Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 3, Issue 6, 2025

HAND GESTURE TO VOICE CONVERSION USING ARTIFICIAL INTELLIGENCE

Awais Ahmad[1], Khalid Manzoor[2], Muhammad Suliman[3], Muzammil Islam[4], Bilal Ur Rehman[*5], Humayun Shahid[6], Muhammad Amir[7], Kifayat Ullah[8]

[1,2,3,4,*5,7,8]*Departement of Electrical Engineering, University of Engineering and Technology, Peshawar, Pakistan*
[6]*Departement of Telecommunication Engineering, University of Engineering and Technology, Taxila, Pakistan*

DOI: https://doi.org/10.5281/zenodo.15709950

Keywords
Machine Learning (ML), Speech recognition, Mean Average Precision.

Article History
Received on 12 May 2025
Accepted on 12 June 2025
Published on 21 June 2025

Corresponding Author:
Bilal Ur Rehman

Abstract
Communication forms one of the basics of interaction among people but becomes a significant problem for deaf and mute individuals. Conventional approaches, such as sign language and lip reading, tend to be restricted regarding availability and precision. The proposed project will close this communication gap with the help of an image-processing-based hand gesture recognition system. This system allows hand signals to be captured by a webcam, converted to text, and then to speech, providing a readily understandable input/output medium. This work captures the hand gestures, preprocesses the model to be invariant to lighting and background variations, and tests with measures such as IOU (Intersection Over Union), MAP (Mean Average Precision), MAE (Mean Absolute Error), and RMSE (Root Mean Square Error). Further, the proposed approach provides better results for hand gestures using image processing by employing deep-learning algorithms for feature extraction and real-time recognition.

1. INTRODUCTION

Nonverbal communication is essential in our daily contact, conveying around 65% of messages, whereas verbal communication constitutes about 35% of our exchanges [1]. Gestures can be categorized into hand and arm gestures, necessitating the recognition of hand positions, sign languages, and amusement applications. Head and facial motions encompass nodding or shaking, directing eye gaze, and opening the mouth to articulate, winking, and analogous acts. Body gestures encompass the movement of the whole body. Body gestures encompass the movement of the whole body. Practical and dependable gesture detection algorithms are essential for successful human-computer interaction (HCI). These identification methods are alternatives to commonly used HCI devices such as the mouse and keyboard [2]. Hand gesture recognition is a significant area of research within automated recognition systems and is crucial for Human-Computer Interaction (HCI). Hand gesture recognition is highly beneficial for applications requiring genuine human-machine interaction. The advancement of hand gesture detection systems, including applications for sign language, is crucial for overcoming the communication barrier faced by individuals unfamiliar with sign language. Automated technology that translates hand gestures into written or spoken language can help mitigate this communication barrier for persons unfamiliar with sign language. Vision-based hand gesture recognition systems are advantageous in communication, education, and rehabilitation domains [3]. The technology may also aid when a human interpreter cannot translate sign language. Hand gesture detection is a complex challenge due to several

factors. Several factors were overlooked during the development of the hand gesture detection system, including environmental noise, heterogeneity among signers, and linguistic variety. These elements should be included to mitigate issues in the segmentation and tracking process [4].

2. LITERATURE REVIEW

Hand gesture identification is the technology that interprets hand movements utilized in the signing process into a result, text, or sound. The classification of the sign language hand gestures can also be characterized by the technique utilized in recording them. Among them is the vision-based system, where one or more cameras are used to detect the gestures, and the device-based system, where special electronic gloves with sensors are used to interface the user with the system. The efficiency of device-based solutions is well-known. However, the practical applicability of these solutions is limited by the fact that a person has to wear a rather large device when talking to the system. However, systems powered by vision do not have this problem, enabling users to engage with the system in an even more natural manner. Concerning its application, it provides a broader array of usage in outside environments. The intuitive design of the vision-based system was evaluated using datasets comprising dynamic hand motions in sign language, encompassing both isolated and continuous signals [5]. The dense vectors encapsulate the intricate semantic information in sign language video frames. The decoder module inputs dense vectors and generates the intended spoken text. The framework of Sign Language Translation (SLT) consists of three parts. The spatial and word-embedding components convert sign language video frames and spoken text into feature or dense vectors. The encoder-decoder module forecasts the spoken text and adjusts its transmission characteristics using reverse propagation to reduce the divergence between the intended and output messages. The spatial embedding level extracts features from the visual data of sign language, resulting in several network designs, including 2D-CNN [6], 3D-CNN [7], and GCN [8]. Moreover, the multi-cue network has been employed particularly for visual data in sign language and its applicability in generic systems. The several components of the hint, including facial expressions, bodily posture, and gestural movements, are amalgamated by pertinent fusion algorithms and subsequently processed through the tokenization layer.

Signed language is a unique visual medium utilized by individuals who are congenitally deaf (congenital DHH) or who have experienced hearing loss later in life (acquired DHH) [9]. It entails the utilization of manual gestures and facial expressions to enhance visual communication. The manual information relates to the hands' shape, orientation, position, and movement. In contrast, the non-manual material includes body position, arm movements [10], eye gaze, lip configuration, and facial expressions [11]. Sign language does not provide a direct and precise translation of spoken discourse. It has independent grammar, semantic organization, and unique linguistic logic [12]. Varying hand and body movements correspond to distinct units of significance. The World Federation of the Deaf's figures indicate that there are around 70 million individuals who are deaf or hard of hearing, and there are more than 200 distinct sign languages worldwide [13]. Hence, enhancing the translation technology for sign language might effectively close the communication divide between those who are deaf or hard of hearing (DHH) and those who are not. The approach presented by the author in [14] employs a Faster R-CNN model to localize and recognize hand gestures in sign language films. This method integrates a 3D convolutional neural network (3D-CNN) with a Long Short-Term Memory (LSTM) based encoder-decoder architecture to recognize sign language. The 3D-CNN accurately identifies and acknowledges hand motions in the video, while the LSTM-based encoder-decoder architecture manages the time-related connections seen in sign-language sequences. This combination enables precise identification of both the spatial position and semantic significance of manual motions in sign language movies. Although many previous studies concentrate on identifying individual gestures, their practicality in real-world situations is still restricted. The author in [15] conducted a comprehensive assessment of the most advanced methods employed in the latest studies on hand motion and recognition of signs. This

evaluation encompassed several aspects, including data collecting, pre-processing, segmentation, feature extraction, and classification. Analysed was a detailed evaluation of the cutting-edge approaches utilized in present-day studies on hand-gestures and sign language identification. In [16], the author performed a comprehensive analysis of scholarly articles in the field of sign language recognition (SLR) that had been published from 2007 to 2017. Their analysis examined the investigation landscape across six important aspects: methods for gathering data, differentiation between static and dynamic signs, signing mode (individual or two-handed), the use of single or double hands for forming gestures, using methods to categorize data, as well as levels of recognition accuracy.

