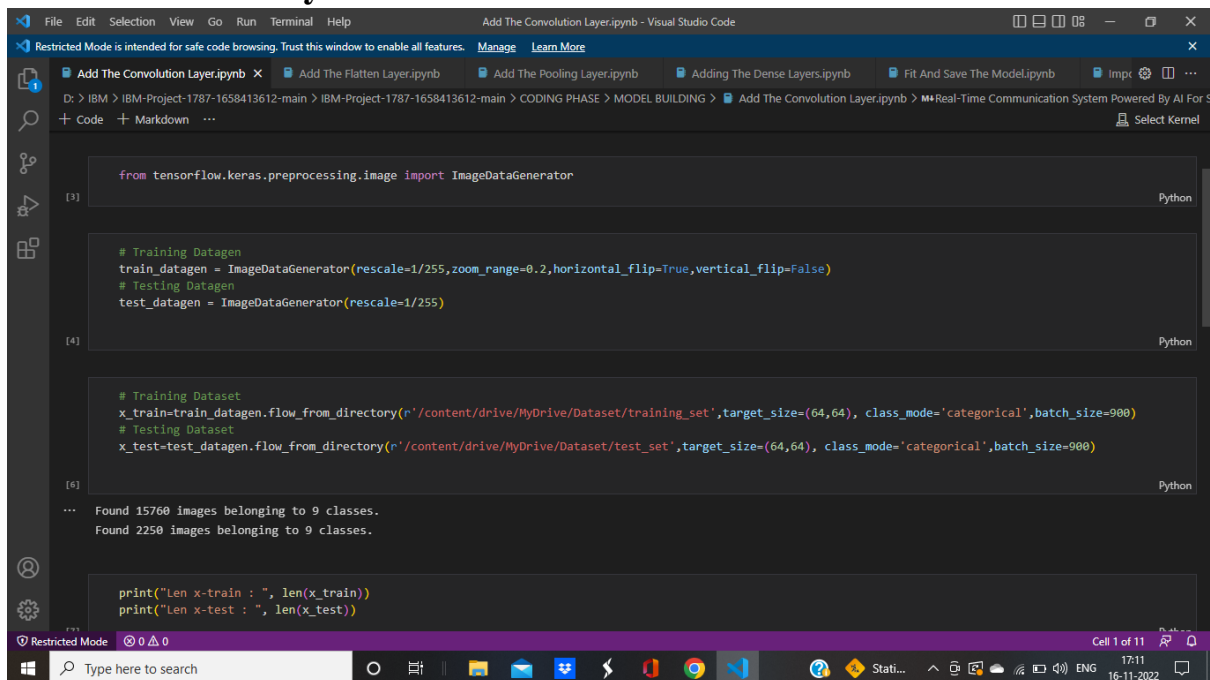


SPRINT -2

DESCRIPTION : As the next part of developing phase we have completed building and testing the model

MODEL BUILDING :

