Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Sign Language Recognition in Real-Time: A Deep Learning Framework for Human-Computer Interaction

Madhan L[1], Mythily M[2*], Beaulah David[3], Rexie.J.A.M.[4], S. Deepa Kanmani[5], Dr. G. Naveen Sundar[6]
[1]*Computer Science and Engineering, Karunya Institute of Technology and Sciences, Coimbatore, Tamil Nadu 641114, India*
[2*]*Computer Science and Engineering, Karunya Institute of Technology and Sciences, Coimbatore, Tamil Nadu 641114, India*
mythily.m@gmail.com
[3]*Computer Science and Engineering, Nehru Institute of Technology Coimbatore, Tamil Nadu 641105, India*
[4]*Computer Science and Engineering, Karunya Institute of Technology and Sciences, Coimbatore, Tamil Nadu 641114, India*
[5]*Department of Information Technology, Sri Krishna College of Engineering and Technology, Coimbatore, Tamil Nadu 641008, India*
[6]*Computer Science and Engineering, Karunya Institute of Technology and Sciences, Coimbatore, Tamil Nadu 641114, India*
naveensundar@karunya.edu

ARTICLE INFO

Received: 28 Dec 2024
Revised: 18 Feb 2025
Accepted: 26 Feb 2025

ABSTRACT

Sign language serves as a vital means of communication for individuals who are deaf, hard of hearing, or unable to speak verbally, enabling them to connect with the world through a structured system of visual and manual gestures. This project addresses the challenge of real-time sign language recognition. A custom dataset was created featuring 11 specific gestures representing commonly used symbols, captured using a webcam. The system processes these gestures in real-time, instantly displaying their corresponding letters on a screen. This initiative, rooted in human-computer interaction (HCI), focuses on enabling seamless communication through real-time hand gesture interpretation. The primary objective of this research is to analyze multiple deep learning models, including the Pre-Trained SSD MobileNet V2 architecture, Convolutional Neural Networks (CNNs), Hybrid models, and YOLO, to determine the most effective model for sign language recognition

Keywords: Object detection with Convolutional Neural Networks (CNN), SSD-MobileNet, Hybrid Model, YOLO, deep learning, real time processing, Human-Computer Interaction (HCI)

I.INTRODUCTION

Communication is an essential aspect of human existence. People convey their thoughts and intentions by speaking, and others respond by listening and engaging in conversation. However, there are still many that cannot speak or hear; this definitely makes communication a problem to them, and they communicate in the presence of others as well within their community. Unfortunately, most people cannot read, write, or interpret this so that usually a trained interpreter is needed for proper communication. This solution, however, can be expensive and impractical in many situations, leaving these individuals in difficult circumstances. To address this issue, a system is needed that can recognize and interpret sign language without the need for a mediator. Artificial Intelligence, a branch of computer science focused on mimicking human intelligence, offers a promising solution. Within AI, computer vision, which is the study of extracting meaningful information from images, is one area that allows machines to interpret visual data such as sign language gestures.

This work utilizes various deep learning techniques, including SSD-MobileNet, YOLO, Hybrid models, and CNNs, to effectively recognize sign language from a unique dataset. These methods have demonstrated impressive performance, achieving high mean average precision compared to other approaches in the field. The success of our system depends on image processing techniques. Improving the quality of images through background removal and light adjustment will improve recognition accuracy significantly. This focus on image processing is important because it allows our models to better interpret the visual data associated with sign language gestures. The deep learning methods in this research are going to work to analyze and classify sign language gestures much more efficiently and precisely. Therefore, have a tendency to make an even stronger framework for sign language recognition with the use

of advanced algorithms, so as to perform well in the real-world environment. The brief concept of deep learning methods used in this work:

- SSD-MobileNet: This is a lightweight object detection framework optimized for real-time usage on devices with limited computational resources, combining the Single Shot Multi-Box Detector (SSD) with a MobileNet backbone for efficiency.
- YOLO is a very fast object detection model. It regards the detection problem as one regression problem and predicts bounding boxes along with class probabilities in one pass through the network, which makes it very efficient for real-time applications.
- CNN-Convolutional Neural Network: A deep learning architecture that is designed to process grid-like data, such as images. CNNs are heavily employed in tasks like image classification and object detection with the use of shared weights. This reduces parameters and allows better efficiency for image-related tasks.
- Hybrid Model: The hybrid model is a deep learning framework that combines two distinct architectures, ResNet and MobileNet. It processes input images simultaneously through two parallel sub-networks.

This research introduces a unique dataset designed to address the scarcity of resources in this field. It was carefully captured with diverse backgrounds and environment condition during data collection. This dataset will fill a critical gap for scarcity of resources in future work within the domain of sign language recognition. The results clearly demonstrate improvements in performance when compared to basic methods. Deep learning algorithms have been used increasingly in recent years to develop systems that can recognize sign language and support those who have difficulty speaking or hearing. In recent years to develop system that can recognize sign language and support those who have difficulty speaking or hearing using Deep Learning algorithms. In [2], the authors presented the Indian sign language letters using only one transfer learning model named MobileNet, and yielded 97.26% accuracy. In [3], the authors researched the Bangla sign alphabets and digits and recognized them using a 2-dimensional CNN model and their accuracy rate is 99.57% with a validation loss of 0.56%. In [4], authors proposed deep CNN-based Bangla sign language character classifications on the Ishara-Lipi dataset and exhibited 95.71% accuracy. In [5], the authors worked on Arabic character sign language using a transfer learning model named EfficientNetB4 and achieved a 98% trained accuracy along with 95% tested accuracy. To create our sign classification model, in this work used the deep learning (DL) approach on various well-know models. And employed four accessible deep learning models named YOLO, CNN, Hybrid and SSD MobileNet with pre-trained (trained on ImageNet) weights. On both the validation and training sets, YOLO achieves the greatest accuracy. The training length outperforms the CNN and SSD MobileNet in terms of timing. In author [8], different performance evaluation metrics in comparing the various performances related to deep learning techniques on object detection algorithms, has in this work explained AP (average precision) as compared to precision and recall relationships with this also proposing a standard implementation that can be adapted into being used as benchmarked among different datasets with minimum adaptation in its respective annotation files. Author [10], explained the Convolutional Network architecture and improved the architecture to withstand and improve the custom-trained models' accuracy by adjusting its parameters with deep design which plays a major role in deep visual recognition architecture. Author [11] worked on sign language such as British and Greek using the Random-Forest algorithm with low-level visual features. They achieved an average F1 score of 95%, which indicates that sign language can be identified with high accuracy using low-level visual features. In Author [12] highlights how customizing Deep Neural Networks (DNNs) for pose estimation enables precise localization of articulated objects. Such methods can be adapted or serve as a foundation for recognizing sign gestures. Author [13] worked on Indian Sign Language recognition using a combination of LSTM and GRU layers, achieving 97% accuracy on their custom IISL2020 dataset, which included 11 static signs collected under natural conditions. Their feedback-based learning approach processed video inputs to recognize isolated signs, addressing communication gaps for individuals with speech or hearing impairments. This work aligns with my research, which employs deep learning techniques to recognize unique gestures in custom datasets. Author [14] explained the impact of convolutional network depth on accuracy in large-scale image recognition. By using 3x3 convolutional filters and systematically increasing network depth to 16-19 layers, they achieved state-of-the-art performance with better accuracy and simplicity. Author [15] developed a real-time sign language recognition system using You Only Look Once (YOLO), an object detection method based on Convolutional Neural Networks (CNN). By retraining the Yolov3 pre-trained model with customized channels and classes, they tailored it to recognize signs from the Indonesian Sign Language

