* Using the image resolution is 100x100 which gets the whole required data within this resolution from the image.

| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| --- | --- | --- | --- |
| **1** | CNN (Conv3D)  (Batch Size: 64  Total Frames: 16  Total Epochs:20) | Training: 88.18  Validation: 58.33  Trainable Params: 2,067,013 | * The model is Overfitted so let’s try to start experimenting with batch size, frame size, and epochs by considering trainable params as well * Experiment with increasing the frames and epochs |
| **2** | CNN (Conv3D)  (Batch Size: 64  Total Frames: 30  Total Epochs:30) | Training: 89.99  Validation: 44.44  Trainable Params: 9,005,637 | * Validation accuracy got reduced Try to reduce the trainable params and remodel the network * Model is overfitted more than the previous model due to an increase in total frames required & epochs |
| **3** | CNN (Conv3D)  (Batch Size: 64  Total Frames: 30  Total Epochs:20) | Training: 90.90  Validation: 50  Trainable Params: 5,036,005 | * Trainable params got reduced and model training, validation accuracy got increased. * Model is Still Overfitting * Need to Experiment by Increasing the epochs and reducing the frame size with the current model |
| **4** | CNN (Conv3D)  (Batch Size: 64  Total Frames: 20  Total Epochs:30) | Training: 85.90  Validation: 41.66  Trainable Params: 5,036,005 | * Reduction of frame size and increase of epochs doesn’t help us. It reduced the accuracies. * Model is Still Overfitting though * Let’s Experiment with the combination of 20 frames and 20epochs |
| **5** | CNN (Conv3D)  (Batch Size: 64  Total Frames: 20  Total Epochs:20) | Training: 81.81  Validation: 44.44  Trainable Params: 8,108,005 | * A combination of frame size 20 and epochs 20 doesn’t help better. * The overfitting problem still remains the same. Including with that model parameters also got increased much * Let’s try experimenting by moving Batch Normalization before the Max Pooling and reducing kernel size may help us in reducing params |
| **6** | CNN (Conv3D)  (Batch Size: 64  Total Frames: 20  Total Epochs:20) | Training: 98.18  Validation: 38.88  Trainable Params: 1,300,565 | * Used kernel size as (2,2,2) and pool size as (2,2,2) * Model params got reduced and training accuracy is enormously increased. * Overfitting still exists. * Let’s try to deeper the network with dropouts and normalization which can improve the validation accuracy |
| **7** | CNN (Conv3D)  (Batch Size: 64  Total Frames: 20  Total Epochs:25) | Training: 78.63  Validation: 41.66  Trainable Params: 397,477 | * Added dropouts of 0.25 in dense layers * Added more layers which helps in increasing the learning of the model * On creating the deep network, model parameters got reduced training accuracy got reduced and validation got increased a little bit. * Overfitting got reduced which can be observed by a difference in accuracies of train and validation sets * Let’s experiment by using this model by making more deeper by adding a layer with LSTM. Since LSTM is mostly used in video processing |
| **8** | Conv2D + LSTM  (Batch Size: 64  Total Frames: 20  Total Epochs:30) | Training: 77.72  Validation: 36.11  Trainable Params: 3,083,141 | * Train and validation accuracy doesn’t get improved well. * Let’s experiment by reducing the batch size for the current model |
| **9** | Conv2D + LSTM  (Batch Size: 32  Total Frames: 20  Total Epochs:30) | Training: 92.38  Validation: 68.05  Trainable Params: 3,083,141 | * On reduction of batch size, raining accuracy got increased well and validation too. * Model overfitting still exists. * Let’s Experiment with increasing the epochs so that model gets trained by 5 more epochs. This helps in more learning the model which in turn can increase validation accuracy |
| **10** | Conv2D + LSTM  (Batch Size: 32  Total Frames: 20  Total Epochs:35) | Training: 92.85  Validation: 80.55  Trainable Params: 3,083,141 | * On increasing the epochs training accuracy doesn’t change much but validation accuracy got increased. * Overfitting got reduced to some extent * Let’s experiment with Transfer learning to reduce the overfitting for more extent |
| **11** | Transfer Learning (LSTM)  (Batch Size: 32  Total Frames: 20  Total Epochs:35) | Training: 100  Validation: 81.94  Trainable Params: 1,380,869 | * Using mobilenet model * LSTM with less batch size doesn’t give validation accuracy much. * Overfitting got reduced by some extent * Let’s try to increase the validation accuracy by increasing the batch size which helps the model to learn more. |
| **12** | Transfer Learning (LSTM)  (Batch Size: 64  Total Frames: 20  Total Epochs:35) | Training: 100  Validation: 88.88  Trainable Params: 1,380,869 | * Increasing the batch size increased the validation accuracy more. * Overfitting got reduced. * Let’s try to reduce params using GRU since GRU uses fewer params |
| **13** | Transfer Learning (GRU)  (Batch Size: 64  Total Frames: 20  Total Epochs:35) | Training: 100  Validation: 80.55  Trainable Params: 1,053,701 | * Model Params got reduced & Training accuracy remains good. * Validation accuracy got reduced. * Let’s experiment with reducing the batch size and selecting only required frames. |
| 14 | Transfer Learning (GRU)  (Batch Size: 5  Total Frames: 16  Total Epochs:20) | Training: 100  Validation: 86.66  Trainable Params: 1,053,701 | * Model Params got reduced & Training accuracy remains good. Validation accuracy also got increased. * Overfitting of the model got reduced by some extent * Let’s check with 30 epochs as well which may increase validation accuracy. |
| **15** | **Final Model**  Transfer Learning (GRU)  (Batch Size: 5  Total Frames: 16  Total Epochs:30) | **Training: 100**  **Validation: 86.66**  **(@Epoch 19)**  Trainable Params: 1,053,701 | * **Model Params got reduced & Training accuracy remains good.** * **Validation accuracy also got increased.** * **Overfitting of the model got reduced by some extent** |

**Final Model:**

* **Transfer Learning(GRU)[i.e Experiment-15]**

**By,**

**Vinyas Shukla,**

**Srikanth Padmanabhuni**