SOCIAL SCIENCE REVIEW ARCHIVES
ISSN Online: 3006-4708
ISSN Print: 3006-4694
https://policyjournalofms.com

Auslan Sign Language Image Recognition Using Deep Neural Network

Shahnaz Aqsa Qambrani[1], Faiza Ahmed Dahri[2], Shabana Bhatti[3], Santosh Kumar Banbhrani[4]

[1,2,3,4] Department of Information and Computing, Faculty of Science and Technology, University of Sufism and Modern Sciences, Bhitshah Sindh Pakistan, Email: [1]aqsa999qambrani@gmail.com, [2]faizadahri8@gmail.com, [3]shabobhatti9@gmail.com, [4]banbhrani@gmail.com
Corresponding Author: Santosh Kumar Banbhrani, [4]banbhrani@gmail.com

DOI: https://doi.org/10.70670/sra.v3i3.1008

Abstract
Sign language recognition improves accessibility for the deaf and hard-of-hearing by translating hand gestures into machine-interpretable labels. This paper presents a hybrid pipeline for static Auslan digit recognition (classes 0–2) that combines convolutional neural networks (CNNs) for automated feature extraction with classical classifiers, Support Vector Machine (SVM) and Random Forest (RF). A grayscale dataset of 6,000 images (2,000 per class) was pre-processed using Canny edge detection to emphasize contour information, then resized for model inputs. Two CNN feature-extractors were trained and their flattened feature vectors fed to an RBF-kernel SVM and a 100-tree Random Forest. Experimental evaluation shows the CNN + Random Forest hybrid attained the highest validation accuracy (99.75%), outperforming the baseline end-to-end CNN (≈95%) and the CNN+SVM (99.67%). The trained pipeline was also integrated into a Mediapipe-based real-time testing setup to demonstrate practical applicability. Results indicate that combining deep feature extraction with ensemble/classical classifiers improves robustness and generalization for static gesture recognition. Future work will expand class coverage, incorporate dynamic gesture modelling, and investigate model compression for embedded deployment.

Keywords: Sign Language Recognition; Auslan; Convolutional Neural Network; Random Forest; Support Vector Machine (SVM); Canny Edge Detection; Media Pipe; Real-Time Recognition.

Introduction

Sign language serves as the primary mode of communication for deaf and hard-of-hearing communities worldwide. Automated Sign Language Recognition (SLR) systems aim to bridge the communication gap between signers and non-signers by translating hand gestures into machine-interpretable labels. With the rapid advancements in deep learning and computer vision, SLR has emerged as an active research area, showing promising progress in both isolated and continuous recognition tasks. Over the years, researchers have proposed various approaches to enhance the robustness and scalability of SLR. Pu et al. [4] developed an iterative alignment network for continuous recognition, while Hu et al. [15] introduced CorrNet, a correlation-based architecture for spatio-temporal modeling. Gloss-free translation methods such as Yin et al. [13] and Zhou et al. [14] advanced the field by eliminating the need for intermediate gloss annotations. More recently, Gong et al. [23] demonstrated that large language models (LLMs) can serve as effective sign language translators, highlighting the potential of multimodal reasoning. Despite these advancements, static sign recognition—an essential component for real-time interactive systems—remains a critical challenge. Convolutional Neural Networks (CNNs) have been the dominant approach for static gesture classification due to their ability to extract discriminative features. For

example, Mureed et al. [1] implemented a CNN-based system for Auslan sign recognition, reporting a validation accuracy of 95%. While this demonstrates the viability of CNNs, end-to-end models often plateau in performance and may struggle to generalize across variations in hand shapes, lighting conditions, or backgrounds. To overcome these limitations, hybrid frameworks combining deep feature extraction with classical machine learning classifiers have been explored in recent studies. Building on this motivation, our work proposes a CNN + Random Forest (RF) hybrid pipeline for static Auslan digit recognition (classes 0–2). The proposed system applies Canny edge detection during preprocessing to emphasize contours, followed by CNN-based feature extraction. These features are then classified using both Support Vector Machines (SVMs) and RF classifiers, with the CNN + RF model achieving a peak validation accuracy of 99.75%, significantly outperforming the baseline CNN (~95%) [1]. Furthermore, the pipeline was integrated into a Mediapipe-based real-time recognition system, demonstrating its practical applicability in assistive technologies.

The key contributions of this paper are as follows:

1. We propose a hybrid CNN + classical classifier pipeline for static Auslan digit recognition.
2. We demonstrate that the hybrid approach achieves state-of-the-art accuracy (99.75%), surpassing the CNN baseline [1].
3. We validate real-time applicability by deploying the pipeline in a Mediapipe-based interactive setup.

