**.DOWNLOAD THE DATASET**

In [ ]:

**from** google.colab **import** drive

drive**.**mount('/content/drive')

Mounted at /content/drive

In [ ]:

**from** tensorflow.keras.models **import** Sequential

**from** tensorflow.keras.layers **import** Convolution2D,MaxPool2D,Flatten,Dense

In [ ]:

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

**2.IMAGE AUGUMENTATION**

In [ ]:

train\_datagen **=** ImageDataGenerator(rescale**=**1.**/**255,shear\_range**=**0.2,zoom\_range**=**0.2,horizontal\_flip**=True**,vertical\_flip**=True**)

In [ ]:

test\_datagen**=**ImageDataGenerator(rescale**=**1.**/**255)

In [ ]:

x\_train **=** train\_datagen**.**flow\_from\_directory(r"/content/drive/MyDrive/dataset/Training",target\_size**=**(64,64),batch\_size**=**32,class\_mode**=**"categorical")

Found 1238 images belonging to 4 classes.

In [ ]:

*#load your images data*

x\_test **=** test\_datagen**.**flow\_from\_directory(r"/content/drive/MyDrive/dataset/Testing",target\_size**=**(64,64),batch\_size**=**32,class\_mode**=**"categorical")

Found 326 images belonging to 4 classes.

In [ ]:

x\_train**.**class\_indices

Out[ ]:

{'bears': 0, 'crows': 1, 'elephants': 2, 'rats': 3}

**3.CREATE MODEL**

In [ ]:

*#initialize the model*

model**=**Sequential()

**4.ADD LAYERS(Convolution,MxPooling,Flatten,Dense-(Hidden Layers),Output)**

In [ ]:

*#add convolution layer*

model**.**add(Convolution2D(32,(3,3),input\_shape**=**(64,64,3),activation**=**'relu'))

In [ ]:

*#add max pooling layer*

model**.**add(MaxPooling2D(pool\_size**=**(2,2)))

In [ ]:

model**.**add(Flatten())

In [ ]:

*#hidden layers*

model**.**add(Dense(units**=**300,kernel\_initializer**=**"random\_uniform",activation**=**"relu"))

model**.**add(Dense(units**=**200,kernel\_initializer**=**"random\_uniform",activation**=**"relu"))

*#output layer*

model**.**add(Dense(units**=**4,kernel\_initializer**=**"random\_uniform",activation**=**"softmax"))

**5.COMPILE THE MODEL**

In [ ]:

*#compile the model*

model**.**compile(loss**=**"categorical\_crossentropy",optimizer**=**"adam",metrics**=**['accuracy'])

**6.FIT THE MODEL**

In [ ]:

model**.**fit\_generator(x\_train,steps\_per\_epoch**=**39,epochs**=**25,validation\_data**=**x\_test,validation\_steps**=**10)

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:1: UserWarning: `Model.fit\_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.

"""Entry point for launching an IPython kernel.

Epoch 1/25

39/39 [==============================] - 213s 5s/step - loss: 1.3571 - accuracy: 0.3086 - val\_loss: 1.2797 - val\_accuracy: 0.3844

Epoch 2/25

39/39 [==============================] - 31s 796ms/step - loss: 1.2132 - accuracy: 0.4338 - val\_loss: 0.9831 - val\_accuracy: 0.5469

Epoch 3/25

39/39 [==============================] - 31s 794ms/step - loss: 0.9853 - accuracy: 0.5792 - val\_loss: 0.8243 - val\_accuracy: 0.6500

Epoch 4/25

39/39 [==============================] - 31s 790ms/step - loss: 0.8966 - accuracy: 0.6284 - val\_loss: 0.7700 - val\_accuracy: 0.6781

Epoch 5/25

39/39 [==============================] - 31s 793ms/step - loss: 0.8226 - accuracy: 0.6656 - val\_loss: 0.6223 - val\_accuracy: 0.7656

Epoch 6/25

39/39 [==============================] - 31s 800ms/step - loss: 0.7507 - accuracy: 0.6922 - val\_loss: 0.5325 - val\_accuracy: 0.8344

Epoch 7/25

39/39 [==============================] - 31s 796ms/step - loss: 0.7334 - accuracy: 0.6931 - val\_loss: 0.6391 - val\_accuracy: 0.7563

Epoch 8/25

39/39 [==============================] - 31s 800ms/step - loss: 0.6739 - accuracy: 0.7246 - val\_loss: 0.4539 - val\_accuracy: 0.8188

Epoch 9/25

39/39 [==============================] - 31s 795ms/step - loss: 0.6430 - accuracy: 0.7528 - val\_loss: 0.5661 - val\_accuracy: 0.7250

Epoch 10/25

39/39 [==============================] - 31s 793ms/step - loss: 0.5744 - accuracy: 0.7617 - val\_loss: 0.3414 - val\_accuracy: 0.8875

Epoch 11/25

39/39 [==============================] - 31s 792ms/step - loss: 0.5035 - accuracy: 0.8013 - val\_loss: 0.5984 - val\_accuracy: 0.7781

Epoch 12/25

39/39 [==============================] - 31s 790ms/step - loss: 0.4987 - accuracy: 0.8053 - val\_loss: 0.3194 - val\_accuracy: 0.8781