In a recent study, Aloysius and Geetha [17] conducted a review of the continuous sign-language recognition (CSLR) system that relies on vision-based methods. Additionally, the author in another study [18] specifically examined suggested models for sign-language recognition that also rely on vision-based techniques. Researchers have identified a vacuum in previous studies since they did not analyze the limitations and future direction of vision-based hand gesture detection systems. The present investigation aims to fill this gap by conducting a comprehensive analysis of the available literature to assess the advancements made in vision-based hand gesture recognition systems. A temporal semantic pyramid model was proposed in a study by the author in [19]. This model cuts the video into sections of many degrees of detail. Such an approach is better than the traditional uniform segmentation algorithms because it gathers rich features within each frame and broader semantic information across the entire video sequence. It applies a time-based hierarchical pattern learning technique to extract features at different levels of video hierarchy. Such processes enable the representation to attend to fine details within each frame, like the position of individual fingers, and coarser contextual information across the entire film, like the overall direction of hand motion. Use these tactics to address the problem of high accuracy in accomplishing segmentation. The authors [20], have successfully transitioned and established a solid basis for the development of advanced sign language recognition. Nevertheless, segmentation in isolation may not fully encompass the intricate nature of sign language, since it frequently depends on a combination of manual gestures and bodily stance to express significance. The author in [21] introduced a model that takes into account the structure of a skeleton. This method directly uses human body skeletons, obtained from the video frames, as a means of representing body positions. Through the examination of these skeletal structures, the model can comprehend the spatial connections between bodily joints, therefore acquiring significant data for the identification of gestures that include certain body motions. Furthermore, divide the movie into segments to examine the time-based patterns of the movements inside each section. Multimodal fusion techniques can be used to combine information from different sources, potentially leading to more robust recognition [22].

The author discusses the issue of recognizing German sign language in [23]. Expanding on the idea of utilizing body position data, a resilient key-point normalization technique was suggested, particularly designed for sign language detection. Their methodology is around standardizing the placement of crucial points, which frequently correlate to body joints or specific positions on the hand, by utilizing a framework including the neck and shoulder. The normalized key-points are subsequently utilized as the input sequence for a transformer network, a robust deep-learning architecture proficient at processing sequential data. The author mentioned in reference [24] and reference [25] may be discussing the issue of signals that have identical hand forms but different lip motions. However, depending exclusively on body position and hand motions may not be enough for all sign languages. The user highlights this limitation, and a model of a multichannel transformer is proposed to capture the data of different articulators, which are the body parts taking part in voice production. Their approach consists of analyzing information each articulator provides, including hand gestures, facial expressions, and even tongue movements. Through the conjunction of information in these various sources, the model can pick up a richer and fuller representation of the sign being conveyed. In [26], the author proposed one of the first frameworks to use graph neural networks

(GNNs) to recognize sign language. Graph Neural Networks (GNNs) are a type of deep learning architecture specifically designed to process graph-structured data. Further, the authors have worked on sign-language translation systems. These systems are meant to assist in interacting with sign and spoken or written language. A befitting example is a Deep ASL system used to translate American Sign Language (ASL). Deep ASL represents a sensor-based approach in which data on hand shape, relative position, and motion is captured using a Leap Motion sensor attached to the user [27,28]. The author proposed connectionist temporal fusion (CTF) architecture to solve sign-language translation (SLT) [29]. Their approach introduces the problem of matching frames by exploiting several modules to track instant and long-lasting interconnections in the sign-language video. The CTF method initially performs a C3D-ResNet to isolate visual information of video clips. The information obtained is then entered in the subsequent distinct boxes. Besides that, the author of [30] proposes three pooling methods to reduce the number of less informative clips, improving the H-LSTM model's performance. The H-LSTM model demonstrates the possibility of better accuracy in sign language translation by incorporating hierarchical information and addressing the issue of frame-level alignment problems. The substantial development in sign-language translation (SLT) was made by Camgoz et al. [31]. They were the first to propose a deep-learning model directly translating sign-language movies into text-written language. This approach eliminates the need for intermediate processing steps, which could lead to a simplified and effective translation process. In [32], the author sought to improve end-to-end SLT models' performance through gated recurrent units (GRUs). GRUs, a specific type of recurrent neural network (RNN) architecture, have been famous for their effectiveness in dealing with long dependencies in sequential data. In this study, a hand gesture recognition system will be optimized through the application of image processing techniques to improve communication. In the suggested method, the hand gestures' motion is captured by a webcam and converted to text and then to speech, providing a natural and easy-to-use communication medium. The hand gestures' motion is captured by the system, and the model is preprocessed to deal with changes in lighting and background. The results of the suggested approach are improved compared to the current approach stated in the existing literature.

3. METHODOLOGY

This module discusses the methodology for identifying hand signals through image processing. The method follows a clearly outlined scheme of steps, namely, pre-processing, feature extraction, and detection. Each of the stages is essential to the detection system of hand gestures to increase the overall accuracy and robustness. The system architecture is represented in Figure 1 below.

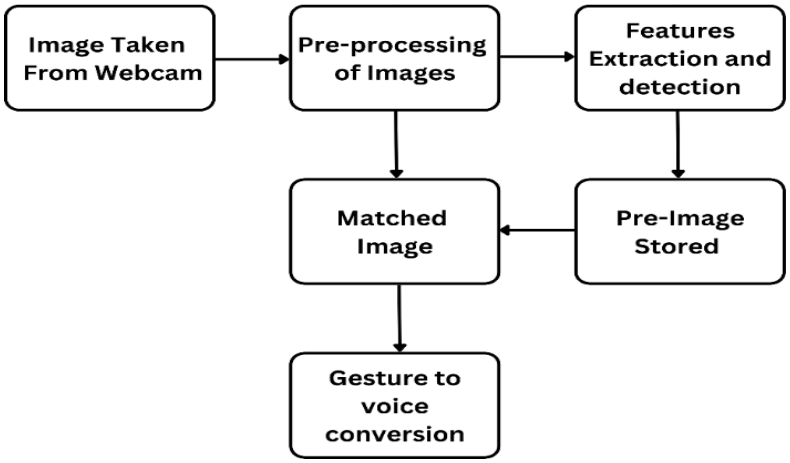

Figure 1: System Architecture

3.1 Pre-processing: Image Acquisition, Enhancement, and Annotation

The system begins with the initial stage of picture collection, which involves capturing digital photos that feature hand motions or things of interest. The method used to obtain the image might differ based on the specific application. Typical methods involve using a camera, a specialized depth sensor, or accessing a picture from an existing database. A high-resolution camera was used to capture a dataset of local fruits and vegetables from several grocery stores in this specific system. The background has been maintained as a consistent color (namely, white) to enhance the precision of the system. To enhance the model's generalizability, photographs have been taken from various perspectives and under diverse lighting circumstances. After capturing the picture, pre-processing techniques are used to improve the quality of the image and make it easier to process in the next phases. These methods target different flaws in images that might impede precise identification. Typical pre-processing methods are reducing noise, normalizing data, and segmenting it.