Related Work

A literature survey of the existing methods is explicated in this section.

Machine Learning

Before the rise of deep learning, traditional machine learning methods were widely used for sign language recognition. These approaches relied on handcrafted features extracted from images or videos, which were then classified using conventional algorithms. While effective for small-scale datasets, their performance often degraded on larger vocabularies or more complex gestures.

Support Vector Machine (SVM): Support Vector Machines (SVMs) have been widely adopted due to their strong performance in high-dimensional feature spaces. Early studies demonstrated that SVMs, when combined with handcrafted descriptors such as Histograms of Oriented Gradients (HOG), achieved competitive accuracy in static gesture recognition tasks. More recently, Kalandar and Dworakowski **[32]** used wearable flex sensors for dynamic sign language recognition and reported 99% accuracy with SVMs. Similarly, Alnujaim et al. **[35]** provided a comprehensive review of Arabic Sign Language systems, highlighting SVM as one of the most reliable classifiers with accuracies reaching up to 99% in fingerspelling and around 96% in dynamic gesture recognition. These results confirm that SVM remains a competitive option for sign language recognition tasks, especially in resource-constrained setups.

K-Nearest Neighbor (K-NN): K-NN classifiers are popular for their simplicity and ease of implementation. Early studies applied K-NN on geometric hand features, obtaining reasonable accuracy on small datasets. With the advancement of sensing technologies, newer works have tested K-NN on larger and more complex inputs. For instance, Kalandar and Dworakowski **[32]** demonstrated that K-NN achieved around 98% accuracy for wearable-sensor-based sign language recognition. In the context of Arabic Sign Language, Alnujaim et al. **[35]** also reported promising results with K-NN, although its performance was generally lower than that of SVM. Despite these successes, K-NN is less favoured in modern SLR pipelines due to scalability issues and its reliance on distance metrics, which make it sensitive to intra-class variations.

Random Forest (RF): Random Forests (RFs) are ensemble methods that combine multiple decision trees, improving robustness against noise and overfitting. RF has been successfully used in various sign language recognition studies. Kalandar and Dworakowski [32] reported 99%

accuracy for dynamic ASL word recognition using RF, highlighting its effectiveness compared to K-NN. Nagesh et al. [33] also compared RF with SVM and K-NN for gesture recognition using sensor-based inputs (flex, accelerometer, and gyroscope data), finding that RF delivered the best performance with 98.5% accuracy. Furthermore, Ronchetti et al. [34] applied RF as a baseline method for Argentinian Sign Language handshape recognition, showing competitive performance against novel classifiers such as ProbSom. These studies demonstrate that RF remains a strong candidate for both vision-based and sensor-based SLR, particularly when combined with deep learning feature extractors. In our work, we leverage RFs alongside CNN-extracted features, outperforming CNN-only baselines [1].

Deep Learning: The advent of deep learning transformed sign language recognition by eliminating the need for handcrafted features and enabling end-to-end learning from raw data. Deep architectures capture complex spatial and temporal dependencies, making them suitable for both static and continuous sign recognition.

Long Short-Term Memory (LSTM): LSTM networks have been widely applied to model temporal dynamics in continuous sign language recognition. Recent studies, such as Aloysius et al. [18], proposed ConSignformer, an adaptation of the Conformer architecture for continuous recognition, combining attention mechanisms with recurrent layers. Similarly, Zuo et al. [26] investigated online continuous SLR pipelines integrating RNNs and transformers for real-time performance.

Convolutional Neural Network (CNN): CNNs remain the most commonly used architecture for static sign recognition due to their powerful feature extraction capabilities. For example, Mureed et al. [1] implemented a CNN-based Auslan recognition model, achieving 95% validation accuracy. While effective, these results highlight the limitations of end-to-end CNN models. Other studies, such as Sandoval-Castaneda et al. [12], applied self-supervised video transformers to enhance isolated sign recognition by combining CNN features with transformer layers, reinforcing CNNs as a foundation for hybrid and multimodal approaches.

Gated Recurrent Unit (GRU): GRUs, a simplified alternative to LSTMs, have also been explored in sequence modeling. Their reduced parameterization enables computational efficiency while capturing temporal patterns. Continuous SLR studies often employ GRUs alongside CNNs or attention modules to balance performance and efficiency [18]. Although less common than LSTMs, GRUs are gaining interest in lightweight, real-time recognition models.

