

```

import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import tensorflow as tf
import keras
from keras.models import Sequential
from keras.layers import Dense, Conv2D, MaxPool2D , Flatten
from keras.preprocessing.image import ImageDataGenerator

from keras.applications.vgg16 import VGG16
from keras.models import Model
from keras.optimizers import Adam
from keras.callbacks import ModelCheckpoint, EarlyStopping
from PIL import Image
import matplotlib.pyplot as plt

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).

```

trdata = ImageDataGenerator()
traindata = trdata.flow_from_directory(directory=r"/content/drive/MyDrive/dataset/train",
tsdata = ImageDataGenerator()
testdata = tsdata.flow_from_directory(directory=r"/content/drive/MyDrive/dataset/test")

```

```

tf.keras.applications.vgg16.VGG16(
    include_top=False,
    weights='imagenet',
    input_tensor=None,
    input_shape=None,
    pooling='avg',
    classes=2,
    classifier_activation='softmax'
)

```

```

model = Sequential()
model.add(tf.keras.applications.vgg16.VGG16(
    include_top=False,
    weights='imagenet',
    input_tensor=None,
    input_shape=None,
    pooling='avg',
    classes=2,
    classifier_activation='softmax'
))
model.add(Flatten())
model.add(Dense(units=4096, activation="relu"))
model.add(Dense(units=4096, activation="relu"))

```

! 0s completed at 11:33 PM



```
opt = Adam(lr=0.0001)
model.compile(optimizer=opt, loss=keras.losses.categorical_crossentropy, metrics=

for layer in model.layers[0].layers:
    layer.trainable = False

model.summary()
```

```
Found 840 images belonging to 2 classes.
Found 188 images belonging to 2 classes.
Downloading data from https://storage.googleapis.com/tensorflow/keras-applic
58889256/58889256 [=====] - 0s 0us/step
Model: "sequential"
```

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 512)	14714688
flatten (Flatten)	(None, 512)	0
dense (Dense)	(None, 4096)	2101248
dense_1 (Dense)	(None, 4096)	16781312
dense_2 (Dense)	(None, 2)	8194

```
=====
Total params: 33,605,442
Trainable params: 18,890,754
Non-trainable params: 14,714,688
```

```
/usr/local/lib/python3.8/dist-packages/keras/optimizers/optimizer_v2/adam.py
super(Adam, self).__init__(name, **kwargs)
```

```
import cv2
from google.colab.patches import cv2_imshow

file_name = '/content/drive/MyDrive/dataset/train/African/af_tr248.jpg'

image = cv2.imread(file_name)
cv2_imshow(image)
```





```
hist = model.fit(traindata, validation_data = testdata, epochs = 30, batch_size
```

```
Epoch 1/30
27/27 [=====] - 215s 7s/step - loss: 1.0368 - acc: 0.0000
Epoch 2/30
27/27 [=====] - 15s 544ms/step - loss: 0.6970 - acc: 0.0000
Epoch 3/30
27/27 [=====] - 15s 541ms/step - loss: 0.6932 - acc: 0.0000
Epoch 4/30
27/27 [=====] - 15s 547ms/step - loss: 0.7029 - acc: 0.0000
Epoch 5/30
27/27 [=====] - 15s 555ms/step - loss: 0.6935 - acc: 0.0000
Epoch 6/30
27/27 [=====] - 15s 552ms/step - loss: 0.6933 - acc: 0.0000
Epoch 7/30
27/27 [=====] - 15s 556ms/step - loss: 0.6930 - acc: 0.0000
Epoch 8/30
27/27 [=====] - 15s 561ms/step - loss: 0.6928 - acc: 0.0000
Epoch 9/30
27/27 [=====] - 15s 560ms/step - loss: 0.6925 - acc: 0.0000
Epoch 10/30
27/27 [=====] - 15s 565ms/step - loss: 0.6922 - acc: 0.0000
Epoch 11/30
27/27 [=====] - 15s 563ms/step - loss: 0.6923 - acc: 0.0000
Epoch 12/30
27/27 [=====] - 15s 563ms/step - loss: 0.6920 - acc: 0.0000
Epoch 13/30
27/27 [=====] - 15s 562ms/step - loss: 0.6917 - acc: 0.0000
Epoch 14/30
27/27 [=====] - 15s 561ms/step - loss: 0.6916 - acc: 0.0000
Epoch 15/30
27/27 [=====] - 15s 562ms/step - loss: 0.6914 - acc: 0.0000
Epoch 16/30
27/27 [=====] - 16s 619ms/step - loss: 0.6915 - acc: 0.0000
Epoch 17/30
27/27 [=====] - 15s 563ms/step - loss: 0.6963 - acc: 0.0000
Epoch 18/30
27/27 [=====] - 15s 565ms/step - loss: 0.6926 - acc: 0.0000
Epoch 19/30
27/27 [=====] - 15s 574ms/step - loss: 0.6924 - acc: 0.0000
Epoch 20/30
27/27 [=====] - 15s 564ms/step - loss: 0.6923 - acc: 0.0000
Epoch 21/30
27/27 [=====] - 15s 562ms/step - loss: 0.6923 - acc: 0.0000
Epoch 22/30
27/27 [=====] - 15s 560ms/step - loss: 0.6922 - acc: 0.0000
Epoch 23/30
27/27 [=====] - 15s 566ms/step - loss: 0.6921 - acc: 0.0000
Epoch 24/30
27/27 [=====] - 15s 562ms/step - loss: 0.6923 - acc: 0.0000
```