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

(BISINDO) dataset. The system's ability to bridge communication gaps was demonstrated when it processed video input at 8 frames per second with 72.97% accuracy and 100% accuracy on picture data. In order to improve Human-Computer Interaction (HCI), the author [16] investigated gesture recognition as a way to allow gadgets to decipher human body languagespecifically, hand gestures without the need for physical contact. Three steps are used in their vision-based system: learning, detection, and recognition. The system identifies hand motions, preprocesses the input, and records real-time gestures using a webcam. A 3D Convolutional Neural Network recognizes gestures and identifies fault spots and finger placements. This method is appropriate for real-time applications as it delivers excellent detection accuracy and performance without the need for gloves or a consistent backdrop. The development and methods of hand gesture recognition systems were examined by the author [17], who also highlighted the uses of these systems in Human-Computer Interaction (HCI). They discussed the benefits and drawbacks of several strategies, including data gloves, vision-based systems, and colored-marker solutions. The study suggested using running average principles and contour-based features for efficient hand identification, highlighting issues including illumination variability, backdrop complexity, and real-time processing. In order to demonstrate how gesture-based interfaces might improve user engagement with gadgets, the authors also examined well-known gesture recognition systems, such as those that use neural networks, segmentation techniques, and skin-color detection. A gesture detection technique utilizing 3D point cloud data obtained from Kinect or ToF cameras was suggested by the author [18]. The technique makes use of a Viewpoint Feature Histogram (VFH) descriptor, which is made more unique by breaking the picture up into smaller cells. The system uses various classifiers to identify hand postures and dynamic gestures from sign language datasets: dynamic time warping or hidden Markov models for dynamic gestures, and a nearest neighbor classifier with city-block distance for postures. Results from experiments show that this method works well for increasing the accuracy of both static and dynamic gesture recognition. Using Convolutional Neural Networks (CNNs) and depth pictures taken with a Kinect camera, the author [19] created a system for identifying hand letter and number motions from American Sign Language. A fresh collection of depth pictures of ASL letters and numerals was produced as part of this study. An empirical evaluation and comparison of the suggested approach with a comparable dataset for Vietnamese Sign Language were part of the study. The groundwork for future advancements in a comprehensive transcription system for Sign Language is laid by this effort. In order to meet the demand for technologies that help deaf people communicate, author [20] investigates the use of Convolutional Neural Networks (CNNs) for hand gesture recognition in sign language. The study shows how CNNs can accurately anticipate the indications by efficiently capturing and learning information from hand gesture photographs. The algorithm also forecasts suitable words or phrases by incorporating a suggestion mechanism, which improves communication. Achieving a 91.07% prediction accuracy, the study highlights the potential of CNNs in sign language recognition, offering real-time, accessible communication for individuals with hearing impairments.

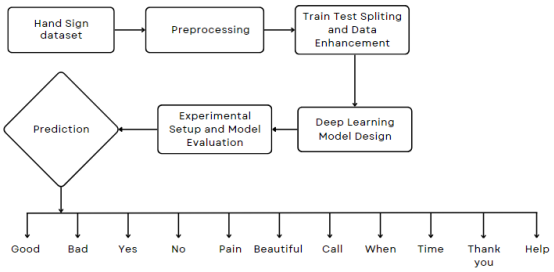

Fig.1 The proposed methodology's block diagram.

II.PROPOSED METHODOLOGY

This study focuses on recognizing sign words using deep learning techniques. To achieve this, four widely recognized algorithms were adapted and optimized to identify 11 unique words. These algorithms were tested on a custom-built dataset, which is essential for the successful implementation of the methodology. Data preprocessing has been considered an essential stage in the improvement of both performance and accuracy of data-driven algorithms. Our dataset, therefore, was rigorously pre-processed to be reliable as it was randomly split between training and test

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

subsets without bias to either. After preparation, the chosen deep learning algorithms were then applied in a way that would help train up models. The performance of the trained models was then tested using different performance metrics. The results were analysed and compared to determine the best algorithm for sign word recognition. Such findings contribute to the development of sign language recognition systems, which will eventually help improve communication accessibility for sign language users. Fig. 1 proposed methodology.

2.1. Dataset description

The dataset contained in this study is a rare and unique resource created specifically for the recognition of sign words. This type of dataset is essential for the effective development of sign recognition systems. It includes the basic inputs that could normally be used to identify sign from hand gestures and each hand gestures are know as classes and the dataset is entirely novel and consists of images representing 11 distinct classes of sign words. These classes include 'Good,' 'Bad,' 'Yes,' 'No,' 'Pain,' 'Beautiful,' 'Call,' 'Help,' 'When,' 'Time,' and 'Thank you.' Each class containing more than 20 images on each class resulting a total of 254 images. A sample of the dataset is illustrated in Fig. 2, which provides a visual representation of the images included in this study.

Fig. 2 Sample data of different classes

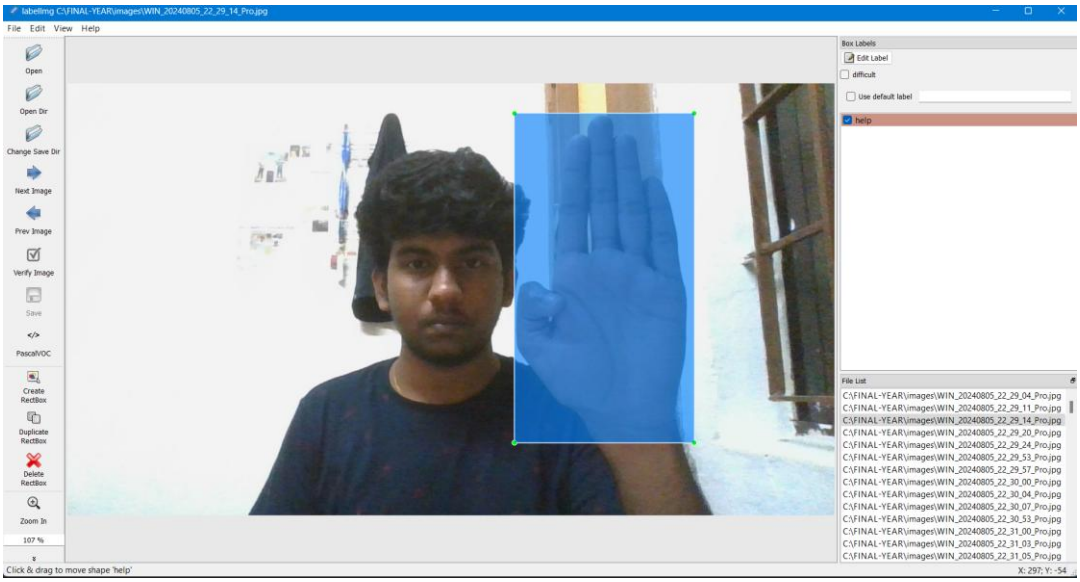

Fig.3 shows a sample instance annotated for a single class using the LabelImg tool.

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

2.2 Data annotation

Data annotation is an essential development step in computer vision models; it enables their interpretation and analysis of wide-ranging visual data with the utmost accuracy. It requires assigning labels to raw data based on the descriptive text that allows models to recognize and respond accurately to visual patterns. In some applications, such as the real-time recognition of hand gestures from sign language, bounding box annotation is especially necessary for database preparation and the bounding box technique has emerged as key facilitator of the annotation process [6]. Bounding boxes are used to delineate rectangular regions around objects of interest in images [7]. Bounding boxes enclose a gesture with spatial information precise enough to include coordinates and dimensions. Annotations are usually stored in text files. In such files, each file contains the class number that will correspond to a specific hand hand-signed word. The labelled data provided during the training phase makes ground truth for models, letting them learn how to effectively recognize and locate hand gestures. In the creation of a sign word dataset with more than 254 images were annotated. Such annotations are very crucial for models to be able to identify and understand hand sign gestures correctly. Annotation was performed with the LabelImg tool-a famous graphical image annotation tool that makes dataset for object wdetection models. LabelImg provides intuitive interface to make user draw rectangle boxes around objects in images and assign the target class labels. Saving annotations is done either in XML (Pascal VOC) or TXT (YOLO) format making it compatible with many widely used deep learning frameworks. In this study, every image in the complete dataset was fully annotated by hand so as not to miss any important information for bounding box placement and labeling that would make all the great training and evaluation possible. Annotation is done as shown in the Fig. 3.