Proposed Methods

This study proposes a hybrid framework for static Auslan digit recognition, designed to leverage the powerful feature extraction capabilities of Convolutional Neural Networks (CNNs) with the classification strengths of traditional machine learning models. We hypothesize that while CNNs excel at learning discriminative features, decoupling the feature extraction and classification phases can enhance robustness and generalization performance. The proposed system employs a CNN as a dedicated feature extractor, the output of which is fed into either a Support Vector Machine (SVM) or a Random Forest (RF) classifier. The following subs detail the architectural pipeline and methodology common to both hybrid models. The dataset used for training and evaluation is described in Section 4.1. The block diagram of proposed method is displayed in figure 1.

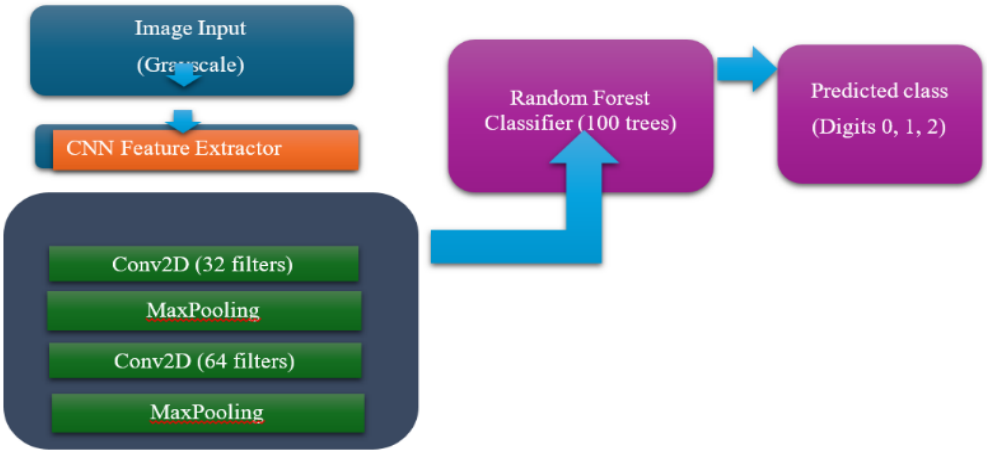


Figure 1: Structure of CNN + RF

CNN + SVM Hybrid Model

In this experiment, a hybrid CNN–SVM pipeline was implemented to evaluate the effectiveness of combining deep feature extraction with a classical machine learning classifier. The workflow consisted of the following steps:

Preprocessing: All input images were converted to grayscale and resized to 64×64 pixels to optimize computational efficiency for the subsequent SVM training. Pixel intensities were normalized to the range [0, 1] and images were reshaped to include a single channel dimension. Canny edge detection was applied to emphasize contour information and suppress background noise.

CNN Feature Extraction: A custom CNN architecture was constructed from multiple convolutional blocks, each comprising a Conv2D layer, Batch Normalization, MaxPooling, and Dropout. The final convolutional output was flattened and fed into a dense layer of 256 neurons, producing a 256-dimensional feature vector for each input image. To establish a strong baseline, this CNN was first trained end-to-end with a softmax output layer, achieving a validation accuracy of approximately 95%, thereby replicating the results of [1]. For the hybrid pipeline, the softmax layer was removed, and the network was truncated after the 256-unit dense layer to function solely as a feature extractor.

SVM Classifier: The 256-dimensional feature embeddings served as input to a Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel. The decision function for the SVM is:

$$\boldsymbol{f(x) = sign(\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b)}$$

where x is the test feature vector, x_i are the support vectors, $y_i \in \{-1, +1\}$ are class labels, α_i are the learned weights, and b is the bias. The kernel function $K(x_i, x)$ defines similarity between samples. In this study, a Radial Basis Function (RBF) kernel was employed:

$$\boldsymbol{K(x_i, x) = exp(-\gamma \parallel x_i - x \parallel^2)}$$

where the parameter γ controls the influence of each support vector, and the regularization parameter C balances the trade-off between margin maximization and misclassification.

Hyperparameters C and γ were optimized using GridSearchCV with 5-fold Stratified Cross-Validation. The best SVM model was retrained on the full training set and evaluated against the test set.

Performance Evaluation: Classification performance was measured using accuracy, precision, recall, and F1-score, allowing direct comparison with both the Random Forest classifier (proposed method) and the baseline CNN [1].

Hybrid CNN + Random Forest (RF) Model:

In the Random Forest experiment, the Convolutional Neural Network (CNN) was not trained end-to-end for classification. Instead, the CNN was employed solely as a feature extractor, while classification was delegated to a Random Forest (RF) ensemble.