The convolution layer :



```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

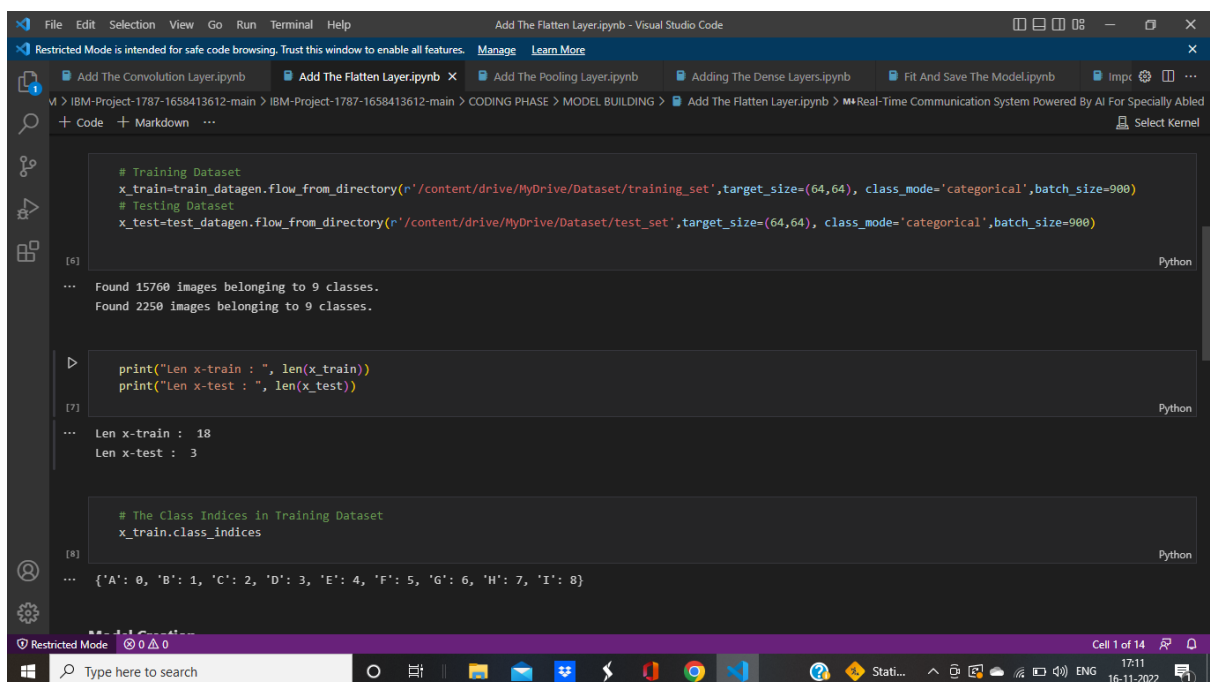
# Training Datagen
train_datagen = ImageDataGenerator(rescale=1/255, zoom_range=0.2, horizontal_flip=True, vertical_flip=False)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)

# Training Dataset
x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/training_set',target_size=(64,64), class_mode='categorical',batch_size=900)
# Testing Dataset
x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/test_set',target_size=(64,64), class_mode='categorical',batch_size=900)

... Found 15760 images belonging to 9 classes.
Found 2250 images belonging to 9 classes.

print("Len x-train : ", len(x_train))
print("Len x-test : ", len(x_test))
```

The flatten layer :



```
# Training Dataset
x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/training_set',target_size=(64,64), class_mode='categorical',batch_size=900)
# Testing Dataset
x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/test_set',target_size=(64,64), class_mode='categorical',batch_size=900)

... Found 15760 images belonging to 9 classes.
Found 2250 images belonging to 9 classes.

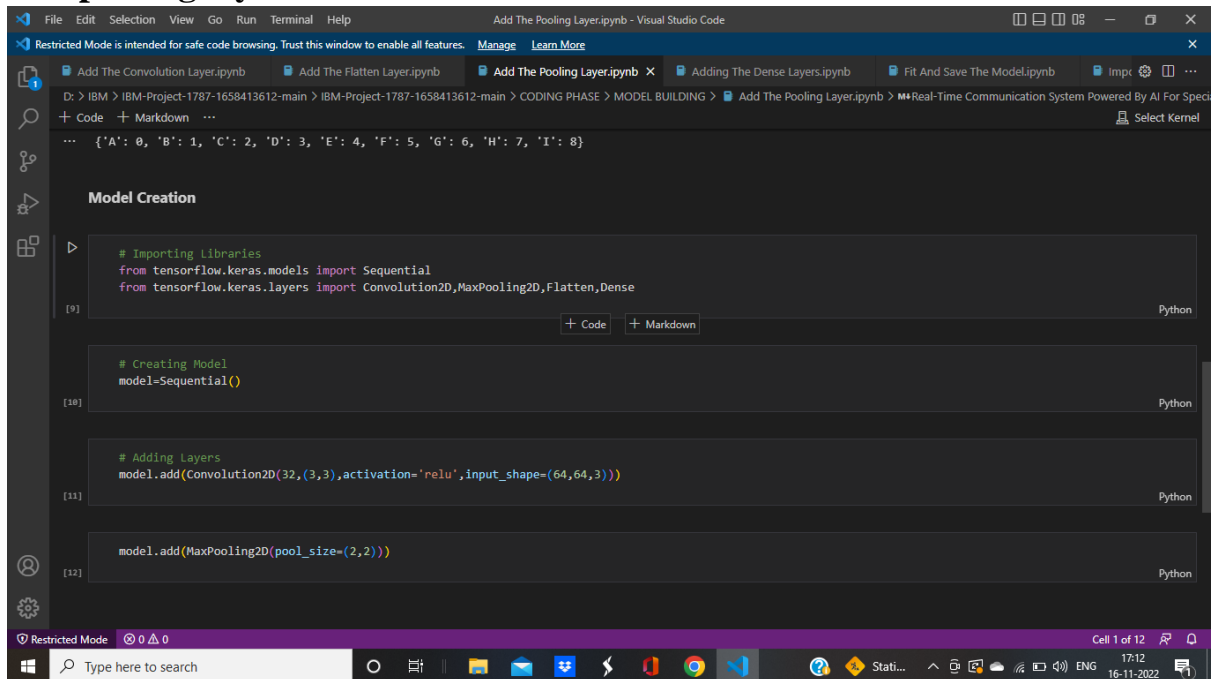
print("Len x-train : ", len(x_train))
print("Len x-test : ", len(x_test))

... Len x-train : 18
Len x-test : 3

# The Class Indices in Training Dataset
x_train.class_indices

... {'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8}
```

