Real-Time Detection of Sign Language Gestures Using the YOLOv11 Object Detection Framework

Donga Saipavan *, Immaraju Srilekha†, G. Abhishikth‡, R Eswar Naik§, and Dr.Sunil CK¶
Department of Computer Science and Engineering, *Indian Institute of Information Technology Dharwad*
Dharwad, India
Email: *21bcs035@iiitdwd.ac.in, †21bec020@iiitdwd.ac.in, ‡21bcs042@iiitdwd.ac.in, §21bec035@iiitdwd.ac.in,
¶ORCID:

***Abstract*—The Sign language is a method of communication through visual hand gestures and signs, and it is generally used by individuals who have hearing, speech, or hearing problems or disabilities. To facilitate the communication with those individuals in the verbal communication in daily life, it's important to know the gestures they are using to communicate. Generally, people who do not have any idea about the sign language gestures in everyday life communication may not be able to comprehend what the gestures are.**

Our work deals with the real-time recognition of hand sign gestures on the American sign language detection system. So, the solution proposed by this study is to detect the gestures or signs of each alphabets. Although there are several existing methods suggested on applying the deep learning methods for sign language gesture recognition, The YOLO model, which has been working well in the real-time detection. Using machine learning and picture classification, computers can comprehend the sign language, which can then be translated by humans.

YOLO(You Only Look Once) YOLOv11 is a type of convolutional neural network (CNN) architecture, a highly advanced object detection algorithm, is used to achieve real-time and high-performance detection. The suggested method to detect the signs in American sign language, the efficiency of the model is higher than few of the other algorithms such as Traditional CNN with both real- time performance and correct hand sign prediction, independent of background. The current algorithms or models are based on machine learning algorithms, CNN algorithms, and some other methods; however, they are not as accurate as YOLO. Yolo v11 has been shown to be better than few other conventional algorithms and even its own previous versions in terms of accuracy as well as speed.

I. Introduction

Communication is the key to human interaction—it enables people to share their ideas, feelings, and maintain relations with each other. In the modern world where time is everything more crucial , good communication is more crucial than ever. For people with hearing or speech impairments, the conversations becomes tough. Sign language is very important in communication for people who suffer from hearing or speech disabilities, but the limited awareness of sign language among common people creates a huge barrier to communication with them.

Many do not know or understand or are not aware of sign language; therefore, a wide gap in communication between people with hearing or speech disability and the rest of normal people in daily life. It's Because they communicate only by using their hand gestures, facial expressions, and by through the body language also, this gap usually limit their abilities and opportunities and complicates, making them unable to fully express themselves in communicating with others.

Imagine that someone who is deaf is trying to ask directions or that a mute person trying to indicate an emergency in an emergency room. The challenge of not being able to communicate with others or understand them can thus be pitched in even the simplest activity. With the aim of not just fulfilling this need, the research and development world is on way to the evolution of sign language recognition systems that helps in translating the signs into text and spoken words, thus instilling much-needed inclusivity and accessibility to communication.

Different sign systems are heterogeneous and geographical like the spoken ones. American Sign Language (ASL) is in American and they possess its own syntax, grammar, and structure that make transcribing interpreters really difficult. This is where artificial intelligence (AI) and deep learning fit in. Methods such as Convolutional Neural Networks (CNNs) and YOLO(You Only Look Once) can analyze and interpret hand gestures with precision. With time and large datasets, the system keeps improving by learning to distinguish between different signing styles and regional distinctions. Apart from individual communication, a real-time sign language recognition system would be a game-changer in public places. Hospitals, banks, government offices, and transport hubs usually do not have sign language interpreters, so hearing-impaired people are left to fend for themselves. In schools, such a system would enable deaf students to be an active part of classrooms, interacting with teachers and fellow students without the need for a human interpreter or translator. In offices, it would enable professionals who are facing speech- or hearing-impairments to communicate better with coworkers.

Technology has revolutionized communication, with voice assistants, smart controls, and translators now integral to daily life. Yet, many of these innovations prioritize convenience over accessibility. Our focus on producing a model which can recognize hand gestures completely word by combining each gesture of the alphabets which could also make sentences.

Across the world, more than 3.6 percentage of the U.S. population, or about 11 million individuals are similarly

challenged in america alone. By developing AI-based sign language translation systems, we can create a world where communication is not a barrier but a bridge—one that unites people, bridges gaps, and induces inclusivity.

This study investigates the possibility of using the deep learning techniques and the image processing for creating an effective sign language recognition system. So, We intend to exploit the current technologies for creating a means by which the use of sign language can make interaction between the challenged people and the society as a whole as smooth and convenient as possible.

II. Literature Review

[1] an Indonesian Sign Language (BISINDO) real-time recognition system based on the YOLOv3 model. Vision-based sign language translation was their area of focus, which is more feasible and economical compared to hardware-based approaches such as gloves or sensors. The YOLO algorithm, which is CNN-based, was employed due to its speed and real-time object detection capability. The dataset was specifically designed with 4,547 images covering 24 static hand gesture classes (excluding motion-requiring letters). When tested on images, the model had 100 percent precision and recall. But for video data, the performance was different — with an F1-score of 84.38 percentage and a processing rate of 8 frames per second. This work emphasized YOLO's superiority in quick detection situations and showed how to successfully apply it for recognizing compound visual gestures using general hardware without special requirements.

[2] This research presented the YOLOv9 model for real-time detection of the American Sign Language (ASL). YOLOv9, published in 2024, is a state-of-the-art advancement of the YOLO family with enhanced accuracy and computational efficiency. The authors discussed the world communication gap experienced by the deaf and challenged people and highlighted the requirement of deep learning-based Sign Language Recognition (SLR). YOLOv9 adopts novel design elements like the Programmable Gradient Information (PGI) and Generalized ELAN (GELAN), which minimize information loss while training and keep significant gradients intact even for deep networks. The model is further aided by reversible operations that boost learning efficacy, particularly in thin networks. The system was evaluated on a data set of 26 ASL alphabets and demonstrated good real-time detection performance.

[3] This paper emphasises about a lightweight and efficient model based on YOLOv5 for hand gesture detection and American Sign Language (ASL) gestures. The authors utilized the MU HandImages ASL dataset with 2,515 images of hand gestures to represent alphabets and digits. The authors' YOLOv5-based model achieved outstanding performances on precision, recall. The model was shown to be effective for real-time recognition and can be deployed on mobile devices with its small model size (167 MB). The authors also critiqued previous hardware-based solutions, including gloves and Kinect sensors, which are accurate but not feasible for everyday use. Deep learning vision models like YOLO, however, provides speed, and accuracy without needing special hardware like other traditional CNN model. Their study validated that YOLOv5's efficiency in finger-spelling vocabulary interpretation using a single camera, bridging the communication gap for the hearing impaired.

