```
from google_drive_downloader import GoogleDriveDownloader as gdd

gdd.download_file_from_google_drive(file_id='176E-pLhoxTgWsJ3MeoJQV_GXczIA6g8D',
                                    dest_path='/content/animals.zip',
                                    unzip=True)
#Downloading the data files
```

Downloading data files using the module Google Drive Downloader

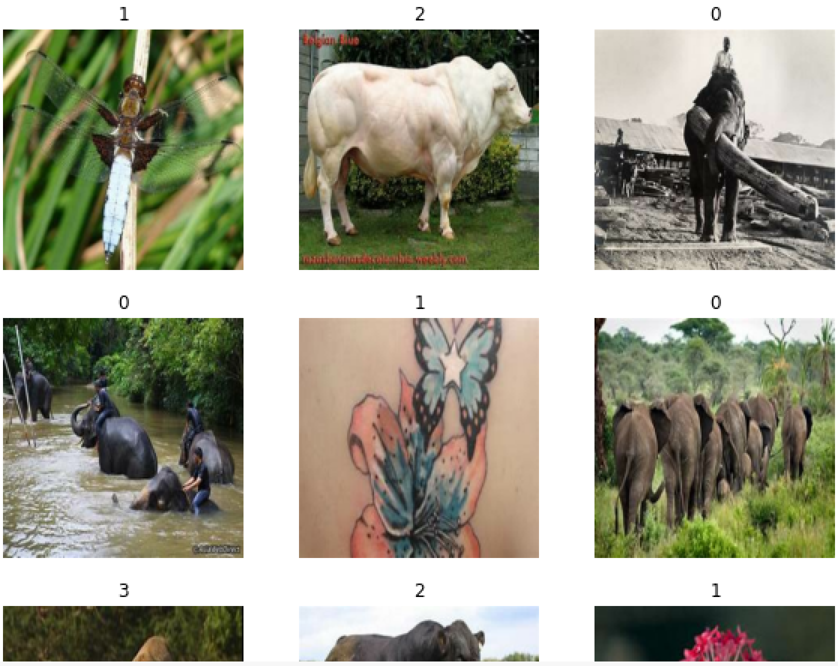
```
#importing necessary libraries
import glob
import cv2
import numpy as np
import matplotlib.pyplot as plt
```

```
import tensorflow as tf
from tensorflow.keras.preprocessing import image_dataset_from_directory                   #importing libraries

image_size = (284, 284)                                                                   #Size of the image
batch_size = 32

train_ds = image_dataset_from_directory(
    "animal_dataset_intermediate/train",
    labels = "inferred",
    label_mode = 'int',
    validation_split=0.2,
    subset="training",                                                                #Obtaining the train set from the Do
    seed=1337,
    image_size=image_size,
    batch_size=batch_size,
)
val_ds =image_dataset_from_directory(
    "animal_dataset_intermediate/train",
    labels = "inferred",
    label_mode = 'int',
    validation_split=0.2,                                                          #Obtaining the validation set from the
    subset="validation",
    seed=1337,
    image_size=image_size,
    batch_size=batch_size
)
```

```
Found 8196 files belonging to 5 classes.
Using 6557 files for training.
Found 8196 files belonging to 5 classes.
Using 1639 files for validation.
```

The data belong to only five clases .Hence the output layer is Built with five perceptrons

```
#visualizing the data
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 10))
for images, labels in train_ds.take(1):
    for i in range(9):
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(images[i].numpy().astype("uint8"))
        plt.title(int(labels[i]))
        plt.axis("off")
```




```
#Augmenting the images
data_augmentation = tf.keras.Sequential(
    [
        tf.keras.layers.experimental.preprocessing.RandomFlip("horizontal"),
        tf.keras.layers.experimental.preprocessing.RandomRotation(0.1),
        tf.keras.layers.experimental.preprocessing.RandomZoom(0.2)
    ]
)

plt.figure(figsize=(10, 10))
for images, _ in train_ds.take(2):
    for i in range(9):
        augmented_images = data_augmentation(images)                                   #visualizes data in different posi
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(augmented_images[0].numpy().astype("uint8"))
        plt.axis("off")
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:14: MatplotlibDeprecationWarning: Adding an axes using
```

