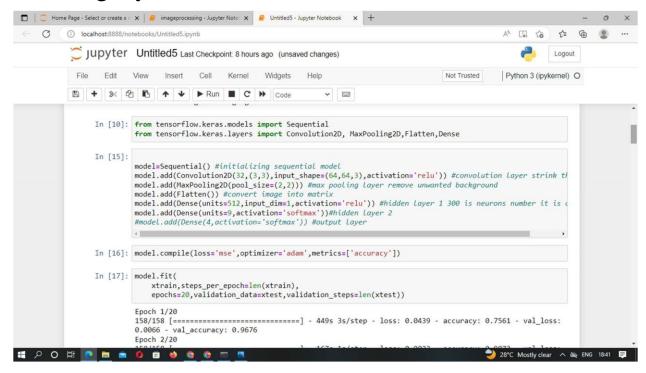| Team id	PNT2022TMID51226
Project Name	REAL TIME COMMUNICATION SYSTEM FOR SPECIALLY AIDED PEOPLES
Date and sprint no	11-11-2022 AND 2

MODEL BUILDING

1)Adding layers

2)Compile and fit the model

Jupyter Untitled5 Last Checkpoint: 8 hours ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel) ○

Code

```python
In [10]: from tensorflow.keras.models import Sequential
         from tensorflow.keras.layers import Convolution2D, MaxPooling2D,Flatten,Dense
```

```python
In [15]: model=Sequential() #initializing sequential model
         model.add(Convolution2D(32,(3,3),input_shape=(64,64,3),activation='relu')) #convolution layer strink th
         model.add(MaxPooling2D(pool_size=(2,2))) #max pooling layer remove unwanted background
         model.add(Flatten()) #convert image into matrix
         model.add(Dense(units=512,input_dim=1,activation='relu')) #hidden layer 1 300 is neurons number it is
         model.add(Dense(units=9,activation='softmax'))#hidden layer 2
         #model.add(Dense(4,activation='softmax')) #output layer
```

```python
In [16]: model.compile(loss='mse',optimizer='adam',metrics=['accuracy'])
```

```python
In [17]: model.fit(
             xtrain,steps_per_epoch=len(xtrain),
             epochs=20,validation_data=xtest,validation_steps=len(xtest))
```

```
Epoch 1/20
158/158 [==============================] - 449s 3s/step - loss: 0.0439 - accuracy: 0.7561 - val_loss:
0.0066 - val_accuracy: 0.9676
Epoch 2/20
```

28°C Mostly clear   ENG 18:41

Jupyter Untitled5 Last Checkpoint: 8 hours ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel) ○

Code

```
158/158 [==============================] - 224s 1s/step - loss: 2.0397e-04 - accuracy: 0.9990 - val_l
oss: 0.0064 - val_accuracy: 0.9689
Epoch 14/20
158/158 [==============================] - 222s 1s/step - loss: 1.5883e-04 - accuracy: 0.9991 - val_l
oss: 0.0050 - val_accuracy: 0.9769
Epoch 15/20
158/158 [==============================] - 217s 1s/step - loss: 2.2415e-04 - accuracy: 0.9988 - val_l
oss: 0.0050 - val_accuracy: 0.9773
Epoch 16/20
158/158 [==============================] - 262s 2s/step - loss: 1.7302e-04 - accuracy: 0.9989 - val_l
oss: 0.0050 - val_accuracy: 0.9769
Epoch 17/20
158/158 [==============================] - 230s 1s/step - loss: 2.5752e-04 - accuracy: 0.9984 - val_l
oss: 0.0050 - val_accuracy: 0.9773
Epoch 18/20
158/158 [==============================] - 176s 1s/step - loss: 1.5972e-04 - accuracy: 0.9990 - val_l
oss: 0.0050 - val_accuracy: 0.9764
Epoch 19/20
158/158 [==============================] - 172s 1s/step - loss: 2.2151e-04 - accuracy: 0.9987 - val_l
oss: 0.0050 - val_accuracy: 0.9773
Epoch 20/20
158/158 [==============================] - 167s 1s/step - loss: 1.2325e-04 - accuracy: 0.9992 - val_l
oss: 0.0049 - val_accuracy: 0.9778
```

```
Out[17]: <keras.callbacks.History at 0x1f743a90280>
```

28°C Mostly clear   ENG 18:41

```
Out[17]:  <keras.callbacks.History at 0x1f743a90280>
```

```
In [18]:  model.save('aslpng1.h5')
```

```
In [3]:  from keras.models import load_model
         import numpy as np
         import cv2
```

```
In [4]:  model=load_model('aslpng1.h5')
```

```
In [5]:  img1 = np.array([np.array([200, 200]), np.array([200, 200])])
         img2 = np.array([np.array([200, 200]), np.array([0, 0])])
         img3 = np.array([np.array([200, 0]), np.array([200, 0])])

         kernel_horizontal = np.array([np.array([2, 2]), np.array([-2, -2])])
         print(kernel_horizontal, 'is a kernel for detecting horizontal edges')

         kernel_vertical = np.array([np.array([2, -2]), np.array([2, -2])])
         print(kernel_vertical, 'is a kernel for detecting vertical edges')

         [[ 2  2]
          [-2 -2]] is a kernel for detecting horizontal edges
         [[ 2 -2]
          [ 2 -2]] is a kernel for detecting vertical edges
```