```
!pip install -q kaggle
```

```
from google.colab import files
files.upload()
```

Choose Files No file chosen

```
{}
```

```
# create a kaggle folder
! mkdir ~/.kaggle
```

```
mkdir: cannot create directory ‘/root/.kaggle’: File exists
```

```
# copy the kaggle.json to folder created
! cp kaggle.json ~/.kaggle/
```

```
# Permission for the json to act
! chmod 600 ~/.kaggle/kaggle.json
```

```
# to list all dataset in kaggle
! kaggle datasets list
```

```
ref                                                       title                                              size
--------------------------------------------------------  ------------------------------------------------  -----
arnabchaki/data-science-salaries-2023                     Data Science Salaries 2023 💸                      25KB
chitrakumari25/corona-virus-latest-data-2023              Corona virus latest data 2023                      10KB
utkarshx27/starbucks-nutrition                            Starbucks Nutrition Facts                           2KB
utkarshx27/monthly-transportation-statistics              Monthly Transportation Statistics                 131KB
iammustafatz/diabetes-prediction-dataset                  Diabetes prediction dataset                       734KB
desalegngeb/students-exam-scores                          Students Exam Scores: Extended Dataset            695KB
utkarshx27/smoking-dataset-from-uk                        Smoking Dataset from UK                            17KB
mykhailozub/500-hotels-from-airbnb-booking-and-hotelscom  7500 hotels from Airbnb, Booking and Hotels.com   428KB
asahu40/walmart-data-analysis-and-forcasting              Walmart Data Analysis and Forcasting              122KB
utkarshx27/real-estate-sales-2001-2020-gl                 Real Estate Sales 2001-2020 GL                     33MB
sougatapramanick/happiness-index-2018-2019                Happiness Index 2018-2019                          15KB
olcaybolat1/dermatology-dataset-classification            Dermatology Dataset (Multi-class classification)    5KB
utkarshx27/suicide-attempts-in-shandong-china             Suicide Attempts in Shandong, China                29KB
r1shabhgupta/best-movies-on-netflix                       Top 100 Movies on Netflix                          32KB
utkarshx27/breast-cancer-dataset-used-royston-and-altman  Breast Cancer Dataset                              10KB
salvatorerastelli/spotify-and-youtube                     Spotify and Youtube                                 9MB
mohamedafsal007/house-price-dataset-of-india              House Price dataset of India                      480KB
ahmedshahriarsakib/usa-real-estate-dataset                USA Real Estate Dataset                             2MB
utkarshx27/arrests-for-marijuana-possession               Arrests for Marijuana Possession                   32KB
muhammadjawwadismail/hr-competency-scores-for-screening   HR Competency Scores for Screening                 13KB
```

```
! kaggle datasets download -d iamsouravbanerjee/animal-image-dataset-90-different-animals
```

```
Downloading animal-image-dataset-90-different-animals.zip to /content
 99% 647M/656M [00:05<00:00, 93.9MB/s]
100% 656M/656M [00:06<00:00, 107MB/s]
```

```
!unzip /content/animal-image-dataset-90-different-animals.zip
```

```
        inflating: animals/animals/zebra/133f1c1e4a.jpg
        inflating: animals/animals/zebra/14fa6cfe2b.jpg
        inflating: animals/animals/zebra/170fd0528c.jpg
        inflating: animals/animals/zebra/19e238a4fc.jpg
        inflating: animals/animals/zebra/1a3a28f74e.jpg
        inflating: animals/animals/zebra/1bc0f43045.jpg
        inflating: animals/animals/zebra/21e2c20f1e.jpg
        inflating: animals/animals/zebra/28d35d124d.jpg
        inflating: animals/animals/zebra/29b3af6957.jpg
        inflating: animals/animals/zebra/2a285c7664.jpg
        inflating: animals/animals/zebra/2a63e3f385.jpg
        inflating: animals/animals/zebra/2ca8813e35.jpg
        inflating: animals/animals/zebra/2cc1ae5608.jpg
        inflating: animals/animals/zebra/2eab8fe49e.jpg
        inflating: animals/animals/zebra/30c6501e99.jpg
        inflating: animals/animals/zebra/35aecee037.jpg
        inflating: animals/animals/zebra/37aaaec2e6.jpg
        inflating: animals/animals/zebra/39e7a4f1c8.jpg
        inflating: animals/animals/zebra/3c97493537.jpg
        inflating: animals/animals/zebra/3ce736b281.jpg
        inflating: animals/animals/zebra/470f427469.jpg
        inflating: animals/animals/zebra/489e17d69e.jpg
        inflating: animals/animals/zebra/49f81f233b.jpg
        inflating: animals/animals/zebra/4c66fba940.jpg
        inflating: animals/animals/zebra/4f5b41a0f6.jpg
        inflating: animals/animals/zebra/545d133e93.jpg
        inflating: animals/animals/zebra/54b4bd58d6.jpg
        inflating: animals/animals/zebra/5c81996bec.jpg
        inflating: animals/animals/zebra/60af7b19e1.jpg
        inflating: animals/animals/zebra/611b81638f.jpg
        inflating: animals/animals/zebra/649e04b7c0.jpg
        inflating: animals/animals/zebra/667edbff5d.jpg
        inflating: animals/animals/zebra/67dce4e834.jpg
        inflating: animals/animals/zebra/6e7a163893.jpg
        inflating: animals/animals/zebra/6eebb7baff.jpg
        inflating: animals/animals/zebra/733f88ba78.jpg
        inflating: animals/animals/zebra/779ee4ab23.jpg
        inflating: animals/animals/zebra/785be6d4b5.jpg
        inflating: animals/animals/zebra/7ae826e4ac.jpg
        inflating: animals/animals/zebra/7b7d8759f1.jpg
        inflating: animals/animals/zebra/7d35713fda.jpg
        inflating: animals/animals/zebra/7f1352d1e8.jpg
        inflating: animals/animals/zebra/7f3f1ddc56.jpg
        inflating: animals/animals/zebra/7f892f3977.jpg
        inflating: animals/animals/zebra/81b246ab40.jpg
        inflating: animals/animals/zebra/872a4e6a25.jpg
        inflating: animals/animals/zebra/8bc28d0165.jpg
        inflating: animals/animals/zebra/94cf6638fd.jpg
        inflating: animals/animals/zebra/96e65ae7f7.jpg
        inflating: animals/animals/zebra/9c23e26c9d.jpg
        inflating: animals/animals/zebra/9e2a9415dc.jpg
        inflating: name of the animals.txt
```