CONVERSION OF IMAGES INTO NUMERICS

```
rain_datagen = tf.keras.preprocessing.image.ImageDataGenerator(rescale = 1./255., # rescaling
                                 rotation_range = 40,  # for augmentation
                                 width_shift_range = 0.2,
                                 validation_split = 0.2,
                                 height_shift_range = 0.2,
                                 shear_range = 0.2,                                                                  #CC
                                 zoom_range = 0.2,
                                 horizontal_flip = True)

al_datagen = tf.keras.preprocessing.image.ImageDataGenerator(rescale = 1.0/255.,validation_split = 0.2)



rain_generator = train_datagen.flow_from_directory("animal_dataset_intermediate/train/",
                                                  batch_size = 32,
                                                  subset="training",
                                                  class_mode = 'binary',
                                                  target_size = (284, 284))


al_generator = val_datagen.flow_from_directory("animal_dataset_intermediate/train/",
                                              subset = "validation",
                                                  batch_size = 32,                                          #splitting
                                                  class_mode = 'binary',
                                                  target_size = (284, 284))
```

```
Found 6558 images belonging to 5 classes.
Found 1638 images belonging to 5 classes.
```

```
print(train_generator[0])
```

```
(array([[[[0.33867058, 0.30729803, 0.29553333],
          [0.339035  , 0.30766246, 0.29589772],
          [0.3393994 , 0.30802685, 0.29626215],
          ...,
          [0.5737814 , 0.6561343 , 0.72280097],
          [0.57906955, 0.66142255, 0.7280892 ],
          [0.5803166 , 0.66266954, 0.72933626]],

         [[0.3710886 , 0.33971605, 0.32795134],
          [0.36489365, 0.3335211 , 0.3217564 ],
          [0.3586987 , 0.32732615, 0.31556144],
          ...,
          [0.5739754 , 0.6563283 , 0.722995  ],
          [0.57967544, 0.6620284 , 0.72869503],
          [0.57784176, 0.6601947 , 0.72686136]],

         [[0.399498  , 0.36812544, 0.35636073],
          [0.4005912 , 0.36921865, 0.35745394],
          [0.40168443, 0.3703119 , 0.35854718],
          ...,
          [0.58654916, 0.6689021 , 0.73556876],
          [0.5940623 , 0.67641526, 0.7430819 ],
          [0.59580296, 0.6781559 , 0.74482256]],

         ...,

         [[0.27058825, 0.29803923, 0.32941177],
          [0.27058825, 0.29803923, 0.32941177],
          [0.3064663 , 0.3339173 , 0.36528984],
          ...,
          [0.5529412 , 0.58431375, 0.627451  ],
          [0.5529412 , 0.58431375, 0.627451  ],
          [0.5529412 , 0.58431375, 0.627451  ]],

         [[0.27058825, 0.29803923, 0.32941177],
          [0.27058825, 0.29803923, 0.32941177],
          [0.30941296, 0.33686393, 0.36823648],
          ...,
          [0.5529412 , 0.58431375, 0.627451  ],
          [0.5529412 , 0.58431375, 0.627451  ],
          [0.5529412 , 0.58431375, 0.627451  ]],

         [[0.27058825, 0.29803923, 0.32941177],
          [0.27102068, 0.29847166, 0.3298442 ],
          [0.30980393, 0.3372549 , 0.36862746],
          ...,
          [0.5529412 , 0.58431375, 0.627451  ],
          [0.5529412 , 0.58431375, 0.627451  ],
          [0.5529412 , 0.58431375, 0.627451  ]]],


        [[[0.22603038, 0.30446175, 0.16720685],
          [0.23319457, 0.31162596, 0.17437105],
          [0.24328896, 0.32172033, 0.18446542],
          ...,
          [0.2581841 , 0.28013638, 0.19292408],
          [0.2521137 , 0.2709249 , 0.1850487 ],
          [0.24604328, 0.26312006, 0.17811106]],

         [[0.21900064, 0.29743204, 0.1601771 ],
          [0.22187996, 0.30031133, 0.16305643],
          [0.231682  , 0.31011337, 0.17285846],
          ...,
          [0.23545812, 0.24951059, 0.16601376],
          [0.22938772, 0.24170576, 0.15907612],
          [0.22686046, 0.23803464, 0.1556817 ]],

         [[0.2178811 , 0.29631248, 0.15905757],
          [0.2187777 , 0.29720905, 0.15995418],
          [0.22968382, 0.30811518, 0.17086029],
          ...,
          [0.22534831, 0.2350103 , 0.15265736],
          [0.22448109, 0.2332759 , 0.15092295],
          [0.22361389, 0.2315415 , 0.14918855]],

         ...,

