Detailed Observation Table

Experiment Name	Model	Result			Decision + Explanation
		Total Parameters	Best Validation Accuracy	Corresponding Training Validation	
Without Data Augmentation					
Model 1	Conv3D	892,101	76%	95%	Starting with - image size = 100X100 - frames to sample = 15 - Batch size = 20 - No.of epochs = 15 - Filter size (3,3,3) It's definitely overfitting and increase in validation loss after epoch 6 indicates that the system needs optimization
Model 2	Conv3D	892,101	90%	85%	Changing - Batch size = 10 - No.of epochs = 25 - Dense neurons = 256 - Dropout is 0.5 Same number of parameters as model 1 but adding dropout rate to 0.5 removed overfitting. However, the validation loss is not unstable and oscillates. We would prefer a model to converge and show stable loss. We will set
Model 3	Conv3D	1,336,629	92%	85%	Changing - filter size (2,2,2) - dense neurons = 64 - dropout = 0.25 - learning rate = 0.0003 - No. of epochs = 30 The model learnt effectively and better accuracy on validation set suggest it is not overfitting and performs well on validation data.
Model 4	Conv3D	2,130,549	91%	80%	We added convolutional layer so that it can even extract more features and do well on unseen data too. We also added dropout layers to avoid overfitting due to added complexity. Even if the model is not overfitting and promising validation accuracy. The validation loss oscillates

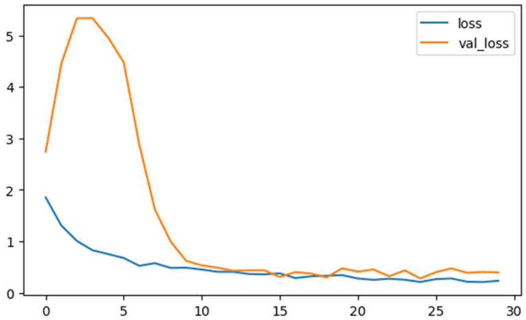
					vividly and does not shows a good learning pattern. Adding more layers was not a good idea.
Model 5	Conv3D	444,613	87%	84%	We changed - Num_epochs = 22 as most learning is getting done after 15 - Dense neurons = 32 - Filter size = (3,3,3) - Now the parameters have dramatically reduces to 450K. The model show good learning pattern on 20% of the total parameters of Model 3. Even though the Model 5 shows lower loss and higher accuracy that are desirable. Model 3 still seems to generalize better to unseen data. With 92% of validation accuracy.
Model 6	Conv + RNN with LSTM	1,657,445	90%	98%	This model is a combination of CNN and RNN layers. The CNN layers are used to extract spatial features and the LSTM layer is used to extract temporal features. The CNN layers are wrapped in TimeDistributed layers to allow the model to process each frame independently. The extracted features are then flattened and passed through an LSTM layer. The output of the LSTM layer is then passed through two dense layers with dropout in between for regularization. - Hyperparameters: - Image size: 100x100 - Frames to sample: 15 - Batch size: 10 - Number of epochs: 20 - Learning rate: Default (0.01) It seems to be overfitting because the training loss is significantly lower than the validation loss.
With Data Augmentation					
Final Model : Model 7 (Extension of Model 3 with Augmentation)	Conv3D	1,336,629	93%	92%	This is model 3 architecture and doing as well as model 3. This is the model we are looking for as it is doing better on augmented data too.
Model 8 (Extension of Model 5	Conv3D	444,613	87%	92%	Initially, the loss is high, but as the training progresses, the loss gradually decreases,

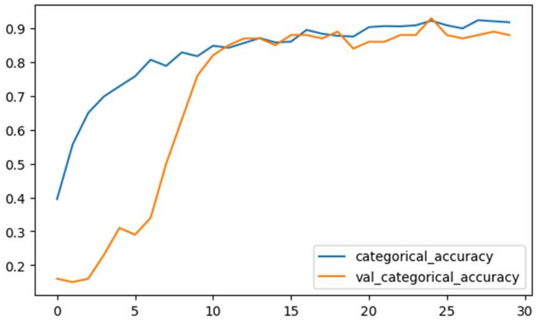
with Augmentation)					signifying improvement. However, the validation loss oscillates and also it overfits.
Model 9 (Extension of Model 6 with Augmentation)	Conv3D	1,657, 445	96%	98.87%	Validation loss is oscillating a lot and is not smooth. The difference between accuracy of training and validation is also fluctuating. The accuracy numbers should not be the only parameter considered for deciding on the model
Model 10	CNN +RNN with LSTM	1,657, 445	94%	99.77%	Validation loss increases after epoch # 10. The categorical and validation accuracy isn't getting better in comparison to Model 3 or 7 (which were our promising models)
Model 11	Transfer Learning CNN + RNN with LSTM	3,840, 453	91%	97.21%	Transfer learning on mobilenet gave high and oscillating validation loss and the accuracy was unstable too even though it seemed that it boosted the model's accuracy
Model 12	Transfer Learning CNN + RNN with GRU	3,451, 205			ResourceExhausted Error. Even though GRU cells create fewer number of parameters (almost equal to Model 11). The intermediate tensors during backpropagation ca cause memory exhaustion

Conclusion

After these experiments, we selected Model 7 – Conv3D with data augmentation. The model is - model-00025-0.20884-0.92232-0.27682-0.93000.h5

We tested our model and the model made prediction with 90% accuracy.





The steep slope of the validation loss curve early on indicates effective learning. After epoch 10, we reached saturation point suggesting diminishing returns. The validation loss remains stable suggesting that the model has converged without significant divergence or oscillation.

Further ideas

We were not able to test some of these ideas due to limitations like time, computing capacity etc.

- Experimenting with different activation technique other than ‘relu’. Few models could have done better on the loss curve. Other like Leaky ReLU which introduces a small gradient for negative input values, preventing the dying ReLU problem but we did not go for it as training process takes longer
- Experimenting using SGD which is the most basic form of gradient descent and claims to generalize better, instead we used ADAM as it converges faster adaptive learning and we were short of computational capacity. We made sure to run sample models and tune hyper parameters to have ADAM perform better.
- We could have used more data augmentation technique beyond geometrical augmentation.
- Using GRU cells instead of LSTM: This was attempted but ran into resource exhaustion error.
- Experimenting with other pre-trained models like ResNET to identify initial features. We could not attempt it due to lack of resources and time.