**Overview of the models:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model No** | **Model** | **Trainable Params** | **Train Accuracy** | **Validation Accuracy** | **Experiments** | **Decision + Explanation** |
| **1** | **Conv3D** | **1,047,141** | **0.79** | **0.58** | **Kernel Size:** (3,3,3) and (2,2,2) for after initial layers.  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  As batch labels were already one hot encoded, Categorical Accuracy was used.  **Image Size:** 100\*100 (To reduce the computation)  **Frame Input:** 20  **Batch Size:** 30  **Epochs:** 10 (Took small number to check the model behavior, size, change in loss, computation time and change in different metrics i.e., training and validation. | **Started with less complex model with a smaller number of layers without any Regularization or Batch Normalization that results in overfitting.** |
| **2** | **Conv3D** | **1,047,141** | **0.93** | **0.58** | **Kernel Size:** (3,3,3) and (2,2,2) for after initial layers.  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 20  **Batch Size:** 30  **Epochs:** 10 | **As previous model was overfit with less validation accuracy, We**  **added dropout and batch normalization on each Conv3D layers to overcome overfitting.**  **Though it increased the training accuracy, there was not much improvement in validation accuracy** |
| **3** | **Conv3D** | **568,421** | **0.95** | **0.68** | **Kernel Size:** (3,3,3) for all Conv3D layers  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 20  **Batch Size:** 50 (To check increment in the model performance)  **Epochs:** 10 | **Added more layers (Conv3D and Dense), increased kernel size for all layers and removed dropout after each Conv3D layer except the last layer.**  **Validation Accuracy doesn’t show any improvement but validation loss was decreased whereas training accuracy was increased.** |
| **4** | **Conv3D** | **568,421** | **0.39** | **0.18** | **Kernel Size:** (3,3,3) for all Conv3D layers  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 20  **Batch Size:** 50  **Epochs:** 10 | **Added regularizations after each layer to overcome overfitting.**  **Validation and Training accuracies dropped significantly as model went underfit.** |
| **5** | **Conv3D** | **568,421** | **0.82** | **0.55** | **Kernel Size:** (3,3,3) for all Conv3D layers  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 20  **Batch Size:** 50  **Epochs:** 10 | **Removed Dropout after each Conv layer except the last layer and Dense layers to retain the learning features and readjusted order of batch normalization and activation**  [**Reference**](https://stackoverflow.com/questions/39691902/ordering-of-batch-normalization-and-dropout)  **Training loss as well as Validation loss increased wrt previous model but not better than model 3 yet.** |
| **6** | **Conv3D** | **4,386,181** | **0.21** | **0.21** | **Kernel Size:** (3,3,3) for all Conv3D layers  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 20  **Batch Size:** 50  **Epochs:** 10 | **Increased dimensions of feature map on each Conv layer that increased the total parameters. Added dropout prior to pooling layer in the last conv layer. Reduced the number of Dense layers**  **Model went untrained. There was no difference in training and validation accuracy and loss possibly because of adding dropout prior to the pooling reason.** |
| **7** | **Conv3D** | **8,401,029** | **0.19** | **0.23** | **Kernel Size:** (3,3,3) and (2,2,2) after initial layers  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 20  **Batch Size:** 50  **Epochs:** 10 | **Decreased kernel size after the starting layer and removed last Conv layer. Reordered dropout after pooling layer.**  **Model went untrained with worse result than previous.** |
| **8** | **Conv3D** | **899,893** | **0.99** | **0.63** | **Kernel Size:** (3,3,3) and (2,2,2) after initial layers  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 15  **Batch Size:** 50  **Epochs:** 10 | **Added Conv layer with less Feature Maps and Decreased Kernel size. Batch Normalization was added after each Conv layer to improve performance stability.**  **In comparison to model 3, training accuracy and loss was better whereas no improvement in validation accuracy and loss** |
| **9** | **Conv3D**  **(Best among all Conv3D models wrt loss and accuracy and model output file .h5 is attached with the Attachment model-00020-0.43257-0.82428-0.49072-0.82000.h5)** | **899,125** | **0.81** | **0.82** | **Without Augmentation:**  **Kernel Size:** (3,3,3) and (2,2,2) after initial layers  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 25  **Batch Size:** 30  **Epochs:** 20 (As model performance was better than others)  **With Augmentation:**  **Kernel Size:** (3,3,3) and (2,2,2) after initial layers  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100 (Tried with different image res that were higher that resulted with OOM error for all other possible combinations)  ***Frame Input:*** *10 & 15* (To match available computation power)  **Batch Size:** 30  **Epochs:** 20 (As model performance was better than others) | **Removed Batch Normalization and added Dropout in the dense layer without Augmentation and with Augmentation (Rotation, Scaling, Erosion, Opening and Thresholding)**  **Validation loss acquired without Augmentation was lesser than previous model.**  **Difference in train and validation accuracy was high.**  **Validation loss acquired with Augmentation was least till now.**  **Difference in train and validation accuracy was very less.** |