         [[0.3803922 , 0.44705886, 0.32156864],
          [0.3803922 , 0.44705886, 0.32156864],
          [0.3803922 , 0.44705886, 0.32156864],
          ...,
          [0.41960788, 0.45882356, 0.3137255 ]
```

```
         [0.41960788, 0.45882356, 0.3137255 ],
         [0.41960788, 0.45882356, 0.3137255 ],
         [0.41960788, 0.45882356, 0.3137255 ]],

        [[0.3803922 , 0.44547552, 0.31919366],
         [0.3803922 , 0.4437411 , 0.31659204],
         [0.3803922 , 0.4420067 , 0.31399044],
         ...,
         [0.41960788, 0.45882356, 0.3137255 ],
         [0.41960788, 0.45882356, 0.3137255 ],
         [0.41960788, 0.45882356, 0.3137255 ]],

        [[0.38062552, 0.43956572, 0.30945393],
         [0.3823599 , 0.44216734, 0.3068523 ],
         [0.38409433, 0.44476894, 0.30425072],
         ...,
         [0.41960788, 0.45882356, 0.3137255 ],
         [0.41960788, 0.45882356, 0.3137255 ],
         [0.41960788, 0.45882356, 0.3137255 ]]],


       [[[0.54303575, 0.51558477, 0.49205533],
         [0.5483254 , 0.5237202 , 0.5030366 ],
         [0.5349078 , 0.5147752 , 0.4985641 ],
         ...,
         [1.        , 0.9960785 , 1.        ],
         [1.        , 0.9960785 , 1.        ],
         [0.9997271 , 0.9955325 , 0.9994541 ]],

        [[0.49938545, 0.47040293, 0.44610775],
         [0.5151942 , 0.48774317, 0.46421376],
         [0.5532105 , 0.5257595 , 0.5022301 ],
         ...,
         [1.        , 0.9960785 , 1.        ],
         [1.        , 0.9960785 , 1.        ],
         [0.99737155, 0.9908214 , 0.994743  ]],

        [[0.5681377 , 0.5328436 , 0.5053926 ],
         [0.5190383 , 0.48678032, 0.46084738],
         [0.49220327, 0.46441776, 0.4407211 ],
         ...,
         [1.        , 0.9960785 , 1.        ],
         [0.9987005 , 0.9934794 , 0.99740094],
         [0.99289095, 0.9850478 , 0.9889694 ]],

        ...,

        [[0.8921108 , 0.8685814 , 0.81339806],
         [0.6574387 , 0.6445406 , 0.58609486],
         [0.53217024, 0.47741783, 0.41528755],
         ...,
         [0.46181142, 0.4400094 , 0.3571896 ],
         [0.47067237, 0.45114863, 0.37275928],
         [0.47514486, 0.46009362, 0.38394058]],

        [[0.7491366 , 0.7322518 , 0.67513496],
         [0.57203895, 0.534563  , 0.47376165],
         [0.47368863, 0.41616577, 0.35342064],
         ...,
         [0.46563515, 0.43781248, 0.3493913 ],
         [0.4552605 , 0.4318207 , 0.34572545],
         [0.4642055 , 0.443002  , 0.3613792 ]],

        [[0.61190766, 0.5917081 , 0.53223574],
         [0.50425464, 0.44274488, 0.3799998 ],
         [0.42929527, 0.37848958, 0.3157445 ],
         ...,
         [0.45561835, 0.4200477 , 0.32565352],
         [0.47382382, 0.4427257 , 0.3526668 ],
         [0.46264255, 0.4360169 , 0.34819427]]],


       ...,


       [[[0.44626984, 0.5564653 , 0.33430934],
         [0.44095936, 0.56197846, 0.33161002],
         [0.3736617 , 0.51079935, 0.27322263],
         ...,
         [0.49803925, 0.5568628 , 0.34117648],
         [0.5020997 , 0.559729  , 0.3445204 ],
         [0.5428296 , 0.5884795 , 0.37806267]],
```

```
        [[0.41283852, 0.51829135, 0.30292723],
         [0.46162605, 0.5737411 , 0.34870577],
         [0.41888484, 0.5447027 , 0.31241477],
         ...,
         [0.51319945, 0.5675641 , 0.35366136],
         [0.5539294 , 0.59631467, 0.38720363],
         [0.58584946, 0.61449337, 0.41017416]],

        [[0.43547806, 0.5318751 , 0.3233028 ],
         [0.42540282, 0.53299   , 0.31474653],
         [0.47095573, 0.58545387, 0.3576938 ],
         ...,
         [0.5649341 , 0.6040357 , 0.39623055],
         [0.59368455, 0.61841094, 0.41539758],
         [0.6366623 , 0.6479026 , 0.4505702 ]],

        ...,

        [[0.4512042 , 0.36585277, 0.20322232],
         [0.40089077, 0.31793517, 0.15051301],
         [0.35775027, 0.2753973 , 0.10318339],
         ...,
         [0.5896049 , 0.69773644, 0.43778834],
         [0.48274726, 0.61441374, 0.3432618 ],
         [0.32800898, 0.47257036, 0.19819468]],

        [[0.38738045, 0.3050275 , 0.13629949],
         [0.34665054, 0.2642976 , 0.09077778],
         [0.32156864, 0.2392157 , 0.0627451 ],
         ...,
         [0.5923665 , 0.68539363, 0.4324965 ],
         [0.56081206, 0.6766217 , 0.4128345 ],
         [0.43667865, 0.5721842 , 0.30007252]],

        [[0.33555079, 0.25319785, 0.0783722 ],
         [0.32156864, 0.2392157 , 0.0627451 ],
         [0.31961256, 0.23778442, 0.06198175],
         ...,
