

## Dataset

```
!unzip '/content/drive/MyDrive/archive (4).zip'
```

```
Archive: /content/drive/MyDrive/archive (4).zip
  inflating: test_data/test_data/blastl/DSC_6396.jpg
  inflating: test_data/test_data/blastl/DSC_6397.jpg
  inflating: test_data/test_data/blastl/DSC_6398.jpg
  inflating: test_data/test_data/blastl/DSC_6399.jpg
  inflating: test_data/test_data/blastl/DSC_6400.jpg
  inflating: test_data/test_data/blastl/DSC_6401.jpg
  inflating: test_data/test_data/blastl/DSC_6402.jpg
  inflating: test_data/test_data/blastl/DSC_6403.jpg
  inflating: test_data/test_data/blastl/DSC_6405.jpg
  inflating: test_data/test_data/blastl/DSC_6406.jpg
  inflating: test_data/test_data/blastl/DSC_6407.jpg
  inflating: test_data/test_data/blastl/DSC_6408.jpg
  inflating: test_data/test_data/blastl/DSC_6409.jpg
  inflating: test_data/test_data/blastl/DSC_6410.jpg
  inflating: test_data/test_data/blastl/DSC_6411.jpg
  inflating: test_data/test_data/bonegl/DSC_4587.jpg
  inflating: test_data/test_data/bonegl/DSC_4588.jpg
  inflating: test_data/test_data/bonegl/DSC_4589.jpg
  inflating: test_data/test_data/bonegl/DSC_4590.jpg
  inflating: test_data/test_data/bonegl/DSC_4591.jpg
  inflating: test_data/test_data/bonegl/DSC_4592.jpg
  inflating: test_data/test_data/bonegl/DSC_4593.jpg
  inflating: test_data/test_data/brhkyt/D72_0473.jpg
  inflating: test_data/test_data/brhkyt/D72_0474.jpg
  inflating: test_data/test_data/brhkyt/D72_0475.jpg
  inflating: test_data/test_data/brhkyt/D72_0477.jpg
  inflating: test_data/test_data/brhkyt/D72_0478.jpg
  inflating: test_data/test_data/brhkyt/D72_0479.jpg
  inflating: test_data/test_data/cbrtsh/_D32_10310.jpg
  inflating: test_data/test_data/cbrtsh/_D32_10311.jpg
  inflating: test_data/test_data/cbrtsh/_D32_10312.jpg
  inflating: test_data/test_data/cbrtsh/_D32_10313.jpg
  inflating: test_data/test_data/cbrtsh/_D32_10314.jpg
  inflating: test_data/test_data/cbrtsh/_D32_10317.jpg
  inflating: test_data/test_data/cbrtsh/_D32_10318.jpg
  inflating: test_data/test_data/cmnmyrn/DSC_2443.jpg
  inflating: test_data/test_data/cmnmyrn/DSC_4681.jpg
  inflating: test_data/test_data/cmnmyrn/DSC_5137.jpg
  inflating: test_data/test_data/cmnmyrn/DSC_7625.jpg
  inflating: test_data/test_data/cmnmyrn/P1050277.jpg
  inflating: test_data/test_data/cmnmyrn/_D32_12426.jpg
  inflating: test_data/test_data/cmnmyrn/_D32_12427.jpg
  inflating: test_data/test_data/cmnmyrn/_D32_12428.jpg
  inflating: test_data/test_data/gretit/11620454726_31a35c26da_o.jpg
  inflating: test_data/test_data/gretit/11776135285_ccf938fa2e_o.jpg
  inflating: test_data/test_data/gretit/11905645146_6a5d4ff9f9_o.jpg
  inflating: test_data/test_data/gretit/8537646712_0b282c4c6a_o.jpg
  inflating: test_data/test_data/gretit/D72_0693.jpg
  inflating: test_data/test_data/gretit/D72_0694.jpg
  inflating: test_data/test_data/gretit/D72_0695.jpg
  inflating: test_data/test_data/hilpig/DSC_6359.jpg
  inflating: test_data/test_data/hilpig/DSC_6362.jpg
  inflating: test_data/test_data/hilpig/DSC_6364.jpg
  inflating: test_data/test_data/hilpig/DSC_6368.jpg
  inflating: test_data/test_data/hilpig/DSC_6403.jpg
  inflating: test_data/test_data/hilpig/DSC_6404.jpg
  inflating: test_data/test_data/hilpig/P1000319.jpg
```

