```
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).
```

```
!unzip  '/content/drive/MyDrive/Colab  Notebooks/birds_dataset.zip'
```

```
  inflating: train_data/train_data/himgri/12266086526_82cd337667_o.jpg
  inflating: train_data/train_data/himgri/IMG_5463.JPG
  inflating:  train_data/train_data/hsparo/100_4757.JPG
  inflating: train_data/train_data/hsparo/100_4758.JPG
  inflating: train_data/train_data/hsparo/100_5039.JPG
  inflating: train_data/train_data/hsparo/100_5040.JPG
  inflating: train_data/train_data/hsparo/100_5041.JPG
  inflating: train_data/train_data/hsparo/100_5048.JPG
  inflating: train_data/train_data/hsparo/100_5049.JPG
  inflating: train_data/train_data/hsparo/100_5050.JPG
  inflating: train_data/train_data/hsparo/100_5572.JPG
  inflating: train_data/train_data/indvul/DSC_0502.jpg
  inflating: train_data/train_data/indvul/DSC_0571e.jpg
  inflating: train_data/train_data/indvul/DSC_0572.jpg
  inflating: train_data/train_data/indvul/DSC_0576e.jpg
  inflating: train_data/train_data/indvul/DSC_0582.jpg
  inflating: train_data/train_data/indvul/DSC_0583e.jpg
  inflating: train_data/train_data/indvul/DSC_0584.jpg
  inflating: train_data/train_data/indvul/DSC_0616c.jpg
  inflating:  train_data/train_data/indvul/DSC_0617.jpg
  inflating:  train_data/train_data/jglowl/12152151476_7a1524aabb_o.jpg
  inflating: train_data/train_data/jglowl/DSC01335.jpg
  inflating:  train_data/train_data/jglowl/DSC01336.jpg
  inflating:  train_data/train_data/jglowl/_D32_10285.jpg
  inflating:  train_data/train_data/jglowl/_D32_10578.jpg
  inflating:  train_data/train_data/jglowl/_D32_10583.jpg
  inflating: train_data/train_data/lbicrw/100_4037.JPG
  inflating:  train_data/train_data/lbicrw/100_4912.JPG
  inflating:  train_data/train_data/lbicrw/100_4913.JPG
  inflating:  train_data/train_data/lbicrw/100_4914.JPG
  inflating:  train_data/train_data/lbicrw/100_4915.JPG
  inflating:  train_data/train_data/lbicrw/100_4916.JPG
  inflating:  train_data/train_data/mgprob/100_5587.JPG
  inflating:  train_data/train_data/mgprob/100_5588.JPG
  inflating:  train_data/train_data/mgprob/100_5589.JPG
  inflating:  train_data/train_data/mgprob/100_5590.JPG
  inflating:  train_data/train_data/mgprob/100_5592.JPG
  inflating:  train_data/train_data/mgprob/100_5762.JPG
  inflating:  train_data/train_data/rebimg/100_5744.JPG
  inflating:  train_data/train_data/rebimg/100_5745.JPG
  inflating:  train_data/train_data/rebimg/100_5746.JPG
  inflating:  train_data/train_data/rebimg/100_5748.JPG
  inflating:  train_data/train_data/rebimg/100_5749.JPG
  inflating:  train_data/train_data/rebimg/100_5750.JPG
  inflating:  train_data/train_data/rebimg/100_5751.JPG
  inflating:  train_data/train_data/rebimg/100_5752.JPG
  inflating:  train_data/train_data/rebimg/100_5754.JPG
  inflating:  train_data/train_data/rebimg/100_5755.JPG
  inflating:  train_data/train_data/wcrsrt/100_4452.JPG
  inflating:  train_data/train_data/wcrsrt/100_4453.JPG
  inflating:  train_data/train_data/wcrsrt/100_4454.JPG
  inflating:  train_data/train_data/wcrsrt/100_4455.JPG
  inflating:  train_data/train_data/wcrsrt/100_4456.JPG
  inflating:  train_data/train_data/wcrsrt/100_4457.JPG
