Journal of Statistics & Management Systems
ISSN 0972-0510 (Print), ISSN 2169-0014 (Online)
Vol. 29 (2026), No. 5, pp. 529–552
DOI : 10.47974/JSMS-1584

Hand gesture recognition using hybrid deep learning models for deaf communication support

Gautam Govind [§]
Dhruv Gupta [†]
Kavita Jhajharia *
Department of Information Technology
Manipal University Jaipur
Jaipur 303007
Rajasthan
India

Abstract

Hand gesture recognition has become an indispensable part of human-computer interaction. It supports intuitive, contactless, and accessible means of communication. This work proposes a robust deep learning-based system for the real-time static recognition of hand gestures to assist communication with the deaf and hard-of-hearing. We propose hybrid architecture that consists of the Swin Transformer for contextual attention, ResNet34 for spatial feature extraction, and a BiLSTM layer for temporal understanding. The model was trained and evaluated using a subset of the HaGRID dataset, consisting of 18 different classes of hand gestures. It achieves a training accuracy rate of 99.3% and a testing accuracy rate of 98.03%, thereby demonstrating its excellence in performance amid different lighting and backgrounds. Our approach proposes a scalable solution that can be integrated into assistive technology for gesture-to-text or gesture-to-speech conversion, thus bridging the communication gap for deaf individuals.

Subject Classification: *68T07, 68T10.*

Keywords: *Hand gesture recognition, Swin transformer, ResNet34, Deep learning, Human-computer interaction.*

[§] *E-mail:* govindgautam933@gmail.com
[†] *E-mail:* dhruvagupta0706@gmail.com
* *E-mail:* Kavita.jhajharia@jaipur.manipal.edu (Corresponding Author)

Introduction

Human-computer interaction (HCI) has traversed a long journey, starting from the initial days of punch cards and command-line interfaces, the rise of graphical user interfaces, and now, to more intuitive and natural modalities comprising speech, eye tracking, and gesture recognition. Among these, hand gestures become a free communication channel that is extremely powerful and universally understood. Gestures provide a robust interface under awkward conditions such as loud environments or accents and ensure that people with disabilities involving speech or hearing are included. Day-to-day communication using hand gestures, for example, sign language—forms the basis of interaction for the deaf and hard-of-hearing communities. That makes this domain a crucial focus area of accessible technology.

Traditionally, hand gesture recognition systems depend mainly on handcrafted features such as color histogram, contour, background subtraction, or motion tracking. While these approaches provided some basis in the early days, they proved extremely sensitive to changes in illumination, skin colors, hand orientations, occlusions, and environmental noise. Because of this, they were not scalable for real-world, unconstrained conditions. Deep learning came as a paradigm shift wherein models are enabled to learn hierarchical and spatially invariant features directly from data. Consequently, some convolutional neural networks (CNNs) such as AlexNet, VGG, and ResNet presented significant breakthroughs in image classification and object detection, making way for hand gesture recognition in practice. Yet, CNNs tend to ignore useful global contexts when discerning between similar gesture classes, being largely focused on local spatial patterns.

The ushering of transformer-based architectures such as the Vision Transformer (ViT) and Swin transformer has opened new vistas in visual understanding through the form of attention mechanisms. Long-range dependencies and spatial relationships are coupled with such models to act upon the drawbacks of global reasoning in CNNs. The temporal learning approaches with BiLSTM (Bidirectional Long Short-Term Memory) are able to capture the sequence dynamics of input features even in static image classification and thus offer a new dimension to the hybrid architectures, which ultimately strengthen gesture recognition.

As far as the real world is considered, assistive technology systems are as concerned for latency and computational efficiency as for other algorithms for models. Models must make a prediction in real-time under

different circumstances in a resource-constrained environment. This research focuses on developing a real-time, available, and efficient hand gesture recognition system configured to assist the deaf community in translating gestures into interpretable commands or text.

To develop a deep learning-based hand gesture recognition system that can overcome both the technical and social challenges of interpreting gestures. The system is able to identify 18 unique static hand gestures using a hybrid neural network architecture. It should be able to work reliably within the three-dimensional world where lighting, background textures, and hand positions can differ significantly from one another. Furthermore, the solution is designed for a real-time application so as to incur low latency during inference but at the same time for high classification accuracy. The intent behind it is not only to achieve prowess performance in well-prepared datasets but also to create a system that would generalize well to the unseen conditions of live video streams.

Another important goal is to enhance accessibility for the deaf-and-hard-of-hearing population by creating a harmony between gesture-based communication and digital systems. The feasibility of the proposed system on consumer-grade hardware is assured through computationally efficient and highly accurate models. OpenCV is used for video capture and display, allowing real-time performance, which is made possible by the optimized inference pipelines for deploying these works on edge devices, thus ensuring an even greater degree of outreach for this technology.

1.3 *Overview of the Proposed System*

The suggested system is one designed as a hybrid deep learning model that makes use of the most advantageous factors from different learning paradigms. The main architectural component consists of a Swin Transformer, a hierarchical attention-based mechanism that captures the global spatial patterns and context from input images quite well. Complementing the Swin Transformer is the ResNet34 backbone, one of the most well-known CNNs that leverage the use of residual connections to widely extract fine-grained local features from images. The outputs of both models are concatenated to result in a rich and diverse feature representation.

The classification accuracy is augmented, and relationships among spatial and channel-wise features are captured by passing the concatenated vector through a bidirectional LSTM. In turn, not only are the input frames LSTM-propaganated static gesture images, but the LSTM also looks at

modeling contextual dependencies across feature vectors, leading to a generalized and discriminative network. Final classification is done in the fully connected layer, which provides output probabilities for 18 classes of gestures.

OpenCV is used for live video input via the webcam and for real-time gesture detection and visualization. Each frame is pre-processed and fed to the trained model, with predicted gestures being overlaid onto the video stream as feedback. The entire system is trained on the HaGRID dataset-a large high-resolution gesture dataset with diverse samples under varying conditions. Model performance is evaluated using standard measures, such as precision, recall, F1-score, and the confusion matrix.