         [0.5131282 , 0.5948356 , 0.34646633],
         [0.62595814, 0.72378415, 0.46896744],
         [0.5320192 , 0.6555069 , 0.38788068]]],


       [[[0.29702953, 0.38123244, 0.34272963],
         [0.2885784 , 0.37018147, 0.3467408 ],
         [0.27743313, 0.3563501 , 0.34177873],
         ...,
         [0.29406002, 0.30128092, 0.25011232],
         [0.28738207, 0.29425153, 0.24519175],
         [0.28070414, 0.28722212, 0.24027118]],

        [[0.2903495 , 0.36147937, 0.31057623],
         [0.31232172, 0.38290998, 0.33585113],
         [0.29432318, 0.35513455, 0.3625636 ],
         ...,
         [0.23137257, 0.23529413, 0.20392159],
         [0.23137257, 0.23529413, 0.20392159],
         [0.23137257, 0.23529413, 0.20392159]],

        [[0.319202  , 0.38979024, 0.33788118],
         [0.3008194 , 0.37140763, 0.3243488 ],
         [0.2850828 , 0.3467569 , 0.34426865],
         ...,
         [0.22721489, 0.23113646, 0.20132303],
         [0.22440314, 0.22832471, 0.1995657 ],
         [0.22159138, 0.22551295, 0.19780834]],

        ...,

        [[0.36078432, 0.4039216 , 0.2392157 ],
         [0.36078432, 0.4039216 , 0.2392157 ],
         [0.36078432, 0.4039216 , 0.2392157 ],
         ...,
         [0.43921572, 0.38823533, 0.18039216],
         [0.43921572, 0.38823533, 0.18039216],
         [0.43921572, 0.38823533, 0.18039216]],

        [[0.36078432, 0.4039216 , 0.2392157 ],
         [0.36078432, 0.4039216 , 0.2392157 ],
         [0.36078432, 0.4039216 , 0.2392157 ],
         ...,
         [0.43921572, 0.38823533, 0.18039216]
```

```
        [0.43921572, 0.38823533, 0.18039216],
        [0.43921572, 0.38823533, 0.18039216],
        [0.43921572, 0.38823533, 0.18039216]],

       [[0.36078432, 0.4039216 , 0.2392157 ],
        [0.36078432, 0.4039216 , 0.2392157 ],
        [0.36078432, 0.4039216 , 0.2392157 ],
        ...,
        [0.43921572, 0.38823533, 0.18039216],
        [0.43921572, 0.38823533, 0.18039216],
        [0.43921572, 0.38823533, 0.18039216]]],


      [[[0.7372549 , 0.7019608 , 0.6666667 ],
        [0.7372549 , 0.7019608 , 0.6666667 ],
        [0.73713815, 0.7023113 , 0.6673676 ],
        ...,
        [0.77647066, 0.80392164, 0.8431373 ],
        [0.77662444, 0.8040754 , 0.8432911 ],
        [0.77997285, 0.80742383, 0.8466395 ]],

       [[0.6457929 , 0.6079582 , 0.56504226],
        [0.7013317 , 0.6650397 , 0.62675196],
```

```
#Building of a CNN model with multiple layer architecture
from tensorflow.keras.layers import Input, Conv2D, Dense, Activation, Flatten, Dropout, MaxPooling2D, BatchNormalization#
INPUT_SHAPE = (284, 284, 3)
from tensorflow.keras import regularizers

# define sequential model
model = tf.keras.models.Sequential()
# define conv-pool layers - set 1
model.add(tf.keras.layers.Conv2D(filters=64, kernel_size=(3, 3), strides=(1, 1),
                                activation='elu', padding='valid',kernel_initializer='he_uniform', kernel_regularizer=regu
model.add(tf.keras.layers.MaxPooling2D(pool_size=(2, 2)))
# define conv-pool layers - set 2
model.add(tf.keras.layers.Conv2D(filters=32, kernel_size=(3, 3), strides=(1, 1),
                                activation='elu', padding='valid',kernel_initializer='he_uniform', kernel_regularizer=regu
model.add(tf.keras.layers.MaxPooling2D(pool_size=(2, 2)))
model.add(tf.keras.layers.Conv2D(filters=16, kernel_size=(3, 3), strides=(1, 1),
                                activation='elu', padding='valid',kernel_initializer='he_uniform', kernel_regularizer=regu
model.add(tf.keras.layers.MaxPooling2D(pool_size=(2, 2)))
model.add(tf.keras.layers.Conv2D(filters=8, kernel_size=(3, 3), strides=(1, 1),
                                activation='elu', padding='valid',kernel_initializer='he_uniform', kernel_regularizer=regu
model.add(tf.keras.layers.MaxPooling2D(pool_size=(2, 2)))

# add flatten layer
model.add(tf.keras.layers.Flatten())

# add dense layers with some dropout
model.add(tf.keras.layers.Dense(512, activation='relu'))
model.add(BatchNormalization())
model.add(tf.keras.layers.Dropout(rate=0.3))
model.add(tf.keras.layers.Dense(256, activation='relu'))
model.add(BatchNormalization())
model.add(tf.keras.layers.Dropout(rate=0.3))

# add output layer
model.add(tf.keras.layers.Dense(5, activation='softmax'))
```

```
# compile model
from tensorflow.keras import optimizers
optimizer = optimizers.Adam(lr = 0.001, decay = 1e-5)
model.compile(optimizer=optimizer,
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

