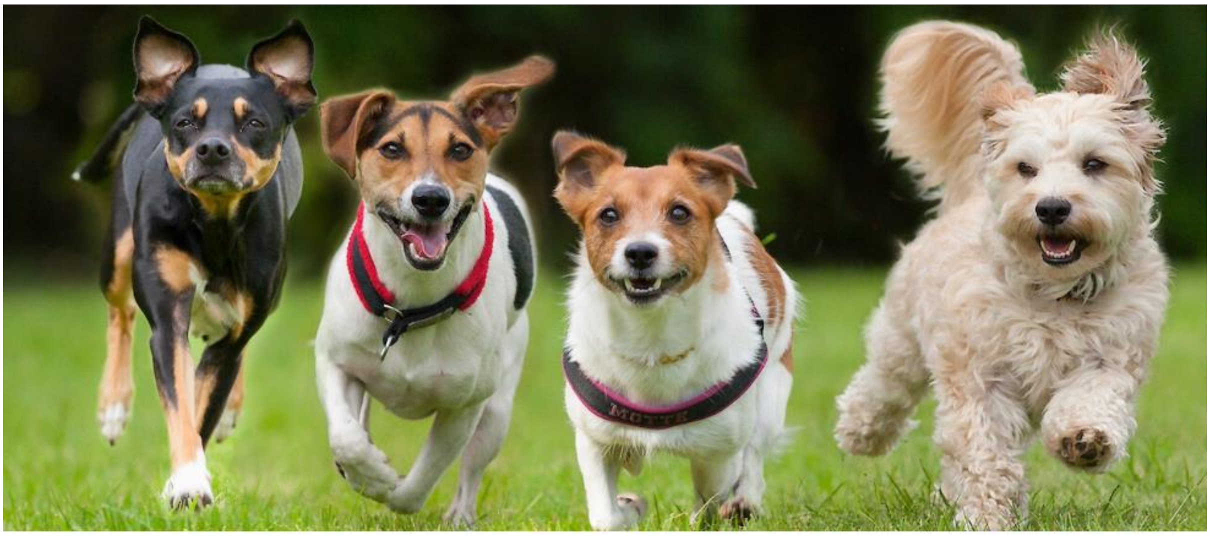
Number of Objects and Area Calculation

1. Step 1: Importing Python Libraries

```
import cv2 as cv
import matplotlib.pyplot as plt
import cvlib as cvl
from numpy.lib.polynomial import poly
from cvlib.object_detection import draw_bbox
from google.colab.patches import cv2_imshow
```

2. Step 2: Importing Image from Google Drive

```
image = cv.imread("/content/drive/MyDrive/SoC Image Processing and
Object Detection/Hero Dog Pedigrees.webp")
cv2_imshow(image)
```



3. Defining function to detect number of objects and labels using model Yolov4 with confidence of 0.5 and threshold of 0.3

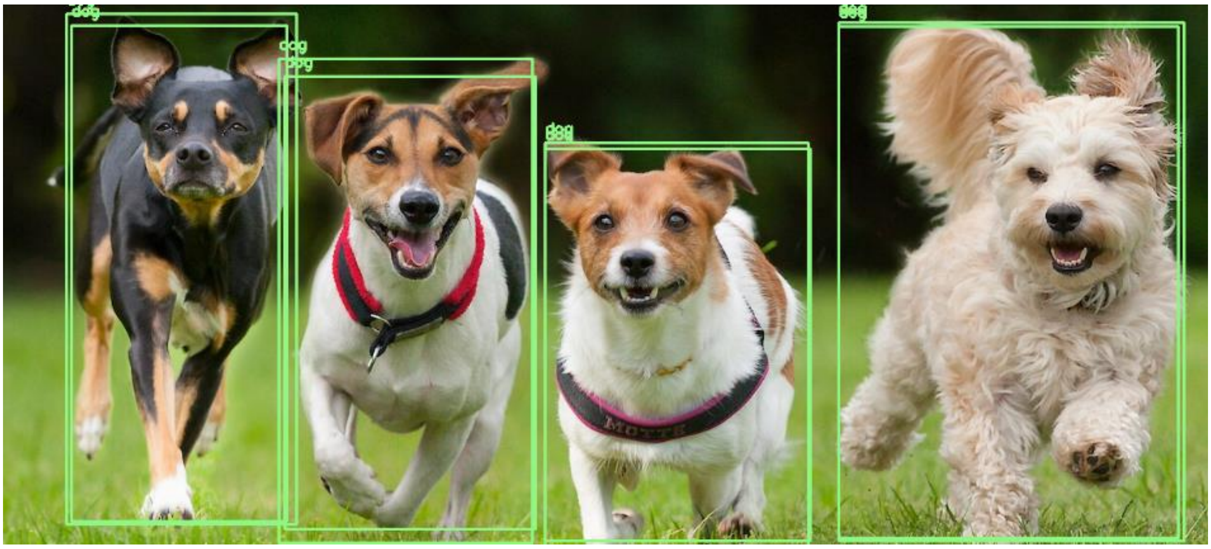
```
def count_object(image):
box, labels, count = cvl.detect_common_objects(image, confidence=0.5,
nms_thresh=0.3, model='yolov4', enable_gpu=False)
output = draw_bbox(image, box, labels, count)
print(r'Number of objects :', len(labels))
print('Labels of the object : ', labels)
cv2_imshow(output)
```

Number of Objects and Area Calculation

4. Calling the function to get the output as number of objects and label of the objects

```
count_object(image)
```

```
Number of objects : 4
Labels of the object :  ['dog', 'dog', 'dog', 'dog']
```



5. Computing the area of the objects in an image by measuring the area of an object in a given image by converting it to grayscale, applying inverse binary thresholding, finding contours, calculating the contour areas, and displaying the binary image.

```
def object_area(image_path):
    # Load the image
    image = cv.imread(image_path)

    # Converting the image to grayscale
    gray = cv.cvtColor(image, cv.COLOR_BGR2GRAY)

    # Applying inverse binary thresholding to obtain a inversed binary
image using otsu thresholding method
    _, binary = cv.threshold(gray, 0, 255, cv.THRESH_BINARY_INV |
cv.THRESH_OTSU)

    # Finding contours in the binary image
    contours, hierarchies = cv.findContours(binary, cv.RETR_EXTERNAL,
cv.CHAIN_APPROX_SIMPLE)

    total_area = 0
    # Iterating through the whole area
    for contour in contours:
        # Calculate the area of each contour
        area = cv.contourArea(contour)
        total_area += area

    # Printing the measured area
    print("Object area:", total_area)
    cv2_imshow(binary)

# Providing the path of the image
```

Number of Objects and Area Calculation

```
image_path = "/content/drive/MyDrive/SoC Image Processing and Object
Detection/Hero Dog Pedigrees.webp"

# Calling the function to measure the area of the object in the image
      object_area(image_path)
```

```
Object area: 328658.5
```