1.4 *Significance and Contributions*

Its broad strokes concerning contributions to the domain of gesture recognition and assistive technological devices can be summed into three: it presents a wholly different hybrid architecture involving CNN and Transformer models alongside a BiLSTM to equip the model for enhanced spatial and contextual learning; this goes to prove that single-model approaches might be limiting on the whole and high in its response when working on real-world challenging datasets. The project aims at real-time usability using OpenCV to build a system working on consumer hardware; thus, its applicability is broadened to practical, deployable uses outside proof-of-concept in lab settings.

The system will be developed entirely with accessibility in mind for the deaf and hard-of-hearing people. Such a system would start to revolutionize communication interfaces and learning aides and open the gates for future action in smart environment control through seamless translation of hand gesture into machine-readable format. Very simply put, all this would bring technical innovation into the fold of social impact: a mouthful bringing this evolution in human-computer interaction.

Literature Review

In their 2012 paper, Rautaray [1] proposed a real-time hand gesture recognition system for dynamic applications in virtual environments. The system uses computer vision methods like Haar-like features for gesture classification and CAMShift for hand tracking. Using seven preset static hand gestures, it allows users to interact with virtual objects through natural hand motions, replacing the traditional mouse. It is implemented

in OpenCV and OpenGL with a focus on inexpensive hardware, such as a simple webcam. The authors note the costs and ease of operation of the system, particularly for disability use, while stressing that the system needs to be more robust in noisy or dim-light conditions.

Dhuzuki et al. [2] invented a hand gesture recognition system based on deep learning in 2025 for Rehabilitation Internet-of-things (RIoT) scenarios. Patients can do hand rehabilitation exercises in the convenience of their own homes while therapists can monitor the patients' progress without having them physically present with the system that combines MediaPipe and web assembly (WASM) to give real-time feedback. Their method uses pre-trained deep learning models with adaptive algorithms that put emphasis on high accuracies, low latencies, and affordability. Further research would focus on integrating CNNs and RNNs via network linking to enhance classification performance as initial results showed potential for stroke rehabilitation applications.

Singh and Singh's [3] DyHand system offers a robust solution for dynamic hand gesture identification based on BiLSTM and soft attention mechanisms. Unlike traditional approaches that rely on hand-crafted features, DyHand uses a dynamic skeleton representation to account for gesture variability within and across classes. In experiments conducted on standard datasets DHG-14/28 and SHREC'17, the model achieved recognition accuracies of 97.14% and 96.42% for 14 and 28 gesture classes, respectively. With the inclusion of temporal sequence learning and attention mechanisms, the model demonstrates promise for real-time human-computer interaction applications concerning intricate hand motions.

Methodology of detection of hand gestures from visual input is presented in a broad fashion by Linardakis, Varlamis, and Papadopoulos [4], describing the fast-moving field and introduces new challenges while providing a wide and current perspective. This includes classification into input modality (RGB, depth, and multi-view video) and implications of gesture classification as well as three-dimensional (3D) hand pose estimation with respect to those modalities. Highlighted by the authors are outstanding challenges such occlusion handling, generalization among users, and computing efficiency for real-time systems, aside from summarizing worthy datasets and their applications. A good reference material for researchers who want to know what is happening at the current trends and what next would be this study in gesture detection through visual data.

By employing extensive RGB image datasets, the researcher Le [5] has developed a deep learning model based on cross-evaluation to contend with the many challenges of real-world hand gesture recognition (HGR). The proposed framework takes the recently published TQU-HG dataset that consists of 60,000 RGB images from 15 gestures filmed under difficult conditions of low light and low resolution for examining state-of-the-art CNN and transformer-based models. In this research, the robust comparisons of models such as ResNext50 and YOLO-NAS are made with parameter optimization on both TQU-HG and HaGRID datasets. The superior accuracy performance exceeding 98% with multiple cross-tested models on common gesture labels lends credence to the applicability of the framework to real-world HGR systems. The study also highlights the salient role of cross-dataset generalization of DL-based HGR in expanding the horizon of publicly available datasets.

Conventional hand gesture recognition (HGR) approaches based on closed-set assumptions have the following shortcomings, which are recognized by Zhou, Xu, and Cheng [6] design of an open-set recognition framework that can tackle unconstrained views, unknown gestures, and new unseen hand forms. Their framework includes a joint-weighted classification algorithm, which enhances few-shot incremental learning through an improved cosine similarity metric, and a viewpoint effect elimination network, which extracts view-invariant features. The scientists subsequently presented the Open Hand Gesture (OHG) dataset with multiple gesture classes and viewing angles to further endorse the real-world applicability. Their experimental results set new benchmarks for flexible and adaptive HGR systems by exhibiting better performance on open-set and few-shot learning paradigms.

To improve the real-time interpretation of sign language, Suthar and Ali [7] examined integrating vision-based hand gesture detection with Arduino Leonardo microcontrollers. Their approach indicates the capacity of accelerometer sensors built into the Arduino to monitor dynamic movements of the right hand and communicate that data to a computer for processing. Then, a deep learning network interprets this data to recognize specific motions in terms of words. The study demonstrates how the Arduino embedded system may affordably and accessibly support the sign language communication medium when used with low-cost and lightweight hardware setups. Therefore, such an approach holds great promise for providing uninterrupted communication for the speech- and hearing-impaired in everyday situations.

The CNN4-M and sEMG hand gesture detection algorithms by Chen et al. [8] take raw numerical greyscale (RNG) images of sEMG signals directly as input for classification. Instead of using conventional time-frequency transformations such as STFT or wavelet transformations, they changed sEMG voltage data into single-channel grey-scale images by applying linear and exponential operations. Their method had a remarkable increase in accuracy as compared to traditional machine learning and deep learning on publicly accessible and self-collected datasets, while preserving the texture of the original signal. Achieving exceptional recognition accuracies of 98.03%, 99.95%, and 98.07%, the architecture of CNN4-M trained on these RNG images proved its robustness and suitability for sEMG-based gesture classification for practical applications such as control of prosthetics and rehabilitation.