Data Augmentation

```
#import necessary libraries

from tensorflow.keras.preprocessing.image import ImageDataGenerator



# data augmentation for training variable

train_datagen = ImageDataGenerator(rescale = 1./255,zoom_range = 0.2,horizontal_flip=True)



# data augmentation for testing variable
```

```
test_datagen = ImageDataGenerator(rescale = 1./255)
```

```
# data augmentation on the training data

xtrain = train_datagen.flow_from_directory('/content/animals',
                                           target_size =(64,64),
                                           class_mode = 'categorical',
                                           batch_size = 100)
```

```
     Found 5400 images belonging to 1 classes.
```

```
xtest = test_datagen.flow_from_directory('/content/animals',
                                           target_size =(64,64),
                                           class_mode = 'categorical',
                                           batch_size = 100)
```

```
     Found 5400 images belonging to 1 classes.
```

CNN Model Building

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

```
## adding layers

model = Sequential() # Initializing sequential model
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3))) # convolution layer
model.add(MaxPooling2D(pool_size=(2, 2))) # Max pooling layer
model.add(Flatten()) # Flatten layer
model.add(Dense(300,activation='relu')) # Hidden layer 1
model.add(Dense(150,activation='relu')) # Hidden layer 2
model.add(Dense(4,activation='softmax')) # Output layer
```

```
# Compile the model

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

