Record of our experiments on the given problem statement.

We have concluded that CNN+RNN model is the best fitting model in this case with accuracy of 87.69%.

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| **1** | **Conv3D** | **Accuracy: 0.55** | **Resize the images into (120,120), so we were loosing some important informations. We used every 3rd alternate image in the sequence. Kernel size = (3,3,3), no. of kernels in different layers= (32, 64, 128), dense connection in different layers = (32,64,128,5)**  **Activation Function = Relu** |
| **2** | **Conv3D** | **Accuracy: 0.58** | **To capture more information, we resized the image to (220,220), which improved the accuracy.**  **Used MaxPooling Layers to reduce the feature map size, Maxpooling = (2,2,2)** |
| **3** | **Conv3D** | **Accuracy: 0.58** | **To capture more information we used every alternate image, so 15 image was used for each sequence. This caused more training parameters.** |
| **4** | **Conv3D** | **Accuracy: 0.609** | **Used ReduceLROnPlateau to change the learning rate by a factor of 0.2 in case the val\_loss does not improve in 3 epochs** |
| **5** | **Conv2D + RNN** | **Accuracy: 0.84** | **Used Conv2D+RNN with kernel=(3,3), maxpooling=(2,2). Added Conv2D, Maxpooling in TimeDistributed cell**  **Kernel size = (3,3), no. of kernels in different layers= (32, 64, 128), dense connection in different layers = (32,64,128,5)** |
| **6** | **Conv2D + RNN** | **Accuracy: 0.74** | **Try changing the normalization of image, and re run with same setup** |
| **8** | **Conv2D + LSTM** | **Accuracy: 0.58** | **Used LSTM(512) to avoid the Vanishing gradient issue, with no dense connection in between** |
| **9** | **Conv2D + LSTM** | **Accuracy: 0.8** | **Used LSTM(512) to avoid the Vanishing gradient issue, with no dense connection in between** |
| **9** | **Conv2D + GRU** | **Accuracy: 0.85** | **Used GRU(150) to avoid using too many training parameters in LSTM and increased the no. of layers for conv2D in multiple of 2, starting from 32 till 1024** |
| **10** | **Conv2D + GRU** | **Accuracy: 0.8769** | **Used Timedistributed layers for Conv2D (3,3) in multiple of 2 starting from 32 till 1024, each Conv2D followed by Maxpooling2D (2,2).**  **Then added GRU(150), followed by softmax.** |
|  |  |  |  |
| **Final Model** | **Conv2D + GRU** | **Accuracy: 0.8769** |  |