The authors present a deep-learning system based on 3D Convolutional Neural Networks (3DCNNs) designed for robust hand gesture detection for sign language interpretation. The system utilizes RGB data to address both temporal and spatial dependencies of video-based gesture inputs for real-world application scenarios without the need for wearing technology and additional sensors. Transfer learning was adopted for optimal performance with labeled data, which were scarce in some applications. The method was tested on three different datasets with the recognition accuracies being 98.12% for signer-dependent mode and 84.38% for signer-independent mode. A parallel 3DCNN structure was then employed to allow for even more extensive capture of temporal variations occurring across several segments of video input with `fusion techniques´ that include Multilayer Perceptron, LSTM, and autoencoder. This study has shown that region-based preprocessing along with spatiotemporal modeling has been able to deliver comparable or even higher accuracy than some state-of-the-art methods while maintaining computational economy [9].

With the rise of robots in everyday life, Qi et al. [10] explored how computer vision-based hand gesture recognition can enhance natural human-robot interaction. Their review focuses on systems using monocular and RGB-D cameras, covering all key step-gesture detection, segmentation, feature extraction, and classification. The study evaluates existing methods, outlines algorithmic approaches, and discusses current challenges such as lighting variations and real-time responsiveness. The authors emphasize the potential of gesture-based communication to bridge the gap between humans and machines in intuitive ways.

Real-time gesture recognition has significantly advanced with deep learning techniques like YOLOv3, which offers fast and accurate detection without the need for heavy preprocessing. A model based on YOLOv3 and DarkNet-53 has demonstrated over 97% accuracy in classifying hand gestures, performing effectively even in low-resolution and complex backgrounds. Its superiority over traditional models like SSD and VGG16 highlights its potential for practical deployments in assistive technologies and human-computer interaction [11].

Deep learning models for hand gesture recognition could also work well with electromyography-(EMG)-based datasets, after first converting them into spectrograms via the Short-Time Fourier Transform (STFT). In recognition of this, a 50-layer ResNet CNN achieved an extraordinary test accuracy of 99.59% and an F1-score of 99.57% on recognition from spectrogram images produced from the 4-channel sEMG signals on seven gestures. This technique well demonstrates the utility of this syncretism of deep residual networks and frequency-domain transformation for high-precision gesture classification [12].

Materials and Methodology

2. Dataset Preparation

2.1 *Dataset Collection*

In this research, datasets were derived from the HaGRID (Hand Gesture Recognition Image Dataset), a large-scale publicly available dataset for real-world hand gesture recognition tasks. Comprising more than 550,000 high-resolution images, each image is annotated with gesture labels, bounding boxes, and metadata. The dataset was collected with diversity in mind: gestures performed by various individuals of different ages, genders, and backgrounds, under diverse lighting conditions, backgrounds, and camera angles. This very rich diversity qualifies HaGRID as the most potent dataset for training models that are expected to generalize well in the presence of real-world variability [13].

Instead, the most adopted gestures, i.e. call, fist, ok, thumbs-up, peace, stop, and their inverted counterparts, depict a great variety of hand motions and orientations. Each class denotes a distinct gesture; thus, this dataset bears a very wide spectrum of communicative and culturally relevant hand signs. This variety will allow for a more robust gesture recognition system, extremely useful for assistive applications, and

human-computer interaction where the accuracy of the system with respect to different user input will be pertinent.

Gesture recognition models depend heavily on high-quality shall be well annotated for effective learning: solving those problems requires high-quality datasets like HaGRID. The dataset gives balanced and sufficient examples across all gesture classes so that problems regarding overfitting would be avoided, and fairness would be promoted. Thus, the model trained on this dataset is relatively good at distinguishing among differences in hand shape, pose, and orientation, meaning that it can robustly perform both fine and gross hand motion recognition.

Besides, the distribution of image samples over the 18 gesture classes are shown in Figure 1, which depicts the balanced nature of the dataset. So, in every gesture class, there are nearly 27,000 to 30,000 samples that do not give any significant class imbalance. This even distribution serves an important purpose during training because it normalizes the process of learning; it prevents the model from being biased by any class and does not let one class dominate the decision boundaries when the model is being defined. Therefore, balance plays a very important part in the overall high classification accuracy across all gestures.

Figure 1

Bar graph showing the number of images per gesture class in the HaGRID dataset. The distribution confirms a well-balanced dataset with nearly equal representation across all 18 gesture categories.

2.2 *Data Annotation Parsing*

Data annotation parsing comprises extracting the required label information from JSON annotation files accompanying the images in the HaGRID dataset. Each of the images in the data set has an assigned unique ID, which can be used to label the image to its corresponding gesture class label. These are the ground truth used to guide the learning process of the model during training. Precisely, the labels for the gesture in the JSON files tell what hand gesture the image represents and make up an important component of the structure of the data set.

2.3 *Model Architecture*

The depicted hand gesture recognition system is a hybridized deep learning scheme that amalgamates the core competencies of three robust architecture that are Swin Transformer, ResNet34, and Bidirectional Long Short-Term Memory (BiLSTM). This multi-stream architecture allows the model to generalize global and local spatial features as well as the temporal relationship adding robustness and accuracy in classifying gestures. Swin Transformer is used for capturing global patterns in terms of hierarchy and context through attention mechanisms while ResNet34 performs local fine-grained extraction using deep convolutional layers and residual connections. The outputs of both feature extractors are concatenated in one high-dimensional feature vector later passed to a BiLSTM layer that learns bidirectionally dependencies in a static image input, thus focusing on the temporal dynamics and contextual continuity.