[4] This paper presents an enhanced variant of YOLOv4-CSP for Turkish Sign Language hand gesture recognition. Through the addition of CSPNet, Mish activation function, and transformer block, the model enhanced learning efficiency and detection performance. In contrast to conventional CNN models with problems of gradient as the layers deepened, the proposed approach added improvements to preserve training stability and speed. The dataset comprised 1,500 hand-annotated Turkish signs data has been split accordingly into training, test, and validation sets. Their model performed remarkable results on precision, recall, all at the speed of 9.8 milliseconds per prediction. This makes the system eminently suitable for real-time scenarios. Their enhanced model surpassed other variations such as YOLOv3 and YOLOv4 in terms of speed and accuracy, even when subject to challenging backgrounds in the image datasets.

[5] Effective communication is key in everyday life, but for persons with hearing or speech impairments, a complete challenge. In view of this headache, a group headed by Adewale and Olamiti (2018) developed a system capable of converting American Sign Language (ASL) into text and speech using machine learning. Their focus of study was one in which hand gestures were captured, image segmentation done, and features extracted for matching and classification using FAST and SURF algorithms. Finally, K-Nearest Neighbor classification got used on the data set for checks on gesture recognition matching, against their code.

The results from the system achieved 92% success rate using supervised learning and 78% success rate using unsupervised learning, therefore, AI-based sign language translation could be very promising. Such a system could then be utilized in bridging the gap in communication for the hearing-impaired: allowing them to carry on with day-to-day interactions in a streamlined manner. The aforementioned technologies would allow sign language application to operate efficiently without a human interpreter in public venues of contact.

[6] The people facing the problems like hearing and speaking disabilities, the sign language is a crucial and helps for communication. However, most of the people does not know or understand the sign language, communication , makes everyday interactions for those people a big challenge. Addressing this issue, Seviappan et al. (2023) developed a deep learning system that translates these American Sign Language (ASL) signs into text. The approach in the paper combines MediaPipe for hand gesture detection with (RNN) and (LSTM) models to provide accurate gesture classification.

Thus, it can translate both static and dynamic extends, very crucial in capturing the complex things of American Sign Language (ASL). The LSTM model, designed for processing sequential data, provides an opportunity for learning long-term

patterns in hand gestures. The model has been trained and tested for the recognition of 26 different gestures with good accuracy, bridging the communication gap with challenged people.

[7] This paper highlights how using the deep learning towards helps in creating accessible technologies that have the potential to significantly improve the communication with the deaf and mute people. Translating sign language into text, helps the individuals to communicate more effectively in any environment ranging from classrooms to public places.

The people suffering with speech and hearing disabilities, communication would be a significant issue because of lack of knowledge about sign language to the people. To overcome this, Deshpande et al. (2023) proposed a sign language recognition system based on Convolutional Neural Networks (CNNs) to convert American Sign Language (ASL) signs or their gestures into the text. They recorded real-time hand movements through a camera, processes the images via filters, and classifies them via a CNN-based model.

For increasing accuracy, they have also applied Gaussian blur filtering and color segmentation to remove background noise or white noise and concentrate on hand movement. And also training the model with a large dataset helps in recognizing various ASL alphabets with good accuracy, therefore makes the communication simple for the deaf and mute impaired people in our society. The study in the paper, emphasizes there is need for utilizing the deep learning to develop solutions, bridging the gap between sign language users and with rest of the normal people in our society.

[8] The people having with deaf or speech-impaired, communication makes a daily challenge for them, particularly when we talk with each others and the people who does not have any knowledge about the sign language. To tackle this problem, a study has been presented at the the 12th International Conference on Cloud Computing, Data Science, and Engineering (2022) that incorporated the latest in image processing as well as machine learning to convert hand movements to text.

Their model in their research paper uses Convolutional Neural Networks (CNNs) to precisely identify most of the hand gestures, providing high accuracy in translation. Their model uses feature extraction and real-time processing to improve the efficiency and for the accuracy of sign language detection in the real time.

Their research highlights about the importance of the artificial intelligence as technologies in order to help in facilitating in communication for the deaf and mute people in the daily life. This could be useful for the deaf and disabled persons in many areas such as hospitals, schools, and workplaces, where accessibility is most important for this barrier-free communication.

[9] The sign language is a necessary thing for the communication among the people with hearing and speech disabilities, but most people do not know it, and therefore there are barriers to communication. In order to overcome these challenges faced by these people, the authors Vedak et al. (2019) suggested a sign language interpreter based on image processing and machine learning. Their system records hand gestures through the use of a webcam, analyzes the images through preprocessing methods such as edge detection and feature extraction, and identifies them through template matching. The identified text is then translated to speech for communication.

The model was trained using a set of 6,000 images of English alphabets with an accuracy of 88 percentage in identifying hand signs. By using Support Vector Machines (SVM) and Google's Text-to-Speech API, their ML model offers real-time sign language detection/Translation into the text and audio. The study shows how the ML/AI-driven tools can make things more easier for the hearing- or speech-impaired individuals to communicate more effectively in diverse real-world contexts.

[10] Communication is integral to human communication, but it becomes really hard for those having hearing and speaking disabilities due to the non-spread awareness about sign language. Sheela et al. (2022) addressed this issue by developing an American Sign Language (ASL) Translator, which translates voice into animated gesture signs and from animated gesture signs to voice. Their method applies speech recognition, text-to-speech conversion, and animation approaches to make signing users communicate directly with common users in real-time.

The system converts spoken words through a speech-to-text model, translates them into equivalent ASL gestures, and renders them as animations. On the other hand in their model, the ASL signs were detected and has been translated into text and speech also for the improvement, and accessibility. The dataset of this research consists of 76 ASL gestures and 100 animated sign language videos representing frequent words utilized in everyday conversations. This study indicates how important are the AI/ML based solutions in order to fill the communication gap and enhancing accessibility for hearing and speech-impaired individuals, especially in environments such as hospitals, offices, and public services.

[11] The Sign language detection/recognition and the translation are used to be the main fundamentals in making the smooth communication for deaf and speech-impaired people and it is an issue for those people. So, to address the issue, Natarajan et al. (2022) created an end-to-end deep learning system for sign language recognition, translation, and video synthesis. Their solution incorporates MediaPipe for pose detection and a CNN + Bi-LSTM hybrid model for text synthesis with more than 95 percentage classification accuracy. Moreover, they use Neural Machine Translation (NMT) and a Dynamic Generative Adversarial Network (GAN) to produce sign language videos from spoken sentences.

So, The H-DNA (Hybrid Deep Neural Architecture) framework suggested works well with multilingual and multimodal sign language data and it is a real-time solution for the communication. Large-scale testing with benchmark datasets showed better outcomes in recognition accuracy, translation quality, and video generation. The study indicates the potential of AI-based sign language processing to close the communication

gap and enhance accessibility for the challenged or impaired people.

[12] The traditional sign language detection/recognition systems usually depend on few computer vision techniques, which were also affected by various privacy issues. so, to address this, the authors had created EarHear in 2024, a new contactless CSLR and CSLT system that is based on acoustic-sensing technology and entirely avoids the cameras. Filtering out background noise through differential-Doppler signal processing, the system runs a Vision Transformer (ViT) model to make the model have high-efficiency gesture recognition. This is also integrated with a large-scale language model for translating sign language to natural text, and provides a much more precise and natural interaction.