Preprocessing: Each image from the Auslan static digit dataset (classes 0–2) was converted to grayscale and resized to 150×150 pixels to provide higher-resolution input for the feature extractor. Canny edge detection was applied to emphasize hand contours and eliminate background noise. Finally, pixel intensities were normalized to the range [0,1].

CNN Feature Extractor: The CNN feature extractor was intentionally kept lightweight to reduce overfitting while retaining discriminative power. It comprised two convolutional layers with 32 and 64 filters, respectively, each followed by max pooling operations. The final feature maps were flattened into a one-dimensional vector representation. These embeddings captured local texture and contour patterns essential for static gesture classification.

Random Forest (RF) Classifier: The extracted feature vectors served as input to a Random Forest classifier with 100 estimators. Random Forest is an ensemble learning method that aggregates multiple decision trees to improve classification robustness and reduce variance. Each decision tree is trained on a random subset of the training data (bootstrapping) and uses a random subset of features at each split, thereby promoting model diversity.

The final class prediction $\hat{y}$ is obtained by majority voting across all trees:

$$\hat{y} = mode\{h1(x), h2(x), \dots, h_T(x)\}$$

where:

- h_t(x) = prediction of t^{th}(x) decision tree,
- T = total number of trees,
- $\hat{y}$ = final predicted class (majority vote).

The probability of class c can also be estimated as:

$$P(y = c|x) = \frac{1}{T}\sum_{t=1}^{T} I\left(h_t(x) = c\right)$$

where I(.) is the indicator function (1 if true, 0 otherwise).

Systems Implementation and Evaluation

This section details the experimental setup used to validate the proposed hybrid models, including the dataset description, baseline for comparison, evaluation metrics, and implementation specifics.

Datasets

The dataset employed in this study is the **AUSLAN Sign Language (Fingerspelling) Dataset**, which is publicly available on Kaggle. It contains more than 71,000 images of Auslan fingerspelling gestures, covering both alphabetic letters and digits. The images were captured under varied but controlled conditions to account for natural variations in hand orientation and background, while

ensuring clarity. A Gaussian blur filter was applied in the original dataset to enhance feature extraction. For this work, a subset of the dataset was selected, consisting of three-digit cl**asses (0, 1, and 2)**. Each class included 2,000 images, resulting in a total of **6,000 grayscale images**. Preprocessing steps included:

- Conversion to grayscale,
- Resizing to the required input dimensions,
- Normalization of pixel values to the range [0,1]
- In the Random Forest experiment, Canny edge detection was additionally applied to highlight hand contours.

This subset was used for both training and evaluation of the proposed hybrid frameworks.

Figure 2: Image of 0, 1, 2 Digit

Baseline Methods

The baseline for comparison in this study was derived from the original Auslan CNN framework, where an end-to-end Convolutional Neural Network (CNN) was trained for classification without any hybridization. In this approach, the CNN directly mapped preprocessed images to class labels through its convolutional, pooling, and fully connected layers, concluding with a softmax output layer. This baseline model achieved a validation accuracy of approximately 95% [1], as reported in prior work, and served as a reference point for assessing the effectiveness of hybrid approaches. The primary objective of introducing Support Vector Machine (SVM) and Random Forest (RF) classifiers was to determine whether CNN-based deep feature extraction, when coupled with classical ensemble or margin-based classifiers, could surpass the performance of a purely end-to-end CNN.

Evaluation Metrics

To comprehensively evaluate model performance, four metrics were considered: accuracy, precision, recall, and F-score. These metrics allow for assessment not only of overall classification success but also of the model's ability to correctly identify individual classes while minimising false predictions. Since the dataset used in this study is balanced across three gesture classes (0, 1, 2), both macro- and micro-averages were considered to ensure fair performance reporting.

Accuracy: Accuracy measures the proportion of correctly classified samples out of the total number of samples. It provides a general performance indicator, although in highly imbalanced datasets it can be misleading. In this study, because each class has the same number of samples, accuracy remains a reliable overall metric.

$$\textbf{Accuracy} = \frac{\textit{Number of Correct Predictions}}{\textit{Total Number of Predictions}} = \frac{\sum_{k=1}^{k} TP_k}{N}$$

Precision: Precision quantifies how many of the predicted positive instances for a given class are actually correct. High precision corresponds to fewer false positives, meaning the model is more confident and selective in its predictions.

$$\textbf{Precision}_k = \frac{TP_k}{TP_k + FP_k}$$

Recall: Recall (or sensitivity) measures the proportion of actual positive samples that were correctly identified by the model. High recall reduces the number of missed cases (false negatives).