20BBS0217 VISHWAS SAPROO AI EXTERNSHIP ASSIGNMENT 3

```
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force\_remount=True).

## Data Augmentation

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

```

train_gen = ImageDataGenerator(rescale=1./255, horizontal_flip=True, shear_range = 0.2)
test_gen = ImageDataGenerator(rescale=1./255)

train = train_gen.flow_from_directory('/content/train_data/train_data',
                                     target_size=(224,224),
                                     class_mode = 'categorical',
                                     batch_size=8)

test = train_gen.flow_from_directory('/content/test_data/test_data',
                                    target_size=(224,224),
                                    class_mode = 'categorical',
                                    batch_size=8)

Found 150 images belonging to 16 classes.
Found 157 images belonging to 16 classes.

```

## ▼ CNN

```

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

model = Sequential()
model.add(Convolution2D(30, (3,3), 20, activation = 'relu', input_shape = (224,224,3)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(80, activation='relu'))
model.add(Dense(16, activation='softmax'))

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

model.fit(train, batch_size=8, validation_data = test, epochs=10)

Epoch 1/10
19/19 [=====] - 130s 7s/step - loss: 2.7577 - accuracy: 0.0933 - val_loss: 2.6848 - val_accuracy: 0.1465
Epoch 2/10
19/19 [=====] - 121s 7s/step - loss: 2.6480 - accuracy: 0.1667 - val_loss: 2.6321 - val_accuracy: 0.1401
Epoch 3/10
19/19 [=====] - 120s 7s/step - loss: 2.5455 - accuracy: 0.2000 - val_loss: 2.6060 - val_accuracy: 0.1656
Epoch 4/10
19/19 [=====] - 121s 7s/step - loss: 2.4714 - accuracy: 0.2933 - val_loss: 2.5620 - val_accuracy: 0.1592
Epoch 5/10
19/19 [=====] - 88s 5s/step - loss: 2.3712 - accuracy: 0.2933 - val_loss: 2.6019 - val_accuracy: 0.2102
Epoch 6/10
19/19 [=====] - 91s 5s/step - loss: 2.3123 - accuracy: 0.3000 - val_loss: 2.5381 - val_accuracy: 0.2357
Epoch 7/10
19/19 [=====] - 120s 7s/step - loss: 2.2334 - accuracy: 0.2867 - val_loss: 2.5214 - val_accuracy: 0.2102
Epoch 8/10
19/19 [=====] - 120s 7s/step - loss: 2.1525 - accuracy: 0.3533 - val_loss: 2.4988 - val_accuracy: 0.2548
Epoch 9/10
19/19 [=====] - 89s 5s/step - loss: 2.1047 - accuracy: 0.3267 - val_loss: 2.6255 - val_accuracy: 0.1592
Epoch 10/10
19/19 [=====] - 91s 5s/step - loss: 2.0745 - accuracy: 0.3467 - val_loss: 2.5463 - val_accuracy: 0.1720
<keras.callbacks.History at 0x7f73345e0310>

```

## ▼ VGG16 Model

```

from tensorflow.keras.layers import Dense, Flatten, Input
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np

from tensorflow.keras.applications.vgg16 import VGG16, preprocess_input

vgg = VGG16(include_top=False, weights='imagenet', input_shape=(224,224,3))

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16\_weights\_tf\_dim\_ordering\_tf\_kernels\_notop.58889256/58889256 [=====] - 2s 0us/step

```

```

for layer in vgg.layers:
    layer.trainable = False

x = Flatten()(vgg.output)

prediction = Dense(16,activation='softmax')(x)

model2 = Model(inputs=vgg.input,outputs=prediction)

model2.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

model2.fit_generator(train,validation_data=test,epochs=10,steps_per_epoch=len(train),validation_steps=len(test))