The evaluation of EarHear conducted on 15 sentences of the Chinese sign language reveals that it has a recognition accuracy of 93.38 percentage and 80.73 percentage BLEU-1 translation score. Unlike conventional vision-based methods, this shows a higher robustness against light interference and thus an ability to work well in real-life settings. This research highlights an exciting opportunity towards acoustic-based sign language recognition that enables inclusivity and privacy-friendly efficient communication avenues for the hearing impaired community.

III. American Sign Language (ASL) Representation

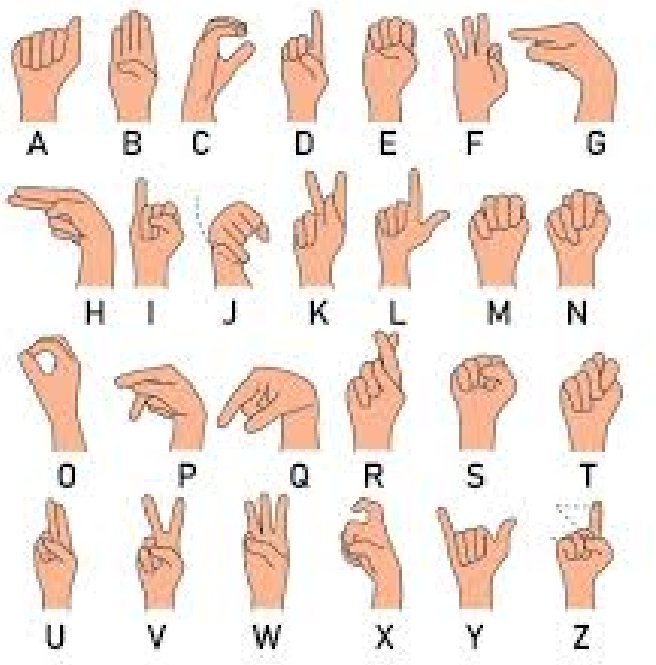

Fig. 1. Sign Language Gestures

American Sign Language (ASL) represents each letter of the English alphabet with a unique hand gesture. Users can spell words and communicate letter-by-letter using this technique, known as *finger-spelling*. This method is particularly useful for spelling names, technical terms, or words that lack specific ASL signs.

The ASL alphabet chart shown above in Fig. 1 [13] serves as a valuable reference for learners, providing a structured approach to mastering the fundamentals of sign language communication.

IV. Methodology

A. CNN

1) Dataset Used

The dataset for this project consists of images representing various English alphabet letters (A-Z) and digits (0-9) in sign language. All the images in the dataset have been scaled to a resolution of 300 × 300 pixels to enhance data extraction and comprehension.

2) Image Preprocessing

To improve the quality of images for further processing, the following preprocessing steps are applied:

- **Grayscale Conversion**: Convert images to grayscale to simplify the feature extraction process.
- **Noise Reduction**: Apply Gaussian blurring to reduce noise in the images.
- **Thresholding**: Use adaptive thresholding techniques to make the hand gesture stand out from the background.
- **Resize and Normalize**: Resize images to a consistent size (e.g., 128 × 128) and normalize pixel values (scale between 0 and 1) for the neural network to process them.

3) Data Splitting

During training our model, it's important to test how well it performs on new, unseen data. To do this, the dataset would be divided into two parts: one for training and another for testing. Training a model on the same data it's evaluated on wouldn't be realistic because, in real-world situations, the model needs to make predictions on entirely new inputs.

By splitting the data, we get a fair and unbiased way to measure the model's performance. Typically, most of the data is used for training only, while a smaller portion is set aside for testing. Randomly selecting data for both sets ensures they are similar, helping to minimize inconsistencies. Once the model is trained, it's tested on the separate dataset, where the correct answers are already known. This helps in determining how accurate the model's predictions are and whether it can effectively handle new data.

- **Training set:** In the supervised learning, each data point includes one or more input variables along with a corresponding output variable.
  For this project, the dataset contains **20,943 records**, which are split into training and testing sets. The test size is set to **0.2**, meaning **20%** of the data is reserved for evaluation, while the remaining **80%** is used for training. These records span across **10 different categories**, ensuring that the model learns to recognize a diverse range of patterns.
- **Testing set:** A separate collection of data points has been used to assess the performance of the model using predefined evaluation metrics. To ensure an **unbiased evaluation**, it is essential that the test set remains completely distinct from the training set. If training data were

included in the test set, it would be difficult to determine whether the model has truly learned to generalize or if it has simply memorized patterns from the training phase.

For this project, the dataset consists of 26,719 records, with 5,236 of them allocated to the test set (`xtest` and `ytest`), following a test size of 0.2. These records belong to 10 different animal categories, and a Convolutional Neural Network (CNN) model is employed to classify the animals based on the given data.

4) Convolutional Neural Network

A CNN is a type of artificial neural networks specifically designed to use easy numerical operations in data analysis and feature extraction on images. CNNs utilize convolutional layers in certain stages of the network in place of fully connected layers. These layers detect the significant image patterns and features, such as edges, textures, and shapes. The convolution operates on an input image across multiple layers, followed by an activation function: ReLU (Rectified Linear Unit)-to introduce non-linearity. Following this, there are further pooling layers to shrink the spatial sizes of images thus retaining only the critical information while avoiding the curse of overfitting. This hierarchically embedded set of features is recursively repeated until the last segment of the network, i.e., the fully connected layers, where classification takes place.

In every layer of a CNN, several filters, or kernels, move over the input image for some level of feature detection. During training, the CNN learns to generate the best set of these filter values so that it can detect certain features in images. In early layers, the network looks for simple textures and edges, and, as we move up, the network can combine these low-level features to represent high-level patterns, such as facial features or the general shape of an object. This hierarchical feature extraction allows CNNs to achieve local invariance-i.e., they are able to identify objects no matter their position in an image. The pooling layers demonstrate this by toning down the key feature responses, making the model relatively lung cancer-resistant to slight variations in object positioning. The other asset of CNNs is compositionality, whereby features from lower layers are combined to form higher-level representations.

The system architecture of a CNN consists of multiple layers. There are a few major components, that includes the Convolutional (CONV) Layer, Pooling (POOL) Layer, and Fully Connected (FC) Layer. The convolutional layer does its work of feature extraction. The pooling layer reduces the size of the image data set while preserving the most critical details. Finally, a fully connected layer takes the learned features and performs a final prediction, for example, cat or dog classification. A typical CNN architecture proceeds through the following structure: INPUT $\rightarrow$ CONV $\rightarrow$ ReLU $\rightarrow$ POOL $\rightarrow$ FC $\rightarrow$ SOFTMAX, which computes the final probability distribution over different classes.