The pooling layer :



```
File Edit Selection View Go Run Terminal Help
Add The Pooling Layer.ipynb - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
D: > IBM > IBM-Project-1787-1658413612-main > IBM-Project-1787-1658413612-main > CODING PHASE > MODEL BUILDING > Add The Pooling Layer.ipynb > M4Real-Time Communication System Powered By AI For Spec
+ Code + Markdown ...
... {'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8}

Model Creation

# Importing Libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D, MaxPooling2D, Flatten, Dense

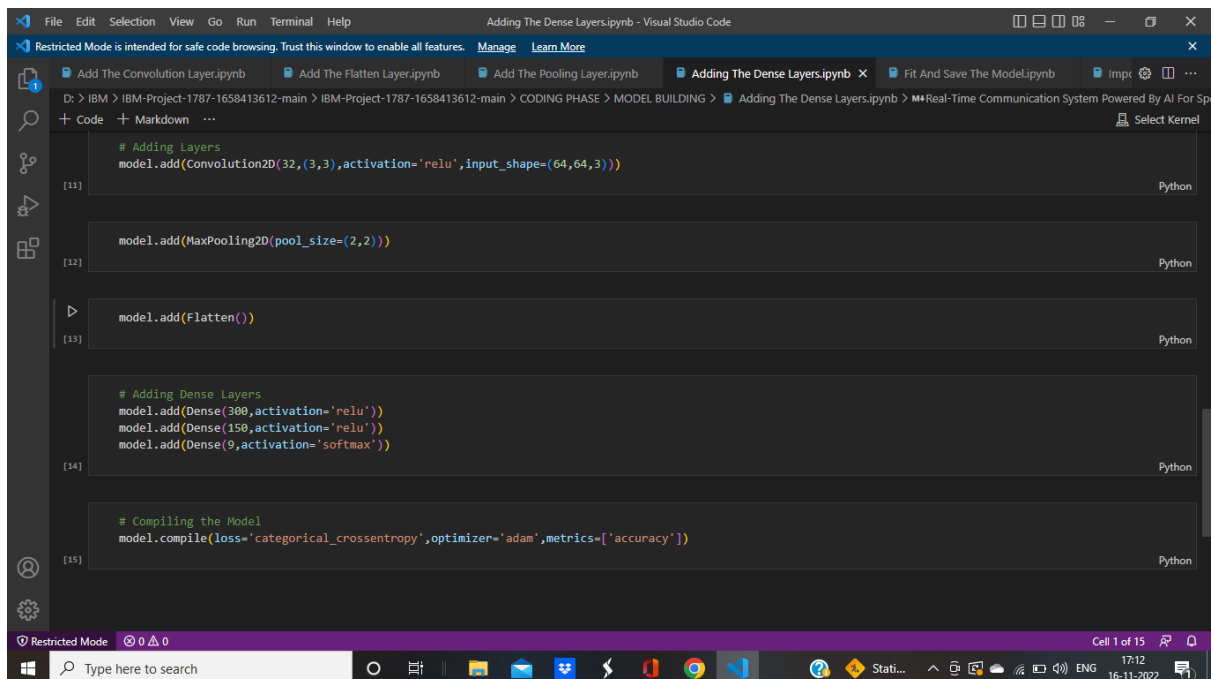
# Creating Model
model=Sequential()

# Adding Layers
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))

model.add(MaxPooling2D(pool_size=(2,2)))

Cell 1 of 12
17:12
16-11-2022
```