<ipython-input-20-93be31fd6a36>:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use
    model2.fit_generator(train,validation_data=test,epochs=10,steps_per_epoch=len(train),validation_steps=len(test))
Epoch 1/10
19/19 [=====] - 97s 5s/step - loss: 3.6755 - accuracy: 0.2533 - val_loss: 3.3230 - val_accuracy: 0.2866
Epoch 2/10
19/19 [=====] - 93s 5s/step - loss: 0.8783 - accuracy: 0.7800 - val_loss: 3.3977 - val_accuracy: 0.3185
Epoch 3/10
19/19 [=====] - 91s 5s/step - loss: 0.2841 - accuracy: 0.9267 - val_loss: 3.2138 - val_accuracy: 0.3312
Epoch 4/10
19/19 [=====] - 91s 5s/step - loss: 0.2410 - accuracy: 0.9333 - val_loss: 2.8432 - val_accuracy: 0.4331
Epoch 5/10
19/19 [=====] - 121s 7s/step - loss: 0.1539 - accuracy: 0.9467 - val_loss: 3.0529 - val_accuracy: 0.3694
Epoch 6/10
19/19 [=====] - 121s 7s/step - loss: 0.1044 - accuracy: 0.9733 - val_loss: 2.9716 - val_accuracy: 0.3758
Epoch 7/10
19/19 [=====] - 90s 5s/step - loss: 0.1110 - accuracy: 0.9600 - val_loss: 3.1626 - val_accuracy: 0.4140
Epoch 8/10
19/19 [=====] - 92s 5s/step - loss: 0.0937 - accuracy: 0.9667 - val_loss: 3.0451 - val_accuracy: 0.3694
Epoch 9/10
19/19 [=====] - 91s 5s/step - loss: 0.0615 - accuracy: 0.9800 - val_loss: 3.0596 - val_accuracy: 0.4204
Epoch 10/10
19/19 [=====] - 93s 5s/step - loss: 0.0234 - accuracy: 0.9933 - val_loss: 3.0704 - val_accuracy: 0.4268
<keras.callbacks.History at 0x7f73341fffd0>

```

## bold text##Testing and results

```
train.class_indices
```

```

{'blasti': 0,
 'bonegl': 1,
 'brhkyt': 2,
 'cbrtsh': 3,
 'cmnmyn': 4,
 'gretit': 5,
 'hilpig': 6,
 'himbul': 7,
 'himgri': 8,
 'hsparo': 9,
 'indvul': 10,
 'jglowl': 11,
 'lbicrw': 12,
 'mgprob': 13,
 'rebimg': 14,
 'wcrsrt': 15}

```

```

output = ['blasti',
 'bonegl',
 'brhkyt',
 'cbrtsh',
 'cmnmyn',
 'gretit',
 'hilpig',
 'himbul',
 'himgri',
 'hsparo',
 'indvul',
 'jglowl',
 'lbicrw',
 'mgprob',

```

```
'rebing',
'wcrsrt']

img = image.load_img("/content/abc.jpg",target_size=(224,224))
img = image.img_to_array(img)
img = np.expand_dims(img,axis=0)
pred1 = np.argmax(model.predict(img))
print(output[pred1])

1/1 [=====] - 0s 108ms/step
blasti

img1 = image.load_img("/content/abcd.jpg",target_size=(224,224))
img1 = image.img_to_array(img1)
img1 = np.expand_dims(img1,axis=0)
pred1_1 = np.argmax(model.predict(img1))
print(output[pred1_1])

1/1 [=====] - 0s 20ms/step
rebimg

img = image.load_img("/content/spa.jpg",target_size=(224,224))
img = image.img_to_array(img)
img = np.expand_dims(img,axis=0)
pred2 = np.argmax(model2.predict(img))
print(output[pred2])

1/1 [=====] - 1s 660ms/step
cmnmyn
```