```
model.summary()
```

```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d_12 (Conv2D)           (None, 282, 282, 64)      1792      
_________________________________________________________________
max_pooling2d_12 (MaxPooling (None, 141, 141, 64)      0         
_________________________________________________________________
conv2d_13 (Conv2D)           (None, 139, 139, 32)      18464     
_________________________________________________________________
max_pooling2d_13 (MaxPooling (None, 69, 69, 32)        0         
_________________________________________________________________
conv2d_14 (Conv2D)           (None, 67, 67, 16)        4624      
_________________________________________________________________
max_pooling2d_14 (MaxPooling (None, 33, 33, 16)        0         
_________________________________________________________________
conv2d_15 (Conv2D)           (None, 31, 31, 8)         1160      
_________________________________________________________________
max_pooling2d_15 (MaxPooling (None, 15, 15, 8)         0         
_________________________________________________________________
flatten_4 (Flatten)          (None, 1800)              0         
_________________________________________________________________
dense_13 (Dense)             (None, 512)               922112    
_________________________________________________________________
batch_normalization_7 (Batch (None, 512)               2048      
_________________________________________________________________
dropout_9 (Dropout)          (None, 512)               0         
_________________________________________________________________
dense_14 (Dense)             (None, 256)               131328    
_________________________________________________________________
batch_normalization_8 (Batch (None, 256)               1024      
_________________________________________________________________
dropout_10 (Dropout)         (None, 256)               0         
_________________________________________________________________
dense_15 (Dense)             (None, 5)                 1285      
=================================================================
Total params: 1,083,837
Trainable params: 1,082,301
Non-trainable params: 1,536
_________________________________________________________________
```

```
es_callback = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=2,
                                               restore_best_weights=True,                        #Use of Early Stopping
                                               verbose=1)
```

