Deep Learning Based Sign Language to Text Conversion Using GAN Gesture Recognition

Dr. K. Geetha,
Head of the Department
Department of Information Technology
M.A.M. College of Engineering and Technology, Tiruchirappalli, India
Email: hod.it@mamcet.com

Abu Thahir A
Department of Information Technology
M.A.M. College of Engineering and Technology,
Tiruchirappalli, India
Email: abuthahir.it21@mamcet.com

Shiva Narayanan S
Department of Information Technology
M.A.M. College of Engineering and Technology,
Tiruchirappalli, India
Email: shivanarayanan.it21@mamcet.com

Dhishwanth N
Department of Information Technology
M.A.M. College of Engineering and Technology,
Tiruchirappalli, India
Email: dhishwanth.it21@mamcet.com

Thipur Jiues D
Department of Information Technology
M.A.M. College of Engineering and Technology,
Tiruchirappalli, India
Email: thipurjiues.it21@mamcet.com

Abstract

There is a shortage of sign language interpreters and limited support for basic services, exacerbating communication barriers. Promoting sign language awareness among the public is crucial, but the process is labor-intensive and often ineffective. Finding accurate sign representations in video requires complex pre-processing. To address this, propose a deep learning model trained on sign language, converting it into text. By using Transfer Learning, we have improved accuracy from 87.9% with CNN to 91.6%.Sign language recognition through video motion analysis and deep learning is vital for enhancing communication and accessibility for sign language users. Recognizing signs is challenging, as they often lack direct spoken language equivalents. To improve accuracy, large annotated datasets of sign language videos are necessary for training deep learning models. One effective method for this is the YOLO (You Only Look Once) algorithm.

Keywords: Sign Language Recognition ,Deep Learning, GAN, CNN ,Gesture Recognition, Feature Extraction, Real-Time Processing, MediaPipe , OpenCV, LSTM (Long Short-Term Memory),Text-to-Speech (TTS),Deaf Communication, Gesture-to-Text Conversion.

1. Introduction

Sign language is a vital communication method for millions of deaf and hard-of-hearing individuals, relying on hand gestures, facial expressions, and body movements. However, a significant challenge exists in bridging the communication gap between sign language users and non-signers, as most people lack the skills to interpret it. This often results in social and professional barriers, limiting opportunities and accessibility. Sign language is the primary mode of communication for people with hearing impairments. Continuous sign language recognition is a challenging task, as it involves identifying sign gestures without prior knowledge of the timing between consecutive signs. Most modern techniques focus on extracting visual features, often neglecting the use of text or contextual information that could enhance recognition accuracy. Additionally, the potential of deep generative models to produce realistic sign language images has not been thoroughly explored. To address this gap, the Recognition System for Indian Sign Language presents a novel approach for continuous sign language recognition using a generative adversarial network (GAN) architecture. This system is trained on a self-created dataset that includes numbers from 0 to 9, letters from A to Z, and 50 distinct words represented by static gestures. The proposed method leverages the capabilities of a GAN to generate realistic sign language gestures, which are then used to train a recognition model. By training these networks concurrently, the generator improves its ability to create increasingly realistic gestures, while the discriminator enhances its accuracy in distinguishing between real and synthetic gestures. In this system, Deep learning-based systems can now interpret sign language gestures and convert them into text in real-time, offering a transformative solution to enable seamless communication and improve accessibility.

2 Related Work

Various techniques have been explored for sign language recognition. Traditional methods used handcrafted features and classifiers like SVM, Random Forest, and K-Nearest Neighbors. These were limited in handling large variability in gestures. More recently, deep learning models such as CNNs and RNNs have shown superior performance in gesture classification tasks. However, they rely heavily on large datasets, which are often unavailable. GANs help overcome this by generating realistic synthetic data.The research explores various methods for data augmentation, highlighting Generative Adversarial Networks (GANs) as a recent algorithm used for complex data augmentation. It discusses a technique for extracting distinctive and invariant features from images, which can reliably match different views of an object or scene. The study presents an object recognition approach that employs a fast nearest-neighbor algorithm, the Hough transform, and a least-squares solution. Additionally, the research outlines a method for object and scene retrieval that searches for and localizes objects in video using viewpoint-invariant region descriptors and temporal continuity. Pre-computed matches and inverted file systems are utilized, along with document rankings, to identify and return key frames or shots.

3. Proposed Methodology

The proposed system aims to convert sign language gestures into textual form using a deep learning-based architecture that integrates LSTM, GAN, and YOLO models. The system workflow is divided into seven core modules: Input Capture, Pre-processing, Feature Extraction, Gesture Recognition, Text Translation, Output Feedback, and User Interface (UI).The Input Capture module uses a webcam to record hand gestures in real time. YOLO (You Only Look Once), a real-time object detection algorithm, is utilized to detect and localize hand regions accurately within each video frame. The detected hand gestures are cropped and sent to the Pre-processing module, where techniques such as resizing, normalization, and background subtraction are applied to enhance input quality.For Feature Extraction, a CNN-based encoder is used to extract spatial features from the gesture images. These features are then passed into an LSTM (Long Short-Term Memory) network in the Gesture Recognition module, which captures the temporal dependencies of gesture sequences and classifies them accordingly.To improve gesture variability and robustness, a Generative Adversarial Network (GAN) is used to synthetically augment gesture datasets, increasing training diversity. The recognized gestures are translated into readable text in the Text Translation module.The final output is displayed in real time via a Flask-based web application, using HTML and CSS for the frontend. The Output Feedback module presents the recognized text to the user, ensuring smooth and accessible interaction, particularly for deaf and hard-of-hearing users.

