|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
|  |  |  |  |
| **1** | **Conv3D** | **Distorted images** | **Experiment with crop size and resize (1) -**  **Since images are of two different dimensions, we first tried to resize the image (without cropping) so as to make all the images of size 120\*160.**  **When displaying the images, we notice that the 360\*360 size images are distorted as their aspect ratio has changed (we resized without cropping).**  **This is fixed by first cropping 120\*160 image and then resizing it to n\*n size so that the aspect ratio of the image is maintained.** |
| **2** | **Conv3D** | **Model runs out of memory or takes too much time with large number of frames.**  **Reduction in image size allowed the model to train.** | **Experiment with number of frames –**  **Each gesture case has 30 image samples, the movement in these images are very slow and hence we don’t have to pick all of them. If we pick all 30 frames then the model runs out of memory or takes too much time**  **We can choose to pick few of these images such that the movement is still recognizable at the same time training time and memory usage is limited.**    **We fist picked every 3rd image between 0 to 10 and 20 to 30 and every alternate one between 10 -20.**  **This way would pick a greater number of images which are at the center of the video. But unfortunately, the model didn’t perform that well.**  **So as an alternate we picked every alternate image between 0-30 i.e. 15 frames were picked.** |
| **3** | **Conv3D** | **Overfitting** | **Experiment with the trainable data-**  **To start with we used very few trainable data for conv3d model. This is to check the model performance on small amount of data. We noticed that accuracy is less and unstable. Moreover, there was overfitting.** |
| **4** | **Conv3D** | **Too many parameters and overfitting**  **Train accuracy around 0.65**  **and validation accuracy around 0.48** | **Experiment with crop size and resize (2) –**  **Initially we resized the images to 120\*120 for Conv3D models. This resulted in too many parameters (5Millions +) and bit of overfitting.**  **To overcome this, we reduced the image size to 75\*75 for Conv3D.** |
| **5** | **Conv3D** | **Train accuracy – 0.68**  **Validation accuracy 0.63** | **Increase the amount of trainable data -**  **Subsequently, we increased the amount of training data fed to the model which brought stability and increased the accuracy slightly.** |
| **6** | **Conv3D** | **Increase in accuracy.**  **Validation accuracy changed from 0.65 to 0.73** | **Experiment with filter size –**  **Filter size of 5\*5\*5 had high number of training parameters but the accuracy was also low (in mid 60s).**  **To increase the accuracy we reduced the filter size to 3\*3\*3, this of course decreased the number of training parameters as expected but this also impacted the accuracy in positive direction.** |
| **7** | **Conv3D** | **Max accuracy achieved changed with the increase in the number of layers but so does the number of the parameters** | **Experiment with number of layers –**   1. **Three layers and softmax–**   **32 >> 64>> 128(dense) 🡪 softmax**  **Each layer is followed by batch normalization and maxpooling.**  **The accuracy achieved was 0.61**   1. **Four layers and softmax –**   **16>>32>>64>>128(dense) 🡪 softmax**  **Each layer is followed by batch normalization and maxpooling.**  **In this case the accuracy increased to 0.72.**   1. **Four layers (increased neurons) and softmax –**   **32>>64>>128>>256(dense) 🡪 softmax**  **Each layer is followed by batch normalization and maxpooling.**  **Accuracy is significantly increased to 0.81** |
| **8** | **Conv3D** | **SGD with LR = 0.01 gave the best accuracy for our model.** | **Experiment with Optimizer –**  **We experimented with different optimizer and using different hyperparameter, some of the examples and accuracy achieved below –**   1. **SGD (lr = 0.001)**   **Accuracy – 0.58**   1. **SGD (lr=0.01, momentum=0.9, nesterov=False**   **Accuracy – 0.77**  **(changed the momentum to 0.85 but no significant change in accuracy)**   1. **SGD (lr = 0.01)**   **Accuracy - 0.81**   1. **Adam(lr = 0.01)**   **Accuracy – 0.75** |
| **9** | **CNN+RNN** | **Resnet50 + GRU became too complex for the data. The model was overfitting.**  **Best accuracy with decent amount of training parameters was achieved via VGG16 +GRU.** | **Experiment with CNN+RNN Model –**  **Since the accuracy from the Conv3D so far was not that great so we tried various flavors of CNN+RNN as well. Here are the models we tried and**   1. **Resnet50+GRU(64)**   **Very large (huge) number of trainable parameters – approx. 45 million.**  **At the same time the model was highly overfitting with training accuracy of 0.95 and test accuracy of 0.76.**  **Clearly this architecture is too complex for this use case.**   1. **VGG16+LSTM(64)**   **Trainable parameters were still large at 20millions and validation accuracy was at 0.75**   1. **VGG16+GRU(64)**   **Trainable parameter approx. to 4.8 million and the validation accuracy was 0.78** |
| **10** | **Both** | **Increase Accuracy** | **Experiment with sorting of image before feeding them to model -**  **We are getting decent accuracy here for both CNN+RNN and Conv3D model. But the model can still be improved.**  **While predicting the gesture we displaced the images and notice that the images are not in sequence. If the images are not fed in sequence to the model**  **then it would be very difficult for model to interpret the images and come to a correct conclusion.**  **To correct this, we are sorting the sequence of the images so that they are in proper sequence for gesture.**  **With this experiment the accuracy shot up for both the models.**  **Accuracy: from 0.78 to 0.90 for CNN+RNN and**  **0.71 to 0.87 for Conv3D** |
| **Final Model** | **Conv3D** | **Accuracy: 0.91** | **Criteria to choose final model -**  **We have tried both Conv3D and CNN+RNN model.**  **The final decision to choose the model is done based on following factors –**  **1. The number of parameters and prediction time (reflex time in this case) -**  **Conv3D -**  **Trainable params: 2,935,493**  **Prediction time - 00.011445 (i.e.. 0.011 milliseconds)**  **CNN+RNN -**  **Trainable params: 4,829,701**  **Prediction time - 02.086714**  **Clearly the number of parameters and prediction time both are significantly less for COnv3D.**  **Note that prediction time here is the reflex time of TV once gesture is made. This reflex time should be as less as**  **possible to have smooth operations.**  **2. The Accuracy of the model -**  **Conv3D -**  **Train accuracy: 97.26%**  **Validation accuracy: 91.00%**  **CNN+RNN -**  **Train accuracy: 96.43%**  **Validation accuracy: 95.00%**  **The accuracy (train and validation) for CNN+RNN are only slightly better than the Conv3D.**    **3. The models size -**  **Conv3D -**  **Model Size - 23.5 MB**  **CNN+RNN -**  **Model size - 97.6 MB**  **Model size of the Conv3D is less which makes it preferable to be used for TV as it will consume less memory.**    **Since Conv3D prediction is way faster than the CNN+RNN so we prefer Conv3D so that the reflex time of the TV to hand gesture is fast. Having a fast reflex is key for the smooth operations.** |