  inflating:  train_data/train_data/wcrsrt/100_4458.JPG
  inflating:  train_data/train_data/wcrsrt/100_4459.JPG
  inflating:  train_data/train_data/wcrsrt/100_4460.JPG
  inflating:  train_data/train_data/wcrsrt/100_4461.JPG
```

```
import tensorflow as tf
tf.keras.backend.clear_session()
```

```
# Data Augmentation
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
from tensorflow.keras.models import Sequential
import cv2 as cv
```

```
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=(120, 120),
                                        class_mode='categorical',
                                        batch_size=8)
test  =  test_gen.flow_from_directory('/content/test data/test data',
                                      target_size=(120, 120),
                                        class_mode='categorical',
                                        batch_size=8)
```

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

```
print(train.class_indices)
print(test.class_indices)
```

```
{'blasti': 0, 'bonegl': 1, 'brhkyt': 2, 'cbrtsh': 3, 'cmnmyn': 4, 'gretit': 5, 'hilpig': 6, 'himbul': 7, 'himgri': 8, 'hsparo': 9, 'indvul': 1
{'blasti': 0, 'bonegl': 1, 'brhkyt': 2, 'cbrtsh': 3, 'cmnmyn': 4, 'gretit': 5, 'hilpig': 6, 'himbul': 7, 'himgri': 8, 'hsparo': 9, 'indvul': 1
```

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

```
model = Sequential([
    Convolution2D(20,(3,3),activation = 'relu',input_shape=(120,120,3)),
    MaxPooling2D(2,2),
    Flatten(),
    Dense(45,activation = 'relu'),
    Dense(16,activation = 'softmax')
])
```

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

```
model_fit = model.fit(train,epochs =20,validation_data = test ,batch_size=5)
```

```
Epoch 1/20
19/19 [==============================] - 96s 5s/step - loss: 4.1707 - accuracy: 0.0533 - val_loss: 2.7471 - val_accuracy: 0.0764
Epoch 2/20
19/19 [==============================] - 83s 5s/step - loss: 2.6931 - accuracy: 0.1467 - val_loss: 2.7058 - val_accuracy: 0.1083
Epoch 3/20
19/19 [==============================] - 83s 5s/step - loss: 2.6182 - accuracy: 0.1600 - val_loss: 2.6560 - val_accuracy: 0.1274
Epoch 4/20
19/19 [==============================] - 82s 4s/step - loss: 2.5492 - accuracy: 0.2467 - val_loss: 2.6966 - val_accuracy: 0.1720
Epoch 5/20
19/19 [==============================] - 82s 4s/step - loss: 2.4447 - accuracy: 0.3067 - val_loss: 2.6792 - val_accuracy: 0.2548
Epoch 6/20
19/19 [==============================] - 83s 5s/step - loss: 2.4244 - accuracy: 0.2867 - val_loss: 2.6652 - val_accuracy: 0.2038
Epoch 7/20
19/19 [==============================] - 82s 4s/step - loss: 2.2520 - accuracy: 0.3667 - val_loss: 3.0739 - val_accuracy: 0.1911
Epoch 8/20
19/19 [==============================] - 83s 5s/step - loss: 2.2176 - accuracy: 0.3400 - val_loss: 2.6261 - val_accuracy: 0.2293
Epoch 9/20
19/19 [==============================] - 82s 4s/step - loss: 2.0222 - accuracy: 0.3867 - val_loss: 2.6453 - val_accuracy: 0.2548
Epoch 10/20
19/19 [==============================] - 82s 4s/step - loss: 1.8719 - accuracy: 0.4067 - val_loss: 2.8523 - val_accuracy: 0.2229
Epoch 11/20
