Gesture Recognition write table below explain the different experiments conducted in Conv2D and Conv3D networks to classify Hand recognition out of the 30 images from each 663 videos into 5 different classes/categories.

Below table give the brief understanding of the results (Training Vs Validation – Loss/Accuracy) comparison with a simple trend chart across each epochs in different network models for various batch sizes and few changes in network structure to attain better result.

Note: I have zipped the H5 file of Final Model (**3DCONV\_EXPERIMENT\_03.ipynb**) in the google drive link as below:

<https://drive.google.com/file/d/1Kd2lmR5I33P0LaMdZyeMlRb9kKyOLWW3/view?usp=sharing>

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| **1** | **Conv2D** | loss: 1.64  categorical\_accuracy: 0.23  val\_loss: 1.60  val\_categorical\_accuracy: 0.23 | IPython Notebook: **2DCONV\_RNN\_EXPERIMENT\_01.ipynb**  Parameters:  **batch\_size = 5**  **epochs = 10**  **There is lot of variation in training and validation accuracy** |
| **2** | **Conv2D** | loss: 0.74  categorical\_accuracy: 0.78  val\_loss: 0.91  val\_categorical\_accuracy: 0.71 | IPython Notebook: **2DCONV\_RNN\_EXPERIMENT\_02.ipynb**  Parameters:  **batch\_size = 15**  **epochs = 20**  **After Increasing the batch size to 15 and epochs to 20, I could see there is a huge difference (Overfitting) between training and validation accuracy after 8th epochs.** |
| **3** | **Conv2D** | loss: 0.83  categorical\_accuracy: 0.75  val\_loss: 1.08  val\_categorical\_accuracy: 0.63 | IPython Notebook: **2DCONV\_RNN\_EXPERIMENT\_03.ipynb**  Parameters:  **batch\_size = 20**  **epochs = 10**  **After increasing the batch size to 20 and epochs to 10, still there is a lot of difference between validation and training accuracy after 5th epochs** |
| **4** | **Conv2D** | loss: 0.93  categorical\_accuracy: 0.69  val\_loss: 1.11  val\_categorical\_accuracy: 0.59 | IPython Notebook: **2DCONV\_RNN\_EXPERIMENT\_04.ipynb**  Parameters:  **batch\_size = 10**  **epochs = 10**  **After setting the batch size and epochs to 10, the training and validation accuracy is more or less in sync will little difference. But the accuracy is below 70%. So, decided to try Conv3D network.** |
| **5** | **Conv3D** | loss: 7.57  categorical\_accuracy: 1.00  val\_loss: 8.28  val\_categorical\_accuracy: 0.72 | IPython Notebook: **3DCONV\_EXPERIMENT\_01.ipynb**  Parameters:  **batch\_size = 10**  **epochs = 10**  **Hidden layer Count = 4**  **Training and validation accuracy are very low.** |
| **6** | **Conv3D** | loss: 7.99  categorical\_accuracy: 1.0  val\_loss: 8.88  val\_categorical\_accuracy: 0.77 | IPython Notebook: **3DCONV\_EXPERIMENT\_02.ipynb**  Parameters:  **batch\_size = 10**  **epochs = 10**  **Hidden layer Count = 3**  **Decided to reduce 4 Hidden layers to 3 to avoid overfitting.**  **Still could not achieve best validation accuracy.** |
| **Final Model - 7** | **Conv3D** | loss: 9.0572  categorical\_accuracy: 1.0  val\_loss: 9.75  val\_categorical\_accuracy: 0.82 | IPython Notebook: **3DCONV\_EXPERIMENT\_03.ipynb**  With Parameters:  **batch\_size = 20**  **epochs = 10**  **Hidden layer Count = 3**  **Increased the batch size to 20.**  **Was able to achieve a validation accuracy to 82%.** |