3.1 Image Acquisition

A standard webcam is combined with OpenCV and MediaPipe libraries to capture real-time gesture inputs. The system records video at **30 frames per second (FPS)** with a resolution of **640x480 pixels**. Each captured image frame at time t is denoted as:

$$I_t \in R^{H \times W \times 3}$$

where $H=480H = 480H=480$, $W=640W = 640W=640$, and 3 denotes the RGB color channels.MediaPipe aids in detecting hand landmarks and gesture regions, providing a focused input for downstream processing.

3.2 Preprocessing

To ensure consistency and enhance important features, each image frame undergoes multiple preprocessing steps:

- **Grayscale Conversion**: Converts RGB to grayscale for simplicity and efficiency:

$$I_{gray} = 0.299R + 0.587G + 0.114B$$

- **Gaussian Blur**: Applied to reduce noise and smooth the image. The 2D Gaussian function is:

$$G(x, y) = (1 / 2\pi\sigma^2) * e^{-(x^2 + y^2) / 2\sigma^2}$$

where σ\sigmaσ is the standard deviation controlling the blur level.

- **Background Removal**: Achieved through **GrabCut algorithm** or **HSV color masking**, to isolate the hand gesture and eliminate irrelevant background elements.

- **Adaptive Thresholding**: Converts the image into a binary form for easier contour detection, adjusting thresholds dynamically based on local pixel intensities.

3.3 GAN-Based Data Augmentation

The GAN consists of a generator G(z) and a discriminator D(x). The generator produces synthetic gesture images from noise, while the discriminator distinguishes real from fake images.

Objective Function :

$$\min_G \max_D V(D,G) = E_{x \sim p_{data}}[\log D(x)] + E_{z \sim p_z}[\log(1 - D(G(z)))]$$

Through iterative training, the GAN improves in generating realistic gesture images, increasing the dataset from **5,000 to 25,000 samples**, thus enhancing model generalization and accuracy.

3.4 CNN-Based Feature Extraction

A **Convolutional Neural Network (CNN)** is designed to extract key spatial features from the gesture images. The architecture includes:

- **Conv2D Layer** (3x3 kernel) with **ReLU activation**
- **MaxPooling Layer** (2x2) for downsampling
- **Flatten Layer**
- **Fully Connected Dense Layer** with **Softmax output**

The core operations include:

- **Convolution Operation**: $Z_{i,j}^k = \sigma(\Sigma X_{i+m,j+n} * W_{m,n}^k + b^k)$

  Softmax: $P(y=c \mid x) = e^{zc} / \Sigma e^{zi}$

The output is a gesture class label gig_igi, which maps to a corresponding word or phrase.

3.5 Text Mapping and Feedback

Each recognized gesture gig_igi is mapped to a predefined text output using a **mapping function**:Ti = M(gi)

Where MMM is a dictionary or trained model mapping gestures to language tokens.

For continuous gesture sequences (i.e., sentences), an **LSTM (Long Short-Term Memory)** network is used to preserve temporal context and predict word order:

LSTM: h_t = f(Wx_t + Uh_{t-1} + b)

Where:

- xtx_txt: input at time step ttt
- hth_tht: hidden state
- WWW, UUU, bbb: trainable parameters

Finally, the text output is converted to audible speech using **Text-to-Speech (TTS)** tools like **Google TTS** or **pyttsx3**, enabling verbal communication and feedback.

4. Model Training and Evaluation

The deep learning model was trained on an augmented dataset of hand gesture images, optimized to achieve high performance in real-time gesture recognition and translation tasks. Below are the key training configurations and evaluation metrics:

4.1 Training Configuration

Parameter	**Value**
Dataset Size	25,000 images (augmented using GAN)
Number of Epochs	50
Batch Size	32
Learning Rate	0.0002

The model was trained for 50 epochs with a batch size of 32, ensuring efficient GPU utilization. A learning rate of 0.0002 was selected after experimentation, allowing stable convergence without overshooting during backpropagation. The dataset was augmented using GANs to simulate varied gestures, improving the model's ability to generalize to unseen data.