Digital pictures are vulnerable to noise that may be introduced during the process of capture or transmission. Methods such as median filtering or Gaussian filtering can effectively remove or reduce undesired fluctuations in the signal, therefore enhancing the clarity of the image. In situations when the item being studied does not fill the whole image, segmentation techniques are used to separate the area of interest (ROI) that contains the object. This allows for a more focused examination of the relevant information and reduces the computing complexity. Commonly employed techniques include thresholding, edge detection, and region-based segmentation.

To accurately annotate the dataset, each picture has been meticulously annotated using the Labelling software. Labeling is a Python-based graphical image annotation application that uses a graphical interface to save annotations as XML files in PASCAL VOC format and YOLO format. Figure 2 illustrates the graphical user interface (GUI) of the Labelling program. For every gesture a number of images have been taken and added to the dataset as shown in Table 1.

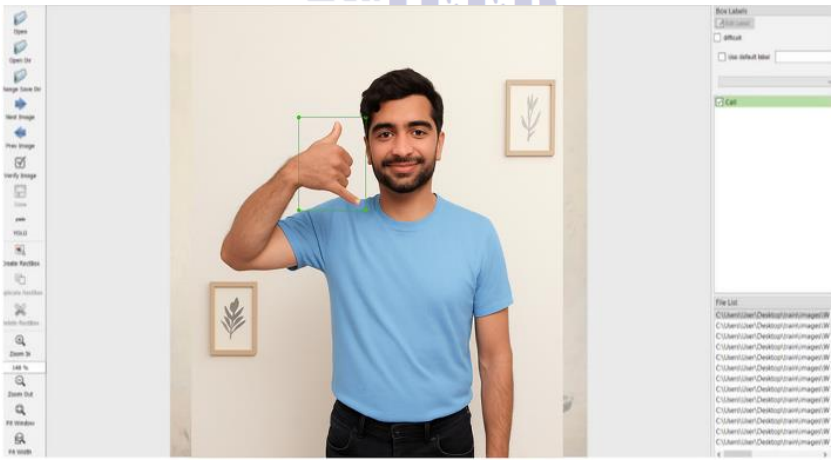

Figure 2: Graphical User Interface of Labeling Software

Table 1: Provides a summary of the number of images per class for the collected dataset.

Gestures	Number of Images
Hello	220
Call	225
Dislike	210
Thank you	213
Ready	202

Sorry	203
No	208
Please	213

3.2 Classification: Assigning Gestures to their Symbolic Meaning

The classification stage utilizes the retrieved information to classify the hand gesture into a pre-established category. This process effectively converts the visual depiction of the hand into its matching symbolic significance. Employing the right classification method is essential for getting a high level of accuracy in recognition. In this context, we examine two important methods of categorization:

Machine learning and deep-learning are tightly interconnected disciplines, where deep learning is a distinct and specialized subset of machine learning. The main goal of deep learning is to acquire complex and layered representations and abstractions from data, allowing the model to comprehend detailed patterns and relationships within the dataset. The notion of deep learning, also known as deep structured learning, was first developed in 2006 by respected academics in the domain of machine-learning. During this era, Geoffrey Hinton and Ruslan Salakhutdinov put forward techniques for unsupervised pre-training and fine-tuning to tackle the issue of the vanishing gradient problem. After their discovery, deep-learning became widely recognized and studied. In 2007, a strategy called greedy layer-wise training was used to improve the starting weights for deep networks. The use of the Rectified Linear Unit (ReLU) in 2011 enhanced deep learning models by retaining more information over several layers and addressing the problem of the vanishing gradient. In 2012, the dropout approach was established as a solution to address overfitting, resulting in improved performance of deep networks. The availability of high-performance computational hardware, especially GPUs, has significantly improved the utilization of deep neural networks in computer vision tasks. Currently, the main areas of emphasis in retail product recognition using deep-learning are:

3.2.1 Image Classification:

This essential computer vision problem is classifying diverse pictures into several categories. Deep-learning algorithms have exceeded the ability of humans in the task of picture categorization.

3.2.2 Detection of Objects:

Categorizing pictures involves the identification of objects using rectangular bounding boxes. Several scientists and developers have most recently created and improved deep-learning frame-works to accelerate training and prediction processes by the rapid growth of deep learning. The most employed frameworks that facilitate the application of deep-learning techniques for scientists are Caffe [15], Tensor Flow [16], MXNet [17], and PyTorch [18]. Recognition of hand motions may employ both classic machine-learning and deep-learning methods.

3.2.3 Supervised Machine Learning Algorithms:

Hand gesture recognition can utilize techniques such as Support Vector Machines (SVM) and K-Nearest Neighbors (KNN). These algorithms necessitate a labeled training dataset that includes images and their related hand motion annotations. During the training process, the algorithm acquires knowledge about the distinctive characteristics of various hand movements. After undergoing training, the model has the ability to categorize unfamiliar hand motions by analyzing their characteristics and comparing them to the previously acquired patterns.

3.2.4 Deep Learning Classifiers:

Convolutional Neural Networks (CNNs) are highly efficient at recognizing hand gestures. Convolutional Neural Networks (CNNs) can autonomously acquire significant characteristics directly from visual input, hence preventing the necessity for laborious feature engineering. Deep-learning techniques often need bigger training datasets in comparison to classic machine-learning algorithms. However, they excel in achieving higher-level of accuracy for complicated hand gesture categorization tasks due to their capacity to grasp nuanced data correlations.