| **10** | **Conv3D** | **172,901** | **0.85** | **0.69** | **Kernel Size:** (3,3,3) and (2,2,2) after initial layers  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 20  **Batch Size:** 30  **Epochs:** 20 | **Changed feature maps of Conv layers, decreased neurons, Added Batch Normalization in the dense layer and reduced training parameters.**  **Parameters were reduced to about 4 times.**  **Training loss was lesser than model 9 and validation loss was increased.** |
| **11** | **Conv3D** | **177,701** | **0.94** | **0.69** | **Kernel Size:** (3,3,3) and (2,2,2) after initial layers  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 20  **Batch Size:** 30  **Epochs:** 20 | **Removed Dropout from the dense layers except the last layer.**  **Training accuracy and training loss improved but no improvement in validation loss and accuracy.** |
| **12** | **Conv3D** | **899,125** | **0.84** | **0.60** | **Kernel Size:** (2,2,2)  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 15  **Batch Size:** 30  **Epochs:** 20 | **The model is similar to model 3 except the first Conv layer.**  **No regularization parameters have been used except on last dense layer and learning rate on optimizer was specified as 0.0001.**  **It was tested with Less Augmented data (Erosion, Opening and Thresholding) to tackle the over usage of memory issue**  **Model was overfit and validation loss and accuracy showed no improvement.** |
| **13** | **Conv2D + GRU** | **2,102,389** | **0.35** | **0.40** | **Kernel Size:** (2,2)  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 15  **Batch Size:** 30  **Epochs:** 10 | **Added 3 layers of Conv2D to keep the model simple with one GRU with smaller number of neurons and Dropout at the last Dense layer.**  **Model performance and loss were not reliable, were fluctuating and not improving.**  **Possibly because the model was naïve.** |
| **14** | **Conv2D + GRU** | **1,266,149** | **0.59** | **0.52** | **Kernel Size:** (3,3) and (2,2) after initial layer  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 15  **Batch Size:** 30  **Epochs:** 10 & 15 | **Added one more layer of Conv2D and removed dropout from the dense layer.**  **There was not much improvement in the result but model started learning.**  **To increase the model performance, No of epochs were increased. Even then, model performance was decreased.** |
| **15** | **Conv2D + GRU** | **1,002,277** | **0.38** | **0.2** | **Kernel Size:** (3,3)  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 15  **Batch Size:** 30  **Epochs:** 10 | **Increased kernel size of every Conv2D layers and added Batch Normalization in the dense layer. Changed the padding type in pooling layers.**  **Model went underfit, training accuracy was higher than validation accuracy whereas overall performance of model was very low as loss didn’t improve throughout the training** |
| **16** | **Conv2D + GRU** | **907,573** | **0.2** | **0.18** | **Kernel Size:** (3,3)  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 15  **Batch Size:** 30  **Epochs:** 10 | **Made the model simpler by removing last Conv2D layer. Also, learning rate was specified in the model as 0.01.**  **Model didn’t learn. Performance of the model was worst of all Conv2D models.** |
| **17** | **Conv2D + GRU** | **512,855** | **0.86** | **0.59** | **Kernel Size:** (3,3)  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 15  **Batch Size:** 20  **Epochs:** 10 | **Added 2 layers at the start and 1 layer at the end to make the model a bit complex as in model 13 that had shown an improvement. Also,**  **changed pooling padding to ‘same’ with stride (2,2)**  **Training and validation accuracy as well as training and validation losses significantly improved wrt previous models but the model went overfit.** |
| **18** | **Conv2D + GRU** | **509,333** | **0.99** | **0.78** | **Kernel Size:** (3,3)  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 25  **Batch Size:** 30  **Epochs:** 20 | **Removed the last layer and added GRU layer with a greater number of neurons after the Dense layer (This model of out of convention)**  **There was good improvement with training and validation accuracy and loss but the model still overfits.** |
| **19** | **Conv2D + GRU** | **983,909** | **0.99** | **0.80** | **Kernel Size:** (3,3)  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 25  **Batch Size:** 30  **Epochs:** 20 | **Added dropout after every dense layer and last Conv2D layer**  **Model’s validation accuracy and loss improved slightly but couldn’t deal with overfitting.** |
