Team Id :PNT2022TMID46670

Model Building

Adding The Flatten Layer

In [ ]:

```
# importing numpy as np import numpy
as np
```

In [ ]:

```
# declare flatten np gfg = np.array([[6, 9, 12], [8,
5, 2], [18, 21, 24]])

# using array.flatten() method flat_gfg
= gfg.flatten(order='A') print(flat_gfg)
```

In [ ]:

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

In [ ]:

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

In [ ]:

```
# Training Dataset
x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/t
raining_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/tes
t_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.
```

In [ ]:

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

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

In [ ]:

```
# The Class Indices in Training Dataset x_train.class_indices
```

Out[ ]:

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

Model Creation

In [ ]:

```
model = Sequential() for i, feat in
enumerate(args.conv_f):     if i==0:
        model.add(Conv2D(feat, input_shape=x[0].shape, kernel_size=3,
padding = 'same',use_bias=False))     else:
        model.add(Conv2D(feat, kernel_size=3, padding =
'same',use_bias=False))         model.add(BatchNormalization())
model.add(LeakyReLU(alpha=args.conv_act))
model.add(Conv2D(feat, kernel_size=3, padding =
'same',use_bias=False))         model.add(BatchNormalization())
model.add(LeakyReLU(alpha=args.conv_act))
model.add(Dropout(args.conv_do[i]))
```

In [ ]:

```
model.add(Flatten())
#Input code here   denseArgs =
{'use_bias':False} for i,feat in
enumerate(args.dense_f):
model.add(Dense(feat,**denseArgs))
model.add(BatchNormalization())
model.add(LeakyReLU(alpha=args.dense_act))
model.add(Dropout(args.dense_do[i])) model.add(Dense(1))
```

In [ ]:

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

In [ ]:

```
# Creating Model model=Sequential()
```

In [ ]:

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

In [ ]:

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

In [ ]:

```
model.add(Flatten())
```

In [ ]:

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