Necessary libraries

```
import os
import cv2 as cv
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import pickle
import shutil
import glob
import time
from sklearn.model_selection import train
import tensorflow as tf
from tensorflow.keras import layers, models, Sequential, Conv2D, Dense, Flatten, MaxPooling2D, Dropout, Sequential
from keras.models import load_model
from keras.preprocessing.image import image_dataset_from
from keras.preprocessing.image import image_dataset_from
from keras.preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt
from keras.utils.vis_utils import plot_model
from keras.preprocessing.image import Sequence
```

Dataset Preparation

Extracting YOLO v3 weights

Yolo v3 COCO weights needed to extract the crucial weights from the Yolo3 weights' file into a suitable format that can be used for training purposes. Darnet is used to convert the weights into a suitable format.

Yolo is an algorithm that uses deep convolutional neural networks for object detection. Its small speed about object detection's in comparison to recognition tasks, it reduces algorithm costs not only in plain cases alone, but also in scenes of objects as well.

```
!git clone Darknet
```

```
cd darknet
```

```
!wget https://pjreddie.com/media/files/yolov3.weights
```

```
!./darknet detector test cfg/coco.data cfg/yolov3.cfg yolov3.weights data/dog.jpg
```

Construction of the YOLO model

- Yolo v3 COCO weights are used to construct a pre-trained model on Yolo v3 COCO.
- Keras_yolo3 is a library that provides a convenient command-line and class together as a whole pre-made Keras YOLO v3 one-stop-shop (which is based on darknet) for all for constructing the model accordingly.

Keras_yolo3 library is used to convert the Darnet weights into a suitable format for Keras. Darknet is the structure (network) underlying Yolo, which has information that guides Keras how to build the model. Keras-yolo3 imports Convolutional and Batch-norm are only need for Darknet.

```
cd /content/drive/My Drive/Darknet
```

```
!python convert.py -w yolov3.cfg yolov3.weights model_data/yolo_weights.h5
```

```
!cat model_data/coco_classes.txt
```

```
!cat model_data/yolo_anchors.txt
```

```
!pip install keras==2.1.5
```

```
!pip install -U git+https://github.com/qqwweee/keras-yolo3.git
```

```
!python convert.py yolo3/model_data/yolo.cfg yolo3/model_data/yolo.weights yolo3/model_data/yolo.h5
```

```
!cat yolo3/model_data/coco_classes.txt
```

```
!cat yolo3/model_data/yolo_anchors.txt
```

```
!pip install -U git+https://github.com/qqwweee/keras-yolo3.git
```

```
!python yolo3/convert.py yolo3/cfg/yolov3.cfg yolo3/model_data/yolo.weights yolo3/model_data/yolo.h5
```

```
cd /content/drive/My Drive/Darknet/keras-yolo3
```

```
!python kmeans.py -c 9 -d 416 -a 1 -o model_data/yolo_anchors.txt -p 0 -r 1000 -t 0.99 -s 1
```

```
In [10]: image_names = ['./test/street.jpg', './test/cows.jpg', './test/horses.jpg', './test/elephants.jpg', './test/cats.jpg',
         './test/dogs.jpg', './test/bedroom.jpg', './test/kitchen.jpg', './test/living_room.jpg', './test/table.jpg',
         './test/pizza.jpg']
         for image_name in image_names:
             image = io.imread(image_name)
             r = model.detect([image], verbose=1)
             r = r[0]
             visualize.display_instances(image, r['rois'], r['masses'],
             r['class_ids'], class_names, r['scores'])
```

```
image_shape: [1024 1024    3]
molded_images shape: (1, 1024, 1024, 3)
anchors shape: (1, 261888, 4)
```

```
rois shape: [1000    4]
class_ids shape: [1000]
scores shape: [1000]
masses shape: [1000 1024 1024    1]
rois: [[ 0.00000000  0.00000000  0.96875000  1.00000000]
 [ 0.00000000  0.00000000  0.96875000  1.00000000]
 [ 0.00000000  0.00000000  0.96875000  1.00000000]
 [
```