```
# Train the model

model.fit_generator(xtrain,steps_per_epoch=len(xtrain),
                    epochs=10,validation_data=xtest,validation_steps=len(xtest))
```

```
     <ipython-input-35-d4e3cc611f8d>:3: UserWarning: `Model.fit_generator` is deprecated and will be removed in a futur
       model.fit_generator(xtrain,steps_per_epoch=len(xtrain),
     Epoch 1/10
     54/54 [==============================] - 255s 5s/step - loss: 6942273.0000 - accuracy: 0.2222 - val_loss: 4297699.
     Epoch 2/10
     54/54 [==============================] - 196s 4s/step - loss: 11077621.0000 - accuracy: 0.2593 - val_loss: 1365736
     Epoch 3/10
     54/54 [==============================] - 253s 5s/step - loss: 10686070.0000 - accuracy: 0.2593 - val_loss: 1259082
     Epoch 4/10
     54/54 [==============================] - 253s 5s/step - loss: 20544194.0000 - accuracy: 0.2222 - val_loss: 8496983
     Epoch 5/10
     54/54 [==============================] - 253s 5s/step - loss: 17710036.0000 - accuracy: 0.2593 - val_loss: 7815714
     Epoch 6/10
     54/54 [==============================] - 253s 5s/step - loss: 23104948.0000 - accuracy: 0.2407 - val_loss: 3109276
```

```
Epoch 7/10
54/54 [==============================] - 254s 5s/step - loss: 31592884.0000 - accuracy: 0.2778 - val_loss: 5669662
Epoch 8/10
54/54 [==============================] - 252s 5s/step - loss: 28833070.0000 - accuracy: 0.2407 - val_loss: 2423812
Epoch 9/10
54/54 [==============================] - 195s 4s/step - loss: 27473134.0000 - accuracy: 0.2222 - val_loss: 2133524
Epoch 10/10
54/54 [==============================] - 248s 5s/step - loss: 38020796.0000 - accuracy: 0.2593 - val_loss: 5654919
<keras.callbacks.History at 0x7fe64c741900>
```

```
# Save model

model.save('animal4.h4')
```

```
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _update_step_xla while saving (showing
```

Testing the model

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

```
# Testing 1

img = image.load_img('/content/animals/animals/elephant/032ef781ef.jpg',target_size =(64,64))
img
```

```
x = image.img_to_array(img) # converting the image into array
x = np.expand_dims(x,axis = 0) # expanding dimensions
pred = np.argmax(model.predict(x)) # predicting the higher probability index
op = ['bears','crows','elephants','rats']
op[pred]
```

```
1/1 [==============================] - 0s 179ms/step
'elephants'
```

```
# Testing 2

img = image.load_img('/content/animals/animals/lion/0209bbf635.jpg',target_size =(64,64))
x = image.img_to_array(img) # converting the image into array
x = np.expand_dims(x,axis = 0) # expanding dimensions
pred = np.argmax(model.predict(x)) # predicting the higher probability index
op = ['bears','crows','elephants','rats']
op[pred]
```

```
1/1 [==============================] - 0s 30ms/step
'elephants'
```

```
# Testing 3

img = image.load_img('/content/animals/animals/mosquito/0014c2d720.jpg',target_size =(64,64))
x = image.img_to_array(img) # converting the image into array
x = np.expand_dims(x,axis = 0) # expanding dimensions
pred = np.argmax(model.predict(x)) # predicting the higher probability index
```

```
op = ['bears','crows','elephants','rats']
op[pred]

     1/1 [==============================] - 0s 29ms/step
     'elephants'
```

```
# Testing 4

img = image.load_img('/content/animals/animals/sheep/08b588e8ac.jpg',target_size =(64,64))
x = image.img_to_array(img) # converting the image into array
x = np.expand_dims(x,axis = 0) # expanding dimensions
pred = np.argmax(model.predict(x)) # predicting the higher probability index
op = ['bears','crows','elephants','rats']
op[pred]

     1/1 [==============================] - 0s 30ms/step
     'elephants'
```

✓ 0s completed at 2:42 PM ● ✕

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.