The dense layer :



```
File Edit Selection View Go Run Terminal Help
Adding The Dense Layers.ipynb - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
D: > IBM > IBM-Project-1787-1658413612-main > IBM-Project-1787-1658413612-main > CODING PHASE > MODEL BUILDING > Adding The Dense Layers.ipynb > M4Real-Time Communication System Powered By AI For Sp
+ Code + Markdown ...

# Adding Layers
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))

model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Flatten())

# Adding Dense Layers
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(9,activation='softmax'))

# Compiling the Model
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

Cell 1 of 15
17:12
16-11-2022
```

Fit and save the model :

```
# Fitting the Model Generator
model.fit_generator(x_train, steps_per_epoch=len(x_train), epochs=10, validation_data=x_test, validation_steps=len(x_test))
```

Epoch 1/10
18/18 [=====] - 92s 5s/step - loss: 0.0049 - accuracy: 0.9994 - val_loss: 0.2635 - val_accuracy: 0.9773
Epoch 2/10
18/18 [=====] - 90s 5s/step - loss: 0.0040 - accuracy: 0.9995 - val_loss: 0.2074 - val_accuracy: 0.9773
Epoch 3/10
18/18 [=====] - 87s 5s/step - loss: 0.0041 - accuracy: 0.9995 - val_loss: 0.2460 - val_accuracy: 0.9773
Epoch 4/10
18/18 [=====] - 91s 5s/step - loss: 0.0041 - accuracy: 0.9992 - val_loss: 0.2470 - val_accuracy: 0.9782
Epoch 5/10
18/18 [=====] - 88s 5s/step - loss: 0.0037 - accuracy: 0.9993 - val_loss: 0.2439 - val_accuracy: 0.9782
Epoch 6/10
18/18 [=====] - 88s 5s/step - loss: 0.0024 - accuracy: 0.9997 - val_loss: 0.2852 - val_accuracy: 0.9782
Epoch 7/10
18/18 [=====] - 91s 5s/step - loss: 0.0023 - accuracy: 0.9997 - val_loss: 0.2589 - val_accuracy: 0.9782
Epoch 8/10
18/18 [=====] - 93s 5s/step - loss: 0.0014 - accuracy: 1.0000 - val_loss: 0.2523 - val_accuracy: 0.9782
Epoch 9/10
18/18 [=====] - 92s 5s/step - loss: 0.0013 - accuracy: 0.9999 - val_loss: 0.2269 - val_accuracy: 0.9778
Epoch 10/10
18/18 [=====] - 91s 5s/step - loss: 0.0012 - accuracy: 0.9999 - val_loss: 0.2968 - val_accuracy: 0.9782

<keras.callbacks.History at 0x7fde26f54590>

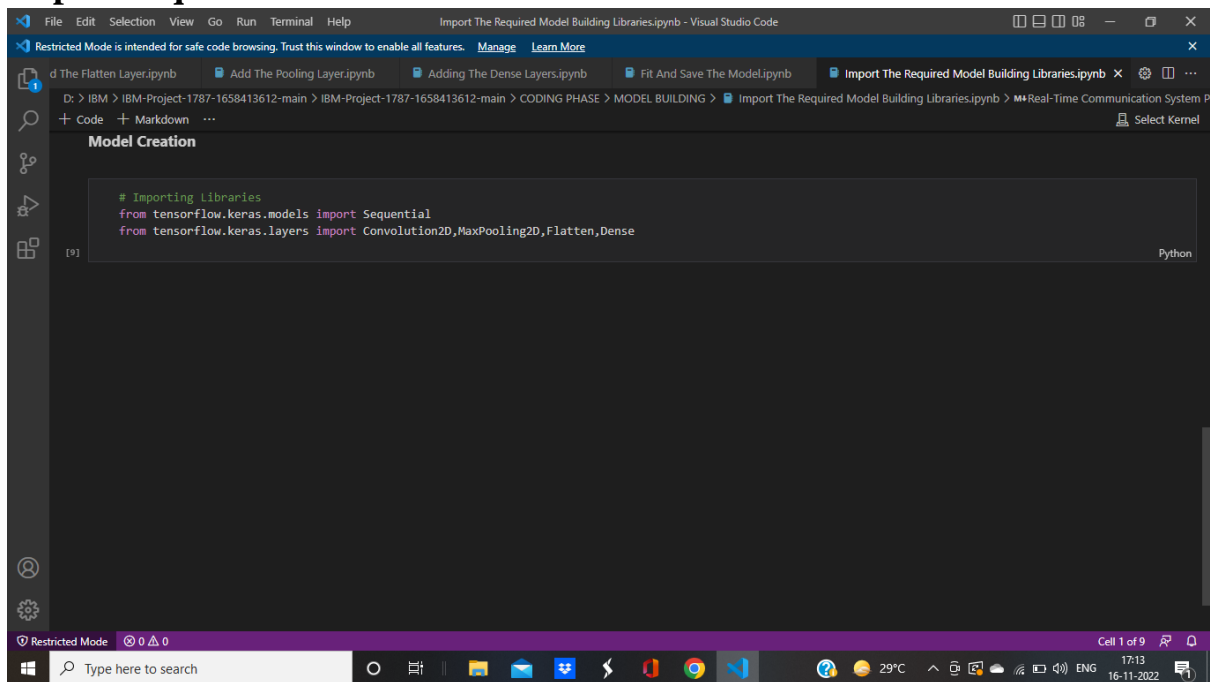
```
model.save("as1_model_04_54.h5")
```