$$\textbf{Recall}_k = \frac{TP_k}{TP_k + FN_k}$$

Macro-averaged

$$\textbf{Recall}_{macro} = \frac{1}{K}\sum_{K=1}^{K} \textbf{Recall}_k$$

Micro-averaged recall

$$\textbf{Recall}_{micro} = \frac{\sum_{k=1}^{k} TP_k}{\sum_{k=1}^{k}(TP_k + FN_k)}$$

F Score: The F-score is the harmonic mean of precision and recall, providing a balanced measure between the two. It is especially useful when both false positives and false negatives need to be minimised.

$$F1_k = \frac{2 \times \textbf{Precision}_k \times \textbf{Recall}_k}{\textbf{Precision}_k + \textbf{Recall}_k}$$

Macro F1 averages across classes:

$$F1_{macro} = \frac{1}{K}\sum_{K=1}^{K} F1_k$$

Micro F1 aggregates counts before computing the harmonic mean:

$$F1_{micro} = \frac{2\sum_{k=1}^{k} TP_k}{2\sum_{k=1}^{k} TP_k + \sum_{k=1}^{k}(FP_k + FN_k)}$$

Experiment Details

The experiments were carried out in a GPU-enabled Google Colab environment (Tesla T4) using Python 3.9. Data preprocessing (grayscale conversion, resizing, normalization, and Canny edge detection) was performed using OpenCV, while model development employed TensorFlow/Keras for CNN construction and training. Classical classifiers, including SVM and Random Forest, along with hyperparameter tuning, were implemented using Scikit-learn. Dataset partitioning, pre-processing steps, and model configurations followed the methodologies described in earlier sections. All models were evaluated on the held-out test set using standard classification metrics to ensure consistent and fair comparison.

Results and Discussion

The performance results of our proposed model are presented in this section. The results were compared with the previously introduced methods, which were tested on the same datasets.

Quantitative Results

The performance of the baseline CNN, CNN–SVM, and CNN–Random Forest models was evaluated in terms of training and validation accuracy. The performance of all models is quantitatively summarized in Table 1.

Figure 3: Performance Comparison of Models

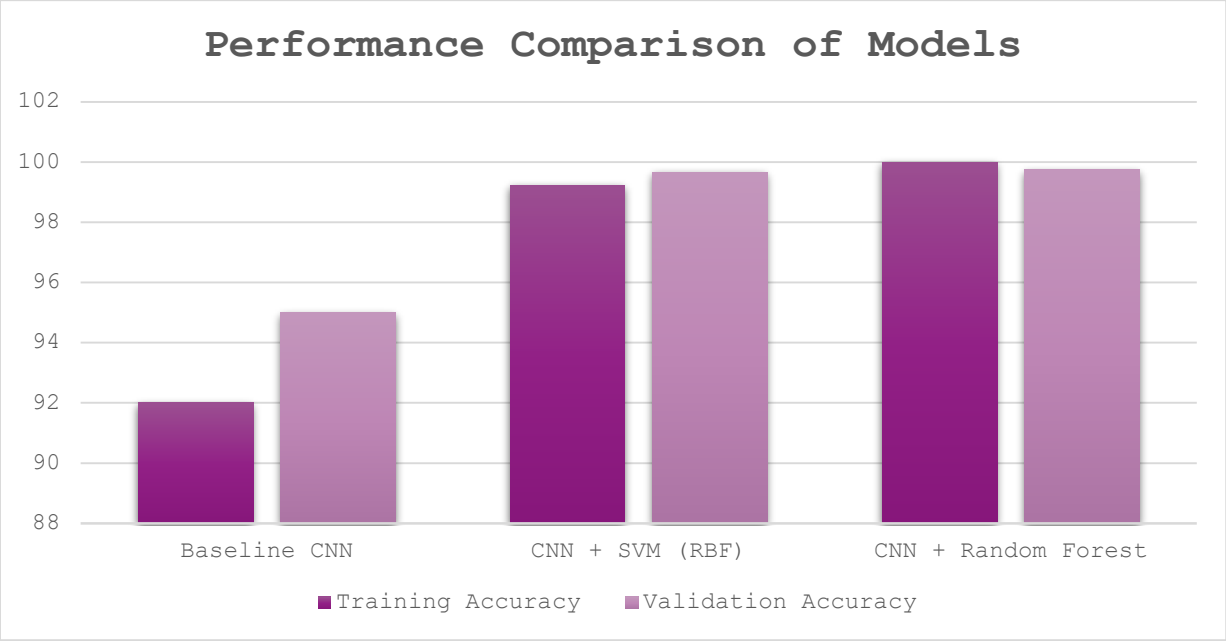


Table 1: Comparison of three approaches

Model	Image Size	Batch Size	Training Accuracy	Validation Accuracy
Baseline CNN	**150x150**	**36**	**92.00%**	**95.00%**
CNN + SVM (RBF)	**64×64**	**32**	**99.23%**	**99.67%**
CNN + Random Forest	**150×150**	**– (tree-based)**	**100.00%**	**99.75%**

Observations

- The baseline CNN achieved a validation accuracy of 95%, which establishes a strong benchmark but shows limitations in handling subtle gesture variations.

