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** |
| Attempt 1 | Conv + GRU | Training Accuracy – 82.07  Validation Accuracy – 79.00 | Conv2D Models with Layers (32,32,32,64,64,128,256) added  With TimeDistributed  Provided Parameters are –  Frames : 10  Images : 100 X 100  Channels : 3 (10,100,100,3)  Trainable Parameters : 1084247  Batch Size : 25  Num of Epochs : 25  Added MaxPooling and BatchNormalization and Dropout  In Each Layers we have used Kernel and bias L2 regularizer of 0.01 |
| Attempt 2 | CONN + GRU | Training Accuracy – 75.86  Validation Accuracy – 72.00 | Conv2D Models with Layers (32,32,32,64,64,128,256) added  With TimeDistributed and appended with a GRU later at the last  Provided Parameters are –  Frames : 10  Images : 100 X 100  Channels : 3 (10,100,100,3)  No of Parameters : 1084247  Batch Size : 25  Epochs : 30  Added MaxPooling and BatchNormalization and Dropout  In Each Layers we have used Kernel and bias L2 regularizer of 0.01 |
| **Final Model** | **Conv2D + GRU** | **Training Accuracy – 94.93**  **Validation Accuracy – 85.83**  **Parameters :**  **1,273,525** | Batch Size : 30; Num of Epoch : 30  Frames : 10; Pixels : 100 X 100; Channels : 3  **Model :**  1. Conv2D - 32 Layers - Same Padding - with Kernal Size (1,1) – TimeDistributed -  Relu Activation  MaxPooling2D - (3,3) with Strides (2,2)  BatchNormalization    2. Conv2D - 32 Layers - Same Padding - with Kernal Size (3,3) – TimeDistributed -  Relu Activation  MaxPooling2D - (3,3) with Strides (2,2)  BatchNormalization  3. Conv2D - 64 Layers - Same Padding - with Kernal Size (3,3) - TimeDistributed and kernel\_regularizer(l2(0.01)) and bias\_regularizer(l2(0.01))  Relu Activation  BatchNormalization  DropOut with 0.25    4. Conv2D - 64 Layers - Same Padding - with Kernal Size (3,3) - TimeDistributed and kernel\_regularizer(l2(0.01)) and bias\_regularizer(l2(0.01))  Relu Activation  MaxPooling2D - (3,3)    5. Conv2D - 64 Layers - Same Padding - with Kernal Size (3,3) - TimeDistributed  Relu Activation with DropOut with 0.25    6. Conv2D - 128 Layers - Same Padding - with Kernal Size (3,3) - TimeDistributed and kernel\_regularizer(l2(0.01)) and bias\_regularizer(l2(0.01))  Relu Activation with DropOut with 0.50    7. TimeDistributed - Conv2D - 256 Layers - Same Padding - with Kernal Size (5,5) and kernel\_regularizer(l2(0.01)) and bias\_regularizer(l2(0.01))  Relu Activation  8. TimeDistributed – Flatten; Dense – 16; DropOut with 0.50  9. GRU with 64 Layers  10. Dense - 5 with softmax activation |
| **Final Model in Conv3D** | **Conv3D** | **Training Accuracy – 91.16**  **Validation Accuracy – 86.67**  Parameters – **185,685** | Batch Size : 30; Num of Epoch : 30  Frames : 10; Pixels : 100 X 100; Channels : 3  **Model Construction :**  1. Conv3D - 8 Layers - Same Padding - with Kernal Size (3,3,3) and activation of  Relu; MaxPooling2D - (2,2,2); Includes BatchNormalization    2. Conv3D - 16 Layers - Same Padding - with Kernal Size (3,3,3) and activation of  Relu; MaxPooling3D - (2,2,2); Includes BatchNormalization    3. Conv3D - 32 Layers - Same Padding - with Kernal Size (3,3,3) and activation of  Relu ; MaxPooling3D - (2,2,2); includes BatchNormalization; Added dropout of  0.1  5. Conv3D - 64 Layers - Same Padding - with Kernal Size (3,3,3) and activation of  Relu; MaxPooling3D - (2,2,1); Includes BatchNormalization; Added dropout of  0.1  5. Flattened the layer  6. Applied Dense layer with 128 neurons ; DropOut with 0.1  7. Applied Dense layer with 64 neurons ; DropOut with 0.1  8. Applied Dense layer with 32 neurons ; DropOut with 0.1  9. Applied Softmax activation with Dense - 5 |