It is, therefore, possible for this architecture to comprehend not only spatial alterations but also detect minor subtleties in hand shapes, which would have otherwise escaped less sophisticated models. The technique spans from preprocessing to annotation segmentation of the data; through training and testing the model for use in real-time applications. Such a system considers real-life applicability with performance capabilities to perform low-latency inference with OpenCV to ensure scale and relevance to integration with accessibility tools such as gesture-to-text or gesture-to-speech systems. Figure 2 illustrates the complete end-to-end architecture of the proposed hybrid model, highlighting the parallel feature extraction using Swin Transformer and ResNet34 followed by sequential learning through the BiLSTM layer.

2.4 *Preprocessing*

Before the images are passed through the model to train, several preprocessing operations are performed on them so that the data is in the

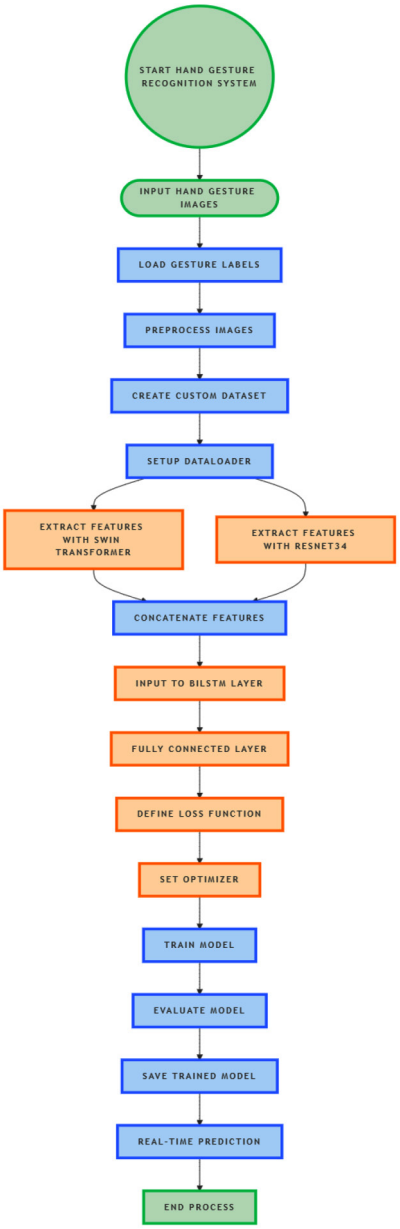

Figure 2

Model architecture combined with Swin Transformer, ResNet34, and BiLSTM for hand gesture recognition.

best format for the model. These preprocessing methods enhance the efficiency of training and the capacity of the model to generalize between different categories of images. The preprocessing operations are as follows:

Image Resizing: All the images are resized to the same resolution of 224×224 pixels. This makes the model input images of the same dimensions, avoiding any problems that could be caused by different image sizes. This is very crucial for deep learning models since they require the same-sized input. 224×224-pixel resolution is a standard resolution used in computer vision operations, providing an optimal trade-off between computational requirements and image detail.

Pixel Normalization: The pixel values of the images are normalized to the interval of [-1, 1]. Normalizing the pixel values ensures that the input features (pixel values) are on the same scale, facilitating quick convergence when training. Models generally learn better when input features are normalized because it negates the possibility of some features over-powering the learning process owing to varying scales.

3. Custom Dataset and DataLoader

There is a custom Dataset class defined in PyTorch to load the images and corresponding labels in an efficient manner during the training time. This class provides an assurance that data is in proper format and can be accessed quickly during the time of training. The data is batched next using a DataLoader, which is used for effective training by parallel loading data and minimizing I/O bottlenecks.

4. Feature Extraction

Feature extraction is an important phase of hand gesture recognition since it allows the model to extract meaningful information from raw image data that will be useful for classification. Two backbone architectures are used for this purpose: Swin Transformer and ResNet34. Both the architectures are such that they extract different types of features from the input images, taking advantage of the different strengths of vision transformers and CNNs.

4.1 *Swin Transformer*

The Swin Transformer is an outstanding vision transformer model that extracts local and global spatial information from input images through hierarchical self-attention techniques. Instead of treating the

complete image as a sequence like the previous transformers, the Swin Transformer parses the image into smaller, non-overlapping windows where each is treated individually. The computational cost of the window-based treatment is much less than that of global self-attention.

Swin Transformer uses a shifting mechanism besides window-based self-attention for creating cross-window connections to depict the broader spatial context more accurately.

Swin transformer hierarchical structure helps the model with multi-scale feature retrieval. Thus, the model can detect both small and large-scale hand patterns, since each layer of the transformer can register information at various resolutions.

4.2 *ResNet34*

Unlike the Swin Transformer, ResNet34 is a modern convolutional neural network (CNN) designed to extract local and spatial features from images. One specialty of the ResNet architecture, which employs the residual connection for achieving that effect, is that it is free of all the downside that attach themselves when training very deep networks because the residual connections allow the network to avoid certain layers and feed the inputs straight down to deeper layers. This condition enables ResNet34 to learn much more discriminative representations-and even more in deeper layers, leaving important information aside-and thus making the performance very competitive.

ResNet34 CNN layers employ convolutional filters to search for a locality featuring edges, textures, and contours. Among the very significant features of ResNet34 is that it uses residual connections, which permit a smooth gradient flow in backpropagation, enabling effective learning with a comparatively deeper model. Hence, ResNet34 has proved to be well-suited for deriving low-level features from images, which can then be used to differentiate various hand movements.

5. Feature Concatenation and Sequence Modeling

By extracting the features through Swin Transformer and ResNet34, they are fused into one feature representation preserving both the local and global information. The two networks' features are concatenated into a single, high-dimensional vector, which, in turn, is fed into the next step of the recognition pipeline: sequence modeling.

This feature vector is then presented to a Bidirectional Long Short-Term Memory (BiLSTM) model. BiLSTM is a type of RNN that sums a

model by utilizing both for mentioned temporal context in its architectural modeling as well as retrospectively past states as inputs to the actual model.

