MTP2 CONCRETE PLAN

GESTURE RECOGNITION USING USING DEEP LEARNING ALGORITHM

Problem Statement-

A significant problem or barrier of in-effective communication between, impaired people, who use only sign language and other community member have no prior understanding of sign language. These communication barriers occur in public services worker who are not familiar with sign languages. This also limits speech impaired person working at different places communicating with people who are unfamiliar with sign languages. Efforts have been made to solve these challenges like interpreters, but these are not very efficient. Advance technology has been proposed like machine learning techniques. This uses special algorithms that translate sign gestures to text or audio output that are universally understandable. Hence research is still in progress, as it has not been established which model can solve problems and used in real time. There is need of an efficient sign language recognition system. The objective SLR system, it is highly independent of the input data type. The input to system should be of any type, either it is video or live communication through camera. The signer must follow no special instructions like, distance from camera, be in a certain frame, with background, any wearable devices while performing gestures, particular camera features, any specific types of cloths, limited to specific sign language, i.e., they are free to perform any sign gestures, etc. For real time conversation the SLR system will be able to communicate with good response time.

Proposed Solution -

The video clips are used for training and may be generated by training framework or provided by user. The training video clips covers the gesture to be recognized from multiple distances and angles. Having a diverse set of visual characteristics in training video clips will enable high accuracy recognition. Each frame of the video is processed and pose estimation is applied to pixels for the body, fingers, and face. This will result in training video clips with overlaid pose estimation pixels. Optical flow is extracted from frames with overlaid pose estimation pixels. After feature extraction line, corner, shape, and edge rendering is performed to allow borders of the shapes in training media to be accurate and enable differentiation of one part from another. This results in very precise feature identification, advantageously enabling far more accurate recognition of movement or flow of objects that occur across time. The extracted and processed features are provided for training a 3DCNN model, as the first bit stream and second bi t stream for a second 3DCNN includes spatial and color information i.e., RGB information. The output layers of 3DCNN s are then fused, thereby enabling the convolution to run across both 3DCNNs, so flow and RGB or spatial information can be processed together as part of the same convolutional kernel. Similarly, recognition process starts from capturing frames from devices with multiple apertures or webcam and other sensors. The frame captured is implemented using its own thread and another different thread is used for recognition system that are ready to accept a frame. The captured frame is preprocessed with pose estimation for the body, face and fingers and resulting pose estimations laid on top existing frame pixels using a transparent layer. The resulting frames are provided to recognition process, and both the 3DCNN begins the recognition process i.e., one from motion or optical flow perspective and the other from a RGB or spatial information perspective. The two 3DCNN are fused together to enable their output layers to be processed jointly and using both their data streams. The recognition results for each frame pixels are provided to RNN which uses Long Short-

term Memory i.e., LTSM to track the recognition process temporally. The RNN with LTSM uses in own feedback loop to track state across more than a single round of recognition.

Technology will be used - Azure Cloud,Tensorflow,Keras,OpenCV,mediapipeline.

Timelines and RoadMap -

Till 15/02/2023 - collecting dataset where different sign languages will be performed

15/02/2023-15/03/2023- Training the model numbers, alphabets and words via vedios dataset dataset created.

15/03/2023 to 26/03/2023 - Training the model for words and phrases and reviewing the performance for the trained vedios.

26/03/2023 to 05/04/2023 - Train the model for all possibe phrases and checking the interactive performance for subtitles for every gesture performed.

05/04/2023 - 09/04/2023 - Testing the trained prases and work and evaluating the performance.

09/04/2023 to 23/04/2023 - Work on final report for all reviews and finding