```
 model.fit_generator(
    train_generator,
    validation_data=val_generator,
    epochs=100,verbose=1,
    callbacks = [es_callback]                                                                    #fitting the model
)
```

```
Epoch 1/100
205/205 [==============================] - 129s 627ms/step - loss: 1.6927 - accuracy: 0.3835 - val_loss: 2.0878 - va
Epoch 2/100
205/205 [==============================] - 127s 620ms/step - loss: 1.3121 - accuracy: 0.4765 - val_loss: 22.9385 - v
Epoch 3/100
205/205 [==============================] - 128s 623ms/step - loss: 1.3606 - accuracy: 0.4597 - val_loss: 1.1973 - va
Epoch 4/100
205/205 [==============================] - 127s 621ms/step - loss: 1.1466 - accuracy: 0.5444 - val_loss: 1.3949 - va
Epoch 5/100
205/205 [==============================] - 127s 621ms/step - loss: 1.1103 - accuracy: 0.5500 - val_loss: 0.9817 - va
Epoch 6/100
205/205 [==============================] - 127s 619ms/step - loss: 1.0505 - accuracy: 0.5871 - val_loss: 1.6704 - va
Epoch 7/100
205/205 [==============================] - 126s 615ms/step - loss: 1.0324 - accuracy: 0.5985 - val_loss: 0.8924 - va
Epoch 8/100
205/205 [==============================] - 126s 617ms/step - loss: 0.9749 - accuracy: 0.6228 - val_loss: 0.9603 - va
Epoch 9/100
205/205 [==============================] - 125s 609ms/step - loss: 0.9549 - accuracy: 0.6325 - val_loss: 0.8783 - va
Epoch 10/100
205/205 [==============================] - 125s 609ms/step - loss: 0.9206 - accuracy: 0.6526 - val_loss: 1.0737 - va
Epoch 11/100
205/205 [==============================] - ETA: 0s - loss: 0.9127 - accuracy: 0.6516Restoring model weights from the
205/205 [==============================] - 124s 606ms/step - loss: 0.9127 - accuracy: 0.6516 - val_loss: 0.9360 - va
Epoch 00011: early stopping
<tensorflow.python.keras.callbacks.History at 0x7f076de46668>
```

```
predictor = model.predict_generator(test_img)
print(predictor)
```

```
[[3.57003580e-03 9.72729683e-01 6.71693415e-04 3.49360076e-03
  1.95349362e-02]
 [9.63128433e-02 3.97134691e-01 2.53202692e-02 7.56583065e-02
  4.05573875e-01]
 [1.07116317e-02 8.04808378e-01 9.90184210e-03 2.79712528e-02
  1.46606937e-01]
 ...
 [1.97279956e-02 8.98142338e-01 1.13927305e-03 7.84919411e-03
  7.31411427e-02]
 [4.06422988e-02 8.77649307e-01 3.08456924e-03 1.03502525e-02
  6.82735294e-02]
 [6.56816959e-02 7.78458536e-01 5.72279189e-03 1.58522148e-02
  1.34284705e-01]]
```

```
model.evaluate(train_generator)
```

```
205/205 [==============================] - 119s 581ms/step - loss: 1.0205 - accuracy: 0.6096
[1.020519733428955, 0.6096370816230774]
```

```
form=np.argmax(predictor,axis=1)
form.shape
```

```
(910,)
```

```
output_result=pd.DataFrame(form)
output_result.head()
```

	0
0	1
1	4
2	1
3	1
4	1

```
output_result= output_result.replace([0,1,2,3,4],["elefante","farfalla","mucca","pecora","scoiattolo"])
print (output_result)
```

```
              0
0      farfalla
1    scoiattolo
2      farfalla
3      farfalla
4      farfalla
..          ...
905    farfalla
906  scoiattolo
907    farfalla
908    farfalla
909    farfalla

[910 rows x 1 columns]
```

```
array=[]
for i in range(len(output_result)):
  array.append(output_result[0][i])
print(array)
```

```
['farfalla', 'scoiattolo', 'farfalla', 'farfalla', 'farfalla', 'farfalla', 'elefante', 'scoiattolo', 'scoiattolo', '
```

```
final_result=pd.DataFrame(array)
final_result.head()
```

	0
0	farfalla
1	scoiattolo
2	farfalla
3	farfalla