Featuring richer semantics by fusing Swin transform and ResNet34 combined with BiLSTM temporal modeling ability helps the hand gesture recognition systems capture both hands pose and the entire movement sequence for increased classification performance.

6. Loss Function & Optimization

The **CrossEntropy Loss** function is applied in training the model for multi-class classification problems. The loss function is widely known and used for classification problems where the point of interest is to predict one out of many possible classes. This function penalizes predictions increasingly based on the model's confidence in its outcomes, and the model learns to become better in making predictions over a period.

Optimizing this model occurs using the **Adam** optimizer, which is very efficient for deep learning models. Convergence with high speed and robustness is provided because Adam dynamically varies the learning rate using estimates of first and second moments of gradients. The learning rate of 0.001 has been applied to this model because it effectively balances the training speed with that of convergence of the model.

The model is **trained for 10 epochs** within a **batch size of 16** so that the model should learn fairly, but it would remain computationally efficient. A relatively small batch size causes the model to be broad with respect to generalization since it updates its weight more often on smaller subsets of data.

8. Real-Time Prediction

Open CV is being used to do the real-time hand gesture detection after model training and testing. Real-time operation proves essential for real-life applications, interpreted as sign language for the hearing impaired and gesture-input command systems.

The system processes live camera input frame-by-frame where each input is analyzed as one of the predefined hand gestures according to the classification model. For every frame, the model predicts the gesture class, thus providing the real-time output of the identified gesture to the user for instant interaction feedback.

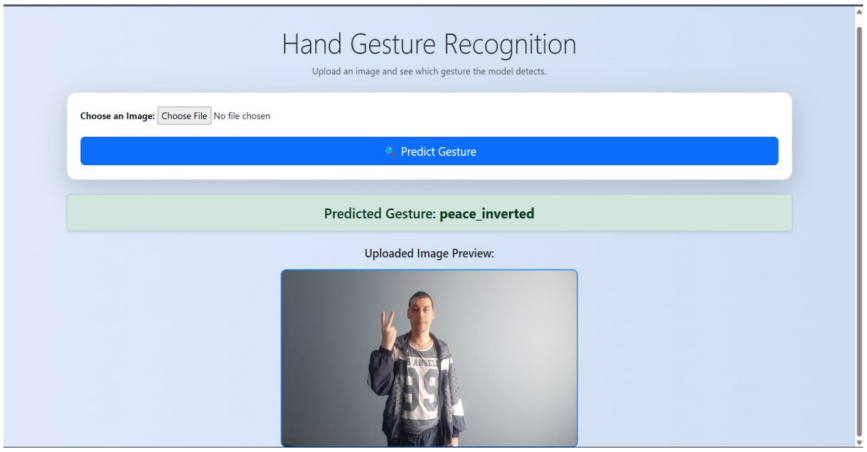

Figure 3

Web interface for image-based hand gesture prediction. The user uploads a static gesture image, and the system displays the uploaded image along with the predicted gesture label using the trained hybrid deep learning model.

9. Web-Based Gesture Prediction Interface

Along with the real-time recognition system, a complementary **web application** was developed, allowing users to upload static images of hand gestures to obtain accurate predictions on the trained hybrid model. Developed in Django with Bootstrap for the web interface, users can easily upload a gesture image and trigger inference from the model to see the predicted class immediately. To clarify things, the prediction output is shown next to the uploaded image. This setup offers a convenient option for webcam recognition, while also making it accessible to mobile and low-end device users. This is especially useful for demonstrations, education, and accessibility testing.

Interface design and functionality shown in Figure 3 presents the prediction output screen, which has a sample gesture image uploaded, and the model predicts successfully.

Results

1. *Evaluation and Performance Metrics*

The range of standard evaluation metrics was assessed after training the model, as an indicator of its effectiveness to generalize regarding unseen data. These metrics provide a clear picture of the overall quality

and credibility of the classifier and are very essential in sensitive applications such as assistive technologies for the deaf and hard-of-hearing. Among the metrics, the most important are **precision**, **recall**, and the **F1-score**, all of which are foundational in evaluating performance for multi-class classification tasks such as hand gesture recognition.

So, **precision** would be the ratio of the true positives to all positives predicted by the model. This really measures how many of those predicted gestures belong to the right class. This becomes crucial particularly when the cost of false positives is high, for instance, on understanding neutral hand position as a command, hence causing unintended interaction. If precision is high, it guarantees that false active events are not likely to be triggered since actions would be wrongly disintegrated from communication.

Recall measures the proportion of true positives among instances that are positive. In essence, it gauges the detection ability of the model concerning all relevant instances of a gesture. High recall is particularly essential in scenarios where missing a gesture may break communication, especially in cases involving users relying exclusively on gesture interaction. For instance, in real-world cases, failing to recognize the sign for `help' can have detrimental consequences.

The **F1-score**, which is calculated as the harmonic means of both precision and recall, constitutes a balanced measure that demonstrates the trade-off between both. It becomes particularly valuable in the presence of an imbalanced dataset or whenever both false positives and false negatives contribute equally to the detriment. A high F1-score implies that the model is not biased towards any given metric and shows its robustness in discriminating actual classes among a variety of scenarios in real-world settings.

Besides computing per-class metrics, **macro-average** and **weighted-average** values were calculated to provide an overview of the performance. Macro averaging computes the unweighted total of precision, recall, and F1-score among all gesture classes thereby giving equal weight to every class without accounting for relative frequency. This becomes important for reviewing the behavior of the model toward gestures that are less frequently performed or poorly represented. In contrast, **weighted averaging** considers the number of instances in each class and reflects the model's performance in a more realistic way, particularly in cases where some gestures are more frequently performed than others. Thus, the two averaging methods give comprehensive evidence about the classification capabilities of the model across the different types of gestures.

