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.

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| **1** | **Conv2D + GRU** | **Generator Code Failed (Checked initially for remaining folders – the code for fetching the folder was wrong)  Therefore, some error was throwing up again and again, had to debug it step by step using print statements etc.** | **So basically after the loop is completed for the batches, whatever is remaining, needs to be allocated within the code file using num\_batch (basically batch after loop + 1) in the imgs code snippet within the generator**  **Initial run of the code had cropping 120,120 but later changed it to 100,100** |
| **2** | **Conv2D + GRU (single GRU layer) (Batch Size = 100)**  **Images sequence used = [0,3,6,9,12,15,18,21,24,27,29]** | **Throws error while doing model.fit()**  **First trial with epochs 20, the results were poor, therefore ran another iteration.**  **However, this code failed in the snippet while ModelCCheckPoint parameter save\_freq = num\_epochs.**  **Due to this, model.fit() failed after just the first epoch. Had to change save\_freq = ‘epoch’ to resolve this issue** | **Image cropping is correct with 100,100,3. However, batch size needs to be reduced. Checked total number of folders, in train it is 663 and in validation folder it is 100. Therefore, a better batch size would be 64, due to which the while loop within the generator runs 10 (while) + 1 times (remaining) for train and twice for validation.**  **Also changed the folder creation code part, to maintain a schema of which structure is generating which models** |
| **3** | **Conv2D + LSTM (Two Layers)  Image Sequence: Thought of running using all the images in the sequence** | **The code failed, not sure, but I assumed the kernel to have died, tried this thrice but had to restart kernel** | **Decided not to iterate with all the 30 images due to processing constraint in my system (Jarvis GPU also got stuck)** |
| **4** | **Conv2D + LSTM (Two Layers)**  **Image Sequence or indexing used**  **[0,2,4,6,8,10,12,14,16,18, 20, 22,24,26,28]** | **Model Trains but takes a long time to run due to the larger number of parameters in LSTM. Kept Save best model parameter to True, and got just one model with an accuracy of 0.30.** |  |
| **5** | **Conv2D + GRU (Two Layers) Image Sequence or indexing used**  **[0,2,4,6,8,10,12,14,16,18, 20, 22,24,26,28]** | **At this point, I realized that I might have to create the generator function again and again if I am changing the sequence. Therefore, re ran all the previous code and updated generator function to have img\_idx passed as an argument to the generator function itself rather than defining it inside the function.** |  |
| **6** | **Conv2D + GRU (Two Layers) Image Sequence or indexing used**  **[0,2,4,6,8,10,12,14,16,18, 20, 22,24,26,28]** | **After re running with the updated generator code, ran the updated model. For some reason, got only 0.39 categorical accuracy from one of the models. Note: Save\_Best\_Model was on for this iteration as well. For some reason, could not save a model but achieved an accuracy of 0.59 (Validation categorical Accuracy)** | **Decided to keep the parameter save\_best as False itself, to get all the models built during the epochs. This can help us to choose and backtrack and go to the model of our choice.**  **Additionally did not re-run the code to save all models, as I was planning to look at the results first with Conv3D model** |
| **7** | **Conv 3D**  **Image Sequence or indexing used**  **[0,2,4,6,8,10,12,14,16,18, 20, 22,24,26,28]** | **Tried running with 100 epochs**  **Code did not run** | **Again tried 75 epochs as well, but code took too much time to even start, hence kept 50 epochs consistent throughout** |
| **8** | **Conv 3D**  **Image Sequence or indexing used**  **[0,2,4,6,8,10,12,14,16,18, 20, 22,24,26,28]** | **Train Categorical Accuracy = 0.78 and Validation Categorical Accuracy = 0.68** | **Considering industry standards with real world data, this can be considered as a very good performing model**  **May be one of the final models** |
| **9** | **Conv3D – but generator code changed to include only grayscale image**  **Sequence remains same as the previous** | **First had to change generator code to get only grayscale image**  **Conv3D architecture remained the same as before except input size with 1 channel instead of 3**  **This was done considering the fact that it is just gesture recognition and color should not bother, would have really reduced the number of trainable parameters**  **Train categorical accuracy: 0.81**  **Validation Categorical Accuracy: 0.48** | **Shows signs of overfitting, and the model did not give results as expected. Therefore, decided to give the final try again with the similar model as Conv3D with colored image** |
| **10** | **Conv3D - re run with a small tweak in model architecture** | **Accuracy: 0.75 (Train)**  **Val Categorical Accuracy: 0.50** | **Did not receive that good results as Experiment No 8. This might have been due to random initial weight assignment (assumption). Maybe using some proper process for initial weight allocation like Xavier Glorat etc might have given consistent results.** |
|  | **Note: The first correctly run code with one layer of GRU gave few models with train categorical accuracy around 0.97 and val categorical accuracy around 0.72** |  | **This still seems to me like a case of overfitting, therefore, final model to be put will be based on experiment 8.** |
| **Final Model** | **Conv3D** | **Train Categorical Accuracy = 0.78 and Validation Categorical Accuracy = 0.68** | **Considering the types of models which goes to production in corporate/industry, this is more likely to be placed in production even though the GRU gave higher accuracy considering 90+ is never usually taken in industry** |