```

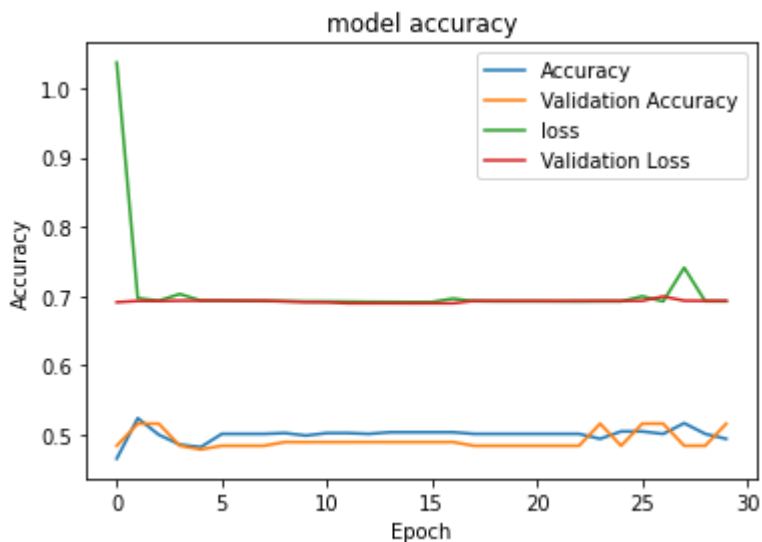
Epoch 25/30
27/27 [=====] - 15s 562ms/step - loss: 0.6923 - ac
Epoch 26/30
27/27 [=====] - 15s 566ms/step - loss: 0.6997 - ac
Epoch 27/30
27/27 [=====] - 15s 563ms/step - loss: 0.6923 - ac
Epoch 28/30
27/27 [=====] - 15s 573ms/step - loss: 0.7410 - ac
Epoch 29/30
27/27 [=====] - 15s 563ms/step - loss: 0.6925 - ac

```

```

plt.plot(hist.history["accuracy"])
plt.plot(hist.history['val_accuracy'])
plt.plot(hist.history['loss'])
plt.plot(hist.history['val_loss'])
plt.title("model accuracy")
plt.ylabel("Accuracy")
plt.xlabel("Epoch")
plt.legend(["Accuracy", "Validation Accuracy", "loss", "Validation Loss"])
plt.show()

```



```
model.save('/content/drive/MyDrive/model/elephant_model.h5')
```

```
model.save_weights('/content/drive/MyDrive/model/elephant_weights.h5')
```

```
width, height = 224, 224
```

```
model_p = '/content/drive/MyDrive/model/elephant_model.h5'
```

```
weight_p = '/content/drive/MyDrive/model/elephant_weights.h5'
```

```
model = tf.keras.models.load_model(model_p)
```

```
model.load_weights(weight_p)
```

WARNING:tensorflow:Error in loading the saved optimizer state. As a result

```
def tpredict(file):
    if not os.path.isfile(file):
        return "Error: file does not exist."
    try:
        test_image = tf.keras.utils.load_img(file, target_size=(width,height))
        test_image = tf.keras.utils.img_to_array(test_image)
        test_image = np.expand_dims(test_image, axis=0)
    except Exception as e:
        return f"Error: {e}"
    if not 'model' in globals():
        return "Error: model is not defined."
    try:
        array = model.predict(test_image)
        result = array[0]
    except Exception as e:
        return f"Error: {e}"
    if result[0] >= 0.5 :
        an = 'African Elephant'
    elif result[1]>= 0.5:
        an = 'Asian Elephant'
    else:
        an = "Not sure"
    return an
```

```
fpredict('/content/drive/MyDrive/elephsnt.jpg')
```

```
1/1 [=====] - 0s 16ms/step
'African Elephant'
```

[Colab paid products](#) - [Cancel contracts here](#)