



Simple Image Detection & Segmentation using Mask RCNN

[Machine Learning Mastery Tutorial](#)

adapted and annotated by Kayla Neal for Pearce Lab



Imports

- ◎ Tensorflow is used as a backend for keras
 - `import tensorflow as tf`
 - `from tensorflow.keras.preprocessing.image import load_img, img_to_array`
- ◎ The [Mask RCNN model](#) can be downloaded from github – not tf2 compatible, [an edited version](#) is.
 - `from Mask_RCNN.mrcnn.config import Config`
 - `from mrcnn import visualize`
 - `from mrcnn.model import MaskRCNN`
- ◎ matplotlib is also used
 - `from matplotlib import pyplot`
 - `from matplotlib.patches import Rectangle`

Define Configuration

- Modify the existing configuration given in config.py of the mrcnn repository
- NUM_CLASSES is defined to be the 80 classes the pretrained coco dataset is trained on PLUS the background
NAME has to be defined

```
# define the test configuration  
class PredictionConfig(Config):  
    NAME = "test"  
    NUM_CLASSES = 80 + 1  
    GPU_COUNT = 1  
    IMAGES_PER_GPU = 1  
  
config = PredictionConfig()
```

create an instance of the new configuration here to use in the compilation later

Bounding Boxes

- ◎ The mrcnn model defines a `display_instances()` function that is used to view the classification, bounding box, and segmentation
- ◎ Make a function that just displays the bounding box:

```
def draw_image_with_boxes(filename, boxes_list):  
    data = pyplot.imread(filename)  
    pyplot.imshow(data)  
    ax = pyplot.gca()  
  
    for box in boxes_list:  
        y1, x1, y2, x2 = box  
        width, height = x2-x1, y2-y1  
        rect = Rectangle((x1,y1), width, height,  
                        fill=False, color='red')  
        ax.add_patch(rect)  
    pyplot.show()
```

Compile Model and Load Weights

- ◎ `rcnn = MaskRCNN(mode = 'inference',
model_dir = './', config = config)`
 - 'inference' for testing, 'training' for training
- ◎ `rcnn.load_weights('mask_rcnn_coco.h5', by_name = True)`
 - The coco dataset is available in the mrcnn repository
 - Contains the weights trained with this data

Class Names for Classification

- Since the coco dataset is being used, a class_names array is created based on the categories coco was trained on and the background:

```
class_names = ['BG', 'person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus',  
               'train', 'truck', 'boat', 'traffic light', 'fire hydrant', 'stop sign',  
               'parking meter', 'bench', 'bird', 'cat', 'dog', 'horse', 'sheep', 'cow',  
               'elephant', 'bear', 'zebra', 'giraffe', 'backpack', 'umbrella',  
               'handbag', 'tie', 'suitcase', 'frisbee', 'skis', 'snowboard',  
               'sports ball', 'kite', 'baseball bat', 'baseball glove', 'skateboard',  
               'surfboard', 'tennis racket', 'bottle', 'wine glass', 'cup', 'fork', 'knife',  
               'spoon', 'bowl', 'banana', 'apple', 'sandwich', 'orange', 'broccoli',  
               'carrot', 'hot dog', 'pizza', 'donut', 'cake', 'chair', 'couch',  
               'potted plant', 'bed', 'dining table', 'toilet', 'tv', 'laptop', 'mouse',  
               'remote', 'keyboard', 'cell phone', 'microwave', 'oven', 'toaster', 'sink',  
               'refrigerator', 'book', 'clock', 'vase', 'scissors', 'teddy bear', 'hair drier',  
               'toothbrush']
```

Predictions

```
img = load_img('elephant.jpeg')
img = img_to_array(img)
results = rcnn.detect([img], verbose = 0)
draw_image_with_boxes('elephant.jpeg', results[0]['rois'])
r = results[0]
display_instances(img, r['rois'], r['masks'], r['class_ids'], class_names, r['scores'])
```

draw_image_with_boxes
will display ONLY the
bounding box overlaid on
the image

display_instances is from
mrcnn.visualize
shows the mask, bounding
box, and classification with
confidence on the image

- ◎ 'elephant.jpeg' is an added image in the working directory
- ◎ rcnn.detect is similar to model.predict from keras
- ◎ r = results[0] is being used as a shortcut