- The CNN + SVM hybrid improved performance substantially, achieving 99.67% validation accuracy. This indicates that SVM effectively separates high-dimensional CNN features using the RBF kernel. However, a slight gap between training (99.23%) and validation (99.67%) accuracy suggests mild overfitting tendencies.
- The CNN + Random Forest model performed the best, achieving 100% training accuracy and 99.75% validation accuracy, indicating robust generalization. RF benefitted from CNN-extracted feature embeddings and leveraged ensemble averaging to minimize misclassifications.
- The nearly perfect accuracy of CNN+RF highlights its suitability for static gesture recognition, though further testing on larger and more diverse datasets would be necessary to confirm scalability.

Qualitative Analysis

Confusion matrices and classification reports were generated for both CNN+SVM and CNN+RF models. Both models demonstrated strong class-wise recognition with minimal false positives and false negatives. The Random Forest in particular showed highly stable predictions across all gesture classes. Additionally, visual inspection of test samples confirmed that Canny edge preprocessing effectively enhanced the distinction between digit gestures by emphasizing contour features. This was especially beneficial for SVM and RF, both of which rely on clear feature boundaries.

Figure 4: Confusion matrix of the CNN+SVM model showing classification results across digit classes.

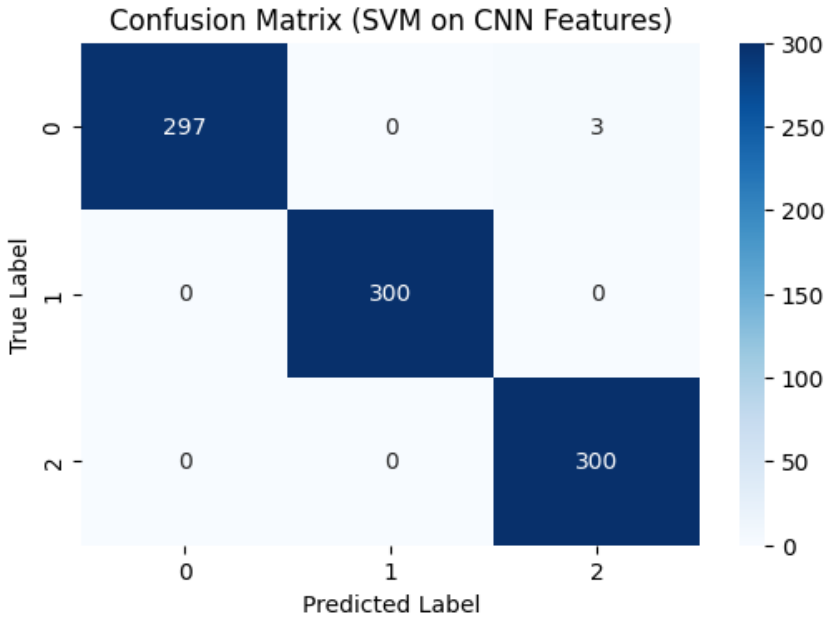


Figure 5: Confusion matrix of the CNN + Random Forest model showing classification results across digit classes.