Epoch 13/25

39/39 [==============================] - 31s 794ms/step - loss: 0.4479 - accuracy: 0.8183 - val\_loss: 0.2687 - val\_accuracy: 0.8906

Epoch 14/25

39/39 [==============================] - 31s 793ms/step - loss: 0.3554 - accuracy: 0.8740 - val\_loss: 0.2047 - val\_accuracy: 0.9312

Epoch 15/25

39/39 [==============================] - 31s 796ms/step - loss: 0.3572 - accuracy: 0.8667 - val\_loss: 0.3596 - val\_accuracy: 0.8313

Epoch 16/25

39/39 [==============================] - 31s 791ms/step - loss: 0.3545 - accuracy: 0.8708 - val\_loss: 0.1499 - val\_accuracy: 0.9625

Epoch 17/25

39/39 [==============================] - 31s 794ms/step - loss: 0.3031 - accuracy: 0.8885 - val\_loss: 0.1655 - val\_accuracy: 0.9406

Epoch 18/25

39/39 [==============================] - 31s 794ms/step - loss: 0.3006 - accuracy: 0.8990 - val\_loss: 0.1121 - val\_accuracy: 0.9656

Epoch 19/25

39/39 [==============================] - 31s 796ms/step - loss: 0.2436 - accuracy: 0.9063 - val\_loss: 0.0975 - val\_accuracy: 0.9563

Epoch 20/25

39/39 [==============================] - 31s 793ms/step - loss: 0.2332 - accuracy: 0.9233 - val\_loss: 0.0822 - val\_accuracy: 0.9844

Epoch 21/25

39/39 [==============================] - 31s 788ms/step - loss: 0.1828 - accuracy: 0.9346 - val\_loss: 0.0978 - val\_accuracy: 0.9625

Epoch 22/25

39/39 [==============================] - 31s 791ms/step - loss: 0.2079 - accuracy: 0.9330 - val\_loss: 0.2019 - val\_accuracy: 0.9312

Epoch 23/25

39/39 [==============================] - 31s 796ms/step - loss: 0.1691 - accuracy: 0.9410 - val\_loss: 0.0647 - val\_accuracy: 0.9781

Epoch 24/25

39/39 [==============================] - 31s 798ms/step - loss: 0.1361 - accuracy: 0.9491 - val\_loss: 0.0550 - val\_accuracy: 0.9750

Epoch 25/25

39/39 [==============================] - 31s 795ms/step - loss: 0.1839 - accuracy: 0.9346 - val\_loss: 0.1726 - val\_accuracy: 0.9312

Out[ ]:

<keras.callbacks.History at 0x7f42189f8dd0>

**7.SAVE THE MODEL**

In [ ]:

model**.**save("animal.h5")

**8.TEST THE MODEL**

In [85]:

*#CNN prediction*

**from** tensorflow.keras.models **import** load\_model

In [ ]:

**from** tensorflow.keras.preprocessing **import** image

In [ ]:

**import** numpy **as** np

In [71]:

model **=** load\_model('animal.h5')

img **=** image**.**load\_img('/content/drive/MyDrive/dataset/Testing/crows/Z1 (28).jpg',target\_size**=**(64,64))

In [72]:

img

Out[72]:

In [73]:

type(img)

Out[73]:

PIL.Image.Image

In [75]:

x**=**image**.**img\_to\_array(img)

In [76]:

x

Out[76]:

array([[[230., 238., 240.],

[235., 239., 242.],

[235., 239., 242.],

...,

[241., 242., 244.],

[242., 241., 246.],

[242., 241., 246.]],

[[234., 238., 241.],

[235., 239., 242.],

[235., 239., 242.],

...,

[240., 241., 243.],

[241., 240., 245.],

[242., 241., 246.]],

[[234., 238., 241.],

[234., 238., 241.],

[234., 238., 241.],

...,

[242., 241., 246.],

[242., 242., 244.],

[242., 242., 244.]],

...,

[[136., 97., 30.],

[147., 112., 56.],

[168., 128., 59.],

...,

[161., 122., 53.],

[159., 124., 58.],

[171., 132., 63.]],

[[136., 99., 29.],

[147., 112., 44.],

[176., 132., 71.],

...,

[166., 128., 65.],

[164., 126., 53.],

[176., 131., 64.]],

[[148., 109., 50.],

[151., 115., 55.],

[191., 143., 79.],

...,

[168., 130., 67.],

[156., 122., 48.],

[160., 121., 46.]]], dtype=float32)

In [77]:

x**.**shape

Out[77]:

(64, 64, 3)

In [78]:

x**=**np**.**expand\_dims(x,axis**=**0)

In [79]:

pred\_prob**=**model**.**predict(x)

In [80]:

pred\_prob

Out[80]:

array([[0., 1., 0., 0.]], dtype=float32)

In [81]:

class\_name**=**['Bear','Crow','Elephant','Rat']

In [82]:

pred\_id**=**pred\_prob**.**argmax(axis**=**1)[0]

In [83]:

pred\_id

Out[83]:

1

In [84]:

print('Predicted animal is',str(class\_name[pred\_id]))

Predicted animal is Crow