**ARTIFICAL INTELLIGENCE AND**

**MACHINE LEARNING (AIML)**

**SECTION-4**

**PROJECT**

**Sign Language Detection**

**(Bridging Communication with ASL Recognition)**

**Problem Statement:** The primary goal of this project is to develop a real-time gesture and sign language recognition system capable of identifying and translating hand gestures (specifically American Sign Language - ASL) into text or speech. This will bridge the communication gap for individuals with hearing impairments, allowing them to communicate more effectively with those unfamiliar with sign language. The system needs to operate efficiently using a webcam or video feed, with fast and accurate gesture recognition, even in dynamic environments with variations in lighting, hand orientations, and background distractions.

**ALOGRITHM:**

**Dataset Preparation**:

* + Use the MNIST dataset (or a similar dataset for sign language) consisting of images representing 24 ASL alphabets (excluding J and Z) in grayscale format.
  + Split the dataset into training and testing sets.

1. **Preprocessing:**
   * Load the dataset and preprocess the images (size 28x28 pixels).
   * Normalize the images and reshape them to fit the CNN model input shape.
   * Augment the data by applying transformations like rotations, scaling, and lighting adjustments for robustness.
2. **Model Building:**
   * Create a Convolutional Neural Network (CNN) using layers such as:
     + Conv2D layers for feature extraction.
     + MaxPooling2D layers to reduce the feature map size.
     + Dropout layers for regularization.
     + Dense layers for classification into 26 output classes (for 26 alphabets).
   * Compile the model using categorical cross-entropy as the loss function and SGD (Stochastic Gradient Descent) as the optimizer.
3. **Model Training:**
   * Train the CNN model on the training dataset.
   * Use the validation dataset to fine-tune the model and ensure generalization.
   * Save the trained model.
4. **Real-Time Gesture Detection (Using OpenCV):**
   * Capture live video frames using a webcam.
   * Preprocess the captured frames:
     + Convert the image to grayscale.
     + Resize the image to 28x28 pixels to match the training dataset format.
   * Feed the processed image into the trained CNN model for gesture prediction.
   * The model outputs a predicted class representing the detected sign language gesture.
5. **Post-Processing and Display:**
   * Use the SoftMax function to obtain probabilities for each alphabet.
   * Display the predicted sign language alphabet on the video feed in real-time.
   * Optionally, convert the predicted text into speech using a text-to-speech engine for audio output.

**Expected Output:**

1. **Real-Time Hand Gesture Detection:**
   * The system will capture video input from a webcam, detecting and highlighting the region of interest where the hand gesture is made.
2. **Sign Language Classification:**
   * For each detected gesture, the CNN model will classify it as one of the 24 ASL alphabets and display the corresponding letter on the screen in real-time.
3. **Dynamic Display and Feedback:**
   * The video feed will overlay the detected sign language letter, updating dynamically as new gestures are performed. This will allow users to see the real-time interpretation of their hand gestures.
4. **Interactive Interface:**
   * The system will handle varying hand orientations, lighting conditions, and other real-world complexities, providing robust and reliable gesture recognition.

By combining deep learning with real-time image processing, HandTalk delivers an accessible and efficient solution to facilitate communication through sign language recognition.

TEAM:

SV KARTHIKEYA - 2320030501

D NAGAARJUNA - 2320090047

PRADEEP - 2320030412