|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| **1** | **Conv3D** | **Stop Iteration Error** | **handled the left over Images which are still left after the number of batches have been exhausted.** |
| **2** | **Conv3D** | **Covolution + pooling hand in hand** | **As it gives us the power of Invariant features.** |
| **3** | **Conv3D** | **Accuracy: 0.62 and val\_accuracy = .42** | **Here the difference was the initial learning rate was 0.02** |
|  | **COnv3D** | **Accuracy: 0.45 and val\_accuracy = .40 after 30 epochs** | **Difference here was the learning rate of 0.001** |
|  | **COnv3D** | **with filter size 2,2,2** | **with filter size 2,2,2**  **accuracy reached to 68 and 48 and rest of the parameters were same.** |
|  |  |  |  |
| **3** | **RNN model with LTM cell** | **Accuracy : 0.54** | **started with learning rate of 0.001 and the model wa built with time Ditributed layer** |
|  |  |  |  |
| **l-1th** | **Conv3D** | **Accuracy: 0.70 and .52** | **here started with learning rate of 0.0004 a after 10 epochs runned previously, the lat learning rate given wa this.** |
| **lth** |  | **Accuracy:34%,**  **val\_accuracy : 35%** | **Here the drop-out used was 0.5** |
|  |  |  |  |
| **Final Model** | **GRU Plus Transfer Leraning** | **Accuracy :- 80**  **val\_accuracy :- 60** | **Here the dropout used was 0.25 and the dense layer used was 16.** |

Reason/Logic :-

Used the conv3D model as the image we have is the RGB image and the filtersize used because of that 3,3,3.

Used the batch Normalisation in the model

Used Convolution +Pooling hand in hand as it gives us the power of Invariant features,

Whether our main feature is present at the right corner or in the left corner, our model will detect that any how

In the generator function make the call for the cropping the image as the main action point is at almost the centre of the image.

Hence it would reduce the training time some how .

After that code for normailization of image would call. as it would uniform all the 663 images to some unique platform.

Handled the left over images by using simple mathematics as there are chances that even after the num\_batches are exhausted there could still be left over images.

Image-resize Have performed for Image\_width and Image\_height as (80,80) and (100,100) and (120,120).

Have acheived the best result in 100,100 shape.

For an 100,100 by 3 would require 30000 neurons which will be equal to the number of weights required

Also for RNN+CNN model used the LSTM cells.

As the LSTM gates would take care of the vanishing gradient problem.

Total numer of parameters to be trained