In [2]:
```
from tensorflow import keras
import tensorflow as tf

from keras.preprocessing import image
from keras.preprocessing.image import ImageDataGenerator

from keras.models import Model
from keras.models import Sequential
from keras.layers import Input, Lambda, Dense, Flatten

from keras.applications.vgg16 import VGG16
from keras.applications.vgg16 import preprocess_input
```

In [8]:
```
ANIMALS_PATH = r"C:\Users\NehaA\Downloads\project"
IMAGE_SIZE = [224, 224]
ANIMAL_TYPES = 90
BATCH_SIZE = 30
EPOCHS = 15
```

In [9]:
```
from keras.applications.vgg16 import VGG16

AnimalModel = VGG16(input_shape = IMAGE_SIZE + [3], weights='imagenet', include_top=False)

# freeze layers of predefined model.
for layer in AnimalModel.layers:
    layer.trainable = False

# add a flatenning layer and output layer.
FlattenedLayer = Flatten()(AnimalModel.output)
OutputLayer = Dense(ANIMAL_TYPES, activation='softmax')(FlattenedLayer)

AnimalModel = Model(inputs=AnimalModel.input, outputs=OutputLayer)

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

# AnimalModel.summary()
```

In [6]:
```
ImageGen = ImageDataGenerator(rescale=1./255, validation_split = 0.2)

TrainGen = ImageGen.flow_from_directory(
                                                directory=ANIMALS_PATH,
                                                target_size=IMAGE_SIZE,
                                                batch_size=BATCH_SIZE,
                                                class_mode='categorical',
                                                shuffle=False,
                                                subset='training',
                                                interpolation='bicubic',
                                        )

TestGen  = ImageGen.flow_from_directory(
                                                directory=ANIMALS_PATH,
                                                target_size=IMAGE_SIZE,
                                                batch_size=BATCH_SIZE,
                                                class_mode='categorical',
                                                shuffle=False,
                                                subset='validation',
                                         interpolation='bicubic',
                                        )
```

```
Found 4320 images belonging to 1 classes.
Found 1080 images belonging to 1 classes.
```

In [10]:
```
import warnings

# Ignore all warnings
warnings.filterwarnings("ignore")

stats = AnimalModel.fit_generator(
                                        generator = TrainGen,
                                        validation_data = TestGen,
                                        epochs = EPOCHS,
                                        steps_per_epoch= len(TrainGen.filenames)//BATCH_SIZE,
                                        validation_steps=len(TestGen.filenames)//BATCH_SIZE
                                    )
```

```
Epoch 1/15
144/144 [==============================] - 550s 4s/step - loss: 444.0057 - accuracy: 0.0000e+00 - val_loss: 443.7668 - val_accuracy: 0.0000e+00
Epoch 2/15
144/144 [==============================] - 555s 4s/step - loss: 444.1506 - accuracy: 0.0000e+00 - val_loss: 443.3059 - val_accuracy: 0.0000e+00
Epoch 3/15
144/144 [==============================] - 532s 4s/step - loss: 444.2957 - accuracy: 0.0000e+00 - val_loss: 443.5928 - val_accuracy: 0.0000e+00
Epoch 4/15
144/144 [==============================] - 523s 4s/step - loss: 444.4206 - accuracy: 0.0000e+00 - val_loss: 443.1876 - val_accuracy: 0.0000e+00
Epoch 5/15
144/144 [==============================] - 538s 4s/step - loss: 444.5385 - accuracy: 0.0000e+00 - val_loss: 443.7982 - val_accuracy: 0.0000e+00
Epoch 6/15
144/144 [==============================] - 808s 6s/step - loss: 444.6337 - accuracy: 0.0000e+00 - val_loss: 443.5831 - val_accuracy: 0.0000e+00
Epoch 7/15
144/144 [==============================] - 635s 4s/step - loss: 444.9206 - accuracy: 0.0000e+00 - val_loss: 443.6951 - val_accuracy: 0.0000e+00
Epoch 8/15
144/144 [==============================] - 743s 5s/step - loss: 445.0455 - accuracy: 0.0000e+00 - val_loss: 443.4697 - val_accuracy: 0.0000e+00
Epoch 9/15
144/144 [==============================] - 540s 4s/step - loss: 445.2161 - accuracy: 0.0000e+00 - val_loss: 443.7672 - val_accuracy: 0.0000e+00
Epoch 10/15
144/144 [==============================] - 562s 4s/step - loss: 445.5118 - accuracy: 0.0000e+00 - val_loss: 443.4320 - val_accuracy: 0.0000e+00
Epoch 11/15
144/144 [==============================] - 523s 4s/step - loss: 445.7509 - accuracy: 0.0000e+00 - val_loss: 443.5370 - val_accuracy: 0.0000e+00
Epoch 12/15
144/144 [==============================] - 533s 4s/step - loss: 445.9718 - accuracy: 0.0000e+00 - val_loss: 443.6737 - val_accuracy: 0.0000e+00
Epoch 13/15
144/144 [==============================] - 811s 6s/step - loss: 446.3512 - accuracy: 0.0000e+00 - val_loss: 443.7073 - val_accuracy: 0.0000e+00
Epoch 14/15
144/144 [==============================] - 919s 6s/step - loss: 446.6984 - accuracy: 0.0000e+00 - val_loss: 444.0745 - val_accuracy: 0.0000e+00
Epoch 15/15
144/144 [==============================] - 546s 4s/step - loss: 447.1590 - accuracy: 0.0000e+00 - val_loss: 443.7911 - val_accuracy: 0.0000e+00
```

In [ ]:
```
AnimalModel.save("AnimalRecognizer.h5")
```

In [ ]:
```
```

In [ ]:
```
```