The model successfully trained at an impressive accuracy of 99.3%, while it recorded a testing accuracy of 98.03%. Clearly, this model has been learned seriously and generalized highly. Moreover, the above results indicate that the model has not only learned the underlying patterns inherent within the data it has been trained for but also has worked effectively on without leakage data, which is essential in real-time deployment. This performance trend is shown in Figure 4, where the training and testing accuracies were recorded over ten epochs. The smooth upward trajectory of both curves, without significant deviation between them, indicates a stable and efficient training process. It is also important to point out both training and testing accuracy deem it impossible for the model to be overfitted and indicate robustness for application in real-world usage.

The Figure 4 demonstrates that the model consistently exhibits high-performance over-all training epochs; this indicates that the learning process is adequately optimized without experiencing problems like vanishing gradients, instability, or noisy updates. The high level of

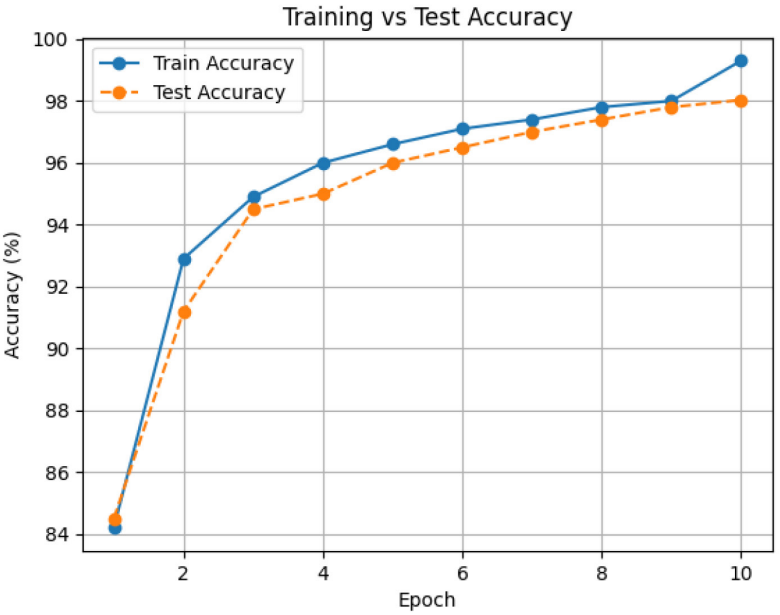

Figure 4

Training vs. Test Accuracy over 10 epochs. The model shows strong performance with training accuracy reaching 99.3% and test accuracy 98.03%, indicating good generalization and minimal overfitting.

accuracy with stability offers a sound premise to implement the model in real-time gesture recognition tasks. Both speed and correctness are brought out as crucial.

At Figure 5, the normalized confusion matrix is displayed on the model's class-wise performance with a view to deducing some strengths and weaknesses. The normalized confusion matrix thus presents a visual summary on how well the model's predictions correspond with the actual gestures happening on the ground. Each row in the normalized confusion matrix represents the actual class, while each column corresponds to the predicted class. The values in the matrix are normalized in such a way that each row sums to 1 (or 100%) for this intuitive understanding of predictions across class boundaries regardless of class size.

Under an ideal situation, the **diagonal cells**-the cells corresponding to the correct classification of each class will generally have the maximum values; it means the model has learnt 'that' gesture well. The brighter or darker, in terms of the colormap, these diagonal cells are, the better is the accuracy on that particular class. **Off-diagonal cells** show areas of the model's mistake, where it has classified one gesture as another. This is particularly insightful in knowing which pairs of gestures are visually or

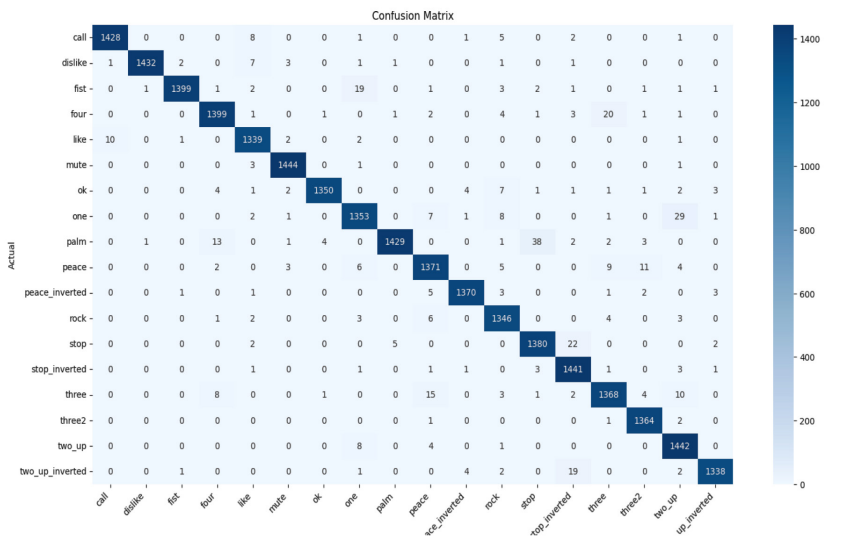

Figure 5

The matrix is normalized and labelled to show the percentage of correct and incorrect predictions per class, thus giving an easy way to identify the misclassified classes and what type of those errors.

spatially similar, such as gestures that use overlapping positions of fingers or similar orientations of the hands.

This confusing scenario could reflect a cell where the true class is "thumbs up" but the predicted class is "fist." It is crucial to find out such patterns for retraining the data and changing the model architecture to best differentiate fine differences. Also, it normalizes the matrix-performances among different gesture classes, even when a few classes are underrepresented, thus mitigating class imbalance effects on skewing interpretation.

Complementing the confusion matrix is the **classification report** shown in **Figure 6**, which gives a numerical evaluation of the performance of the system on each gesture class. The report counts **precision**, **recall**, and **F1-score** for each class. Precision measures how correct predictions were for such a gesture among all the predictions for that gesture; recall measures how many actual instances of the gesture were correctly recognized; while F1-score provides a harmonic balance of the two, thus giving a more holistic view of performance for each gesture category.

