpenCV."

[9] F. Jafari and A. Basu, "Two-Dimensional Parallel Spatio-Temporal Pyramid Pooling for Hand Gesture Recognition," *IEEE Access*, vol. 11, pp. 133755–133766, 2023, doi: 10.1109/ACCESS.2023.3336591.

[10] A. Ivan, T. Jaya, P. Puspitaningayu, A. P. Adiwangsa, and N. Funabiki, "Two-dimensional Human Pose Estimation using Key Points' Angular Detection for Basic Strength Training," *Journal of Intelligent System and Telecommunications*, vol. 1, no. 1, pp. 105–119, 2024, doi: 10.xxxxx/jistel.vXiX.

[11] S. Huse, R. Makode, T. Wankhade, and T. Nachane, "Real-Time ISL Recognition Using CNN and MediaPipe." [Online]. Available: www.ijfmr.com

[12] R. Kumar, S. K. Singh, A. Bajpai, and A. Sinha, "Mediapipe and CNNs for Real-Time ASL Gesture Recognition."

[13] Y. Meng, H. Jiang, N. Duan, and H. Wen, "Real-Time Hand Gesture Monitoring Model Based on MediaPipe's Registerable System," *Sensors*, vol. 24, no. 19, Oct. 2024, doi: 10.3390/s24196262.

[14] Divya Naadem, "Sign Language Detection Using Hand Gestures," *Int J Res Appl Sci Eng Technol*, vol. 10, no. 6, pp. 935–942, Jun. 2022, doi: 10.22214/ijraset.2022.43997.

DOI: https://doi.org/10.26740/jistel.v1n2.p225-238
e- ISSN 3090-613X

[15] P. T. Nguyen, T. H. Nguyen, N. X. N. Hoang, H. T. B. Phan, H. S. H. Vu, and H. N. Huynh, "Exploring MediaPipe optimization strategies for real-time sign language recognition," *CTU Journal of Innovation and Sustainable Development*, vol. 15, no. ISDS, pp. 142–152, Oct. 2023, doi: 10.22144/ctujoisd.2023.045.

[16] B. Qiang *et al.*, "SqueezeNet and Fusion Network-Based Accurate Fast Fully Convolutional Network for Hand Detection and Gesture Recognition," *IEEE Access*, vol. 9, pp. 77661–77674, 2021, doi: 10.1109/ACCESS.2021.3079337.

[17] M. A. Rahim, J. Shin, and K. S. Yun, "Hand gesture-based sign alphabet recognition and sentence interpretation using a convolutional neural network," *Annals of Emerging Technologies in Computing*, vol. 4, no. 4, pp. 20–27, 2020, doi: 10.33166/AETiC.2020.04.003.

[18] T. J. Sánchez-Vicinaiz, E. Camacho-Pérez, A. A. Castillo-Atoche, M. Cruz-Fernandez, J. R. García-Martínez, and J. Rodríguez-Reséndiz, "MediaPipe Frame and Convolutional Neural Networks-Based Fingerspelling Detection in Mexican Sign Language," *Technologies (Basel)*, vol. 12, no. 8, Aug. 2024, doi: 10.3390/technologies12080124.

[19] F. Shah, M. S. Shah, W. Akram, A. Manzoor, R. O. Mahmoud, and D. S. Abdelminaam, "Sign Language Recognition Using Multiple Kernel Learning: A Case Study of Pakistan Sign Language," *IEEE Access*, vol. 9, pp. 67548–67558, 2021, doi: 10.1109/ACCESS.2021.3077386.

[20] S. Shin and W. Y. Kim, "Skeleton-Based Dynamic Hand Gesture Recognition Using a Part-Based GRU-RNN for Gesture-Based Interface," *IEEE Access*, vol. 8, pp. 50236–50243, 2020, doi: 10.1109/ACCESS.2020.2980128.

[21] J. Shin, A. S. M. Miah, K. Suzuki, K. Hirooka, and M. A. M. Hasan, "Dynamic Korean Sign Language Recognition Using Pose Estimation Based and Attention-Based Neural Network," *IEEE Access*, vol. 11, pp. 143501–143513, 2023, doi: 10.1109/ACCESS.2023.3343404.

[22] J. Shin, A. S. M. Miah, Y. Akiba, K. Hirooka, N. Hassan, and Y. S. Hwang, "Korean Sign Language Alphabet Recognition Through the Integration of Handcrafted and Deep Learning-Based Two-Stream Feature Extraction Approach," *IEEE Access*, vol. 12, pp. 68303–68318, 2024, doi: 10.1109/ACCESS.2024.3399839.

[23] B. Subramanian, B. Olimov, S. M. Naik, S. Kim, K. H. Park, and J. Kim, "An integrated mediapipe-optimized GRU model for Indian sign language recognition," *Sci Rep*, vol. 12, no. 1, Dec. 2022, doi: 10.1038/s41598-022-15998-7.

[24] A. R. Verma, G. Singh, K. Meghwal, B. Ramji, and P. K. Dadheech, "Enhancing Sign Language Detection through Mediapipe and Convolutional Neural Networks (CNN)," Jun. 2024, [Online]. Available: http://arxiv.org/abs/2406.03729

[25] Yaseen, O. J. Kwon, J. Kim, S. Jamil, J. Lee, and F. Ullah, "Next-Gen Dynamic Hand Gesture Recognition: MediaPipe, Inception-v3 and LSTM-Based Enhanced Deep Learning Model," *Electronics (Switzerland)*, vol. 13, no. 16, Aug. 2024, doi: 10.3390/electronics13163233.

DOI: https://doi.org/10.26740/jistel.v1n2.p225-238
e- ISSN 3090-613X