```
from zipfile import ZipFile
```

```
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).
```

```
!unzip '/content/drive/MyDrive/Animal_Dataset.zip'
```

```
Archive:  /content/drive/MyDrive/Animal_Dataset.zip
  inflating: dataset/Testing/bears/k4 (100).jpeg
  inflating: dataset/Testing/bears/k4 (100).jpg
  inflating: dataset/Testing/bears/k4 (101).jpeg
  inflating: dataset/Testing/bears/k4 (101).jpg
  inflating: dataset/Testing/bears/k4 (102).jpeg
  inflating: dataset/Testing/bears/k4 (102).jpg
  inflating: dataset/Testing/bears/k4 (103).jpeg
  inflating: dataset/Testing/bears/k4 (104).jpeg
  inflating: dataset/Testing/bears/k4 (105).jpeg
  inflating: dataset/Testing/bears/k4 (106).jpeg
  inflating: dataset/Testing/bears/k4 (107).jpeg
  inflating: dataset/Testing/bears/k4 (108).jpeg
  inflating: dataset/Testing/bears/k4 (109).jpeg
  inflating: dataset/Testing/bears/k4 (110).jpeg
  inflating: dataset/Testing/bears/k4 (71).jpg
  inflating: dataset/Testing/bears/k4 (72).jpeg
  inflating: dataset/Testing/bears/k4 (72).jpg
  inflating: dataset/Testing/bears/k4 (73).jpeg
  inflating: dataset/Testing/bears/k4 (73).jpg
  inflating: dataset/Testing/bears/k4 (74).jpeg
  inflating: dataset/Testing/bears/k4 (74).jpg
  inflating: dataset/Testing/bears/k4 (75).jpeg
  inflating: dataset/Testing/bears/k4 (75).jpg
  inflating: dataset/Testing/bears/k4 (76).jpeg
  inflating: dataset/Testing/bears/k4 (76).jpg
  inflating: dataset/Testing/bears/k4 (77).jpeg
  inflating: dataset/Testing/bears/k4 (77).jpg
  inflating: dataset/Testing/bears/k4 (78).jpeg
  inflating: dataset/Testing/bears/k4 (78).jpg
  inflating: dataset/Testing/bears/k4 (79).jpeg
  inflating: dataset/Testing/bears/k4 (79).jpg
  inflating: dataset/Testing/bears/k4 (80).jpeg
  inflating: dataset/Testing/bears/k4 (80).jpg
  inflating: dataset/Testing/bears/k4 (81).jpeg
  inflating: dataset/Testing/bears/k4 (81).jpg
  inflating: dataset/Testing/bears/k4 (82).jpeg
  inflating: dataset/Testing/bears/k4 (82).jpg
  inflating: dataset/Testing/bears/k4 (83).jpeg
  inflating: dataset/Testing/bears/k4 (83).jpg
  inflating: dataset/Testing/bears/k4 (84).jpeg
  inflating: dataset/Testing/bears/k4 (84).jpg
  inflating: dataset/Testing/bears/k4 (85).jpeg
  inflating: dataset/Testing/bears/k4 (85).jpg
  inflating: dataset/Testing/bears/k4 (86).jpeg
  inflating: dataset/Testing/bears/k4 (86).jpg
  inflating: dataset/Testing/bears/k4 (87).jpeg
  inflating: dataset/Testing/bears/k4 (87).jpg
  inflating: dataset/Testing/bears/k4 (88).jpeg
  inflating: dataset/Testing/bears/k4 (88).jpg
  inflating: dataset/Testing/bears/k4 (89).jpeg
  inflating: dataset/Testing/bears/k4 (89).jpg
  inflating: dataset/Testing/bears/k4 (90).jpeg
  inflating: dataset/Testing/bears/k4 (90).jpg
  inflating: dataset/Testing/bears/k4 (91).jpeg
  inflating: dataset/Testing/bears/k4 (91).jpg
  inflating: dataset/Testing/bears/k4 (92).jpeg
  inflating: dataset/Testing/bears/k4 (92).jpg
```

```
# Data Augmentation
```

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

```
train_gen = ImageDataGenerator(rescale=(1./255),horizontal_flip=True,shear_range=0.2)
test_gen = ImageDataGenerator(rescale=(1./255)) #--> (0 to 255) convert to (0 to 1)
```

```
train = train_gen. flow_from_directory('/content/dataset/Training',
                                       target_size=(120, 120),
```

```
                                        class_mode='categorical',
                                        batch_size=8)
test = test_gen.flow_from_directory('/content/dataset/Testing',
                                       target_size=(120, 120),
                                        class_mode='categorical',
                                        batch_size=8)
```

```
Found 1238 images belonging to 4 classes.
Found 326 images belonging to 4 classes.