The job of the convolutional layer is the very nucleus of CNN. This layer consists of small filters moving along the image, checking different small portions of the image and then extracting vital details. The filter performs an element-wise multiplication with the image pixels and sums these before creating a feature map to highlight where it has detected patterns. For example, the earlier filters may respond to simple patterns, such as hunter vertical or horizontal lines, while the deeper filter layers may recognize more complex patterns, such as eyes or wheels. This automatic detection of features makes CNNs very attractive when it comes to the detection of images.

One other key point making CNNs much more efficient is local connectivity and receptive field. Not every neuron gets connected to the whole image; rather, CNNs connect small small parts of the image, which consequently helps in the reduction of inter-connection and also makes it easier to train the network. Some specific portion of the image is called the receptive field–for example, if the input image is 32×32×3 (height ×width × depth) and the filter size is 3×3, each output neuron of the feature map would analyze a 3×3×3 portion of the image. Following that, as the filter gets applied more and more through the neural layer, the depth of each convolutional layer increases, helping the model learn richer and finer representations.

This systematic methodology enables CNNs to learn spatial hierarchies of features, thus rendering them very useful for image recognition, object detection, and even sign language interpretation. Through decomposition of an image into tiny meaningful parts and sequentially progressing to richer representations, CNNs have revolutionized visual data interpretation in machines, thus allowing for advancements in medical imaging, autonomous vehicles, and facial recognition.

B. YOLO: You Only Look Once

1) Dataset used

This dataset [14] consists of signs of alphabets i.e. from A to Z and 6 words i.e. Hello, I love You, Please, Yes, Thanks, No. There are images along with their respective labels in the data which are necessary for using yolo model, it helps in bounding the hand gestures from the image which also helps in focusing in the hand other than background and unnecessary noise. There are around 2000 images in the dataset along with the respective labels

2) Data Spliting

The dataset consists of about 1,850 labeled images, one for each of the unique data points our model will learn on. We also split the data into three portions: training, validation, and testing, making sure to separate it carefully for effective training and evaluation of performance.

In the dataset, a big part of 85 percent of the images was divided to the training set. A large part of data like this guarantees that the model is provided with sufficient variety and volume to effectively learn patterns. With this training set, we created a solid model, eventually saved as the best.pt file, that identifies the most accurate and optimized weights during learning.

Second, 10 percent of the dataset has been for validation. This part is important in tracking the model's performance

during the training, and it aids in measuring important metrics like validation loss, class loss, mean Average Precision (mAP), and even heat maps, which helps in visually represent how well the model is learning to detect or classify.

Finally, 5 percent of the dataset was kept solely for testing. This last set of images will not be visible by the model at all during training or validation. It gives us an unbiased measure of how well the model can generalize to totally unseen data, and thus provides us with a realistic estimate of its performance in real-world use.

By adhering to this division, we maintained a balanced and efficient training pipeline that enables maximum learning with minimal overfitting, culminating into a well-tuned and reliable model.

C. YOLO model

The YOLO model is among the most widely used and strongest algorithms in real-time object detection. Unlike other traditional object detection systems that classify and localize in independent stages, YOLO solves this problem as a single problem of regression. It scans the whole image in a single pass when using a convolutional neural network (CNN) and predicts bounding boxes and class probabilities directly. This renders YOLO much faster than two-stage detectors such as R-CNN or Fast R-CNN, which have to make multiple passes over an image.

YOLO is exceptionally adept at usage when rapid detection is needed while preserving accuracy, i.e., use in driving on its own, surveillance equipment, and in our scenario, recognition of sign language. Localizing and identifying the objects is made rapidly by it, with minimum usage, rendering it best used for translation from hand motions to text real time.

1) How YOLO works

The entire thought process about YOLO lies in separating the input image into a cell-based grid. Every cell is tasked with detecting objects appearing in its region. More precisely, every grid cell produces a constant number of bounding boxes, their confidence, as well as class probabilities. A bounding box has four elements—center x, center y coordinates, width, and height, all normalized w.r.t. the image size. The confidence levels reflects the probability that the bounding box predicted contains an object and the precision of the bounding box.

For each object that is detected, YOLO computes a final score as the product of the confidence score and the class probability. This enables the model in determining both what object it is (classification) and where it is detected in the image. Moreover, YOLO applies a method known as Non-Maximum Suppression (NMS) in order to remove overlapping bounding boxes and keep only the most probable prediction per object.

2) YOLO for Object Detection of Sign Language

In our sign language translation project, object detection is important in separating the hand gesture from the rest of the image. This enables the model to concentrate on the part of the visual input that matters and disregard irrelevant background noise. The YOLO model was trained to recognize

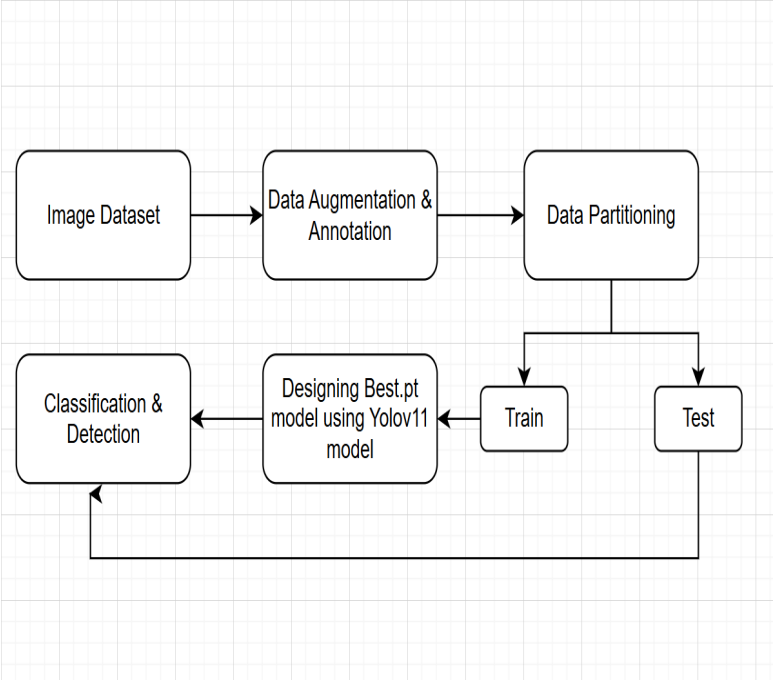

Fig. 2. YOLO Model WorkFlow

both alphabets (A to Z) and particular words like ”Hello,” ”I Love You,” ”Please,” ”Yes,” ”Thanks,” and ”No.”

To do this, we built a bespoke dataset of around 2,000 images, each annotated with a bounding box and its respective gesture class. They are necessary for training YOLO since they contain the ground truth data necessary to instruct the model on how to precisely detect and classify the gestures. Each label file contains the class index and the bounding box coordinates, allowing the model to learn both the appearance and location of each sign.

The bounding boxes on the hand region helps in ignoring all the irrelevant elements in the background. This not only improves detection accuracy but also enhances the model's robustness to variations in lighting, hand size, and background clutter.