```
# To download the csv file locally
from google.colab import files           #importing files library
result.to_csv('cnn_result.csv')     #converting the res dataframe into an prediction_results csv file
files.download('cnn_result.csv')
```

Use of Transfer learning reduces the time spent on building and training a modl from scratch.It reduces time as well as it produces a better accuracy when compared to a CNN model built entirely from scratch.Hence here we use VGG19 model for image recognition

```
import cv2
import pandas as pd
test_file=pd.read_csv('/content/animal_dataset_intermediate/Testing_set_animals.csv')
test_img=[]
for i in test_file['filename']:                                                                    #Loading test data
  n=cv2.imread('animal_dataset_intermediate/test/'+i)
  r=cv2.resize(n,(284,284))
  test_img.append(r)
```

```
from keras.preprocessing.image import ImageDataGenerator                                   #Conversion of images into num
test_data_gen = ImageDataGenerator(rescale=1.0/255.0)
```

```
test_data_gen.fit(test_img)
```

```
test_img[0]
```

```
array([[[118, 147, 151],
        [119, 148, 152],
        [121, 149, 153],
        ...,
        [ 70, 129, 109],
        [ 69, 128, 108],
        [ 69, 128, 108]],

       [[118, 147, 151],
        [120, 149, 153],
        [121, 150, 154],
        ...,
        [ 70, 129, 109],
        [ 70, 129, 109],
        [ 69, 128, 108]],
```

```
test_img = np.array(test_img)
```

```
        [119, 148, 152],
```

```
test_img=test_img/255                                                   #Normalizing
```

```
        [ 72, 128, 109],
```

```
test_img
```

```
array([[[[0.4627451 , 0.57647059, 0.59215686],
         [0.46666667, 0.58039216, 0.59607843],
         [0.4745098 , 0.58431373, 0.6       ],
         ...,
         [0.2745098 , 0.50588235, 0.42745098],
         [0.27058824, 0.50196078, 0.42352941],
         [0.27058824, 0.50196078, 0.42352941]],

        [[0.4627451 , 0.57647059, 0.59215686],
         [0.47058824, 0.58431373, 0.6       ],
         [0.4745098 , 0.58823529, 0.60392157],
         ...,
         [0.2745098 , 0.50588235, 0.42745098],
         [0.2745098 , 0.50588235, 0.42745098],
         [0.27058824, 0.50196078, 0.42352941]],

        [[0.4627451 , 0.57647059, 0.59215686],
         [0.46666667, 0.58039216, 0.59607843],
         [0.47058824, 0.58431373, 0.6       ],
         ...,
         [0.28235294, 0.50196078, 0.42745098],
         [0.27843137, 0.49803922, 0.42352941],
         [0.27058824, 0.49019608, 0.41568627]],

        ...,

        [[0.32156863, 0.63529412, 0.63921569],
         [0.32156863, 0.63529412, 0.63921569],
         [0.32156863, 0.63529412, 0.63921569],
         ...,
         [0.37254902, 0.55686275, 0.61176471],
         [0.37254902, 0.55686275, 0.61176471],
         [0.37254902, 0.55686275, 0.61176471]],

        [[0.31764706, 0.63137255, 0.63529412],
         [0.31764706, 0.63137255, 0.63529412],
         [0.31372549, 0.62745098, 0.63137255],
         ...,
         [0.37254902, 0.55294118, 0.61176471],
         [0.36862745, 0.55294118, 0.60784314],
         [0.37254902, 0.55686275, 0.61176471]],

        [[0.31372549, 0.62745098, 0.63137255],
         [0.31372549, 0.62745098, 0.63137255],
         [0.31372549, 0.62745098, 0.63137255],
         ...,
         [0.36862745, 0.55294118, 0.60784314],
         [0.36862745, 0.55294118, 0.60784314],
         [0.36862745, 0.55294118, 0.60784314]]],


       [[[0.3254902 , 0.63137255, 0.68235294],
         [0.3254902 , 0.63137255, 0.68235294],
         [0.3254902 , 0.63137255, 0.68235294],
         ...,
         [0.27058824, 0.61568627, 0.67843137],
         [0.2745098 , 0.61568627, 0.67843137],
         [0.2745098 , 0.61568627, 0.67843137]],

