Smart Bridge Assignment - 3

Name: Kuchibhatla Mohan Datta

Reg_No : 20MID0012

Build a CNN model for Bird species Bird species classification is the process of using machine learning and computer vision techniques to identify and categorize different species of birds based on their visual characteristics. By analyzing images of birds, models can extract features and patterns to accurately classify bird species. This classification is vital for ecological research, wildlife monitoring, and conservation efforts. Advancements in deep learning and the availability of large annotated datasets have improved the accuracy of bird species classification models. Challenges include variations in lighting, pose, and background clutter. Ongoing research focuses on methods like transfer learning and data augmentation to enhance classification performance and contribute to avian biodiversity understanding and conservation.

Dataset Link: https://www.kaggle.com/datasets/akash2907/bird-species-classification

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

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

```
unzip:  cannot find or open /content/drive/MyDrive/Colab Notebooks/birds_dataset.zip, /content/drive/MyDrive/Colab Notebooks/birds_
```

```
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/drive/MyDrive/BIRDS/train_data/train_data',target_size=(120, 120),class_mode='categorical
test = test_gen.flow_from_directory('/content/drive/MyDrive/BIRDS/test_data/test_data',target_size=(120, 120),class_mode='categorical',ba
```

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

```
# 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
```

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

```
model_fit = model.fit(train,epochs = 49,validation_data=test, batch_size=10)
```

```
Epoch 1/49
19/19 [==============================] - 128s 7s/step - loss: 1.7659 - accuracy: 0.4867 - val_loss: 2.7011 - val_accuracy: 0.1529
Epoch 2/49
```

```
19/19 [==============================] - 129s 7s/step - loss: 1.6338 - accuracy: 0.5333 - val_loss: 2.8207 - val_accuracy: 0.1338
Epoch 3/49
19/19 [==============================] - 130s 7s/step - loss: 1.4189 - accuracy: 0.5933 - val_loss: 2.7950 - val_accuracy: 0.1465
Epoch 4/49
19/19 [==============================] - 109s 6s/step - loss: 1.1738 - accuracy: 0.6867 - val_loss: 2.9514 - val_accuracy: 0.1529
Epoch 5/49
19/19 [==============================] - 109s 6s/step - loss: 1.1368 - accuracy: 0.6800 - val_loss: 2.9679 - val_accuracy: 0.1529
Epoch 6/49
19/19 [==============================] - 128s 7s/step - loss: 1.0236 - accuracy: 0.7533 - val_loss: 2.8528 - val_accuracy: 0.1401
Epoch 7/49
19/19 [==============================] - 110s 6s/step - loss: 0.8899 - accuracy: 0.7333 - val_loss: 2.9291 - val_accuracy: 0.2229
Epoch 8/49
19/19 [==============================] - 107s 6s/step - loss: 0.8204 - accuracy: 0.7800 - val_loss: 2.8495 - val_accuracy: 0.1847
Epoch 9/49
19/19 [==============================] - 127s 7s/step - loss: 0.6876 - accuracy: 0.8067 - val_loss: 3.2249 - val_accuracy: 0.1592
Epoch 10/49
19/19 [==============================] - 129s 7s/step - loss: 0.6471 - accuracy: 0.8000 - val_loss: 3.2429 - val_accuracy: 0.2293
Epoch 11/49
19/19 [==============================] - 128s 7s/step - loss: 0.5130 - accuracy: 0.8733 - val_loss: 3.1915 - val_accuracy: 0.2548
Epoch 12/49
19/19 [==============================] - 128s 7s/step - loss: 0.4683 - accuracy: 0.9000 - val_loss: 3.4248 - val_accuracy: 0.1911
Epoch 13/49
19/19 [==============================] - 128s 7s/step - loss: 0.4194 - accuracy: 0.9133 - val_loss: 3.4828 - val_accuracy: 0.2611
Epoch 14/49
19/19 [==============================] - 108s 6s/step - loss: 0.3926 - accuracy: 0.9200 - val_loss: 3.2596 - val_accuracy: 0.2611
Epoch 15/49
19/19 [==============================] - 128s 7s/step - loss: 0.3609 - accuracy: 0.9400 - val_loss: 3.5108 - val_accuracy: 0.2675
Epoch 16/49
19/19 [==============================] - 128s 7s/step - loss: 0.3104 - accuracy: 0.9533 - val_loss: 3.5579 - val_accuracy: 0.2548
Epoch 17/49
19/19 [==============================] - 128s 7s/step - loss: 0.2926 - accuracy: 0.9533 - val_loss: 3.5536 - val_accuracy: 0.2357
Epoch 18/49
19/19 [==============================] - 129s 7s/step - loss: 0.2523 - accuracy: 0.9533 - val_loss: 3.4559 - val_accuracy: 0.2548
Epoch 19/49
19/19 [==============================] - 111s 6s/step - loss: 0.2135 - accuracy: 0.9667 - val_loss: 3.5900 - val_accuracy: 0.2548
Epoch 20/49
19/19 [==============================] - 129s 7s/step - loss: 0.1894 - accuracy: 0.9733 - val_loss: 3.5701 - val_accuracy: 0.2548
Epoch 21/49
19/19 [==============================] - 108s 6s/step - loss: 0.1619 - accuracy: 0.9867 - val_loss: 3.7074 - val_accuracy: 0.2803
Epoch 22/49
19/19 [==============================] - 129s 7s/step - loss: 0.1534 - accuracy: 0.9933 - val_loss: 3.6128 - val_accuracy: 0.2739
Epoch 23/49
19/19 [==============================] - 108s 6s/step - loss: 0.1263 - accuracy: 0.9933 - val_loss: 3.6510 - val_accuracy: 0.3121
Epoch 24/49
19/19 [==============================] - 107s 6s/step - loss: 0.1405 - accuracy: 0.9733 - val_loss: 4.1337 - val_accuracy: 0.2548
Epoch 25/49
19/19 [==============================] - 110s 6s/step - loss: 0.1223 - accuracy: 0.9933 - val_loss: 3.6016 - val_accuracy: 0.2739
Epoch 26/49
19/19 [==============================] - 129s 7s/step - loss: 0.1148 - accuracy: 0.9867 - val_loss: 3.9174 - val_accuracy: 0.2930
Epoch 27/49
19/19 [==============================] - 108s 6s/step - loss: 0.0861 - accuracy: 0.9933 - val_loss: 3.8961 - val_accuracy: 0.2803
Epoch 28/49
19/19 [==============================] - 128s 7s/step - loss: 0.0778 - accuracy: 1.0000 - val_loss: 3.9142 - val_accuracy: 0.2930
Epoch 29/49
```