3.2.5 Convolutional Neural Networks

Convolutional neural networks (CNN) have demonstrated exceptional performance in challenging image recognition tasks in recent years [21]. The existence of extensive public picture libraries such as ImageNet has made it possible to carry out image recognition on a broad scale. CNNs, like the human-brain, are networks composed of neurons [22]. Neurons consist of weights and biases that organize into layers and activate in a certain sequence to provide an output [23]. By training the networks to a significant volume of data, they may be trained to recognize specific patterns. Convolutional Neural Networks (CNNs) typically have three unique layer types besides the input and output layers. The convolutional, pooling, and fully linked layers exemplify these layers. Each layer performs several operations. These operations finally provide a value at the network's conclusion. In the sub-sequent elucidation of the layers, a picture is used as the input, and the outcome— which may not always be the case— is a prediction of the image's content. Figure 3 depict the visual representation of a CNN as it searches for distinctive characteristics that aid in the recognition of a cartoon.

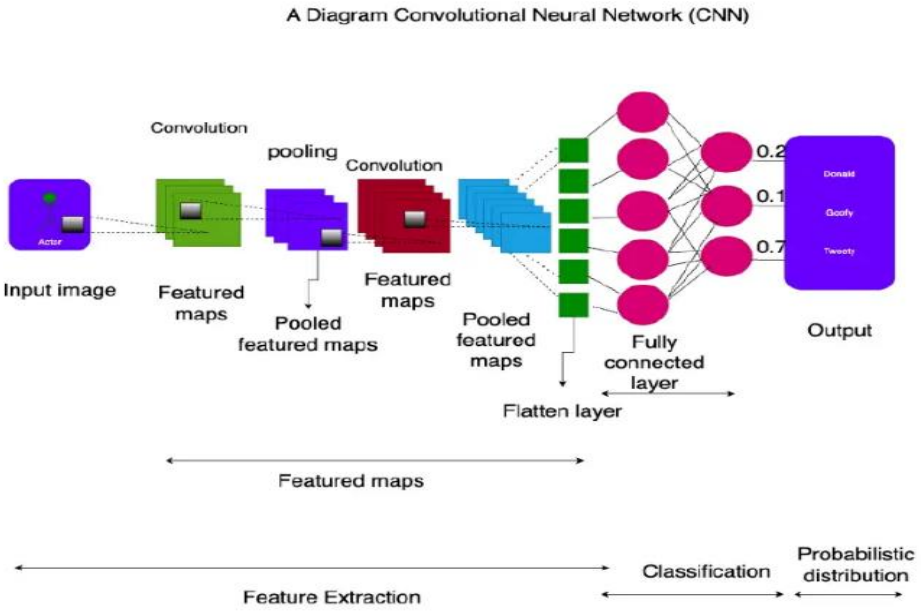

Figure 3: An Overview of CNN's Operation

3.2.6 Convolutional Layer

Every convolutional layer performs a distinct mathematical operation on the pixels of an input picture. The goal is to extract different facets from the image. These attributes may consist of endpoints, edges, or corners. However, not all pixels in a picture are engaged in the process. The mathematical operation is performed on a specific group of pixels that are split by a square shape, referred to as the kernel or filter. Next, the kernel is reapplied to the complete image in the same manner. Stride represents the displacement of the kernel about the prior mathematical operation. To enhance feature extraction, many mathematical processes might be used inside a single layer. The upper layers amalgamate the recovered features from the lower layers as the network advances through its layers, resulting in the creation of feature maps. Each pixel in feature maps indicates the result of a mathematical operation performed by the kernel. Figure 4 depicts an instance of a convolutional layer.

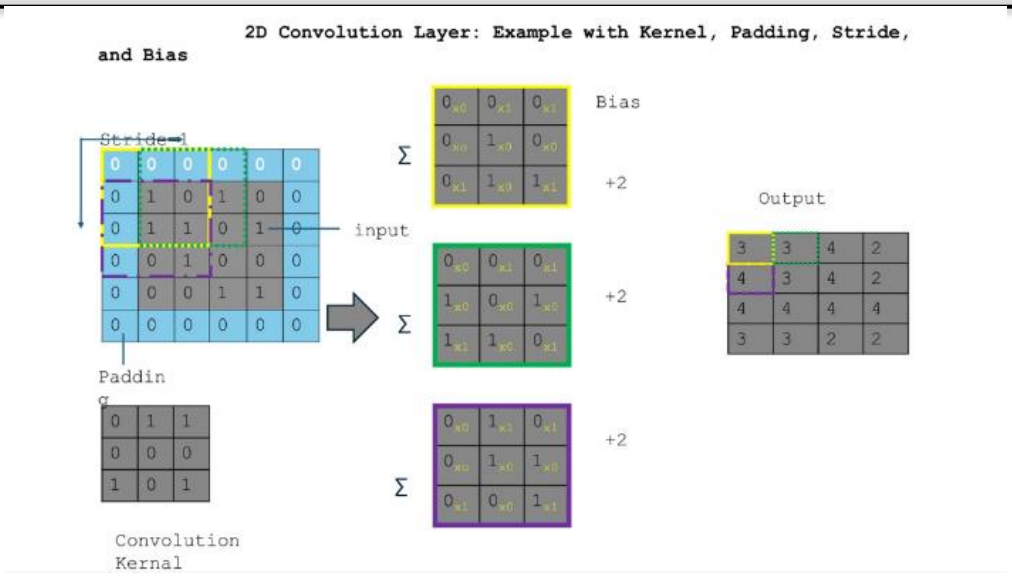

Figure 4: Convolutional Layer Technique

3.2.7 Pooling Layer

The outputs from several neurons are consolidated into a single neuron through the pooling layer. The pooling function is responsible for calculating the value of the combined neuron. Max, Average, and Stochastic are three frequently used pooling functions. The purpose of this layer is to decrease the resolution of the feature maps, hence reducing the vulnerability of the output to shifts and distortions. Figure 5 illustrates a sample of a max pooling layer.

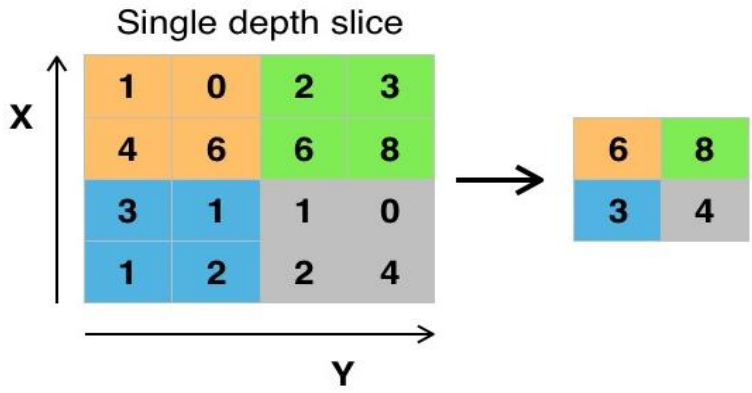

Figure 5: Max Pooling Layer Technique

3.2.8 Fully Connected Layer

The last layer of the network is the fully linked layer. The value of N represents the total number of classes that the network aims to identify. It utilizes the output from the previous layer to generate a vector with N dimensions [24]. The vector contains the probability values for each class. Again, all interconnected neurons are utilized in a mathematical algorithm to accomplish this task. Figure 6 depicts a layer that is entirely linked.