3) Variants of YOLO and How They Have Evolved

Since its release in 2015, YOLO has had numerous versions, each succeeding the other by enhancing accuracy, speed, and architecture.

YOLOv1 (2015): The original YOLO model brought in the idea of single-shot detection of objects. Though the algorithm was speed, it was not too accurate, particularly when it came to detecting smaller objects.

YOLOv2 and YOLOv3: These models have enhanced the accuracy in detecting the signs, brought a few improved networks (such as Darknet-53 and etc), and enabled multi-scale detection, which helped in detecting small and large objects more efficiently.

YOLOv4: Based on YOLOv3 with added features such as the CSPNet backbone, PANet, and the Mish activation

function. This model greatly improved detection performance without compromising speed.

YOLOv5: Created by Ultralytics, this iteration gained popularity because it was easy to use, modular, and PyTorch-integrated. It added several model sizes (s, m, l, x) to trade off accuracy and speed for various hardware.

YOLOv8 and later (YOLOv11): More recent versions such as YOLOv8 and YOLOv11 added transformer-based features, improved feature extraction methods, and optimized training methods. They not only support object detection but also segmentation and tracking, making them extremely versatile in real-world applications. But the Yolov11 version was most accurate among all the previous versions.

Model Size Variants: n, s, m, l, xl Each YOLO version (v8 and v11) provides different model sizes, denoted as n (nano), s (small), m (medium), l (large), and xl (extra large), to suit different hardware and application requirements.

Trade-offs:

- **N and S models**: Faster, but may struggle with highly intricate gestures or background clutter.
- **M models**: Strike a balance, suitable for real-time detection with reasonable accuracy.
- **L and XL models**: Provide superior accuracy, especially for finger spelling, complex signs, or multi-hand coordination, but require more resources.

V. Results

A. CNN

The CNN model has been trained for about 20 epochs using a data set that included 9,583 training samples and 4,107 validation samples. The model showed high learning ability right from the initial steps of training, as it was able to reach a validation accuracy of approx. 88 percent during the first epoch only.

Throughout the training, the model had kept on getting better, and at the 20th epoch, it had achieved a training accuracy of around 97.45 percent and a validation accuracy of 99.88 percent, showing great generalization on new data. The associated validation loss also kept on decreasing, reaching 0.0038, which shows a sign of high confidence and low error in prediction.

This performance demonstrates that the CNN architecture was successful in the sign language detection task, because it can learn automatically spatial hierarchies of features from the input images. From the small gap between the training and validation accuracies indicates that the model did not overfit to the training set very well.

Important Observations:

- High validation accuracy in early epochs
- Final validation accuracy at the end: 99.88 percent
- Final validation loss at the end: 0.0038
- No major overfitting detected.
- The model is appropriate for real-time sign detection system deployment.

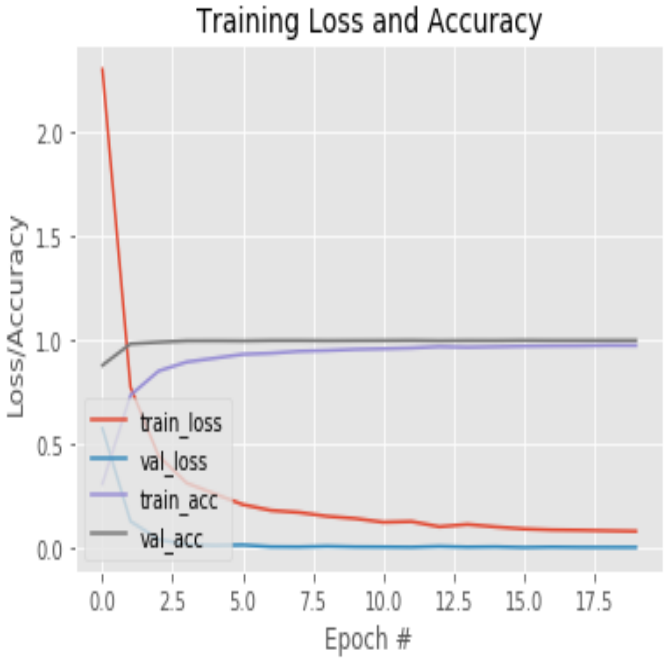

Fig. 3. Training LOss & Accuracy

B. YOLO

The system was trained and tested using the YOLOv11L architecture, which performed excellently on most of sign the language gestures, both alphabets and general phrases. The model achieved a mean Average Precision (mAP@0.5) of 96.8 percentage and a mAP@0.5:0.95 of 80.9 percentage respectively, which indicates high object localization and classification accuracy.

The model was tested on a validation set of 168 images with 168 labeled gestures. It gained an overall precision of 95.3 percent and recall of 92.5 percent respectively, showing the high confidence as well as resilience in predictions.

Class-wise performance individually demonstrates uniformly high mAP values for the majority of classes. The notable results are:

- Alphabet signs like ‘A’, ‘B’, ‘C’, ‘D’, and ‘E’ all attained **mAP@0.5** greater than 99%, and **mAP@0.5:0.95** ranging between 0.774 and 0.895.
- Custom sign gestures such as ‘Hello’, ‘Thank You’, ‘Yes’, and ‘Please’ also performed exceptionally well, with **mAP@0.5** values of 99.5% and **mAP@0.5:0.95** values over 0.80.
- One of the poor-performing classes was ‘I’, which had a low **mAP@0.5** of 54.0%, likely due to sparse or unclear training data, indicating a potential area for dataset refinement.
- The model had a real-time inference rate of around 15 ms per image, making it deployable in interactive scenarios such as real-time sign language translation or detection.

These findings show the ability of the model to identify and classify various sign gestures accurately with high precision,

