Below are the Write-up with explanation for different models tried for this case study:

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| **1** | **Conv3D(16, (3,3,3))**  **Activation function – Relu**  **MaxPooling3D(pool\_size=(2,2,2))**  **Conv3D(32, (3,3,3))**  **Activation function – Relu**  **MaxPooling3D(pool\_size=(2,2,2))**  **Conv3D(64, (3,3,3))**  **Activation function – Relu**  **MaxPooling3D(pool\_size=(2,2,2))**  **Conv3D(64, (3,3,3))**  **Activation function – Relu**  **MaxPooling3D(pool\_size=(2,2,2))**  **X=18,y=120,z=120**  **Num\_epochs = 30**  **batch\_size = 10** | **Val\_loss- Nan** | **Change the activation function to ‘elu’** |
| **2** | **Conv3D(16, (3,3,3))**  **Activation function – elu**  **MaxPooling3D(pool\_size=(2,2,2))**  **Conv3D(32, (3,3,3))**  **Activation function – elu**  **MaxPooling3D(pool\_size=(2,2,2))**  **Conv3D(64, (3,3,3))**  **Activation function – elu**  **MaxPooling3D(pool\_size=(2,2,2))**  **Conv3D(64, (3,3,3))**  **Activation function – elu**  **MaxPooling3D(pool\_size=(2,2,2))**  **X=18,y=120,z=120**  **Num\_epochs = 30**  **batch\_size = 10** | **loss: 0.0217 - categorical\_accuracy: 1.0000**  **val\_loss: 1.0553 - val\_categorical\_accuracy: 0.7400** | **Overfitting as train and validation accuracy has big difference.**  **Updating batch size** |
| **3** | **Conv3D(16, (3,3,3))**  **Activation function – elu**  **MaxPooling3D(pool\_size=(2,2,2))**  **Conv3D(32, (3,3,3))**  **Activation function – elu**  **MaxPooling3D(pool\_size=(2,2,2))**  **Conv3D(64, (3,3,3))**  **Activation function – elu**  **MaxPooling3D(pool\_size=(2,2,2))**  **Conv3D(64, (3,3,3))**  **Activation function – elu**  **MaxPooling3D(pool\_size=(2,2,2))**  **X=18,y=120,z=120**  **Num\_epochs = 30**  **batch\_size = 35** | **loss: 0.3710 - categorical\_accuracy: 0.8944**  **val\_loss: 0.9937 - val\_categorical\_accuracy: 0.7100** | **Overfitting is still there as train and validation accuracy has some difference but it has improved from previous model when we updated the batch size to 35.**  **Adding Batch normalization layer to improve it further** |
| **4** | **Conv3D(16, (3,3,3))**  **BatchNormalization**  **Activation function – elu**  **MaxPooling3D(pool\_size=(2,2,2))**  **Conv3D(32, (3,3,3))**  **BatchNormalization**  **Activation function – elu**  **MaxPooling3D(pool\_size=(2,2,2))**  **Conv3D(64, (3,3,3))**  **BatchNormalization**  **Activation function – elu**  **MaxPooling3D(pool\_size=(2,2,2))**  **Conv3D(64, (3,3,3))**  **BatchNormalization**  **Activation function – elu**  **MaxPooling3D(pool\_size=(2,2,2))**  **X=18,y=120,z=120**  **Num\_epochs = 30**  **batch\_size = 35** | **loss: 0.1453 - categorical\_accuracy: 0.9894**  **val\_loss: 0.9461 - val\_categorical\_accuracy: 0.7100** | **Overfitting has increased after batch normalization as train and validation accuracy has big difference.**  **Adding dropouts** |
| **5** | **Conv3D(16, (3,3,3))**  **BatchNormalization**  **Activation function – elu**  **MaxPooling3D(pool\_size=(2,2,2))**  **Conv3D(32, (3,3,3))**  **BatchNormalization**  **Activation function – elu**  **MaxPooling3D(pool\_size=(2,2,2))**  **Dropout(0.25)**  **Conv3D(64, (3,3,3))**  **BatchNormalization**  **Activation function – elu**  **MaxPooling3D(pool\_size=(2,2,2))**  **Dropout(0.25)**  **Conv3D(64, (3,3,3))**  **BatchNormalization**  **Activation function – elu**  **MaxPooling3D(pool\_size=(2,2,2))**  **Flatten()**  **Dropout(0.5)**  **Dense(128, activation='elu')**  **Dropout(0.5)**  **Dense(5, activation='softmax')**  **X=18,y=120,z=120**  **Num\_epochs = 30**  **batch\_size = 35** | **loss: 1.2755 - categorical\_accuracy: 0.5656**  **val\_loss: 1.9862 - val\_categorical\_accuracy: 0.4100** | **Train and validation accuracy dropped after adding dropouts** |
| **6** | **Conv3D(32, (3,3,3))**  **BatchNormalization**  **Activation function – elu**  **MaxPooling3D(pool\_size=(2,2,2))**  **Conv3D(64, (3,3,3))**  **BatchNormalization**  **Activation function – elu**  **MaxPooling3D(pool\_size=(2,2,2))**  **Dropout(0.25)**  **Conv3D(128, (3,3,3))**  **BatchNormalization**  **Activation function – elu**  **MaxPooling3D(pool\_size=(2,2,2))**  **Dropout(0.25)**  **Conv3D(128, (3,3,3))**  **BatchNormalization**  **Activation function – elu**  **MaxPooling3D(pool\_size=(2,2,2))**  **Flatten()**  **Dropout(0.5)**  **Dense(256, activation='elu')**  **Dropout(0.5)**  **Dense(5, activation='softmax')**  **X=18,y=120,z=120**  **Num\_epochs = 30**  **batch\_size = 35** | **Getting Out of memory error** | **Reducing the batch size to 20** |
| **Final Model** | **Conv3D(32, (3,3,3))**  **BatchNormalization**  **Activation function – elu**  **MaxPooling3D(pool\_size=(2,2,2))**  **Conv3D(64, (3,3,3))**  **BatchNormalization**  **Activation function – elu**  **MaxPooling3D(pool\_size=(2,2,2))**  **Dropout(0.25)**  **Conv3D(128, (3,3,3))**  **BatchNormalization**  **Activation function – elu**  **MaxPooling3D(pool\_size=(2,2,2))**  **Dropout(0.25)**  **Conv3D(128, (3,3,3))**  **BatchNormalization**  **Activation function – elu**  **MaxPooling3D(pool\_size=(2,2,2))**  **Flatten()**  **Dropout(0.5)**  **Dense(256, activation='elu')**  **Dropout(0.5)**  **Dense(5, activation='softmax')**  **X=18,y=120,z=120**  **Num\_epochs = 30**  **batch\_size = 20** | **loss: 0.8529 - categorical\_accuracy: 0.6893**  **val\_loss: 1.1233 - val\_categorical\_accuracy: 0.6700** | **Overfitting issue has been resolved and this will be our final model. We have got 0.68 train accuracy and 0.67 validation accuracy for this model** |