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

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

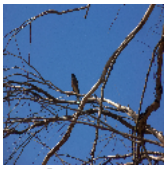
```
import numpy as np
from tensorflow.keras.preprocessing import image
```

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

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

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

```
img2 = image.load_img("/content/drive/MyDrive/BIRDS/train_data/train_data/cmnmyn/100_5763.JPG",target_size=(120,120))
display(img2)
img2 = image.img_to_array(img2)
img2 = np.expand_dims(img2,axis=0)
pred = np.argmax(model.predict(img2))
print(pred)
print(output[pred])
```



```
1/1 [==============================] - 0s 49ms/step
4
mgprob
```

```
img3 = image.load_img("/content/drive/MyDrive/BIRDS/train_data/train_data/gretit/100_5043.JPG",target_size=(120,120))
display(img3)
img3 = image.img_to_array(img3)
img3 = np.expand_dims(img3,axis=0)
pred = np.argmax(model.predict(img3))
print(pred)
print(output[pred])
```

```
1/1 [==============================] - 0s 37ms/step
5
hsparo
```

```
img4 = image.load_img("/content/drive/MyDrive/BIRDS/train_data/train_data/himbul/100_5029.JPG",target_size=(120,120))
display(img4)
img4 = image.img_to_array(img4)
img4 = np.expand_dims(img4,axis=0)
pred = np.argmax(model.predict(img4))
print(pred)
print(output[pred])
```

```
1/1 [==============================] - 0s 26ms/step
7
himgri
```

```
img5 = image.load_img("/content/drive/MyDrive/BIRDS/train_data/train_data/brhkyt/D72_0401.jpg",target_size=(120,120))
display(img5)
img5 = image.img_to_array(img5)
img5 = np.expand_dims(img5,axis=0)
pred = np.argmax(model.predict(img5))
print(pred)
print(output[pred])
```

```
1/1 [==============================] - 0s 30ms/step
2
jglowl
```

✓ 0s completed at 20:37