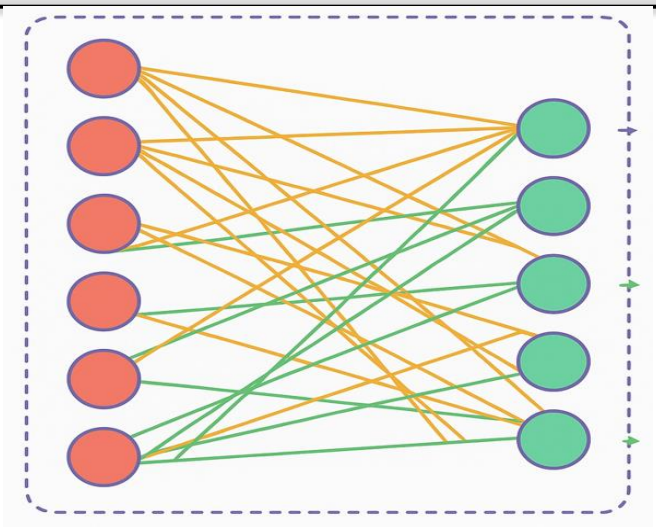

Figure 6: A Fully Connected Layer

4. INTEGRATION AND OPTIMIZATION: TOWARDS ROBUST HAND GESTURE RECOGNITION

The efficacy of a hand motion detection system depends on the interaction between the preprocessing, feature extraction, and detection phases. Pre-processing methods are used to prepare the picture for feature extraction, whereas feature extraction methods generate a streamlined image that is suited for detection. Ultimately, the detection system utilizes these characteristics to allocate the hand motion to its appropriate category.

5. MODEL SELECTION

You Only Look Once (YOLO) serves as the fundamental basis for the proposed system's object detection. The term used to refer to modern real-time object detection technologies is YOLO. Joseph Redmon is credited with developing the idea. A continuous-object identification system can identify many items within a single-frame. YOLO has a distinct approach compared to conventional history recognition algorithms. A singular interconnected system obtains the complete image. Figure 7 illustrates how the inter-connected system partitions the entire image into several sections and predicts ratings of confidence for every single object.

Since the introduction of YOLOv1 in 2015, the method has gained significant interest in the computer vision field. Furthermore, enhanced iterations of the YOLOv2, YOLOv3, YOLOv4, YOLOv5, YOLOv6, YOLOv7, YOLOv8, YOLOv9, and the latest YOLOv10 models have been released. Before delving into the topic, it is crucial to acknowledge that there exist two versions of the YOLOv5 algorithm accessible on the internet. In this discussion, we will specifically focus on the "Official YOLOv5 algorithm". The collective is responsible for the development of the authorized Yolov5 algorithm. YOLOv5 has emerged as a highly popular and successful version of the YOLO series. The fact that the new version was developed by the same team provides a compelling rationale for acquiring knowledge about it. Yolov5 is a real-time object detector that is now revolutionizing the computer vision market due to its exceptional features. Compared to its previous versions, the authentic YOLOv5 provides exceptional velocity and accuracy. The Yolov5 weights are trained only on the COCO data-set from Microsoft, without utilizing any pre-trained weights. YOLOv5 surpasses all prior object detectors in terms of both speed and accuracy. The YOLOv5 architecture design is built around the ELAN (efficient layer aggregation network). ELAN optimizes network design efficiency by effectively managing both the shortest and longest gradient pathways, enabling deep-networks to converge and learn more efficiently.

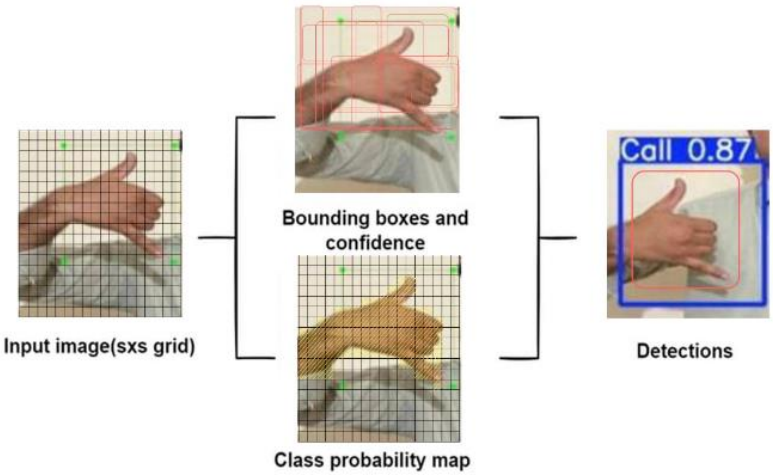

Figure 7: Working Of YOLO Algorithm

5.1 Model Training:

Google Colaboratory is employed for model training due to its ample computational power, including GPU capabilities. Collaboratory, sometimes called "Colab," is a product created by Google Research. Colab allows users to develop and run Python code directly in the browser, rendering it appropriate for machine learning, data analysis, and education activities.

Step 1: First the dataset has been zipped and uploaded to Google Drive.

Step 2: Then the Google Drive has been mounted in colab to access that dataset.

Step 3: Download pre-trained yolov5

```
[ ]  1 from google.colab import drive
     2 drive.mount('/content/drive')

     Mounted at /content/drive
```

Figure 8: Model Training Chart

Step 4: For training this model on a custom dataset changes have been made in the configuration file.

Step 5: Split the dataset into an 80% training set and a 20 %validation set.

Step 6: Train the model and get MAP 99% on our custom dataset shown in Figure 8.