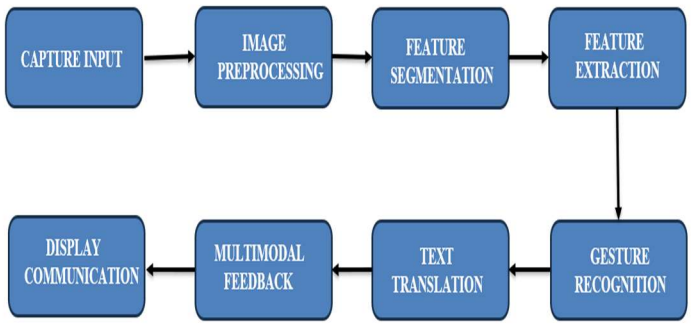
4.2 Evaluation Metrics

Metric	Value
Accuracy	94.7%
Precision	95.2%

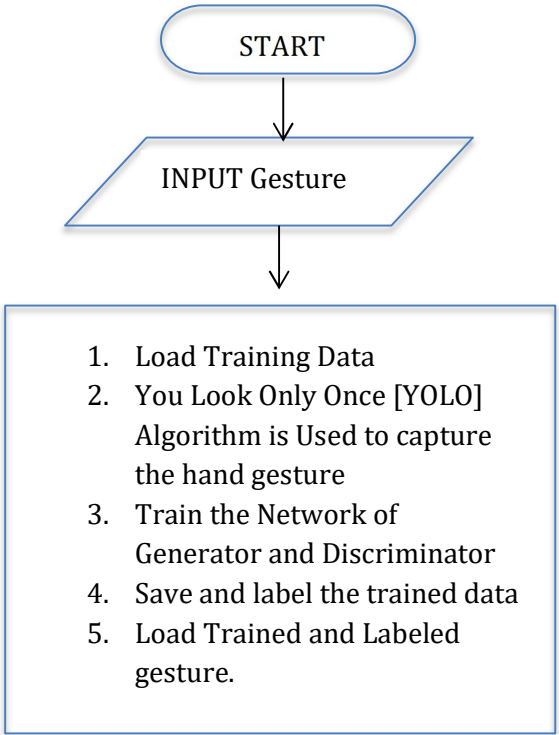
Recall	94.1%
F1 Score	94.6%
Latency	0.48 seconds/frame
Accuracy	94.7%

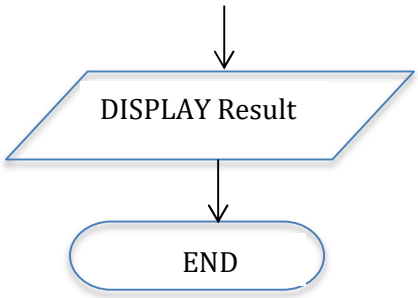
The evaluation of the GAN model is based on the images it generates, which are then compared to the original images from the training dataset. The results indicate an accuracy increase of 94.7%, a precision value of 95.2%, and a latency of 0.48 seconds per frame.

5. System Architecture



7. Working of the System





8. Conclusion and Future Work

The proposed system effectively combines CNNs, GANs, and LSTMs for real-time gesture recognition. The accuracy and robustness make it suitable for integration into accessibility tools and video conferencing platforms like Zoom or Google Meet. Future work includes support for sentence-level translation, multi-language integration, and edge deployment for low-power devices.In the future, this project can be integrated into popular video conferencing platforms like Zoom and Google Meet to enable seamless communication for deaf individuals. By incorporating real-time GAN-based gesture recognition, the system will translate sign language into text within meetings, ensuring that deaf participants can engage equally with others. This innovation will foster inclusivity, breaking communication barriers and allowing deaf individuals to interact effortlessly with people worldwide.

References

[1] Mahesh Kumar N B11 Assistant Professor (Senior Grade), Bannari Amman Institute of Technology, Sathyamangalam, Erode, India, Conversion of Sign Language into Text,2018.

[2]Sharvani Srivastava, Amisha Gangwar, Richa Mishra, Sudhakar Singh*[0000-0002-0710-924X] Sign Language Recognition System using TensorFlow Object Detection API. 2023.

[3]Rosalina; Yusnita, L.; Hadisukmana, N.; Wahyu, R.B.; Roestam, R.; Wahyu, Y. Implementation of Real-Time Static Hand Gesture Recognition Using Artificial Neural Network. In Proceedings of the 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT), Bali, Indonesia, 8–10 August 2017.

[4] Hangun and Eyecioglu, "Performance Comparison of OpenCV CPU and GPU," IJESA, 2017.

[5]Prof. Mrs. Maheshwari Chitampalli*1, Dnyaneshwari Takalkar*2, Gaytri Pillai*3, "Real-Time Sign Language Detection," 2020.

[6] Jebakani C, Rishitha S.P. Department Of Computer Science And Engineering, School Of Computing, Conversion Of Sign Language Into Speech Or Text Using CNN, Sathyabama Institute Of Science And Technology, March 2022

[7]Akash Kamble1, Jitendra Musale2, Rahul Chalavade3, Rahul Dalvi4, Shrikar Shriyal5, Department of Computer Engineering, Conversion of Sign Language to Text, Savitribai Phule Pune University, India.

[8]Jeevana Shree M, Department Of Computer Science And Engineering School Of Computing , Sign Language To Text And Speech Translation Using Machine Learning , Sathyabama Institute Of Science And Technology , March 2021.