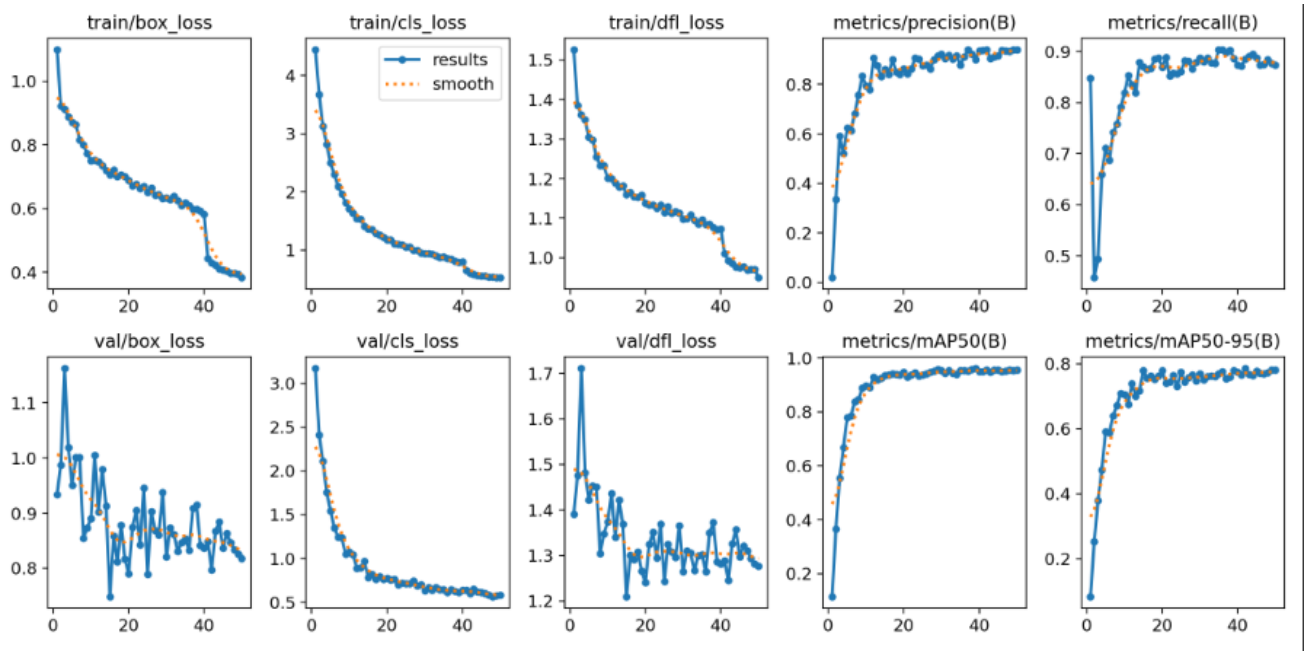

Fig. 4. Training loss and evaluation metrics of the YOLOv11 model across 50 epochs.

rendering it a viable solution for real-world sign language recognition systems.

TABLE I
YOLOv11 Detection Performance for Sign Language Dataset

Class	Precision	Recall	mAP@0.5	mAP@0.5:0.95
A	0.963	0.800	0.906	0.774
B	1.000	0.688	0.995	0.836
C	0.907	1.000	0.995	0.831
D	1.000	0.900	0.995	0.862
E	0.985	1.000	0.995	0.895
F	0.860	1.000	0.995	0.863
G	0.985	1.000	0.995	0.843
H	0.994	1.000	0.995	0.792
I	0.910	0.500	0.540	0.486
J	0.960	1.000	0.995	0.705
K	0.992	0.833	0.955	0.855
L	0.853	0.750	0.945	0.850
M	0.985	1.000	0.995	0.810
N	0.971	1.000	0.995	0.872
O	0.987	1.000	0.995	0.793
P	0.978	0.857	0.953	0.746
Q	0.984	1.000	0.995	0.866
R	1.000	0.947	0.995	0.854
S	0.944	1.000	0.995	0.895
T	0.813	0.833	0.927	0.834
U	0.981	1.000	0.995	0.855
V	0.756	0.622	0.880	0.767
W	0.867	1.000	0.995	0.843
X	0.908	1.000	0.995	0.895
Y	0.981	0.875	0.982	0.749
Z	0.984	1.000	0.995	0.838
Hello	0.982	1.000	0.995	0.834
Thanks	0.991	1.000	0.995	0.807
Please	1.000	1.000	0.995	0.809
Yes	0.983	1.000	0.995	0.810
No	0.984	1.000	0.995	0.627
IloveYou	0.994	1.000	0.995	0.796
Overall	**0.953**	**0.925**	**0.968**	**0.809**

The confusion matrix illustrates that the model's performance in identifying alphabetic signs (A–Z) as well as frequent sign language words like Hello, Iloveyou, No, Please, Thanks, Yes, and a background class for non-sign inputs. The x-axis is for true labels and the y-axis for predicted labels. A dense array of darker squares along the diagonal suggests high model accuracy, and that the majority of signs were well-classified. Misclassifications are indicated by lighter squares away from the diagonal, depicting cases where some signs, like Hello, might have been mistaken with Thanks. The background class was also well-separated, affirming the soundness of the model in differentiating sign gestures from irrelevant inputs. In general, the matrix points out the strength of the model as well as areas where improvement is needed.

The visualization give information regarding the distribution and spatial nature of the dataset applied for sign language detection. The top-left bar plot indicates instances per class and reflects a pretty balanced dataset among most sign classes (alphabets and universal gesture words), with a sharp decline in the number of samples for some classes after index 25—perhaps special gestures or background class. The right top subplot presents bounding box annotations over the whole dataset, with the majority of gestures being centralized but with varying size, reflecting multiple hand placements and scales. The bottom-left heatmap demonstrates spatial distributions of centers of bounding boxes (x, y) where they cluster at the center of the image, implying that signs most frequently occur at the middle of the image. The bottom-right heatmap illustrates bounding box width and height distribution, with a positive visualization between the two as well as a depiction of natural size variations in gestures. Collectively, these plots confirm the consistency of the dataset, enabling successful

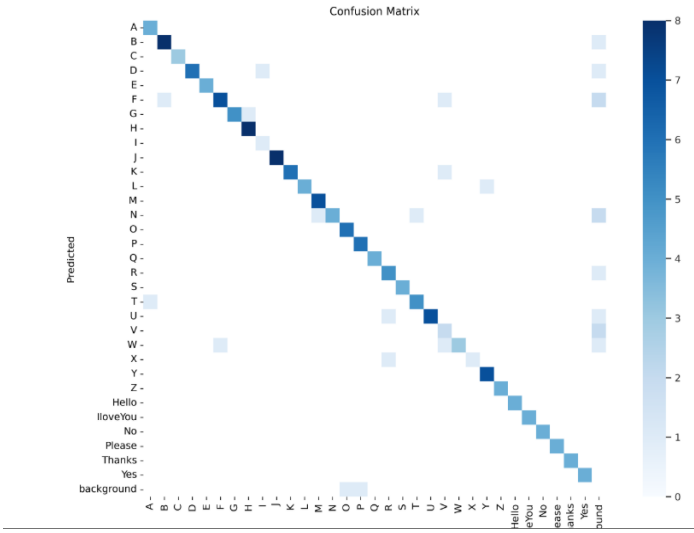

Fig. 5. Confusion Matrix for Sign Language Detection Using YOLOv11

model training.

VI. Comparision

In the Approach and Architecture, CNNs are usually applied for image classification, i.e., they classify the whole image into a single class. In sign language detection, this can be utilized to classify an image of a particular hand gesture. Yet, CNNs do not localize the gesture (i.e., where it is in the image). In YOLOv11, it is a one-stage object detection model. It not just classifies what gesture is occurring but also locates the hands precisely (bounding box). That makes it stronger for applications that require multiple hands, gestures, or continuous detection (such as in video).

YOLOv11 offers in both classification and detecting images in the bounding box, whereas CNNs offer only classification.

In terms of Real-Time Detection, CNN:

CNN models usually have a multi-stage pipeline (classification, etc), which adds latency at inference. As far as sequential processing and performance optimization are concerned, CNNs are generally slower and less responsive for real-time gesture recognition.