2.3. Dataset splitting and enhancement

The dataset utilized in this study comprises 254 images categorized into 11 classes of sign words. This dataset was split into a training set with 204 images and a test set with 50 images. Since larger datasets typically enhance the performance of deep learning models, data augmentation was applied to the training set using techniques such as flipping, rotation, and zooming. It is important to note that data augmentation was not performed on the test set. Data augmentation is a technique that enhances datasets by generating variations of the existing data. Common augmentation methods include flipping, cropping, rotation, color space transformation, and noise injection. These enhancements improve the robustness of the model by exposing it to diverse data during training. Our deep learning models were trained on the augmented training dataset, and the test set was used for validation to evaluate algorithm performance. The distribution of images across the 11 classes in the training set is depicted in Fig. 3. The accompanying bar chart demonstrates that all classes are equally represented in the dataset, ensuring a balanced input for model training.

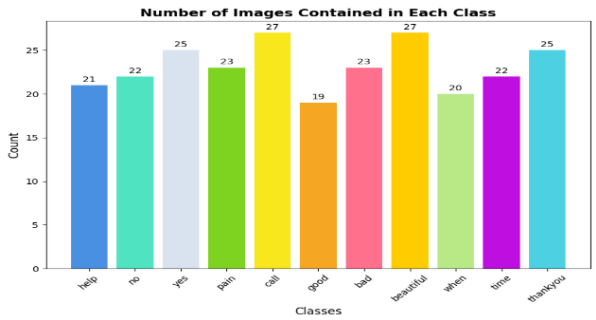

Fig. 4. A bar chart showing number of images of each class.

2.5 Deep Learning model design

This design of the deep learning model integrates YOLO, CNN, and SSD MobileNet to leverage their strengths in image classification and recognition tasks. The workflow begins with input preprocessing, where images are resized and normalized to suit the model requirements. Data augmentation techniques such as rotation, flipping, and scaling are used to enhance diversity in the training dataset. four sub-models were used for feature extraction. YOLO, because

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

it is very fast and efficient in detecting objects and bounding boxes; a custom CNN model, to extract detailed features, classify the object, using a series of convolutional layers, pooling layers, and activation layers; and SSD MobileNet v2 for its lightweight architecture optimized for mobile and resource-constrained environments. In that approach, the extracted features from each one of these models are integrated in a dedicated feature-fusing layer, were feature maps are combined using concatenation, and then 1 x 1 convolutions diminish the dimensionality, making a global average pooling that creates a compact representation from the features, followed finally by a fully connected network processing the fused features made dense with ReLU activation of the layers, dropout regularity of connections among neurons, and finally an output layer of softmax. This integrated approach combines YOLO's speed and the precision of localization, the fine-grained feature extraction from CNN, and the efficiency of SSD MobileNet, therefore representing a strong solution in terms of complex image classification even in the presence of very limited computing resources as shown in the Fig.5 the four deep learning model collect and map the insights from the image to predict the final predicted hand sign gesture.

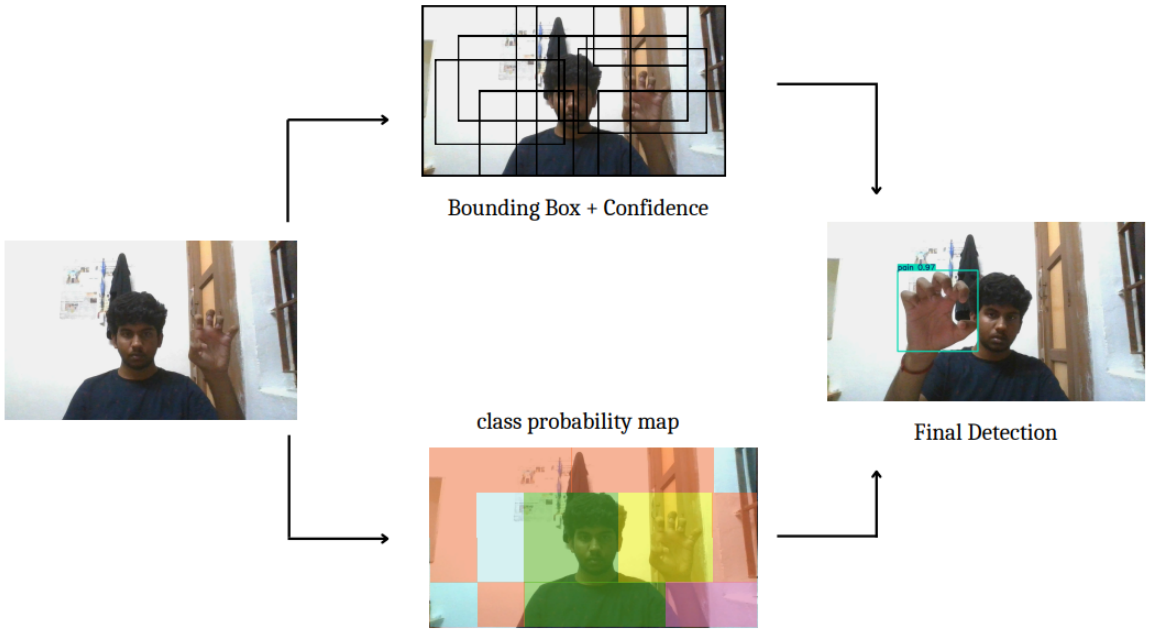

Fig 5. Workflow of Object Detection

The hybrid model in this project combines the strengths of ResNet and MobileNet architectures to improve sign language recognition. It utilizes ResNet blocks for feature extraction through convolutional layers and MobileNet blocks for efficient depthwise separable convolutions, reducing computational cost while maintaining accuracy.

2.6 Methodological Framework for Identification

2.6.1 CNN Architecture

Convolutional neural networks, or CNNs for short, are a type of deep learning model that is mostly used in computer vision applications including object recognition and picture categorization. Typically, the design begins with an input layer that normalizes and resizes incoming pictures. The Convolutional Layer, or Conv2D, is the central part of CNNs. It draws low-level features like edges and textures by applying filters to the input picture. These filters create feature maps as they go across the image. The feature maps are subjected to an Activation Layer (often ReLU) that sets negative values to zero in order to induce non-linearity. The feature maps' spatial dimensions are then decreased using a pooling layer (like MaxPooling2D), which helps preserve crucial information while lowering computational complexity. The feature maps go through a Flatten Layer, which transforms 2D data into a vector with a length of 1D, after a number of convolution and pooling layers. This vector is passed to one or more fully connected (dense) layers, where the model actually performs high-level reasoning and classification. Dropout Layers are included to prevent overfitting by setting a fraction of units randomly to zero during training. The final Output Layer generates the

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

predicted class labels, usually employing the Softmax activation function for multi-class classification. Overall, the CNN architecture enables efficient feature extraction and classification by learning hierarchical patterns, with regularization techniques improving its ability to generalize to new, unseen data. Fig. 6. CNN architecture.

