# Animal Image Dataset (90 Different Animals)

Data Stats and Preprocessing

# Data Source

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

# Data Statistics

1. Dataset Consist of 5400 Animal Images for 90 Animals
2. Each image has dimensions [1200,1200,3]. Meaning a coloured picture with 1200 by 1200 pixel
3. The dataset contains 90 classes pertaining to the 90 Animals

*['beetle', 'parrot', 'fox', 'wolf', 'hedgehog', 'turtle', 'snake', 'shark', 'horse', 'antelope', 'hamster', 'woodpecker', 'eagle', 'chimpanzee', 'bison', 'penguin', 'butterfly', 'sandpiper', 'hornbill', 'lion', 'lobster', 'oyster', 'otter', 'grasshopper', 'raccoon', 'dog', 'pigeon', 'donkey', 'hippopotamus', 'hummingbird', 'bear', 'coyote', 'bee', 'pig', 'rat', 'owl', 'caterpillar', 'koala', 'squid', 'hare', 'whale', 'mouse', 'tiger', 'fly', 'lizard', 'orangutan', 'bat', 'porcupine', 'kangaroo', 'gorilla', 'starfish', 'mosquito', 'crow', 'possum', 'goldfish', 'swan', 'elephant', 'sheep', 'dragonfly', 'zebra', 'seal', 'octopus', 'cow', 'ox', 'wombat', 'reindeer', 'panda', 'duck', 'goose', 'goat', 'badger', 'flamingo', 'crab', 'seahorse', 'rhinoceros', 'dolphin', 'okapi', 'moth', 'sparrow', 'pelecaniformes', 'squirrel', 'leopard', 'jellyfish', 'ladybugs', 'cockroach', 'hyena', 'cat', 'deer', 'turkey', 'boar']*

# Data Load

import os

data\_dir = '../input/animal-image-dataset-90-different-animals/animals/animals'

Name = os.listdir(data\_dir)

N=list(range(len(Name)))

normal\_mapping=dict(zip(Name,N))

reverse\_mapping=dict(zip(N,Name))

# Pre-processing

The dataset involves images. Therefore, our pre-processing steps focus on

1. Rescaling – to adapt the input image for the down the line Machine Learning model training
2. Augmentation- to increase the training dataset

## Image ReScaling

Rescale is a value by which we multiply the data before any further processing. Our original images consist of RGB coefficients in the range 0-255, but such values would be too high for our model to process (at a typical learning rate). Therefore, we aim for values between 0 and 1 instead, scaling by a factor of 1/255.

## Image Rotation – Augmentation

An image can be rotated by 0 to 360 degrees clockwise. The pixel of a given image rotates. To use this argument in the ImageDataGenerator class constructor, we pass the argument rotation\_range. The rotation\_range argument accepts an integer value between **0 to 360**.

## Width Shift Range – Augmentation

In this method of augmentation, we shift the image to the left or right (horizontal shifts). If the value is float and <=1 it will take the percentage of total width as range. Suppose image width is 100px. if width\_shift\_range = 1.0 it will take -100% to +100% means -100px to +100px.

## height\_shift\_range

Another augumentation method, It works the same as width\_shift\_range but shift vertically (up or down).

## shear\_range

shear\_range=**0.1 means shear the image by 10%. We have chosen shear\_range as 20%**

## zoom\_range

Zoom-range value of 0.2 means zoom-in and zoom-out by 20%

## horizontal\_flip

This parameter provides a mirror reflection

## fill\_mode

Fill\_mode is a very important argument. When your image shift by 10% there is some space left over. To fill this space we can use constant, nearest, reflect, or wrap. We have chosen ‘nearest’ value for the fill\_mode argument which will fill the area with the nearest pixel and stretching it

## Preprocess function

tf.keras.applications.resnet.preprocess\_input

The preprocess function here is for Resnet architecture. This function then properly transforms a standard image into an appropriate input.

# Pre-processing Code

The code here pre-process training dataset as well as validation dataset. Keras works with batches of images. So, the first dimension is used for the number of samples (or images) you have.

preprocess\_input = tf.keras.applications.resnet.preprocess\_input

batch\_size=64

*# Rescaling the input image as well as Data Augmentation*

train\_datagen = ImageDataGenerator(rescale = 1./255,

validation\_split=0.2,

rotation\_range=30,

width\_shift\_range=0.2,

preprocessing\_function=preprocess\_input,

height\_shift\_range=0.2,

shear\_range=0.2,

zoom\_range=0.2,

horizontal\_flip=True,

fill\_mode='nearest')

validation\_datagen = ImageDataGenerator(rescale = 1./255,validation\_split=0.2)

data\_dir = "../input/animal-image-dataset-90-different-animals/animals/animals"

train\_generator = train\_datagen.flow\_from\_directory(

data\_dir,

target\_size=(255,255),

class\_mode='categorical',

batch\_size=120,subset = "training")

validation\_generator = validation\_datagen.flow\_from\_directory(

data\_dir,

target\_size=(255,255),

class\_mode='categorical',

batch\_size=120,subset = "validation")