YOLOv11: YOLOv11 detects with the detection architecture feeding the full image in one pass, which actually reduces latency considerably. YOLOv11 is much faster and more efficient for real-time detection.

In Handling the Multiple Gestures

CNN: Restricted to a single class per image. If there are multiple gestures, it cannot recognize or localize them. YOLOv11:

Can recognize multiple hand signs within one frame, even though they may overlap or occur in various parts of the body. This is particularly important for sign language where both hands are used or group communication. OLOv11 is very efficient in multi-object and multi-label detection, as opposed to CNNs.

While discussing the Advanced Features of YOLOv11

YOLOv11 features improvements such as: Context-aware detection: improved occlusion and light finger movement handling.

Multi-scale detection: detects fine hand movements and small gestures more precisely.

Feature	CNN	YOLOv11
Gesture Localization	✗ No	✓ Yes
Real-Time Inference	Slower	✓ Fast
Multiple Gesture Detection	✗ Limited	✓ Strong
Backbone Architecture	CNN only	Transformer + CNN Hybrid
Complexity of Gestures	Average	✓ Handles complex signs

TABLE II
Comparison between CNN and YOLOv11 for Gesture Recognition

VII. Key Contributions

In this project, we gave some of the key contributions: First, we made our own custom dataset by combining 26 alphabets letters and 6 commonly used words in sign language with different types of representations of how to do the hand gestures (as we used various forms and provided visually different examples). Based on the collected dataset, we trained a deep-learning model that was capable of classifying both the alphabets and words made from a static hand gesture. This was accomplished despite using a relatively small dataset because we used an effective pre-processing and robust optimization process. In addition to the above, we tested the conventional method using a Convolutional Neural Network (CNN) and compared it to the YOLO (You Only Look Once) model's framework and the analysis we conducted provided comparisons for detection accuracy and real-time performance.

VIII. Conclusion

In our study of sign language detection, the CNN and YOLO models offer strong backbones to create real-time gesture recognition systems. YOLOv11, with transformer boosts, reveals new avenues for advanced gesture modeling, While CNNs are well suited for simple gesture classification applications, they are wanting in real-time, multi-hand, and continuous detection applications. YOLOv11 is obviously better suited for sign language detection because it has High-speed, inference, Can detect and multiple signs, has a superior edge in real time detection, and for detecting complex gesture nuances

Therefore, YOLOv11 is a more powerful and scalable solution for real-world sign language detection systems.

References

[1] S. Daniels, N. Suciati, and C. Fathichah, "Indonesian sign language recognition using yolo method," in *IOP Conference Series: Materials Science and Engineering*, vol. 1077, no. 1. IOP Publishing, 2021, p. 012029.

[2] A. Imran, M. S. Hulikal, and H. A. Gardi, "Real time american sign language detection using yolo-v9," *arXiv preprint arXiv:2407.17950*, 2024.

[3] T. F. Dima and 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.

[4] M. Alaftekin, I. Pacal, and K. Cicek, "Real-time sign language recognition based on yolo algorithm," *Neural Computing and Applications*, vol. 36, no. 14, pp. 7609–7624, 2024.

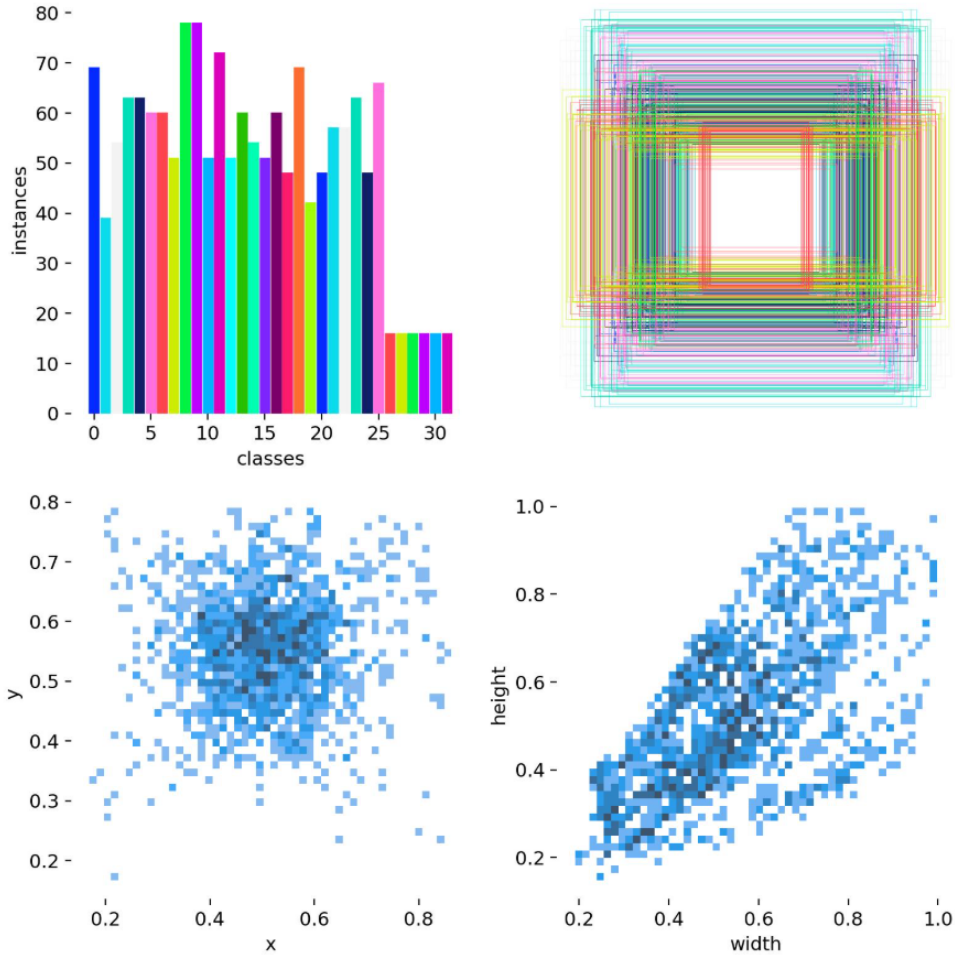

Fig. 6. Visualization of Dataset Annotation

[5] V. Adewale and A. Olamiti, "Conversion of sign language to text and speech using machine learning techniques," *Journal of research and review in science*, vol. 5, no. 12, pp. 58–65, 2018.

[6] A. Seviappan, K. Ganesan, A. Anbumozhi, A. S. Reddy, B. V. Krishna, and D. S. Reddy, "Sign language to text conversion using rnn-lstm," in *2023 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI)*. IEEE, 2023, pp. 1–6.

[7] A. Deshpande, A. Shriwas, V. Deshmukh, and S. Kale, "sign language recognition system using cnn," in *2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*. IEEE, 2023, pp. 906–911.

[8] V. Gupta, M. Jain, and G. Aggarwal, "Sign language to text for deaf and dumb," in *2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 2022, pp. 384–389.

