|  |  |  |  |
| --- | --- | --- | --- |
| **Exp No.** | **Model** | **Result** | **Decision + Explanation** |
| 1 | Pre trained VGG16 + GRU | Overfits the training set. Val\_accuracy : 0.79 | Try with Dropout, batch Normalization and fewer units in GRU layer to reduce overfitting. |
| 2 | Pre trained VGG16 + GRU with dropout and batchNorm after VGG16 layer | Val\_accuracy improved to 0.92 from 0.79 | Since model still overfits, we will add dropout and batchNorm after all layer |
| 3 | Pre trained VGG16 + GRU with dropout and batchNorm after each layer | Accuracy: 0.81 on both train and validation set.Not overfitting. |  |
| 4. | Pre trained VGG16 + CNN3D  -BatchNorm after every CONV layer -Dropout after Dense layer.  -SGD with LR=0.01, Imagesize=200, 200  Number of images=15 | Model was not learning, accuracy fluctuated around 22%. | -Decided to go with a lower learning rate as gradient was not propagating towards global minima.  -Also parameters was around 16 million, to reduce it decided to try lower size images of 120, 120 |
|  | -Changed LR=0.001, imagesize=120,120 | Max of 78% validation accuracy reached. | Didn’t try this model further as it has 16 million parameters.  Decided to go with 3DCNNs |
|  |  |  |  |
| 5. | CNN3D  -CNN(8, 16, 32), DENSE(128)  - BatchNorm after CNN layers and Dropout after Dense layer  - started with small lr=0.0001, SGD  -15 images of size 120,120  -batchsize of 40 with 30 epochs | Max 62% accuracy reached with loss=1.22 | Model was not able to achieve decent accuracy.  Decided to increase one more CNN layer And also using image size as 200, 200 as now parameters are less we can afford using bigger image sizes to don't lose on much information. |
|  | - increase one CNN layer(8, 16, 32, 128)  -imagesize 200, 200 | 77% reached under 30 epochs with 0.75 val loss, training accuracy is 82% seems not overfitting. | - Generalising well with decent accuracy of 77%.  Wanted to try other options to improve accuracy.  Decided to change the optimization algorithm to Adam from SGD. |
|  | - Adam algorithm with ReduceLROnPlateau callback.  - | 69% accuracy reached with 81% training accuracy. seems overfitting and not learning well | - 2 million learning parameters  - reduce image size to 120, 120 as earlier it was not impacting accuracy much. |
|  | - image size 120, 120 | max 75% val accuracy reached with 92% train accuracy. seems overfitting | - decided to change architecture to increase filters.  - changed to 16,32,64,128 CNN layers |
|  | - Architecture changed to 16, 32, 64, 128  -also 15 images are used with 120, 120 that gave us the room to increase batch size to 100 from 20 or 10 before.  -Adam used with callback to reduce lr on plateau. | learning happens smoothly and max of 81% val accuracy achieved with 91% train accuracy. | Decent enough **accuracy of 81%** achieved with only **1.02 million** learning parameters, model size is less of only 4mbs. It can be deployed in small systems for our purpose. |
| **NOTE:** Model seems to learn enough with only 15 frames only instead of taking the whole 30 frames. Even experiments were made to strip first and last 5 images of sequence as they dont have much motion variations. That does not seem to impact model performance much. So most of the experiments were done using 15 alternate frames. That also reduces the learning parameters. | | | |