```

```
train.class_indices
```

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

```
# CNN

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


model = Sequential()
model.add(Convolution2D(20,(3,3),activation='relu',input_shape=(120,120,3)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(45,activation='relu'))
model.add(Dense(4,activation='softmax'))


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


model.fit(train,batch_size=8,validation_data=test,epochs=10)
```

```
Epoch 1/10
155/155 [==============================] - 14s 86ms/step - loss: 1.3693 - accuracy: 0.5679 - val_loss: 0.6885 - val_accuracy: 0.7209
Epoch 2/10
155/155 [==============================] - 13s 83ms/step - loss: 0.5579 - accuracy: 0.8094 - val_loss: 0.2989 - val_accuracy: 0.9172
Epoch 3/10
155/155 [==============================] - 13s 81ms/step - loss: 0.3292 - accuracy: 0.8950 - val_loss: 0.1648 - val_accuracy: 0.9724
Epoch 4/10
155/155 [==============================] - 13s 82ms/step - loss: 0.1703 - accuracy: 0.9612 - val_loss: 0.0568 - val_accuracy: 0.9969
Epoch 5/10
155/155 [==============================] - 13s 82ms/step - loss: 0.0969 - accuracy: 0.9798 - val_loss: 0.0938 - val_accuracy: 0.9816
Epoch 6/10
155/155 [==============================] - 13s 81ms/step - loss: 0.0699 - accuracy: 0.9863 - val_loss: 0.0342 - val_accuracy: 0.9969
Epoch 7/10
155/155 [==============================] - 13s 84ms/step - loss: 0.0655 - accuracy: 0.9871 - val_loss: 0.0421 - val_accuracy: 0.9939
Epoch 8/10
155/155 [==============================] - 12s 80ms/step - loss: 0.0400 - accuracy: 0.9960 - val_loss: 0.0094 - val_accuracy: 1.0000
Epoch 9/10
155/155 [==============================] - 13s 81ms/step - loss: 0.0221 - accuracy: 0.9976 - val_loss: 0.0044 - val_accuracy: 1.0000
Epoch 10/10
155/155 [==============================] - 12s 79ms/step - loss: 0.0248 - accuracy: 0.9960 - val_loss: 0.0058 - val_accuracy: 1.0000
<keras.src.callbacks.History at 0x7b01b020f310>
```

```
model.save('animal.h5')
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3000: UserWarning: You are saving your model as an HDF5 file via `m
  saving_api.save_model(
```

```
# Testing

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


img = image.load_img('/content/drive/MyDrive/WhatsApp Image 2023-10-08 at 3.04.17 PM (1).jpeg',target_size=(120,120))


img
```

```
img = image.img_to_array(img)
img
```

```
array([[[255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.],
        ...,
        [252., 247., 244.],
        [251., 246., 243.],
        [251., 246., 243.]],

       [[255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.],
        ...,
        [252., 247., 244.],
        [251., 246., 243.],
        [251., 246., 243.]],

       [[255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.],
        ...,
        [252., 247., 244.],
        [251., 246., 243.],
        [251., 246., 243.]],

       ...,

       [[255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.],
        ...,
        [254., 253., 251.],
        [254., 253., 251.],
        [254., 253., 251.]],

       [[255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.],
        ...,
        [255., 254., 252.],
        [254., 253., 251.],
        [254., 253., 251.]],

       [[255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.],
        ...,
        [255., 254., 252.],
        [254., 253., 251.],
        [254., 253., 251.]]], dtype=float32)
```

```
img = np.expand_dims(img,axis=0)
img
```

```
array([[[[255., 255., 255.],
         [255., 255., 255.],
         [255., 255., 255.],
         ...,
         [252., 247., 244.],
         [251., 246., 243.],
         [251., 246., 243.]],

        [[255., 255., 255.],
         [255., 255., 255.],
         [255., 255., 255.],
         ...,
         [252., 247., 244.],
         [251., 246., 243.],
         [251., 246., 243.]],

        [[255., 255., 255.],
         [255., 255., 255.],
         [255., 255., 255.],
         ...,
         [252., 247., 244.],
         [251., 246., 243.],
         [251., 246., 243.]],

        ...,
```

```
[[255., 255., 255.],
 [255., 255., 255.],
 [255., 255., 255.],
 ...,
 [254., 253., 251.],
 [254., 253., 251.],
 [254., 253., 251.]],

[[255., 255., 255.],
 [255., 255., 255.],
 [255., 255., 255.],
 ...,
 [255., 254., 252.],
 [254., 253., 251.],
 [254., 253., 251.]],

[[255., 255., 255.],
 [255., 255., 255.],
 [255., 255., 255.],
 ...,
 [255., 254., 252.],
 [254., 253., 251.],
 [254., 253., 251.]]]], dtype=float32)
```

```
np.argmax(model.predict(img))
```

```
1/1 [==============================] - 0s 77ms/step
3
```