Classification Report:

	precision	recall	f1-score	support
call	0.99	0.98	0.99	1423
dislike	1.00	0.99	0.99	1444
fist	1.00	0.98	0.99	1387
four	0.97	0.97	0.97	1357
like	0.97	0.99	0.98	1360
mute	0.99	0.99	0.99	1460
ok	0.99	0.97	0.98	1397
one	0.97	0.97	0.97	1375
palm	0.99	0.95	0.97	1425
peace	0.97	0.98	0.97	1419
peace_inverted	0.99	0.99	0.99	1392
rock	0.97	0.99	0.98	1441
stop	0.96	0.97	0.97	1453
stop_inverted	0.97	0.99	0.98	1467
three	0.97	0.97	0.97	1371
three2	0.98	0.99	0.99	1379
two_up	0.97	0.99	0.98	1479
two_up_inverted	0.99	0.98	0.99	1438
accuracy			0.98	25467
macro avg	0.98	0.98	0.98	25467
weighted avg	0.98	0.98	0.98	25467

Figure 6

A Classification Report is necessary, where model performance is articulated according to gesture class, presenting the precision, recall, and F1-Score for all classes.

This detailed classification table lets one know under which gestures there are highly confident classification, and under which it is best aimed toward improvement for particular gestures. For instance, if there is a high recall with low precision, it might mean that the model is making a prediction of that class very often but fails in accuracy most of the time. On the other hand, if there is high precision with low recall, it indicates that the model will predict that class only when the model is sufficiently sure to do so and might not always get the true cases of that gesture.

Specifically, both the **macro-average** and the **F1-score weighted averages** appear at **0.98**, testifying to the fact that the model maintains reasonably good performances over all gesture classes, regardless of their dataset frequency. A higher macro-average F1-score indicates a constancy in performance across the classes, but the weighted average incorporates class imbalance so that frequent gestures do not overshadow the model performance.

The combo of a confusion matrix and classification report should be considered as diagnostic tools regarding its internal decision process. Such evaluations allow for a necessary analysis with developers and researchers alike to determine where performance could be fine-tuned and to set up any inequitable class performance. This is especially significant in real-time and critical applications such as assistive communication-based tools, where a single act of misclassification can terminate proper interactions. High and homogeneous scores against multiple evaluation measures are clear and undoubted indicators for asserting the reliability and robustness of the proposed hybrid model in diverse and dynamic environments.

Along with the typical evaluation measures, a **Receiver Operating Characteristic (ROC) curve** was created for every gesture class, as shown in **Figure 7**. The ROC depicts the trade-off between the true positive rate (sensitivity) and the false positive rate for different threshold values. The **Area Under Curve (AUC)** metric gives a single scalar value that summarizes the overall performance of the classifier with respect to each class. The AUC value of 1.0 indicates perfectly classifying the data points, whereas 0.5 means random guessing.

From the Figure 7, all the defined classes of gestures almost achieved the values for area under curve (AUC) that are to be closer to **1.00**, where many such as call, dislike, fist, mute, and peace_inverted achieved AUC values of **1.00**. Even for the lowest AUC values among classes, they hover around **0.99**, affirming how well this model understood gestures overall. Furthermore, it acts as an affirmation regarding the system's robustness

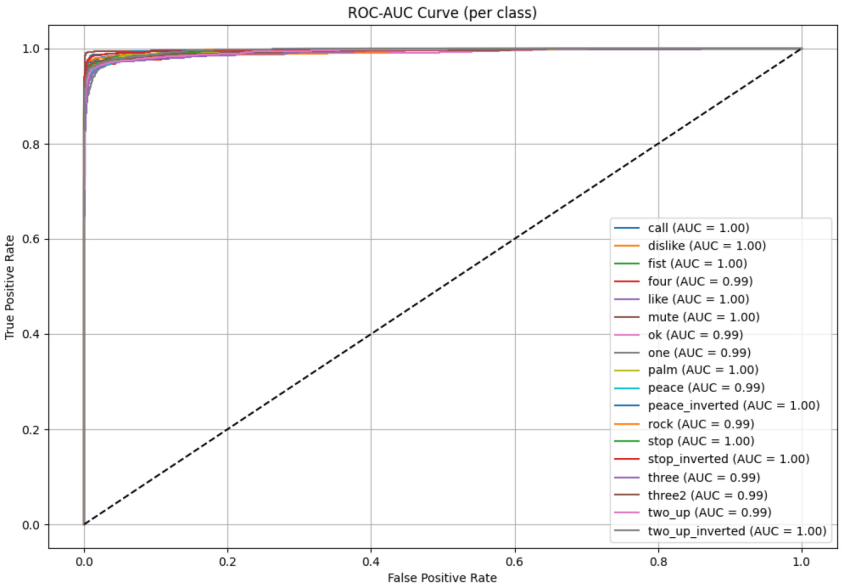

Figure 7

ROC-AUC Curve for each gesture class. Nearly all classes exhibit an AUC score of 0.99 or higher, highlighting the model's strong discriminatory ability and consistent classification performance across the entire gesture set.

and reliability towards more complex and multi-class classification problems.

Discussion

The use of a hybrid deep learning architecture combining Swin Transformer, ResNet34, and BiLSTM for static hand gesture recognition has yielded encouraging results. Such high accuracy scores, 99.3% for the training and 98.03% for the testing dataset, reflected its generalization capabilities. The model received a holistic understanding of gesture semantics due to the spatial and contextual features extracted from the gestures in time-series along with sequence modeling. The real-time inference using OpenCV showed the application potential of the system for deploying aid technology. A few challenges remain, though. Some gestures were misclassified, indicating that improvements for differentiating fine-grained features should be made. Despite the challenges, the current performance of the system on consumer-grade hardware makes it necessary to optimize the architecture for deployment

on mobile and embedded devices for greater accessibility. Exploring dynamic gesture recognition and multilingual gesture-to-text mapping to enhance model applicability would be part of the future work.