2.6.2 YOLOv5n6u Architecture

YOLOv5n6u, or YOLOv5 Nano, is the lightweight and efficient version of the YOLOv5 architecture, designed for real-time object detection on resource-constrained devices such as mobile phones and embedded systems. This model maintains the core capabilities of YOLO fast and accurate object detection—while significantly reducing the model size and computational requirements. YOLOv5n6u splits the image into a grid and makes predictions for multiple bounding boxes and corresponding class probabilities in one forward pass. It does this to enable the simultaneous detection and classification of objects. This makes it useful for tasks such as sign language recognition, which often requires real-time processing. Running efficiently on edge devices, YOLOv5n6u delivers a powerful solution for the detection and classification of sign words in images with speed and accuracy. It is very well-suited for dynamic and complex environments due to its ability to perform

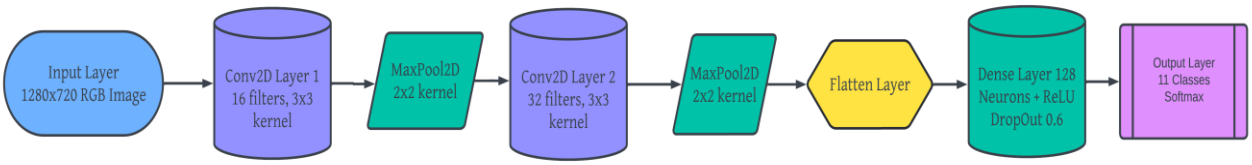

Fig 6. CNN Architecture

detection in a single step, unlike traditional models, which require multiple stages. YOLOv5n6u has a low computational footprint, fast inference time, and can work on various devices, so it is ideal for real-time sign word recognition applications, offering efficient performance without sacrificing accuracy. As shown in the Fig. 7 The YOLO model randomly selects image and performs label encoding and gains insights from the image and computes the loss computation which helps to create a YOLO network to predict the custom hand sign gesture.

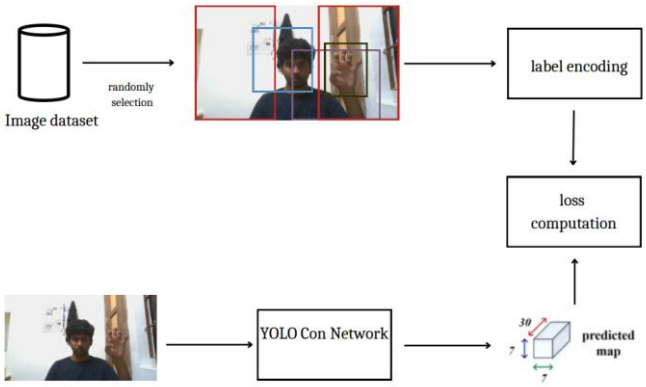

Fig 7. Workflow of YOLO Model

2.6.3 Hybrid Model

The hybrid model combines simplified versions of ResNet and MobileNet, processing input images in parallel to extract complementary features. In the ResNet branch, convolutional layers with batch normalization and ReLU activation are used to identify patterns, while gradually reducing the image size. A global average pooling layer simplifies the output into a feature vector. The MobileNet branch uses depthwise separable convolutions, which are efficient by separating the convolution process into two steps: depthwise and pointwise. It also includes batch normalization and ReLU activation, reducing the input size and creating a feature vector through global average pooling. By merging the feature vectors from both branches, MobileNet's effectiveness and ResNet's powerful feature extraction are combined. To avoid overfitting, this combined output goes via a dropout layer and a thick layer with

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

64 neurons. Ultimately, the input is categorized into one of the 11 target classes by the softmax layer. In order to prevent overfitting and guarantee effective training, the model is trained using the Adam optimizer and sparse categorical cross-entropy loss, with early stopping. This architecture is perfect for real-time sign language recognition since it balances speed and accuracy. The hybrid model architecture is depicted in Fig. 8, which also highlights important elements such as the merged features, the ResNet and MobileNet branches, and the final classification layers.

2.6.4 SSD MobileNet Architecture

The Single Shot MultiBox Detector (SSD) framework and the MobileNetV2 architecture are combined to create the effective item detection model known as SSD-MobileNetV2. Its real-time object detection design makes it ideal for applications running on low-processing-power devices like smartphones and embedded systems. Through the simultaneous prediction of bounding boxes and class labels, the SSD framework detects many objects in an image in a single forward pass. The foundation is MobileNetV2, which offers feature extraction via a thin, mobile, edge-device-optimized architecture. It maintains great accuracy while lowering the computing strain by using depthwise separable convolutions. The model is perfect for real-time applications like security systems, driverless cars, and mobile object detection jobs because of its ability to strike a balance between speed and accuracy. Because of its great efficiency and low latency, SSD-MobileNetV2 is a well-liked option for resource-constrained settings where quick object recognition is essential. Fig 9. MobileNet-SSD Network Architecture.

