Final Year Project – ASL Recognition

**Production Log 1: (Friday, 26/01/2024, 11:00 am)**

This is a real pain having to write down and summarize everything in words, but it is what it is.

I am currently running the project through a single Jupyter Notebook file and using an external hard disk to store all the extracted content and the dataset. Till now, I have been running the project on my Hp Envy Laptop but will later be using the university laptop to train and run the model.

My plan is:

* Stage 1: Create the dataset from videos.
* Stage 2: Train proposed ResNet + BiLSTM model with data.
* Stage 3: Train 1 or 2 other models. (we’ll decide them later)
* Stage 4: Evaluate results.
* Stage 5: Real Time Detection using Webcam. (have found code for this)
* Stage 6: Convert predicted Results to audio. (have found code for this)

I am currently in stage 1 of production. So, this is what I have worked on till now:

1. Decided on top 11 classes to train the model with (classes chosen based on the number of videos per class).

The number of videos per class lies in the range 9 to 14.

1. I have found a method to extract frames from each video.

Since the videos are not of equal lengths, each video will have unequal number of frames extracted. To overcome this, 60 frames will be extracted from each video. If the number of frames extracted is less than 60, then the last frame is repeated until the frame count reaches 60.

1. I have found a way to apply MediaPipe Holistic to a frame.

This detects and draws landmarks for face, hand and pose.

1. As for storing these frames, the folder hierarchy is as follows:

Main frames folder -> folders for actions/classes -> folders for each video in the respective class -> 60 frames of that respective video.

The data set is pretty small now; the total of all videos comes to 124. I will have to perform data augmentation to increase the number of videos. Say each class had 30-35 videos, then the total will be 330 – 385 videos… a bit decent, I guess. Which is why I am currently working on data augmentation.

As for the model implementation, I am thinking of 2 ways:

**Way 1:** Training a model with MediaPipe Keypoint data (This is numeric data with 3D coordinates of the keypoints).

I won’t be able to use a pretrained model for this but will have to train the model from scratch, for which I don’t have enough data. Let’s see about that though.

**Way 2:** Training a model with images.

I will be able to use a pretrained model for this approach. This method can further be split into 2 ways:

**Way A:** Training the model with MediaPipe Landmarks.

**Way B:** Training the model without MediaPipe Landmarks.

I need to try out both methods and not down the inferences. I don’t think we need to apply the face landmarks for this approach though.

To train the ResNet model, I need to resize the extracted frame to 224 x 224. To do that, I am cropping the center square from the frame and then resizing the images. This will make sure that the signer stays in the center of the resized frame and will retain the quality of the frames as the original aspect ratio. (Often when you resize the image to a smaller size, pixels are lost, and this affects the quality of the image… it makes it blurry).

So this is what will do, I will keep different folders for different tasks:

1. Contain original frames.
2. Contain total augmented frames.
3. Contain keypoints (NumPy arrays). [required for Way 1]
4. Contain images with keypoint landmarks.
5. Contain resized images with keypoints.
6. Contain resized images without keypoints.

Right now, I am searching for ways of augmenting video data.