[9] B. K. Akshatharani and N. Manjanaik, "Sign language to text-speech translator using machine learning," *International Journal of Emerging Trends in Engineering Research*, vol. 9, no. 7, 2021.

[10] B. D. Patel, H. B. Patel, M. A. Khanvilkar, N. R. Patel, and T. Akilan, "Es2isl: an advancement in speech to sign language translation using 3d avatar animator," in *2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 2020, pp. 1–5.

[11] D. Das Chakladar, P. Kumar, S. Mandal, P. P. Roy, M. Iwamura, and B.-G. Kim, "3d avatar approach for continuous sign movement using speech/text," *Applied Sciences*, vol. 11, no. 8, p. 3439, 2021.

[12] D. Abdulla, S. Abdulla, R. Manaf, and A. H. Jarndal, "Design and implementation of a sign-to-speech/text system for deaf and dumb people," in *2016 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA)*. IEEE, 2016, pp. 1–4.

[13] T. Bihus, "Dactylic alphabet, asl alphabet vector illustration," https://www.vecteezy.com/vector-art/37899180-dactylic-alphabet-asl-alphabet-vector-illustration, 2025, accessed: 2025-04-18.

[14] A. S. L. YOLOv8, "Sign language detection dataset," https://universe.roboflow.com/american-sign-language-yolov8/sign-language-detection-ucv5d, mar 2024, visited on 2025-04-18. [Online]. Available: https://universe.roboflow.com/american-sign-language-yolov8/sign-language-detection-ucv5d

[15] P. Kapoor, R. Mukhopadhyay, S. B. Hegde, V. Namboodiri, and C. Jawahar, "Towards automatic speech to sign language generation," *arXiv preprint arXiv:2106.12790*, 2021.

[16] M. K. Mounika, B. Hemalatha, M. S. Mounka, and J. Sumanth, "Speech/text to sign language convertor using nlp," *UGC care Group I journal*, vol. 12, no. 08, 2022.

[17] A. S. M. Miah, M. A. M. Hasan, S. Nishimura, and J. Shin, "Sign language recognition using graph and general deep neural network based on large scale dataset," *IEEE Access*, 2024.

[18] D. R. Kothadiya, C. M. Bhatt, H. Kharwa, and F. Albu, "Hybrid inceptionnet based enhanced architecture for isolated sign language recognition," *IEEE Access*, 2024.

[19] D. R. Kothadiya, C. M. Bhatt, T. Saba, A. Rehman, and S. A. Bahaj, "Signformer: deepvision transformer for sign language recognition," *IEEE Access*, vol. 11, pp. 4730–4739, 2023.

[20] B. Natarajan, E. Rajalakshmi, R. Elakkiya, K. Kotecha, A. Abraham, L. A. Gabralla, and V. Subramaniyaswamy, "Development of an end-to-end deep learning framework for sign language recognition, translation, and video generation," *IEEE Access*, vol. 10, pp. 104 358–104 374, 2022.

[21] L. Thomas, "Audibly: Speech to american sign language converter," *Authorea Preprints*, 2023.

[22] A. Shinde and R. Dandona, "Two-way sign language converter for speech-impaired," *International Journal of Engineering Research & Technology*, vol. 9, no. 2, pp. 647–648, 2020.

[23] M. M. Chandra, S. Rajkumar, and L. S. Kumar, "Sign languages to speech conversion prototype using the svm classifier," in *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*. IEEE, 2019, pp. 1803–1807.

[24] V. Aiswarya, N. N. Raju, S. S. J. Joy, T. Nagarajan, and P. Vijayalakshmi, "Hidden markov model-based sign language to speech conversion system in tamil," in *2018 Fourth International Conference on Biosignals, Images and Instrumentation (ICBSII)*. IEEE, 2018, pp. 206–212.

[25] A. Abraham and V. Rohini, "Real time conversion of sign language to speech and prediction of gestures using artificial neural network," *Procedia computer science*, vol. 143, pp. 587–594, 2018.

[26] A. P. Neog, A. Kalita, and M. N. Pandiyarajan, "Speech/text to indian sign language using natural language processing," *Indian Journal of Computer Science and Engineering (IJCSE)*, vol. 14, no. 3, 2023.

[27] J. Peguda, V. S. S. Santosh, Y. Vijayalata, A. Deepa, and V. Mounish, "Speech to sign language translation for indian languages," in *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1. IEEE, 2022, pp. 1131–1135.

[28] M. K. NB, "Conversion of sign language into text," *International Journal of Applied Engineering Research*, vol. 13, no. 9, pp. 7154–7161, 2018.

[29] P. Vijayalakshmi and M. Aarthi, "Sign language to speech conversion," in *2016 international conference on recent trends in information technology (ICRTIT)*. IEEE, 2016, pp. 1–6.

[30] H. ZainEldin, S. A. Gamel, F. M. Talaat, M. Aljohani, N. A. Baghdadi, A. Malki, M. Badawy, and M. A. Elhosseini, "Silent no more: a comprehensive review of artificial intelligence, deep learning, and machine learning in facilitating deaf and mute communication," *Artificial Intelligence Review*, vol. 57, no. 7, p. 188, 2024.

[31] V. Harit, N. Sharma, A. Tiwari, A. K. Yadav, and A. Chauhan, "Speech-to-sign language translator using nlp," in *Federated learning for Internet of Vehicles: IoV Image Processing, Vision and Intelligent Systems*. Bentham Science Publishers, 2024, pp. 301–313.

[32] P. Sharma, D. Tulsian, C. Verma, P. Sharma, and N. Nancy, "Translating speech to indian sign language using natural language processing," *Future Internet*, vol. 14, no. 9, p. 253, 2022.

[33] H. Monga, J. Bhutani, M. Ahuja, N. Maid, and H. Pande, "Speech to indian sign language translator," in *Recent Trends in Intensive Computing*. IOS Press, 2021, pp. 9–15.

[34] A. Ojha, A. Pandey, S. Maurya, A. Thakur, and P. Dayananda, "Sign language to text and speech translation in real time using convolutional neural network," *Int. J. Eng. Res. Technol.(IJERT)*, vol. 8, no. 15, pp. 191–196, 2020.

[35] A. Kamble, J. Musale, R. Chalavade, R. Dalvi, and S. Shriyal, "Conversion of sign language to text," *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 11, no. 5, 2023.

[36] G. B. Regulwar, N. Gudipati, M. Naik, N. Kathirisetty, L. Madiga, and E. R. Kumar, "Audio to sign language translator," in *2023 2nd International Conference on Ambient Intelligence in Health Care (ICAIHC)*. IEEE, 2023, pp. 1–6.

[37] Y. Grover, R. Aggarwal, D. Sharma, and P. K. Gupta, "Sign language translation systems for hearing/speech impaired people: a review," in *2021 International Conference on Innovative Practices in Technology and Management (ICIPTM)*. IEEE, 2021, pp. 10–14.