**General Training:**

All models were trained for 20 epochs.

The ‘**Adam**’ optimiser was used. This was considered best as it computes individual learning rates for different parameters that are adapted based on the gradients.

Loss = Categorical Cross-Entropy

Metrics = Categorical Accuracy

**Model: Conv3D:**

**Model Working:** The filter moves in three directions (cubic filter), convolution happens in the same manner as conv2D. The conv3D model takes in 4-D tensor as input and features are extracted for each frame.

**Architecture:** Conv3D layer with 16 kernels of size (3,3,3) and ReLU activation function, followed by 3 (Conv3D + BatchNormalisation + MaxPooling3D (2,2,2)) blocks with 16, 16, 32 kernels (respectively) of size (3,3,3) and ReLU activation function, Conv3D layer with 32 kernels of size (3,3,3) and ReLU activation function, dropout layer (dropout percentage = 50%), flatten layer, fully connected layer with 64 neurons and final dense layer with 5 neurons and softmax activation function.

**Experiments:** Different values for number of frames and image size were considered to try and achieve maximum accuracy. No. of frames - i) 15 and ii) 30, image sizes - i) 128x128 and ii) 200x200. The findings for each are mentioned below:

**Model 1:** No. of frames = 15 and img size = 128x128 - When the model was trained with these values, training accuracy of 64.91% and validation accuracy of 40.62%;

training loss of 4.4165 and validation loss of 6.4411 was achieved.

There is some overfitting.

Note: An ablation experiment was first done for this model. After training for 1 epoch, a training accuracy of 26.14% and validation accuracy of 17.19% was achieved.

**Model 2:** No. of frames = 30 and img size = 128x128 - A logits size error occurred.

**Model 3:** No. of frames = 15 and img size = 200x200 - A resource exhausted error occurred.

**Model: CNN + RNN**

**Model Working:** The images of a video are passed through a CNN which extracts a feature vector for each image, and then the sequence of these feature vectors is passed through an RNN.

**Architecture:** All layers are time distributed (in CNN architecture), i.e., . Four (Conv2D layers with 16 kernels of size (3,3) and ReLU activation function + MaxPooling2D (2,2)) blocks, last block contains BatchNormalisation layer, flatten layer, LSTM layer with 128 LSTM cells and ReLU activation function, dropout layer with dropout percent = 25%, final fully connected layer with 5 neurons and softmax activation function.

**Experiments:** CNN architecture followed by LSTM, GRU, GRU (with batch Normalisation and increasing dropout), VGG16 (CNN) + LSTM

No. Of frames = 15, Image Size = 128x128

**Model 4:** CNN + LSTM - Training Accuracy: 70.41% , Validation Accuracy: 35.29%

Training Loss:0.7453 , Validation Loss: 1.5182. There is serious overfitting with this model.

**Model 5:** CNN + GRU - Experimented with CNN + GRU (different RNN type) as GRU is better for smaller datasets/sequences and we are using only 15 frames. GRU could give better accuracy. Training Accuracy: 80.40% , Validation Accuracy: 61.72%

Training Loss: 0.9122 , Validation Loss:1.1656.

As expected, CNN + GRU gives better much accuracy than CNN + LSTM. There is some overfitting with this model. Batch normalisation layers have been added before each max pooling layer and increasing dropout percentage have been done to reduce this.

**Model 6:** CNN + GRU with BatchNormalisation - This model was added as Model 5 has overfitting problem and batch normalisation and increased dropout percentage (50%) were added to try and fix this overfitting problem. Training Accuracy: 76.42% , Validation Accuracy: 25%

Training Loss:0.9091 , Validation Loss: 2.1825

Adding the batch normalisation layers only worsened the overfitting problem and the accuracy. More dropout layers could be added instead, batch normalisation could be added after max pooling layers.

**Model 7** (Transfer Learning): VGG16 + LSTM (Could not train as Google Colab limit used up) - Experiment with VGG16 + LSTM to try and achieve better accuracy (VGG16 could be better than CNN). VGG16 followed by LSTM layer with 128 LSTM cells, flatten layer, dropout layer with dropout = 25%, fully connected layer with 128 neurons and final dense layer with 5 neurons and softmax activation function.

**Model Comparison:**

|  |  |  |
| --- | --- | --- |
| **Experiment Number** | **Model** | **Result** |
| **1** | **Conv3D** | **Validation Accuracy:** 40.62% |
| **2** | **Conv2D + LSTM** | **Validation Accuracy:** 35.29% |
| **3** | Conv 2D + GRU  (best ) | **Validation Accuracy:** 61.72% |
| **4** | CNN + GRU with BatchNormalisation | **Validation Accuracy: 25%** |
| **5** | VGG16 + LSTM | **Validation Accuracy: -** |