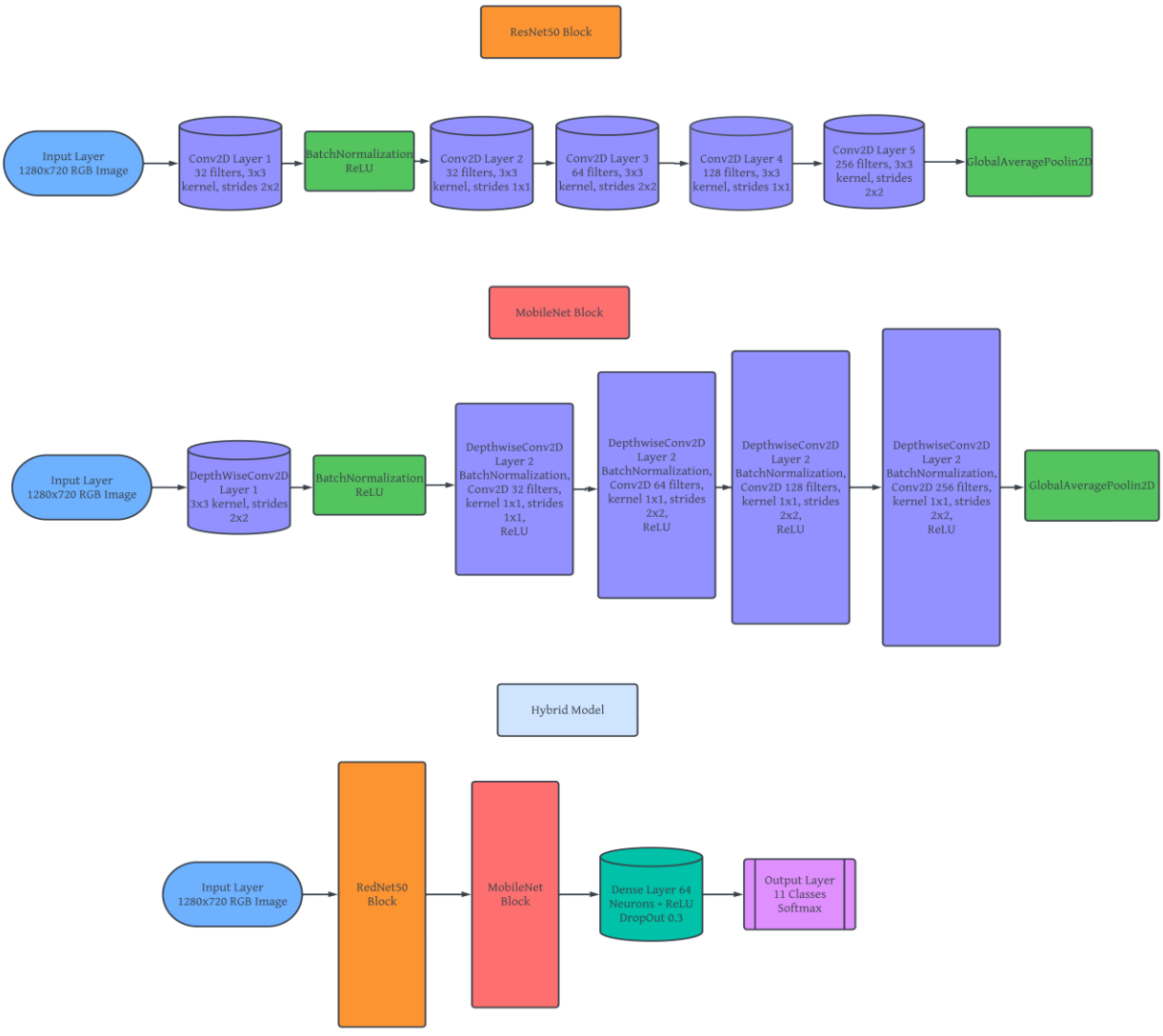

Fig 8. Hybrid Model Architecture

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

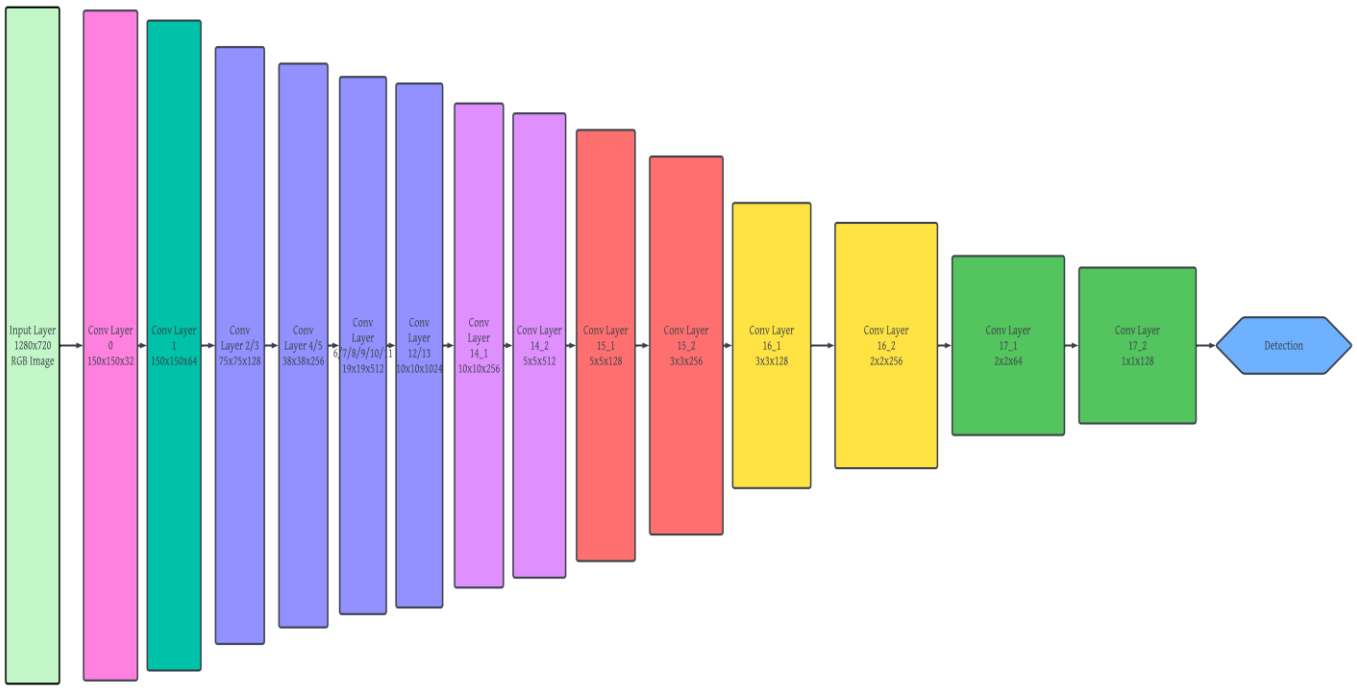

Fig 9. MobileNet-SSd Network Architecture.

III. RESULTS AND DISCUSSION

In this study, a custom dataset of 254 static images representing basic hand gestures was developed, whereby each image was carefully annotated to include critical landmarks; these annotations provided essential spatial information that significantly improved model ability to discern subtle differences between hand gestures. This achieved precision in deep learning models using these annotations during training, which eventually captured the minute structures of every hand gesture for a more robust and accurate detection framework. During the training phase, it used the annotated dataset for training deep learning models for effectively recognizing hand gestures. Integration of landmark data has facilitated more accurate localization and classification of hand gestures. In real-time testing, a webcam was used for dynamic tracking of hand movement, further proving the system's practicality.

As presented in Table 1, the YOLOv5n6u model demonstrated exceptional performance metrics. The precision reached 90%, indicating that 90% of predictions for each class were accurate. The recall rate was 95%, reflecting the model's capability to identify 95% of actual instances for each class. The F1 score reached 98%, 99% of mAP Mean Average Precision, highlighting the model's high accuracy across all classes. The performance of the YOLOv5n6u model is visualized in the Recall-Confidence Curve Fig. 11, the F1-Confidence Curve Fig. 12, and the Precision-Recall Curve Fig. 13. These plots demonstrate consistent results across most sign classes, though slight discrepancies were observed for certain signs, such as "Yes" and "Call," as highlighted in Fig. 10. These discrepancies could potentially be attributed to annotation errors or the difficulty in distinguishing subtle differences between similar gestures. The YOLOv5n6u model emerged as the top performer, achieving a 99% mAP score, as calculated using the formula presented in *(Ep1)*. Its ability to effectively handle occluded and overlapping hand signs, coupled with its computational efficiency, makes it an excellent choice for real-time applications, such as traffic surveillance and autonomous driving systems.

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Metric	Value (Dataset – 11 Classes)
Mean Average Precision(mAP)	99%
Precision	90%
Recall	95%
F1	98%

Table 1: Performance of YOLOv5n6u Model

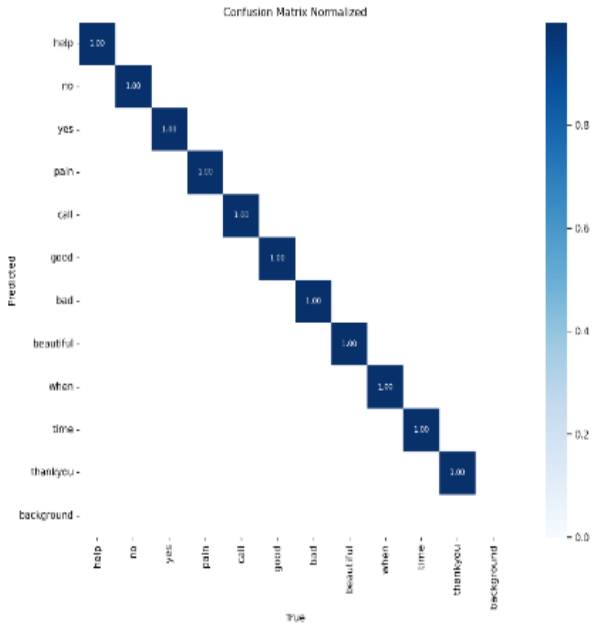

Fig 10. Normalized YOLOv5n6u Confusion Matrix.