19/19 [==============================] - 83s 5s/step - loss: 1.7951 - accuracy: 0.4867 - val_loss: 2.6157 - val_accuracy: 0.2166
Epoch 12/20
19/19 [==============================] - 82s 4s/step - loss: 1.4791 - accuracy: 0.5133 - val_loss: 2.6350 - val_accuracy: 0.2102
Epoch 13/20
19/19 [==============================] - 82s 4s/step - loss: 1.3083 - accuracy: 0.5667 - val_loss: 2.6900 - val_accuracy: 0.2548
Epoch 14/20
19/19 [==============================] - 83s 4s/step - loss: 1.1463 - accuracy: 0.6400 - val_loss: 2.7943 - val_accuracy: 0.2484
Epoch 15/20
19/19 [==============================] - 82s 4s/step - loss: 1.0063 - accuracy: 0.6800 - val_loss: 2.9623 - val_accuracy: 0.2484
Epoch 16/20
19/19 [==============================] - 82s 4s/step - loss: 0.9316 - accuracy: 0.7200 - val_loss: 3.0134 - val_accuracy: 0.2611
Epoch 17/20
19/19 [==============================] - 82s 4s/step - loss: 0.7708 - accuracy: 0.7733 - val_loss: 2.9831 - val_accuracy: 0.2484
Epoch 18/20
19/19 [==============================] - 118s 6s/step - loss: 0.6621 - accuracy: 0.7867 - val_loss: 3.3987 - val_accuracy: 0.2548
Epoch 19/20
19/19 [==============================] - 83s 4s/step - loss: 0.5935 - accuracy: 0.8067 - val_loss: 3.3076 - val_accuracy: 0.2357
Epoch 20/20
19/19 [==============================] - 82s 4s/step - loss: 0.5132 - accuracy: 0.8333 - val_loss: 3.0896 - val_accuracy: 0.2484
```

```
model.save('birds.h5')
```

```
model_new = tf.keras.models.load_model('/content/birds.h5')
```

```
import numpy as np
from tensorflow.keras.preprocessing import image
```

```
output = ['rebimg','wcrsrt','jglowl','ibicrw','mgprob','hsparo',
         'indvul','himgri','himbul','gretit','hilpig','cbrtsh',
         'cmnmyn','bonegl','brhkyt','blasti']
print(output)
```

```
['rebimg', 'wcrsrt', 'jglowl', 'ibicrw', 'mgprob', 'hsparo', 'indvul', 'himgri', 'himbul', 'gretit', 'hilpig', 'cbrtsh', 'cmnmyn', 'bonegl', '
```

```
img1 = image.load_img("/content/train_data/train_data/mgprob/100_5590.JPG",target_size=(120,120))
img1 = image.img_to_array(img1)
img1 = np.expand_dims(img1,axis=0)
pred = np.argmax(model.predict(img1))
print(pred)
print(output[pred])
```

```
1/1 [==============================] - 0s 34ms/step
13
bonegl
```

```
img1 = image.load_img("/content/train_data/train_data/cmnmyn/100_5763.JPG",target_size=(120,120))
img1 = image.img_to_array(img1)
img1 = np.expand_dims(img1,axis=0)
pred = np.argmax(model.predict(img1))
print(pred)
print(output[pred])
```

```
1/1 [==============================] - 0s 20ms/step
4
mgprob
```

```
img1 = image.load_img("/content/train_data/train_data/gretit/100_5043.JPG",target_size=(120,120))
img1 = image.img_to_array(img1)
img1 = np.expand_dims(img1,axis=0)
pred = np.argmax(model.predict(img1))
print(pred)
print(output[pred])
```

```
1/1 [==============================] - 0s 18ms/step
5
hsparo
```

```
img1 = image.load_img("/content/train_data/train_data/himbul/100_5029.JPG",target_size=(120,120))
img1 = image.img_to_array(img1)
img1 = np.expand_dims(img1,axis=0)
pred = np.argmax(model.predict(img1))
print(pred)
print(output[pred])
```

```
1/1 [==============================] - 0s 19ms/step
11
cbrtsh
```