| **20** | **Conv2D + GRU** | **10,609,253** | **0.21** | **0.21** | **Kernel Size:** (3,3)  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 120\*120  **Frame Input:** 15  **Batch Size:** 30  **Epochs:** 20 | **Complex model based on VGG Architecture. In GRU layer, dropout and recurrent dropout was specified as 0.5 along with Batch Normalization.**  **The model went underfit and performance was not reliable.** |
| **21** | **Conv2D + GRU** | **2,907,685** | **0.20** | **0.21** | **Kernel Size:** (3,3)  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 15  **Batch Size:** 40  **Epochs:** 10 | **Complex model based on simpler VGG Architecture.**  **The number of parameter decreased by 5 times as compared to previous model.**  **The model didn’t show any improvement and went underfit.** |
| **22** | **Conv2D + LSTM** | **10,684,165** | **0.21** | **0.23** | **Kernel Size:** (3,3)  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 15  **Batch Size:** 20  **Epochs:** 20 | **Complex model based on VGG architecture (based on 20th model’s architecture with LSTM)**  **The model didn’t show any improvement and went underfit.** |
| **23** | **Conv2D + LSTM** | **10,684,165** | **0.28** | **0.23** | **Kernel Size:** (3,3)  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 15  **Batch Size:** 20  **Epochs:** 20 | **Conv2d architecture based on the VGG Architecture. LSTM is being used and followed by Dense.**  **The model didn’t show any improvement and went underfit.** |
| **24** | **Transfer Learning Conv2D + LSTM** | **3,625,157** | **0.30** | **0.26** | **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 15  **Batch Size:** 20  **Epochs:** 10 | **Using pre trained weights of MobileNetV2(**[**As it has the least number if training params**](https://meet.google.com/linkredirect?authuser=0&dest=https%3A%2F%2Fkeras.io%2Fapi%2Fapplications%2F)**) and added some of our own last layers that includes pooling, LSTM and Dense layers.**  **The model’s performance didn’t show improvement wrt validation.**  **Possibly because pre trained weights were used.** |
| **25** | **Transfer Learning Conv2D + LSTM + Training Weights** | **3,625,157** | **0.75** | **0.51** | **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 15  **Batch Size:** 20  **Epochs:** 10 | **Training weights in the MobileNetV2 and adding some of our own last layers.**  **Model showed significant improvement wrt previous model. Training and validation loss and accuracy were significantly improved but the model remains overfit.** |
| **26** | **Transfer Learning Conv2D + LSTM + Training Weights** | **3,623,877** | **0.92** | **0.79** | **Without Augmentation:**  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 15  **Batch Size:** 20  **Epochs:** 10  **With Augmentation:**  **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 80\*80  **Frame Input:** 15  **Batch Size: 1**0  **Epochs:** 15 | **Removed Batch Normalization after the Dense layer from the previous model and Augmented data was also fed.**  **Model showed good improvement in training and validation metrics but the model still overfits (Significant difference in training and validation accuracy).**  **With Augmentation, validation accuracy and loss hit the plateau and model overfits.** |
| **27** | **Transfer Learning Conv2D + GRU + Training Weights** | **4,328,517** | **0.94** | **0.81** | **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 20  **Batch Size:** 10  **Epochs:** 15 | **Trying the same with GRU instead of LSTM**  **Training and Validation loss and accuracy shows slight improvement. Model performance was better than the previous.** |
| **28** | **Transfer Learning Conv2D + GRU + Training Weights***(Best among all Conv2D and RNN models)*  [Link of the model saved](https://drive.google.com/file/d/18DoQYLwEwXRkIbmJlCSJ984_bC76TDx-/view?usp=sharing) | **4,328,517** | **0.93** | **0.77** | **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 100\*100  **Frame Input:** 15  **Batch Size:** 10  **Epochs:** 25 | **Trying the same with Augmentation**  **Model showed the best performance at 22nd epoch where validation loss was the least and accuracy were the best.** |
| **29** | **Transfer Learning CNN 2D + GRU + Training weights** | **21,755,437** | **0.99** | **0.84** | **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 80\*80  **Frame Input:** 15  **Batch Size:** 10  **Epochs:** 25 | **Using pre trained weights of Xception (As it’s the next light weight model and having high Top-5 accuracy after mobilenet) and added some of our own last layers. Also fed Augmented data.**  **Model showed good improvement in training and validation metrics but the model still overfits (Significant difference in training and validation accuracy).** |
| **30** | **Transfer Learning CNN 2D + GRU + Training weights** | **24,481,669** | **0.99** | **0.79** | **Optimizer:** Adam  L**oss**: categorical\_crossentropy  **Activation Function:** relu  **Metric:** Categorical Accuracy  **Image Size:** 80\*80  **Frame Input:** 15  **Batch Size:** 10  **Epochs:** 25 | **Using pre trained weights of ResNet(Next light model after Xception) and added some of our own last layers. Also fed Augmented data.**  **Model’s performance goes down in comparison with previous model as validation loss and accuracy falls and shows no improvement after a certain epoch.** |