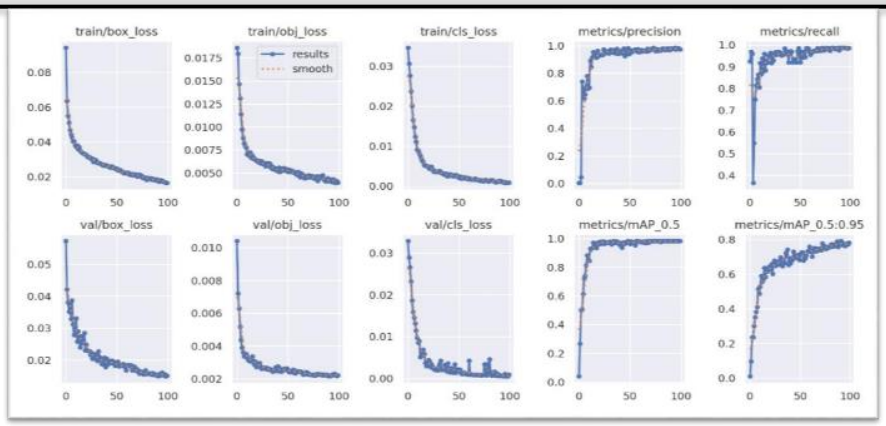

Figure 9 displays the loss curves for both training and validation, as well as the evaluation metrics, of a deep learning model across 100 epochs. The plots offer valuable understanding of the model's performance and learning advancement across several phases of training

Step 7: Evaluate the model

```
[1] !git clone https://github.com/ultralytics/yolov5.git

Cloning into 'yolov5'...
remote: Enumerating objects: 16680, done.
remote: Counting objects: 100% (221/221), done.
remote: Compressing objects: 100% (160/160), done.
remote: Total 16680 (delta 102), reused 131 (delta 61), pack-reused 16459
Receiving objects: 100% (16680/16680), 15.32 MiB | 15.83 MiB/s, done.
Resolving deltas: 100% (11408/11408), done.
```

```
Validating runs/train/FYPL/weights/best.pt...
Fusing layers...
YOLOv5l summary: 267 layers, 46145973 parameters, 0 gradients, 107.8 GFLOPs
               Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 3/3 [00:02<00:00,  1.25it/s]
                 all         86         87       0.98      0.988      0.983      0.791
                Call         86         10      0.968        0.9      0.899      0.829
             Dislike         86          8      0.967          1      0.995      0.924
               Hello         86         18      0.986          1      0.995      0.823
           Thank You         86         22      0.993          1      0.995      0.781
               Sorry         86          4      0.997          1      0.995       0.61
              Please         86         12      0.981          1      0.995      0.819
               Ready         86          4      0.961          1      0.995      0.803
                  No         86          9      0.984          1      0.995      0.737
```

Figure 9: YOLOv5 Model Validation Results

Figure 9 displays the validation outcomes of a YOLOv5 model that was trained specifically for hand gesture recognition. The evaluation encompasses many measures, including precision (P), recall (R), and mean average precision (mAP), calculated at two different intersections over union (IoU) criteria (50% and 95%) for each class and overall efficiency.

Step 8: Then the model has been downloaded on the local computer.

5.2 Parameters:

The Table 2 shows the parameters used to train the proposed frame-work.

Parameters	Object Detection
Train Test Split Ratio	Training set 80%, Validation set 20%
Learning Rate	0.001
Activation Function	ReLU

Batches Size	16

6. DEEP LEARNING FRAMEWORKS

In order to execute the trained model on your local computer, it is necessary to have the following libraries already installed. These are the software libraries: OpenCV, TensorFlow, absl-py, Matplotlib, Pillow, imutils, Qt5, Keras, PyTorch, NumPy, and pip.

6.1 Tensor flow:

TensorFlow is an openly available machine-learning (ML) platform that offers several levels of abstraction. Thus, this framework is strongly recommended for novices and industry experts seeking more control and influence over the network. The neural network may be constructed via the official Keras API, a high-level interface. The Distribution Strategy API is specially tailored to spread training for larger machine learning workloads over several hardware devices without altering the model. In addition, TensorFlow is indifferent to both programming languages and operating systems. It can be employed for desktop, mobile, web, and cloud applications. TensorFlow is interoperable with Python, C++, and R programming languages. The framework further has a tool known as the Tensor Board, which offers an extensive visualization of network modeling and performance.

7. RESULTS AND DISCUSSION

Multiple quantitative metrics were employed to evaluate the performance of the proposed hand gesture recognition and speech conversion system. These metrics provide insight into the model's ability to accurately detect, localize, and classify hand gestures in real time. The evaluation primarily focused on Intersection over Union (IoU) and Mean Average Precision (mAP), standard benchmarks in object detection tasks. The ioU value is between 0 and 1. An IoU equal to 0 means that the predicted and actual boxes do not overlap, whereas an IoU of 1 means a perfect match. The YOLO-based detection module used in this study consistently attained a high IoU of over 0.85 on the test set, demonstrating that it is precise in localizing hand gestures. Such good performance shows that the model effectively isolates gesture areas despite changing background scenarios and plays a part in achieving high accuracy of the entire system. Further, the proposed approach achieved a high mAP score, indicating that the system is strong or accurate in detecting and classifying different hand gestures. An increase in mAP shows that the model consistently identifies positive examples of gestures and significantly reduces false positives. Hence, it reinstates the system's suitability in real-time human-computer interaction, especially with people with speech or hearing impairments. Hence, combining these metrics demonstrates that the system is accurate and trustworthy in detecting and classifying data. The maintainability of the IoU and mAP metrics proves the usefulness of the proposed YOLO-based gesture recognition method, which, combined with pieces of text and speech converters, can create a highly interactive and accessible communication network.

8. DATABASE CREATION AND IMAGE CAPTURING

The system allows the efficient user-controlled database creation since the user can create personalized gesture databases that suit his/her communication requirements. The user can define the number of image samples per gesture, resulting in a rich and representative training set to teach the classification algorithm. The images should be taken with a transparent and clutter-free background to guarantee the best performance of the training and classification. Also, it increases the isolation of the hand gestures, eases the images' pre-processing step, and makes the following recognition processes more accurate.

8.1 Image Recognition and Classification

Once the user clicks on the "Recognize Image" button, the image recognition process starts, whereby the system tries to process the captured image and match it with the specific hand gesture. The gesture detection is conducted with the help of the YOLO (You Only Look Once) algorithm, an advanced object detector in real time. YOLO approaches the task of gesture recognition as a regression task that directly regresses bounding box coordinates and class probabilities using raw image pixels as input. The input image is split into a grid, and each cell

produces bounding boxes and corresponding confidence scores. The overall classification is done by picking the highest confidence score of all detections as the gesture class. Such an approach provides precise and effective multi-gesture recognition in one frame. Successful recognition of a gesture is then translated into a text label that is later synthesized into an audio output- Making it possible to interpret gestures in an audio form. It was trained on a dataset of 1,694 images, including some gathered on Roboflow and others taken manually for certain gestures. Figure 10 demonstrates examples of the gathered dataset, and Figure 11 gives examples of the trained model.