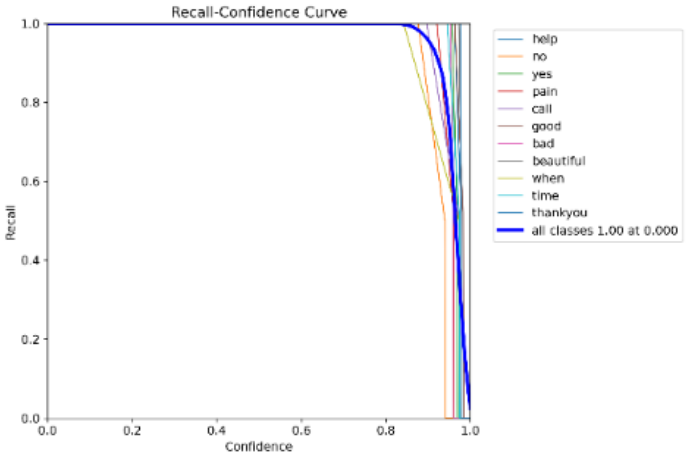

Fig 11. Recall Confidence Curve for YOLO Model

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

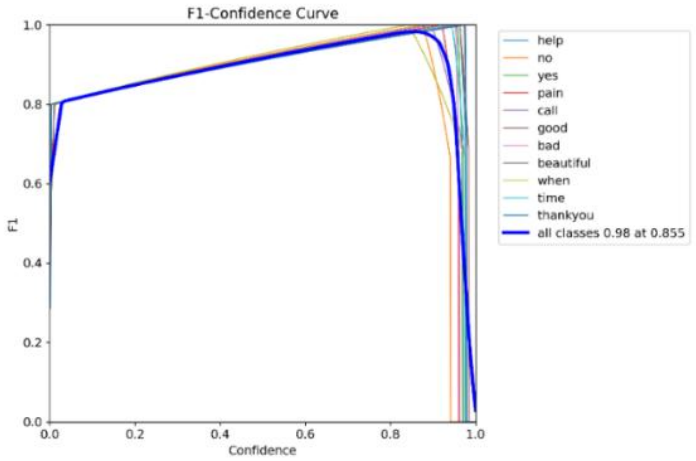

Fig 12. F1 Confidence Curve for YOLO

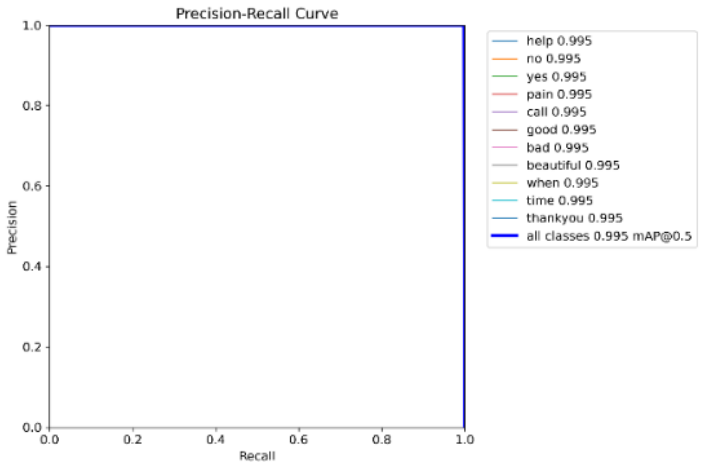

Fig 13. Precision Confidence Curve for YOLO

As shown in Table 2, the CNN model had significant difficulty correctly identifying several sign words. Gestures like "Good," "Bad," "Yes," "No," and "Beautiful" were often misclassified as "Pain." This problem likely occurred because CNNs look at the entire image as a whole, instead of focusing on specific important areas. As a result, overlapping features like hand position or background noise confused the model, making it harder to distinguish between similar gestures see Fig. 16. The training results highlight the model's struggles. It achieved a training accuracy of only 53% with a training loss of 4.7%. The validation accuracy was even lower at 37%, with a validation loss of 5%. Overall, the model's accuracy was just 10%, showing that it had major challenges in recognizing and classifying hand gestures. This poor performance is likely due to the simple design of the CNN model. Without deeper layers or fine-tuning, the model cannot fully understand the small differences between hand gestures. It also struggles to pick up enough important features to work well with different types of gestures. Improving the model by adding more layers, fine-tuning, or using advanced techniques like transfer learning could help overcome these issues.

Metric	Value (Dataset – 11 Classes)
Training Accuracy	53%
Training Loss	4.7%
Validation Accuracy	37%
Validation Loss	5%

Table 2: Performance of CNN Model

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

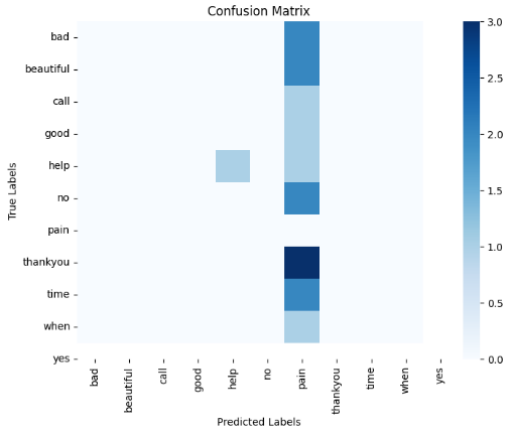

Fig. 16. CNN Model Confusion Matrix

As shown in Table 3, the SSD MobileNet-v2 model performed better than CNNs in recognizing and differentiating gestures. Unlike CNNs, SSD MobileNet focused on specific regions within images by generating bounding boxes to identify and classify hand gestures. This regional approach reduced distractions from irrelevant background elements, minimizing errors such as confusing "Pain" with other signs.

Metric	Value (Dataset – 11 Classes)
Mean Average Precision (mAP)	85%
Classification Loss	26%

Table 3: Performance of SSD-MobileNet-v2 Model

The model achieved a Mean Average Precision (mAP) of 85% and a classification loss of 26%. It used feature pyramids to detect gestures across different scales and positions, capturing small details like hand orientation, finger placement, and movement with high precision. The choice of MobileNet as the feature extractor made the model lightweight, fast, and efficient. The custom-trained SSD MobileNet model achieving mAP score of 85%, it may face difficulties when distinguishing between similar objects. Fig. 17 illustrates the total loss curve of the SSD MobileNet-v2 model *(Ep8)*, while Fig. 12 displays a set of predicted images using the trained model.

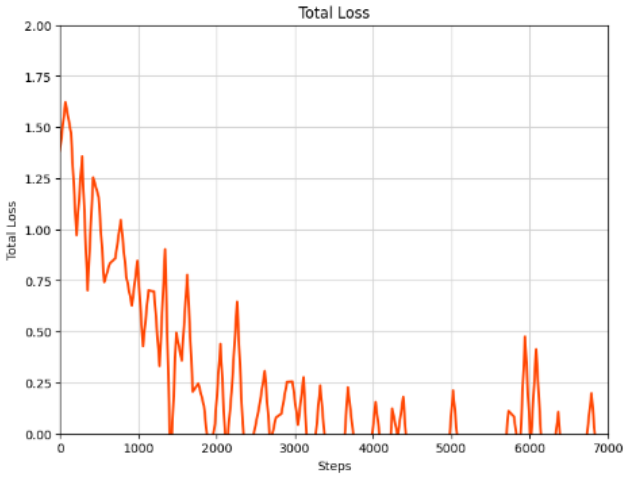

Fig 17. Total Loss of SSD MobileNet-V2

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Metric	Value (Dataset – 11 Classes)
Training Accuracy	13%
Training Loss	8.78%
Validation Accuracy	18%
Validation Loss	9.62%