Saving the Model

OneDrive
Screenshot saved
The screenshot was added to your OneDrive.

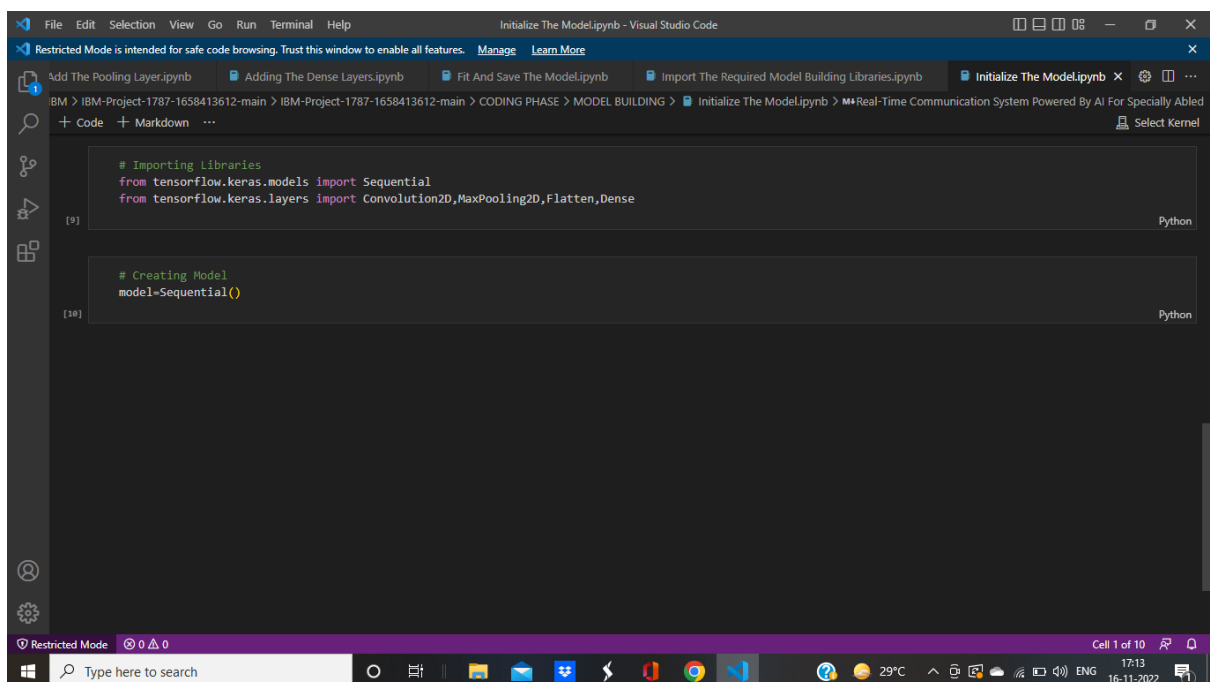
TEST THE MODEL :

Import required libraries :



```
# Importing Libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```

Initialize the model :



```
# Importing Libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense

# Creating Model
model=Sequential()
```