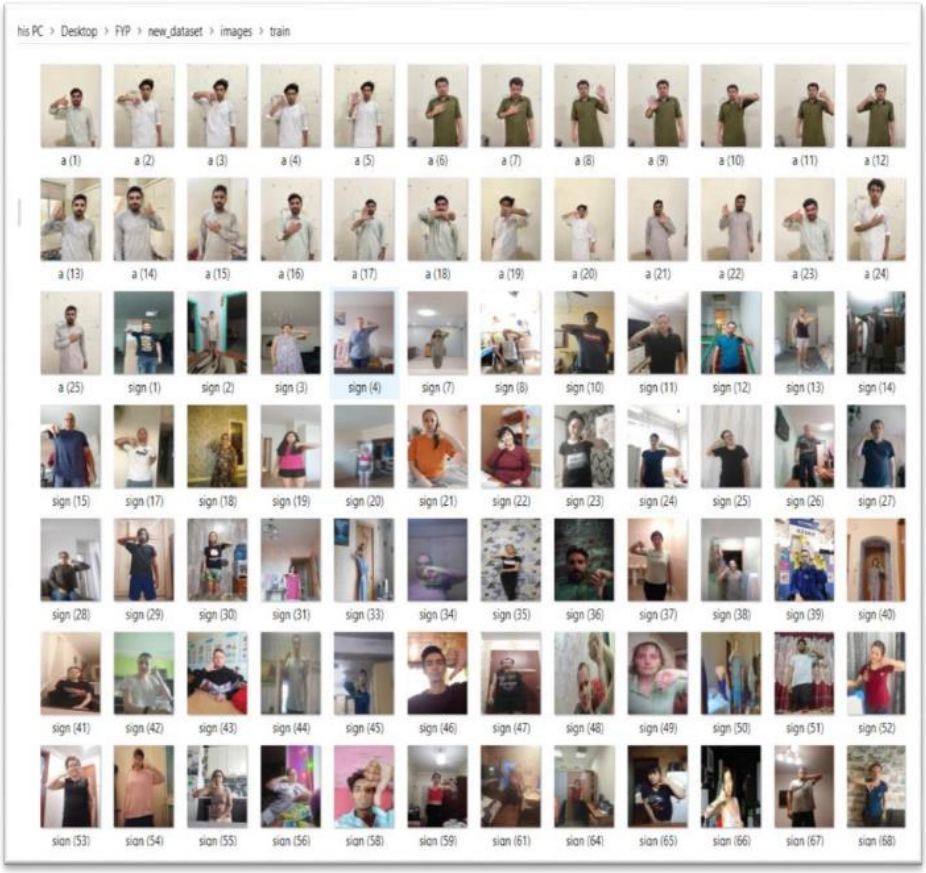

Figure 10: Databases Of Images

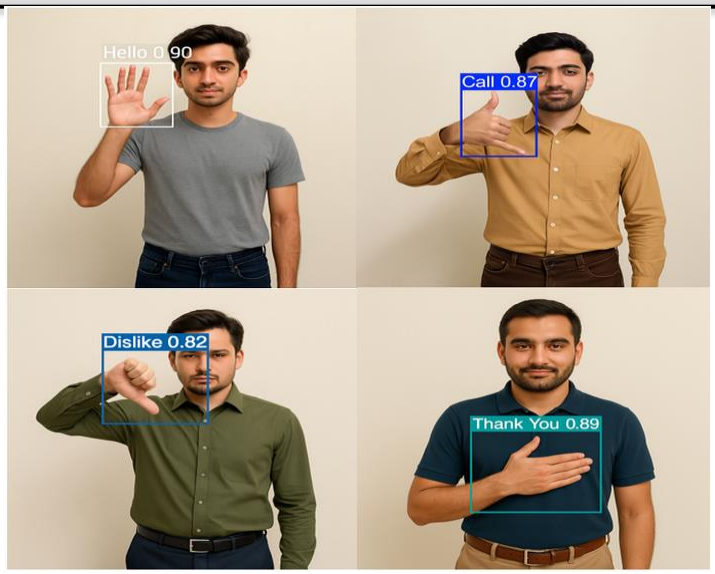

Figure 11: Accuracy comparison

9. CONCLUSION

The study of hand gesture recognition to voice conversion with the help of AI illustrates a breakthrough in the sphere of human-computer interaction, specifically in enabling communication in deaf and mute people. Such a technology with such transformative effects can become the source of inspiration for change. The system uses image processing and deep learning to capture hand gestures, extract relevant features, and translate them into text and synthesized speech. It starts with the high accuracy of gesture recognition based on strong pre-processing and feature extraction, including the YOLO object detector. As a future work, it is possible to introduce more advanced feature extraction, which will be based on deep learning methods, to make real-time gesture recognition practical and integrate this system with at least several applications associated with education, healthcare, and entertainment to enlarge the category of its applicability and impact.

REFERENCES

[1] Shinde, Shweta S., Rajesh M. Autee, and Vitthal K. Bhosale. "Real time two way communication approach for hearing impaired and dumb person based on image processing." Computational Intelligence and Computing Research (ICCIC), 2016 IEEE International Conference on. IEEE, 2016.

[2] Shangeetha, R. K., V. Valliammai, and S. Padmavathi. "Computer vision based approach for Indian Sign Language character recognition." Machine Vision and Image Processing (MVIP), 2012 International Conference on. IEEE, 2012.

[3] Sood, Anchal, and Anju Mishra. "AAWAAZ: A communication system for deaf and dumb." Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), 2016 5th International Conference on. IEEE, 2016.

[4] Ahire, Prashant G., et al. "Two Way Communicator between Deaf and Dumb People and Normal People." Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on. IEEE, 2015.

[5] Ms R. Vinitha and Ms A. Theerthana. "Design And Development Of Hand GestureRecognition System For Speech Impaired People." Kumari, Sonal, and Suman K. Mitra. "Human action recognition using DFT." Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2011 Third National Conference on. IEEE, 2011

[6] P. K. Pisharady and M. Saerbeck, "Recent methods and databases in vision-based hand gesture recognition: A review," Comput. Vis. Image Understand., vol. 141, pp. 152–165, Dec. 2015, doi: 10.1016/j.cviu.2015.08.004.

[7] M. Yasen and S. Jusoh, "A systematic review on hand gesture recognition techniques, challenges and applications," PeerJ Comput. Sci., vol. 5, p. e218, Sep. 2019.