Table 4: Performances of Hybrid Model

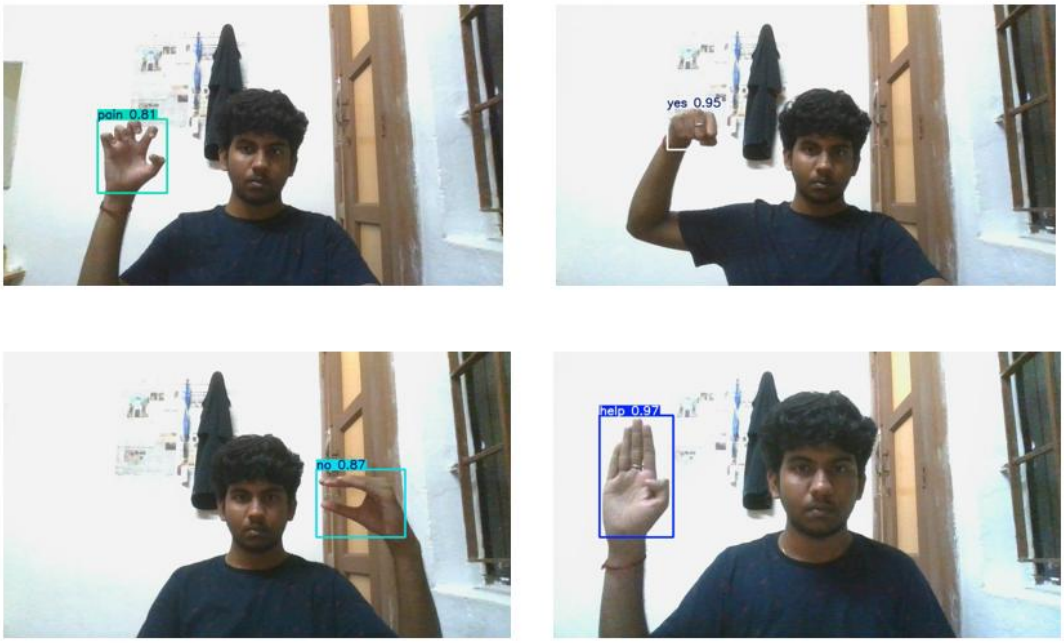

Fig 14. Predicated Image using Trained YOLOv5n6

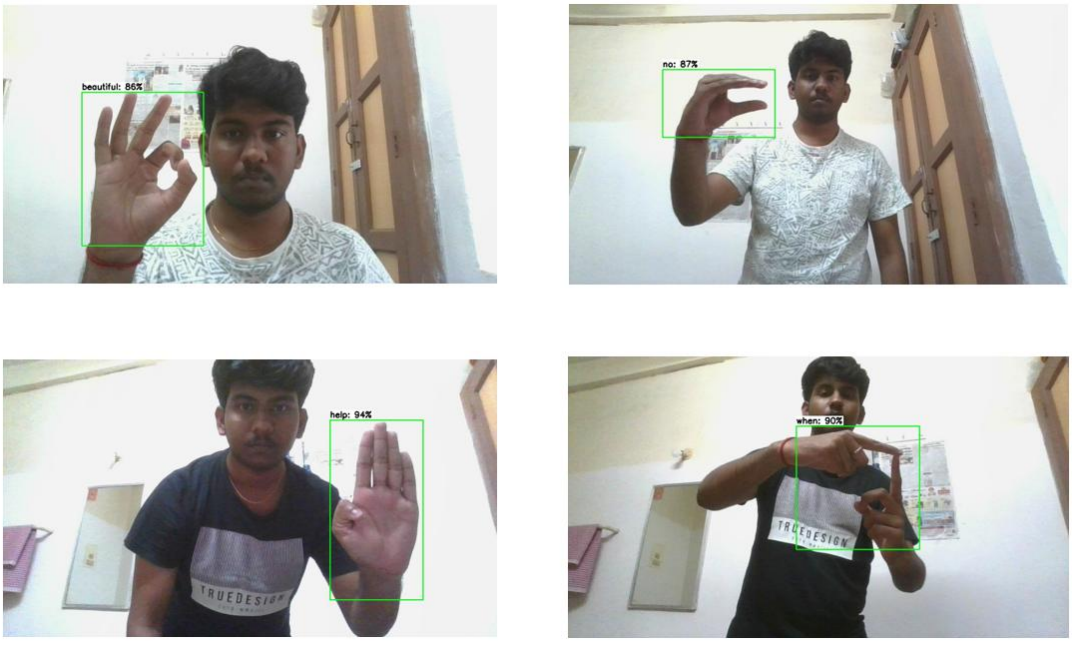

Fig 15. Predicated Image using Trained SSD MobileNet-V2

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

As shown in Table 4, the hybrid model achieved a training accuracy of 13% and a validation accuracy of 18%, showing significant difficulties in classifying gestures effectively. The model's complexity is probably the cause of the poor performance; overfitting may have resulted from merging ResNet and MobileNet blocks, which may have been too sophisticated for the limited dataset with only 11 classes. Furthermore, the use of 720x1280 high-resolution photos created further computational difficulties, making the extraction of significant characteristics more difficult. Another factor was the small size and lack of variety of the dataset; there were insufficient instances or variations, such varied lighting or rotations, to support the model's ability to generalize. Performance may be further impacted by training parameters that have not been tuned, such as learning rate and dropout. Potential overfitting or underfitting was indicated by the model's 8.78% training loss and 9.62% validation loss. Training loss was constant whereas validation loss varied, as seen in Fig. 18, indicating poor generalization. Similarly, while training accuracy improved steadily, validation accuracy stayed low and inconsistent Fig. 19.

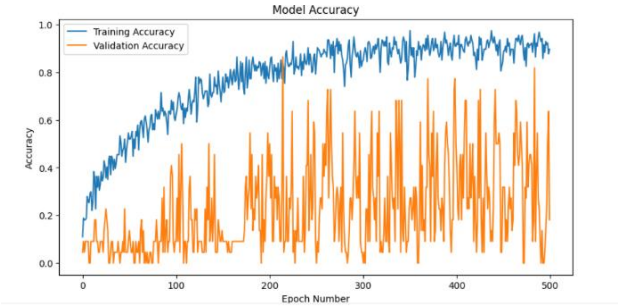

Fig 18. Training and Validation Loss of the Hybrid Model

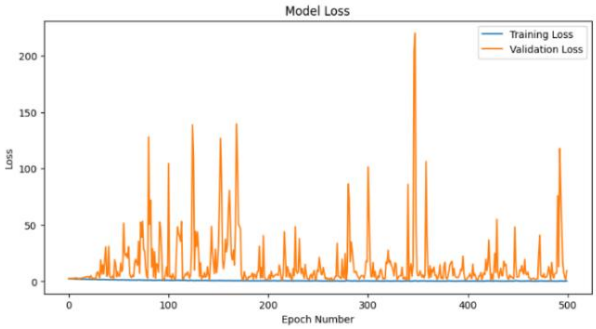

Fig 19. Training and Validation Accuracy of the Hybrid Model

3.1 Evaluation Metrics

In evaluating the models, several performance metrics were utilized to assets the model prediction and its effectiveness. Among these were mAP50-95, which assesses performances over many IoU thresholds, and mAP50, which calculates Mean Average Precision at an IoU threshold of 0.5. To evaluate deep learning model, implementating different performance metrics, will be explained in more depth while preserving the underlying context in the sections that follow.

3.1.1 Mean Average Precision (mAP)

Mean Average Precision (mAP) is commonly used for object detection tasks and measures the precision and recall across different levels of overlap between predicted and ground truth bounding boxes.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \qquad (Eq1)$$

Where the Intersection over Union (IoU) is calculated as the ratio of overlap to the union of predicted and true bounding boxes.

- **mAP@50**: The average of the Average Precision values at an IoU threshold of 0.50 for all classes.

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

$$\text{mAP@50} = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_{\text{i,IoU=0.50}} \quad (Eq2)$$

Where:

- N is the number of classes.
- $\text{AP}_{\text{i,IoU=0.50}}$ is the Average Precision for class iii at IoU = 0.50.
- **mAP@50:95**, also known as mAP@[0.50:0.95], represents the average precision computed at several IoU thresholds, from 0.50 to 0.95, with a step size of 0.05.

$$\text{mAP@[0.50:0.95]} = \frac{1}{\text{T}}\sum_{i=1}^{\text{T}}\left(\frac{1}{N}\sum_{i=1}^{N}\text{AP}_{\text{i,IoU=0.50}}\right) \quad (Eq3)$$

3.1.2 Weighted-Averaged Precision

Weighted Precision computes the precision for each class and takes the weighted average, considering the number of instances (support) for each class. The weighted precision is given by:

$$\text{Precision}_{\text{weighted}} = \frac{1}{\text{T}}\sum_{i=1}^{\text{T}}\left(\frac{\text{support}_i}{\text{total support}} \times \text{Precision}_i\right) \quad (Eq4)$$

Where:

- support_i is the number of true instances for class i.
- Total support is defined as:

$$\text{total support} = \sum_{i=1}^{N}\text{support}_{\text{i}} \quad (Eq5)$$

3.1.3 Weighted-Averaged F1 Score

The F1 score combines Precision and Recall into a single metric, especially useful when balancing these two measures. The weighted-averaged F1 score is calculated as:

$$\text{F1}_{\text{weighted}} = \sum_{i=1}^{\text{N}}\left(\frac{\text{support}_i}{\text{total support}} \times \text{F1}_i\right) \quad (Eq6)$$

3.1.4 Weighted-Averaged Recall

Weighted Recall computes the recall for each class and then takes the weighted average based on the number of true instances for each class. The weighted recall is defined as:

$$\text{Recall}_{\text{weighted}} = \sum_{i=1}^{\text{N}}\left(\frac{\text{support}_i}{\text{total support}} \times \text{Recall}_i\right) \quad (Eq7)$$

3.1.5 Loss Function

Loss functions play a critical role in training machine learning models, guiding the optimization process by quantifying the difference between the predicted outputs and the ground truth. For your project, which involves object detection using TensorFlow Object Detection API and YOLOv8

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

$$\text{Loss}_{\text{classification}} = -\sum_{i=1}^{C} y_i log(y_i) \quad (Eq8)$$

Where C is number of classes, y_i ground truth and y_i predicted probability for class i.

