Sprint 4 SAM-SLR research

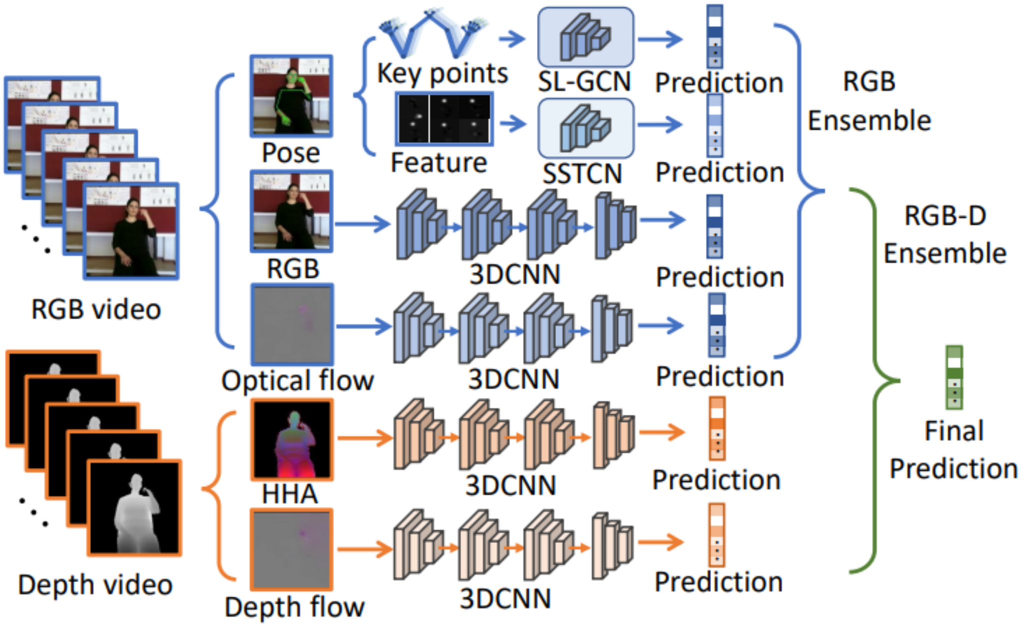
Mat Besch and 581 Team 1

November 2022

1 Introduction

The SAM-SLR model is a model created to attempt to recognize videos of individual signs and determine what word they represent. It stands for Skeleton Aware Multi-modal Sign language Recognition. It uses multiple types of videos to construct its predictions which it then compiles into 1 final prediction. The two types of videos it uses are RGB video like you would see from a regular camera and depth video. The data set this was trained on used a Xbox Kinect to train record its videos which is capable of capturing that depth information but our OAK-D camera is also capable of that same video capture. Once they get the RGB video and its corresponding depth information, they split it up into multiple parts and put those through different types of models, which each generate a prediction which then get combined into one final prediction.

2 Model Details



The model operates by splitting the input video into multiple different parts, getting predictions from each of them, the synthesizing the predictions into one final prediction. First it splits the video into RGB video and depth video. With the RGB video, it further creates three different pathways; pose information like key points and features of the arms and hands, the raw RGB video, and optical flow which shows what changes are made between frames to show motion. The raw RGB video and optical flow were both seperately passed through 4 rounds of 3DCNN (3D convolutional neural network) models after which each generated a prediction. The key points of the video were generated by a pretrained pose estimation network to generate 133 different points on the signer's body, of which 27 were actively used to reduce the amount of noise. This was passed into a SL-GCN (Structure Learning Graph Convolutional network) to generate another prediction which will be synthesized into the final RGB prediction. Finally, they take feature data (different known body parts like the nose, mouth, shoulders, etc) from the RGB video and pass it through a SSTCN (Temporal Convolutional Network) to generate a 4th prediction. Finally the 4 predictions are put together by weighting each of the individual predictions based on experiment data to generate a final RGB prediction.

Next the model moves on to the depth video, which it splits into HHA and Depth flow. Each of these get put through 4 rounds of 3DCNNs to generate a prediction. These predictions are then added to the RGB prediction in a similar fashion as the the 4 predictions were synthesized to generate a final RGB-D prediction.

3 Possible Changes

Some of this seems very complicated and might not be feasible to implement, so here are some suggestions for what we might look to change if this is too difficult to implement. First of all, I don't know how important doing 4 rounds of 3DCNNs is, so doing it with only 1 round to start with might be a good start. Next, we can look at the weights they ended up with for combining the different predictions to see that the skeleton and RGB predictions were both weighted pretty heavily, 1.0 and .9 respectively. However flow and feature data were only weighted at .4, meaning they were less important to the final RGB prediction. For the RGB-D track, the depth prediction was weighted at .4 and the depth flow was weighted at .1, meaning it was not very useful for the final prediction. Given these weights, it seems reasonable to start with the skeleton and RGB models then add the others as necessary to improve performance.