[8] M. J. Cheok, Z. Omar, and M. H. Jaward, "A review of hand gesture and sign language recognition techniques," Int. J. Mach. Learn. Cybern., vol. 10, no. 1, pp. 131–153, Jan. 2017, doi: 10.1007/s13042-017-0705-5.

[9] S. Kausar and M. Y. Javed, "A survey on sign language recognition," in Proc. Frontiers Inf. Technol., 2011, pp. 95–98. [5] H. Cooper, B. Holt, and R. Bowden, "Sign language recognition," in Visual Analysis of Humans. London, U.K.: Springer, 2011, pp. 539–562.

[10] G. Fang, W. Gao, and D. Zhao, "Large vocabulary sign language recognition based on fuzzy decision trees," IEEE Trans. Syst., Man, Cybern. A, Syst. Humans, vol. 34, no. 3, pp. 305–314, May 2004. VOLUME 9, 2021 157433 N. Mohamed et al.: A Review of the Hand Gesture Recognition System: Current Progress and Future Directions

[11] M. Mohandes, M. Deriche, U. Johar, and S. Ilyas, "A signer-independent Arabic sign language recognition system using face detection, geometric features, and a hidden Markov model," Comput. Electr. Eng., vol. 38, no. 2, pp. 422–433, 2012.

[12] S. C. W. Ong, S. Ranganath, and Y. V. Venkatesh, "Understanding gestures with systematic variations in movement dynamics," Pattern Recognit., vol. 39, no. 9, pp. 1633– 1648, Sep. 2006.

[13] B. K. Chakraborty, D. Sarma, M. K. Bhuyan, and K. F. MacDorman, "Review of constraints on vision-based gesture recognition for human– computer interaction," IET Comput. Vis., vol. 12, no. 1, pp. 3–15, Feb. 2018, doi: 10.1049/iet-cvi.2017.0052.

[14] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: A survey," Artif. Intell. Rev., vol. 43, no. 1, pp. 1–54, Jan. 2012, doi: 10.1007/s10462012-9356-9.

[15] M. A. Moni and A. B. M. S. Ali, "HMM based hand gesture recognition: A review on techniques and approaches," in Proc. 2nd IEEE Int. Conf. Comput. Sci. Inf. Technol., 2009, pp. 433–437.

[16] A. Wadhawan and P. Kumar, "Sign language recognition systems: A decade systematic literature review," Arch. Comput. Methods Eng., vol. 28, pp. 785–813, May 2021, doi: 10.1007/s11831-019-09384-

[17] N. Aloysius and M. Geetha, "Understanding vision-based continuous sign language recognition," Multimedia Tools Appl., vol. 79, nos. 31–32, pp. 22177– 22209, Aug. 2020, doi: 10.1007/s11042-020-08961-z. [14] R. Rastgoo, K. Kiani, and S. Escalera,

[18] 'Sign language recognition: A deep survey," Expert Syst. Appl., vol. 164, Feb. 2021, Art. no. 113794, doi: 10.1016/j.eswa.2020.113794.

[19] K. M. Lim, A. W. C. Tan, and S. C. Tan, "A feature covariance matrix with serial particle filter for isolated sign language recognition," Expert Syst. Appl., vol. 54, pp. 208–218, Jul. 2016, doi: 10.1016/j.eswa.2016.01.047.

[20] W. Ahmed, K. Chanda, and S. Mitra, "Vision based hand gesture recognition using dynamic time warping for Indian sign language," in Proc. Int. Conf. Inf. Sci. (ICIS), Aug. 2016, pp. 120– 125.

[21] M. V. D. Prasad, P. V. V. Kishore, D. A. Kumar, and C. R. Prasad, "Fuzzy classifier for continuous sign language recognition from tracking and shape features," Indian J. Sci.
Technol., vol. 9, no. 30, pp. 1–9, Aug. 2016, doi: 10.17485/ijst/2016/v9i30/98726.
[22] O. Koller, J. Forster, and H. Ney, "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers," Comput. Vis. Image Understand., vol. 141, pp. 108–125, Dec. 2015, doi: 10.1016/j.cviu.2015.09.013.
[23] W. Yang, J. Tao, and Z. Ye, "Continuous sign language recognition using level building based on fast hidden Markov model," Pattern Recognit. Lett., vol. 78, pp. 28–35, Jul. 2016, doi: 10.1016/j.patrec.2016.03.030.
[24] K. Tripathi and N. B. G. C. Nandi, "Continuous Indian sign language gesture recognition and sentence formation," Proc. Comput. Sci., vol. 54, pp. 523–531, Jan. 2015.
[25] T. Kim, J. Keane, W. Wang, H. Tang, and J. Riggle, "Lexicon-free fingerspelling recognition from video: Data, models, and signer adaptation," Comput. Speech Lang., vol. 46, pp. 209–232, Nov. 2017, doi: 10.1016/j.csl.2017.05.009.
[26] T.-W. Chong and B.-G. Lee, "American sign language recognition using leap motion controller with machine learning approach," Sensors, vol. 18, no. 10, p. 3554, Oct. 2018, doi: 10.3390/s18103554.
[27] R. Yang and S. Sarkar, "Detecting coarticulation in sign language using conditional random fields," in Proc. 18th Int. Conf. Pattern Recognit. (ICPR), vol. 2, Jan. 2006, pp. 108–112.
[28] S. Tanaka, A. Okazaki, N. Kato, H. Hino, and K. Fukui, "Spotting fingerspelled words from sign language video by temporally regularized canonical component analysis," in Proc. IEEE Int. Conf. Identity, Secur. Behav. Anal. (ISBA), Feb. 2016, pp. 1–7.
[29] N. Singh, N. Baranwal, and G. C. Nandi, "Implementation and evaluation of DWT and MFCC based ISL gesture recognition," in Proc. 9th Int. Conf. Ind. Inf. Syst. (ICIIS), Dec. 2014, pp. 1–7.
[30] S. M. Shohieb, H. K. Elminir, and A. M. Riad, "Signsworld atlas: A benchmark Arabic sign language database," J. King Saud Univ.-Comput. Inf. Sci., vol. 27, pp. 68–76, Jan. 2015, doi: 10.1016/j.jksuci.2014.03.011.
[31] B. Mocialov, G. Turner, K. Lohan, and H. Hastie, "Towards continuous sign language recognition with deep learning," in Proc. Workshop Creating Meaning Robot. Assistants, 2017, pp. 1–5.
[32] M. Fagiani, E. Principi, S. Squartini, and F. Piazza, "Signer independent isolated Italian sign recognition based on hidden Markov models," Pattern Anal. Appl., vol. 18, no. 2, pp. 385–402, May 2015, doi: 10.1007/s10044-014-0400-z.