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.

SGD is giving similar results but taking 4X long-time preferable optimizer is Adam

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| **1** | **Conv3D Batch size-10** | **Training Accuracy – 38.31**  **Validation Accuracy – 42.00**  **Worse than a random model**  **Trainable Parameters – 3,667,509**  **High number of Trainable parameters and low accuracy**  **Not plotted as very poor accuracy** | **Batch Size: 10,**  **Frames: 30,**  **Image Size: 120X120,**  **Channel: 3**  **Model:**  **1. Conv3D - 8 Layers**  **Padding - Same**  **Kernal Size (3,3,3)**  **Relu Activation**  **MaxPooling2D - (2,2,2)**  **Batch Normalization**  **2. Conv3D - 16 Layers**  **Padding- Same**  **Kernel Size (3,3,3)**  **Relu Activation**  **MaxPooling3D - (2,2,2)**  **Batch Normalization**  **3. Conv3D - 32 Layers**  **Same Padding - with**  **Kernel Size (3,3,3)**  **Relu Activation**  **MaxPooling3D - (2,2,2)**  **Batch Normalization**  **4. Conv3D - 64 Layers**  **Same Padding**  **Kernel Size (3,3,3)**  **Relu Activation**  **MaxPooling3D - (2,2,2)**  **Batch Normalization**  **Flatten**  **Dense - 1000**  **Dropout-0.5**  **Dense - 500**  **Dropout-0.5**  **Softmax - Dense 5**  **Worse than a random model omitted** |
| **2** | **Conv3D**  **Batch Size-30** | **Training Accuracy – 76.09**  **Validation Accuracy – 71.67**  **Parameters – 397,269**  **Lower parameters than CNN+Conv model** | **Batch: 30**  **Epoch: 30**  **Frames: 18**  **Image Size: 84 X 84**  **Channels: 3**  **Model:**  **1. Conv3D - 8 Layers**  **Same Padding**  **Kernel Size (3,3,3)**  **Elu Activation**  **MaxPooling2D - (2,2,2)**  **Batch Normalization**  **2. Conv3D - 16 Layers**  **Same Padding**  **Kernal Size (3,3,3)**  **Elu Activation**  **MaxPooling3D - (2,2,2)**  **Batch Normalization**  **3. Conv3D - 32 Layers**  **Same Padding**  **Kernel Size (3,3,3)**  **Elu Activation**  **MaxPooling3D - (2,2,2)**  **Batch Normalization**  **4. Conv3D - 32 Layers**  **Same Padding**  **Kernel Size (3,3,3)**  **Elu Activation**  **MaxPooling3D - (2,2,1)**  **Batch Normalization**  **5. Conv3D - 64 Layers**  **Padding**  **Kernel Size (3,3,3)**  **Elu Activation**  **MaxPooling3D - (2,2,1)**  **Batch Normalization**  **5. Conv3D - 128 Layers**  **Same Padding**  **Kernel Size (3,3,3)**  **Elu Activation**  **MaxPooling3D - (2,2,2)**  **Batch Normalization**  **DropOut - 0.25**  **Flatten**  **Dense - 256**  **DropOut with 0.50**  **Dense - 128**  **DropOut with 0.50**  **Dense - 64**  **DropOut with 0.50**  **Dense - 5 with softmax activation** |
| **3** | **Conv + RNN (GRU)** | **Training Accuracy – 88.70**  **Validation Accuracy – 70.00**  **Trainable Parameters –**  **1,158,645** | **Added Conv2D Models with Layers (32,32,32,64,64,64,128,256)**  **With Time Distributed**  **Input as:**  **Frames: 18**  **Image Size: 84 X 84**  **Channels: 3**  **Batch Size: 30**  **Epochs: 30**  **In Each Layers we have used Kernel and bias L2 regularizer of 0.01**  **Added Max Pooling and Batch Normalization and Dropout** |
| **4.** | **CONV + RNN (LSTM)** | **Training Accuracy – 86.97**  **Validation Accuracy – 78.33**  **Trainable Parameters-**  **1,163,829** | **Batch: 30**  **Epoch: 30**  **Frames: 15**  **Image Size: 84X84**  **Channels: 3**  **Model:**  **1. Time Distributed - Conv2D - 32 Layers - S**  **same Padding**  **Kernel Size (1,1)**  **Relu Activation**  **MaxPooling2D - (3,3)**  **Batch Normalization**  **2. TimeDistributed - Conv2D - 32 Layers**  **Same Padding**  **Kernel Size (3,3)**  **Relu Activation**  **MaxPooling2D - (3,3)**  **Batch Normalization**  **3. TimeDistributed - Conv2D - 64 Layers**  **Same Padding - with**  **Kernal Size (3,3)**  **Bias & kernel regularizer (l2(0.01))**  **Relu Activation**  **Batch Normalization**  **DropOut with 0.25**  **4. TimeDistributed - Conv2D - 64 Layers –**  **Same Padding - with**  **Kernel Size (3,3)**  **Bias & kernel regularizer(l2(0.01))**  **Relu Activation**  **MaxPooling2D - (3,3)**  **5. TimeDistributed - Conv2D - 64 Layers –**  **Same Padding**  **Kernal Size (3,3)**  **Relu Activation**  **DropOut with 0.25**  **6. TimeDistributed - Conv2D - 128 Layers**  **Same Padding**  **Kernal Size (3,3)**  **Bias & kernel regularizer(l2(0.01))**  **Relu Activation**  **DropOut with 0.50**  **7. TimeDistributed - Conv2D - 256 Layers –**  **Same Padding - with**  **Kernal Size (5,5) and**  **Bias & kernel regularizer(l2(0.01))**  **Relu Activation**  **TimeDistributed - Flatten**  **Dense - 16**  **DropOut with 0.50**  **LSTM with 64 Layers**  **Dense - 5 with softmax activation**  **Final Model: If computational constraint not present** |
| **5.** | **Conv3D with Relu function and less images than before** | **Training Accuracy – 78.99**  **Validation Accuracy 73.33**  **Trainable Parameters-**  **380,885** | **Batch: 30**  **Epoch: 30**  **Frames: 15**  **Image Size: 84 X 84**  **Channels: 3**  **Model:**  **1. Conv3D - 8 Layers**  **Same Padding**  **Kernel Size (3,3,3)**  **Relu Activation**  **MaxPooling2D - (2,2,2)**  **Batch Normalization**  **2. Conv3D - 16 Layers**  **Same Padding**  **Kernel Size (3,3,3)**  **Relu Activation**  **MaxPooling3D - (2,2,2)**  **Batch Normalization**  **3. Conv3D - 32 Layers**  **Same Padding**  **Kernel Size (3,3,3)**  **Relu Activation**  **MaxPooling3D - (2,2,2)**  **Batch Normalization**  **4. Conv3D - 32 Layers**  **Same Padding**  **Kernel Size (3,3,3)**  **Relu Activation**  **MaxPooling3D - (2,2,1)**  **Batch Normalization**  **5. Conv3D - 64 Layers**  **Padding**  **Kernel Size (3,3,3)**  **Relu Activation**  **MaxPooling3D - (2,2,1)**  **Batch Normalization**  **6. Conv3D - 128 Layers**  **Same Padding**  **Kernel Size (3,3,3)**  **Relu Activation**  **MaxPooling3D - (2,2,2)**  **Batch Normalization**  **DropOut - 0.25**  **Flatten**  **Dense - 256**  **DropOut with 0.50**  **Dense - 128**  **DropOut with 0.50**  **Dense - 64**  **DropOut with 0.50**  **Dense - 5 with softmax activation**  **Final Model:**  **If computational constraint present** |