3.2 Real-time evaluation

As a part of this work experimental setup, we have used Python 3.10.11 along with TensorFlow at 2.15.1 and PyTorch at 2.2.1 on a system equipped with an 11th Gen Intel Core i5-1135G7 processor. Inference using the proposed model proved fast with an average image detection time of 0.12 seconds. This underlines that the inference stage can be considered as one of the key determinants of the performance of image processing pipeline. For testing the system in real-time situation, we connected the webcam to our custom-built models for hand gesture recognition. This made it possible to perform real time recognition and classify sign language gestures and so distinguishing its capability to act as a practical solution for communication and learning tools for communicate and ISLD or hard of hearing. Our proposed system executed better than conventional systems that have tradeoff issues or latency issues with real time gesture recognition systems to differentiate between dynamic gestures and static or complex. Offered a high detection accuracy rate without sacrificing more time, and produced minimal latency despite the variation of gestures' shapes or positions. This balance of speed and accuracy makes the system well suited to real-time applications where feedback is desirable; for example, sign language interpretation at a conference.

V. CONCLUSION

The comparative analysis of CNN, SSD MobileNet-V2 and YOLOv5n6u for custom sign language recognition revealed significant differences in performance across key metrics. In this work, various models were explored to develop an efficient system for recognizing sign language gestures. With an 99% mAP, the YOLOv5n6u model proved to be the most successful, proving that it could handle overlapping and obscured hand signals while still remaining computationally efficient, which made it appropriate for real-time applications. Additionally, SSD MobileNet demonstrated strong performance, obtaining 85% mAP and demonstrating its effectiveness as a lightweight, quick, and efficient gesture detection model. High-resolution inputs and architectural complexity, on the other hand, presented serious difficulties for the CNN and hybrid models, resulting in less accuracy and generalization compared to YOLOv5n6u and SSD MobileNet Models. The findings highlight how crucial model architecture, variety, and dataset quality are to attaining reliable performance. Future developments might increase recognition accuracy and generalization even further by growing the dataset, improving training methods, and improving model architectures. This would help the sign language recognition system find practical uses.

REFERENCES

[1] T.F. Dima, M.E. Ahmed, Using YOLOv5 algorithm to detect and recognize American sign language, in: 2021 International Conference on Information Technology, ICIT, IEEE, 2021, pp. 603–607.

[2] Mistree K, Thakor D, Bhatt B, Indian alphabets and digits sign recognition using pretrained model. In: smart intelligent computing and application, Volume 2. Springer; 2022, p. 13-20.

[3] Alam M, Tanvir M, Saha DK, Das SK, et al. Two dimensional convolutional neural network approach for real-time Bangla sign language characters recognition and translation. SN Comput Sci 2021;2(5):1-13. 10.1007/s42979-021-00783-6

[4] Angona TM, Shaon AS, Niloy KTR, Karim T, Tasnim Z, Reza SS, et al. Automated bangla sign language translation system for alphabets by means of MobileNet. TELKOMNIKA (Telecommun Comput Electron Control) 2020;18(3):1292-301. 10.12928/telkomnika.v18i3.15311

[5] Zakariah M, Alotaibi YA, Koundal D, Guo Y, Mamun Elahi M. Sign language recognition for Arabic alphabets using transfer learning technique. Comput Intell Neurosci 2022. **https://doi.org/10.1155/2022/4567989**

[6] J. Zhao, M. Glueck, S. Breslav, F. Chevalier, A. Khan, Annotation graphs: A graph based visualization for meta-analysis of data based on user-authored annotations, IEEE Trans. Visual. Comput. Graph. 23 (1) (2016) 261–270. https://doi.org/10.1109/TVCG.2016.2598543.

[7] J. Moehrmann, G. Heidemann, Efficient annotation of image data sets for com puter vision applications, in: Proceedings of the 1st International Workshop on Visual Interfaces for Ground Truth Collection in Computer Vision Applications, 2012, pp. 1–6. 10.1145/2304496.2304498

[8] R. Padilla, S.L. Netto, E.A. Da Silva, A survey on performance metrics for object detection algorithms, in: 2020 International Conference on Systems, Signals and Image Processing, IWSSIP, IEEE, 2020, pp. 237–242.

[9] M. Raza, F.K. Hussain, O.K. Hussain, M. Zhao, Z. ur Rehman, A comparative analysis of machine learning models for quality pillar assessment of SaaS services by multi-class text classification of users' reviews, Future Gener. Comput. Syst. 101 (2019) 341–371. 10.1016/j.future.2019.06.022

[10] W. Du, Y. Wang, and Y. Qiao. Rpan: An end-to-end recurrent pose- attention network for action recognition in videos. In Proceedings of the IEEE International Conference on Computer Vision, pages 3725–3734, 2017.

[11] Gebre, B.G., Wittenburg, P., Heskes, T.: Automatic sign language identification. In: 2013 IEEE International Conference on Image Processing. pp. 2626–2630. IEEE (2013)

[12] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In CVPR. 2014. https://doi.org/10.1109/CVPR.2014.214

[13] D. Kothadiya, C. Bhatt, K. Sapariya, K. Patel, A.-B. Gil-González, J.M. Corchado, Deepsign: Sign Language Detection and Recognition Using Deep Learning, Electronics. 11 (2022) 1780. https://doi.org/10.3390/electronics11111780.

[14] Karen S, Andrew Z. Very deep convolutional networks for large-scale image recognition. 2015, preprint at https://arxiv.org/abs/1409.1556.

[15] Daniels S, Suciati N, Fathichah C. Indonesian sign language recognition using yolo method. In: IOP Conference Series: Materials Science and Engineering. vol. 1077. IOP Publishing; 2021. p. 012029. 10.1088/1757-899X/1077/1/012029

[16] Abhishek B, Krishi K, Meghana M, Daaniyaal M, Anupama H. Hand gesture recognition using machine learning algorithms. Computer Science and Information Technologies. 2020;1(3):116-20

[17] AlSaedi AKH, AlAsadi AHH. A new hand gestures recognition system. Indonesian Journal of Electrical Engineering and Computer Science. 2020;18(1):49-55. http://doi.org/10.11591/ijeecs.v18.i1.pp49-55

[18] Kapuscinski T, Oszust M, Wysocki M, Warchol D. Recognition of hand gestures observed by depth cameras. International Journal of Advanced Robotic Systems. 2015;12(4):36. https://doi.org/10.5772/60091

[19] Vazquez Lopez I. Hand Gesture Recognition for Sign Language Transcription. 2017. https://doi.org/10.18122/B2B136

[20] Juneja S, Juneja A, Dhiman G, Jain S, Dhankhar A, Kautish S. computer Vision-Enabled character recognition of hand Gestures for patients with hearing and speaking disability. Mobile Information Systems. 2021;2021 **https://doi.org/10.1155/2021/4912486**