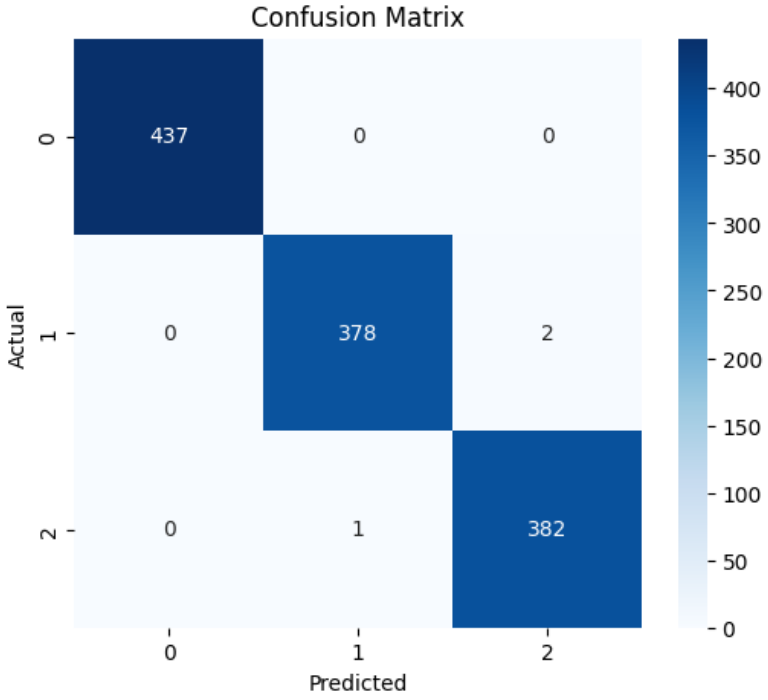
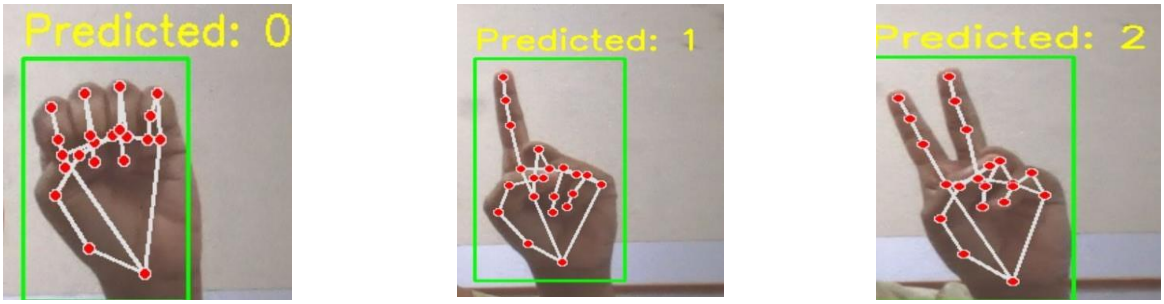


Figure 6: Image of correctly classified digit gestures (0, 1, and 2) from the test set using the proposed hybrid models.

CNN + SVM

CNN + Random Forest



Conclusions and Future Work

This study proposed a hybrid framework for Auslan sign language digit recognition by combining CNN-based feature extraction with classical classifiers (SVM and Random Forest). The results demonstrated that hybridization significantly outperformed the baseline CNN, with CNN+SVM

achieving 99.67% accuracy and CNN+RF achieving 99.75%, compared to the baseline CNN's 95%. Random Forest in particular exhibited highly stable predictions across all gesture classes. These findings confirm that coupling deep feature representations with ensemble learning enhances robustness and generalization for static gesture recognition tasks. The current experiments were limited to three-digit classes (0–2) from the Auslan dataset, future extensions will focus on:

- Expanding the framework to cover the full Auslan alphabet and number set.
- Incorporating dynamic gesture sequences using temporal models (e.g., LSTM, GRU, or Transformers).
- Evaluating performance on larger and more diverse datasets to test scalability.
- Investigating model compression and lightweight architectures for real-time deployment on embedded systems.

References

[1] M. Mureed, M. Atif, and F. A. Abbasi, "Character recognition of Auslan sign language using neural network," *Int. J. Artif. Intell. Math. Sci.*, vol. 2, no. 1, pp. 29–36, 2023.

[2] K. A. Hafez, M. Massoud, T. Menegotti, J. Tannous, and S. Wedge, "American sign language recognition using a multimodal transformer network," in *Proc. IEEE Can. Conf. Electr. Comput. Eng.*, 2024, pp. 654–658.

[3] C. Kakade, N. Kadam, V. Kaira, and R. Kewalya, "Enhancing sign language interpretation with multi-headed CNN, hand landmarks and large language model (LLM)," in *Proc. IEEE Int. Conf. Future Mach. Learn. Data Sci.*, 2024, pp. 527–532.

[4] J. Pu, W. Zhou, and H. Li, "Iterative alignment network for continuous sign language recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4165–4174.

[5] J. Sharma and J. Lande, "Hand gesture recognition using EfficientNetB5: A robust approach for real-time human-computer interaction," in *Proc. Int. Conf. Commun. Electron. Syst.*, 2024, pp. 1519–1523.

[6] S. Siddharth, J. R. Jayashree, and K. N. Shwetha, "Indian sign language recognition system," in *Proc. Int. Conf. Soft Comput. Mach. Intell.*, 2024, pp. 364–367.

[7] A. Tripathi, S. Semwal, S. Makhloga, V. Tomar, and S. Singh, "SLRMPCMC: Sign language recognition using Mediapipe and cross-model comparison," in *Proc. Int. Conf. Electr., Electron. Comput. Technol.*, 2024.