        [[0.3254902 , 0.63137255, 0.68235294],
         [0.3254902 , 0.63137255, 0.68235294],
         [0.3254902 , 0.63137255, 0.68235294],
         ...,
         [0.27058824, 0.61568627, 0.67843137],
         [0.2745098 , 0.61568627, 0.67843137],
         [0.2745098 , 0.61568627, 0.67843137]],

        [[0.3254902 , 0.63137255, 0.68235294],
         [0.3254902 , 0.63137255, 0.68235294],
         [0.3254902 , 0.63137255, 0.68235294],
         ...,
         [0.27058824, 0.61568627, 0.67843137],
         [0.2745098 , 0.61568627, 0.67843137],
         [0.2745098 , 0.61568627, 0.67843137]],

        ...,

        [[0.25098039, 0.63529412, 0.6745098 ],
         [0.25098039, 0.63529412, 0.6745098 ],
         [0.24313725, 0.63921569, 0.6745098 ],
         ...,
         [0.24313725, 0.58039216, 0.62745098]
```

```
         [0.24313725, 0.58039216, 0.62745098],
         [0.25098039, 0.58039216, 0.62745098],
         [0.25098039, 0.58039216, 0.62745098]],

        [[0.24705882, 0.63137255, 0.67058824],
         [0.24705882, 0.63137255, 0.67058824],
         [0.23921569, 0.63529412, 0.67058824],
         ...,
         [0.24313725, 0.58431373, 0.62352941],
         [0.25098039, 0.58431373, 0.62352941],
         [0.25098039, 0.58431373, 0.62352941]],

        [[0.24313725, 0.63137255, 0.66666667],
         [0.24705882, 0.63137255, 0.67058824],
         [0.23921569, 0.63529412, 0.67058824],
         ...,
         [0.23921569, 0.58039216, 0.61960784],
         [0.24705882, 0.58039216, 0.61960784],
         [0.24705882, 0.58039216, 0.61960784]]],


       [[[0.38431373, 0.77647059, 0.70588235],
         [0.38823529, 0.78039216, 0.70980392],
         [0.38823529, 0.78039216, 0.70980392],
         ...,
         [0.57647059, 0.89411765, 0.81960784],
         [0.57647059, 0.89411765, 0.81960784],
         [0.57647059, 0.89411765, 0.81960784]],

        [[0.38431373, 0.77647059, 0.70588235],
         [0.38431373, 0.77647059, 0.70588235],
         [0.38823529, 0.78039216, 0.70980392],
         ...,
         [0.57647059, 0.89411765, 0.81960784],
         [0.57647059, 0.89411765, 0.81960784],
         [0.58039216, 0.89803922, 0.82352941]],

        [[0.38431373, 0.77647059, 0.70588235],
         [0.38431373, 0.77647059, 0.70588235],
         [0.38431373, 0.77647059, 0.70588235],
         ...,
         [0.57647059, 0.89411765, 0.81960784],
         [0.57647059, 0.89411765, 0.81960784],
         [0.57647059, 0.89411765, 0.81960784]],

        ...,

        [[0.49019608, 0.90196078, 0.86666667],
         [0.49019608, 0.90196078, 0.85882353],
         [0.48627451, 0.90196078, 0.85882353],
         ...,
         [0.43529412, 0.76078431, 0.63137255],
         [0.43137255, 0.77254902, 0.63529412],
         [0.42745098, 0.78823529, 0.65098039]],

        [[0.49411765, 0.89803922, 0.86666667],
         [0.49019608, 0.90196078, 0.85882353],
         [0.49019608, 0.90196078, 0.85882353],
         ...,
         [0.44705882, 0.76470588, 0.63529412],
         [0.44313725, 0.77647059, 0.64313725],
         [0.45098039, 0.78823529, 0.65490196]],

        [[0.50196078, 0.90588235, 0.8745098 ],
         [0.49803922, 0.90588235, 0.8627451 ],
         [0.49411765, 0.90196078, 0.85882353],
         ...,
         [0.45882353, 0.76470588, 0.63921569],
         [0.4627451 , 0.78039216, 0.65098039],
         [0.47058824, 0.8       , 0.66666667]]],


       ...,


       [[[0.        , 0.01176471, 0.        ],
         [0.00784314, 0.07843137, 0.06666667],
         [0.09803922, 0.16470588, 0.15294118],
         ...,
         [0.14901961, 0.44705882, 0.35294118],
         [0.11764706, 0.41960784, 0.3254902 ],
         [0.1254902 , 0.43137255, 0.3372549 ]],
```