Conclusion

The research presents a rigorous hand gesture recognition system that uses hybrid deep learning comprising Swin Transformer and ResNet34 deep networks fused for feature extraction and Bi-LSTM for sequence modeling. The model efficiently learned locally and globally in terms of spatial features; whereby CNN-based feature learning was merged with transformer-based attention mechanisms for recognition. The entire dataset preparation and preprocessing pipeline was disciplined so that the system trained on high-quality annotated gesture images. Data augmentation and preprocessing steps used in the system, like resizing and normalization, helped to some extent at improving the generalization capacity. The introduction of a custom Dataset and PyTorch DataLoader improvised data handling efficiently during the training and led to a smooth handling of bulk gesture datasets. The evaluation indicators, in terms of precision, recall, and F1 score, demonstrated consistency so that the model could uphold its classification performance across all gesture categories. The confusion matrix and classification report offered some insights into system performance on separating different gestures at a low misclassification rate. This high consistency with other classes also validates the useful field of the model. The system was further implemented for real-time prediction, wherein live input was to some extent processed using OpenCV. Extra user-friendly features such as live video processing and gesture classification create a real chance for applying the model for gesture identification in sign language and human-computer interaction. Sequence modeling using BiLSTM makes it more logical when it comes to unfolding the possibilities of interpreting dynamic hand gestures for continuous gesture-based command execution. The work thus offers important insights that provide help to propel forward the hand gesture recognition under an amalgam of deep learning systems professing high reliability and efficiency. The findings indicate indeed that hybrid architectures combining CNNs and transformers can dramatically increase recognition performances. Future work might focus on expanding the dataset, enhancing computational efficiency, and adapting the model to mobile and embedded applications to help with its real-world usability.

Financial Grant: None

References

[1] S. S. Rautaray, "Real-time hand gesture recognition system for dynamic applications," *Int. J. UbiComp*, vol. 3, pp. 21–31 (2012), doi: 10.5121/iju.2012.3103.

[2] N. H. Mohd Dhuzuki, A. A. Zainuddin, N. A. S. Kamarul Zaman, A. N. M. Ahmad Razmi, W. S. Kaitane, A. Ahmad Puzi, M. N. Johar, M. Yazid, N. A. Mohd Nordin, S. N. Sidek, and H. F. Mohd Zaki, "Design and implementation of a deep learning-based hand gesture recognition system for rehabilitation Internet-of-Things (RIoT) environments using MediaPipe," *IIUM Eng. J.,* vol. 26, no. 1, pp. 353–372 (Jan. 2025), doi: 10.31436/iiumej.v26i1.3455.

[3] R. P. Singh and L. D. Singh, "Dyhand: Dynamic hand gesture recognition using BiLSTM and soft attention methods," *Vis. Comput.,* vol. 41, pp. 41–51 (Jan. 2025), doi: 10.1007/s00371-024-03307-4.

[4] M. Linardakis, I. Varlamis, and G. Th. Papadopoulos, "Survey on hand gesture recognition from visual input," arXiv:2501.11992 (2025), doi: 10.48550/arXiv.2501.11992.

[5] V.-H. Le, "Selected hand gesture recognition model based on cross-evaluation of deep learning from large RGB image datasets," *Multimedia Tools Appl.,* (Mar. 2025), doi: 10.1007/s11042-025-20743-z.

[6] J. Zhou, C. Xu, and L. Cheng, "Hand gesture recognition from an open-set perspective," *IEEE Trans. Multimedia, early access*, pp. 1–12 (Jan. 2025), doi: 10.1109/TMM.2025.3535363.

[7] B. Suthar and S. I. Ali, "A survey on hand gesture sign language recognition with Arduino Leonardo," in *Proc. Int. Conf. Computational, Communication and Information Technology (ICCCIT)*, Indore, India (Feb. 2025), doi: 10.1109/ICCCIT62592.2025.10927994.

[8] Q. Chen, Q. Tang, M. Zhang, and L. Ma, "CNN-based gesture recognition using raw numerical gray-scale images of surface electromyography," *Biomed. Signal Process. Control,* vol. 101, Art. no. 107176 (Mar. 2025), doi: 10.1016/j.bspc.2024.107176.

[9] M. Al-Hammadi, G. Muhammad, W. Abdul, M. Alsulaiman, M. A. Bencherif, and M. A. Mekhtiche, "Hand gesture recognition for sign language using 3DCNN," *IEEE Access,* vol. 8, pp. 79491–79509 (Apr. 2020), doi: 10.1109/ACCESS.2020.2990434.

[10] J. Qi, L. Ma, Z. Cui, and Y. Yu, "Computer vision-based hand gesture recognition for human–robot interaction: A review," *Complex Intell. Syst.*, vol. 10, pp. 1581–1606 (2024), doi: 10.1007/s40747-023-01183-5.

[11] A. Mujahid, M. J. Awan, A. Yasin, M. A. Mohammed, R. Damaševičius, R. Maskeliūnas, and K. H. Abdulkareem, "Real-time hand gesture recognition based on deep learning YOLOv3 model," *Appl. Sci.*, vol. 11, no. 9, pp. 4164 (May 2021), doi: 10.3390/app11094164.

[12] M. A. Ozdemir, D. H. Kisa, O. Guren, A. Onan, and A. Akan, "EMG-based hand gesture recognition using deep learning," in *Proc. Med. Technologies Congress (TIPTEKNO)*, Antalya, Turkey, pp. 1–4 (Nov. 2020), doi: 10.1109/TIPTEKNO50054.2020.9299264.

[13] A. Kapitanov, K. Kvanchiani, A. Nagaev, R. Kraynov, and A. Makhliarchuk, "HaGRID—Hand gesture recognition image dataset," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV*), pp. 4572–4581 (2024).

Received June, 2025