[8] P. Jeevanandham, G. B. A, K. G. Ms, and A. Hariharan, "Real-time hand sign language translation: Text and speech conversion," in *Proc. Int. Conf. Circuit Power Comput. Technol.*, 2024.

[9] X. Xu and J. Fu, "A two-stage sign language recognition method focusing on the semantic features of label text," in *Proc. CSI Int. Symp. Artif. Intell. Signal Process.*, 2024.

[10] Y. Matveyas et al., "Research and development of sign language recognition system using neural network algorithm," in *Proc. IEEE Int. Conf. Smart Inf. Syst. Technol.*, 2024.

[11] P. Edward, B. Sameeh, and W. Alexan, "Comparative study between CNN and LSTM approaches for sign language recognition," in *Proc. Novel Intell. Leading Emerg. Sci. Conf.*, 2024, pp. 220–222.

[12] M. Sandoval-Castaneda et al., "Self-supervised video transformers for isolated sign language recognition," *arXiv:2309.02450*, 2023.

[13] A. Yin et al., "Gloss attention for gloss-free sign language translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023.

[14] B. Zhou et al., "Gloss-free SLT: Improving from visual-language pretraining," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023.

[15] L. Hu et al., "CorrNet: Continuous sign language recognition with correlation network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023.

[16] T. Lee et al., "Human part-wise 3D motion context learning for sign language recognition," *arXiv:2308.09305*, 2023.

[17] R. Rastgoo et al., "A transformer model for boundary detection in continuous sign language," *arXiv:2402.14720*, 2024.
[18] N. Aloysius et al., "ConSignformer: Adapting conformer for continuous sign language recognition," *arXiv:2405.12018*, 2024.
[19] C. Raude et al., "A tale of two languages: Large-vocabulary CSLR and retrieval (CSLR2)," *arXiv:2405.10266*, 2024.
[20] P. Zhang et al., "EvSign: Sign language recognition and translation with event cameras," *arXiv:2407.12593*, 2024.
[21] H. Ranjbar and A. Taheri, "Continuous sign language recognition using intra-inter gloss attention," *arXiv:2406.18333*, 2024.
[22] X. Shen et al., "MM-WLAuslan: A multi-view multi-modal dataset for word-level Auslan sign language recognition," in *Adv. Neural Inf. Process. Syst.*, 2024.
[23] J. Gong et al., "Large language models are good sign language translators," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024.
[24] B. Zhang et al., "Scaling sign language translation," *arXiv:2407.11855*, 2024.
[25] L. Hu et al., "CorrNet+: Sign language recognition and translation via spatial-temporal correlation," *arXiv:2404.11111*, 2024.
[26] R. Zuo et al., "Towards online sign language recognition and translation," *arXiv:2401.05336*, 2024.
[27] N. Aloysius et al., "A comparative study of continuous sign language recognition methods," *arXiv:2406.12369*, 2024.
[28] W. Zhou et al., "Scaling up multimodal pre-training for sign language understanding," *arXiv:2408.08544*, 2024.
[29] Y. Chen et al., "SignAvatars: Real-time sign language avatars via neural motion diffusion," in *Proc. ACM Multimedia*, 2024.
[30] R. S. Abdul Ameer, M. A. Ahmed, Z. T. Al-Qaysi, M. M. Salih, and M. L. Shuwandy, "Empowering communication: A deep learning framework for Arabic sign language recognition with an attention mechanism," *Computers*, vol. 13, no. 6, p. 153, 2024.
[31] S. M. Mahalingam, N. S. Kumar, C. Harika, C. S. Reddy, and D. P. Kalyan, "Sign to text: Automated sign language interpretation using LSTM and computer vision," in *Proc. Int. Conf. Adv. Comput. Renew. Syst.*, 2024.
[32] B. Kalandar and Z. Dworakowski, "Sign language conversation interpretation using wearable sensors and machine learning," *arXiv:2312.11903*, 2023.
[33] S. Nagesh, S. R. Kannan, and M. Ramachandran, "Performance of classifier for gesture recognition using machine learning techniques," in *Advances in Computational Intelligence and Communication*. Springer, 2024, pp. 141–151.
[34] R. Ronchetti, C. Mansilla, and L. Lanzarini, "Handshape recognition for Argentinian sign language using ProbSom," *arXiv:2310.17427*, 2023.
[35] H. I. Alnujaim et al., "Arabic sign language recognition systems: A systematic literature review," *Sensors*, vol. 24, no. 23, p. 7798, 2024.