Experiment Number	Model	Model Performance		Result	Decision + Explanation	Total Model Parameters
		Training Accuracy	Validation Accuracy			
1	Conv3D			**ResourceExhaustedError** Experimented with batch_size like 10, 20, 30, 40 keeping the image size fixed at 160x160. But GPU memory was exhausted when batch_size was 40.	Based on the experiment - We can use batch_size between 20-30 - Image resolution either 160x160 or 120x120	1,734,661
2	Model-1 (Conv3D)	89.28%	75.83%	As a starting model it performed well and reached val_acc of 75% in the 1st 10 epochs itself. But as we can see the val loss decreased but spiked and started increasing.	Bring in regularization by adding Batch Normalization and dropout layers. Which might help the model to generalize better	1,734,661
3	Model-2 (Conv3D)	99.71%	27.50%	A huge gap can be observed in the training and validation accuracy, which signifies that the model is overfitting to a great extend.	- Reduce the batch_size to 20 - Decrease the filter size of Conv3D layers to (2, 2, 2) - Increasing dropouts to 0.5 - Increasing Sense layer neurons - Also training for more epochs	1,949,509
4	Model-3 (Conv3D)	95.00%	65.00%	There is a slight improvement in the validation accuracy. But the model is still overfitting	- Increasing the depth of the model (adding more layers) - Changing the image resolution to (120x120) - Increasing the filter size back to (3, 3, 3)	3,433,781
5	Model-4 (Conv3D)	85.29%	89.00%	Adding more layers and dropout has made the validation accuracy is higher than the train accuracy. This might be because we have used a high dropout %age (0.5) and as we know that dropouts "switches off" neurons in the training phase but during the validation all the neurons are activated and hence the increase in validation accuracy. But we can see that the model has performed much better as compared to the prevoius once and the overfitting is also reduced	The number of parameters in the model effects the time for inference and as we want to deploy the model for webcams we might need a model with lesser parameter and with better results. And all the models that we have trained so far for Conv3D have more than 1Million parametes. So now we will try reducing the number of parameters (model size).	2,556,533
6	Model-5 (Conv3D)	97.00%	93.00%	We can see that now the model is not overfitting as both the training and validation accuracy are comparable. Moreover we have achieved a pretty decent and our best validation accuracy till now.	Trying to reduce the model parametes further. - Reducing neurons in dense layers - Increasing dropouts	909,637
7	Model-6 (Conv3D)	93.68%	91.00%	We reduced the model parameters even more but the validation accuracy of model 5 was the best.	Now lets proceed and train RNN models, to compare their results with the best Conv3D model i.e. model-5	504,709
8	Model-7 (CNN + SimpleRNN)	92.54%	76.00%	As a stepping stone we tried **SimpleRNN** layer after the extracting features using Conv2D layers but we can see that the model is getting overfit.	SimpleRNN have the problem of vanishing/ exploding gradient in which the gradient information when propogated to many stages tends to vanish or explode. So we will use LSTM/ GRU networks which are designed to overcome these problems.	3,392,869
9	Model-8 (CNN + LSTM)	84.63%	67.00%	Even the custom LSTM model is overfitting. But it is comparable to using SimpleRNN provided it has almost half parameters and might have achieved similar accuray if trained for more epochs.	We can leverage the use of pre-trained CNN networks, as the are trained on a much larger dataset and the feature extracted will be similar. So instead of initializing our model with random initial weights, we will initialize it with pre-trained mobileNet model.	1,657,445
10	Model-9 (CNN + LSTM) with Transfer learning	99.85%	78.00%	As we are using pre-trained mobileNet model trained on 'imagenet' data and we are not training the layers of mobileNet. Hence we can see that the model is overfitting and the validation accuracy is very low.	Train the same model with making mobileNet layers also trainable.	3,001,669
11	Model-10 (CNN + LSTM) with Transfer learning, Training CNN layers	92.54%	94.00%	On training the mobileNet layers we can see that the train and validation accuracy has improved a lot. Even the model is not overfitting now.	We have got a much better model as compared to the model-5 in terms of accuracy but model-5 still has very less parameters as compared to this model.	3,840,453
12	Model-11 (CNN + GRU) with Transfer learning, Training CNN layers	100.00%	99.00%	The best model performace we have achieved.		3,692,869
Final Model	The best model is **Model-11 (CNN + GRU) with Transfer learning,Training CNN layers.** **Reason:** As it out performs all other models and has the highest training and validation accuracy that are 100% and 99% respectively. (**Filename:** "model_5/model_init_2022-02-2015_52_27.164407/model-00016-0.06678-0.99403-0.04007-0.99000.h5")					
	But if we take number of parameters into consideration then **Model-5 (Conv3D)** has also performed pretty decent. Training Accuracy: 97% and Validation Accuracy: 93%. (**Filename:** "model_11/model_init_2022-02-1914_16_54.364892/model-00026-0.09021-0.97382-0.50617-0.93100.h5")					