This is a sample write-up. The write-up need not be in tabular form.

It doesn’t state that ConvLSTM will give you better results than Conv3D. The explanation should be as detailed as possible so that the logic behind the decision is conveyed. Also, there are a lot of things you can experiment with in the generator function and elsewhere. Please do not forget to specify the exact metric values, here Accuracy which drives your decision.

You can draw inspiration from the concepts taught in the Industry demo in CNNs to experiment with the data and different architectures.

**Answer:**

Two model have been built.

* A CNN3D model
* A CNN2D + RNN based model

Summary of models is below:

**Conv3D Models**

|  |  |  |  |
| --- | --- | --- | --- |
| Experiment Number | Model | Result | Decision + Explanation |
| 1 | Conv3D – Tried to reach maximum Batch size | Throws Generator error at batch size 50 | Continued ahead with Batch size 40 |
| 2 | Conv3D – Simple model with 2 3DConv layers | Validation accuracy is not increasing | Model is not learning well.  Number of layers may be very less. |
| 3 | Conv3D – Make a deeper network | Validation accuracy is not increasing | Architecture change could not work well. Batch size should be changed |
| 4 | Conv3D – Previous model with  reduced batch size of 25 | Train accuracy is closed to 100%  Validation accuracy is increased | Learning rate is very less.  Architecture should be changed |
| 5 | Conv3D – 2 Conv 3D layers, the dense layer neurons reduced.  Only one Batch normalization to  the first layer | Train data loss suddenly fall at 2nd epoch.  Validation loss improved | Model is not learning well.  The hyperparameters should be changed and checked. |
| 6 | Conv3D – More images per  video, reduced the image size | Train accuracy is almost 100% from the fifth epoch  Validation accuracy is low | Model is not learning well |
| 7 | Conv3D – elu activation instead of  Relu, to avoid Dying Relu problem  because that might be causing the  dying out of loss. | Validation accuracy is low | Model is not learning well.  The architecture should be changed |
| 8 | Conv3D – the model  Simpler while hyperparameters keep the same | Validation accuracy improved | The improvement is not enough. The model is still not learning well. |
| 9 | Conv3D – the normalization changed in the generator. | Validation accuracy is 85%  Train categorical accuracy is 85.07% | Model is learning.  Problem is previous normalization.  However, Model has too many parameters |
| 10 (Final Model) | Conv3D –the model deeper but trainable  Parameters are less (only 234,549) | Validation accuracy is 73%  Train categorical accuracy is 76.17% | It's performance is not excellent as previous model but it should be considered if we want a small and lightweight model, and occupies less memory.  This one is the best model which runs in a smart TV. |

**Conv2D + RNN**

|  |  |  |  |
| --- | --- | --- | --- |
| Experiment Number | Model | Result | Decision + Explanation |
| 1 | Conv2D+RNN  (GRU) | Train categorical accuracy is 88.24%  Validation accuracy is 74% | The model is learning.  Trainable params are very high: Trainable params is 2,548,485, and not suitable for business |
| 2 | Conv2D+RNN  (GRU) –  Reduced  parameters /  Made deep | Train categorical\_accuracy: 0.8084  Val\_categorical\_accuracy: 0.6800 | Model has less parameters. Perfomance is better than the previous one |
| 3 | Conv2D+RNN  (GRU) with  Transfer  Learning  (MobileNet) | val\_categorical\_accuracy: 0.5100  categorical\_accuracy: 0.5943 | Model is not learning well, it is not used. |