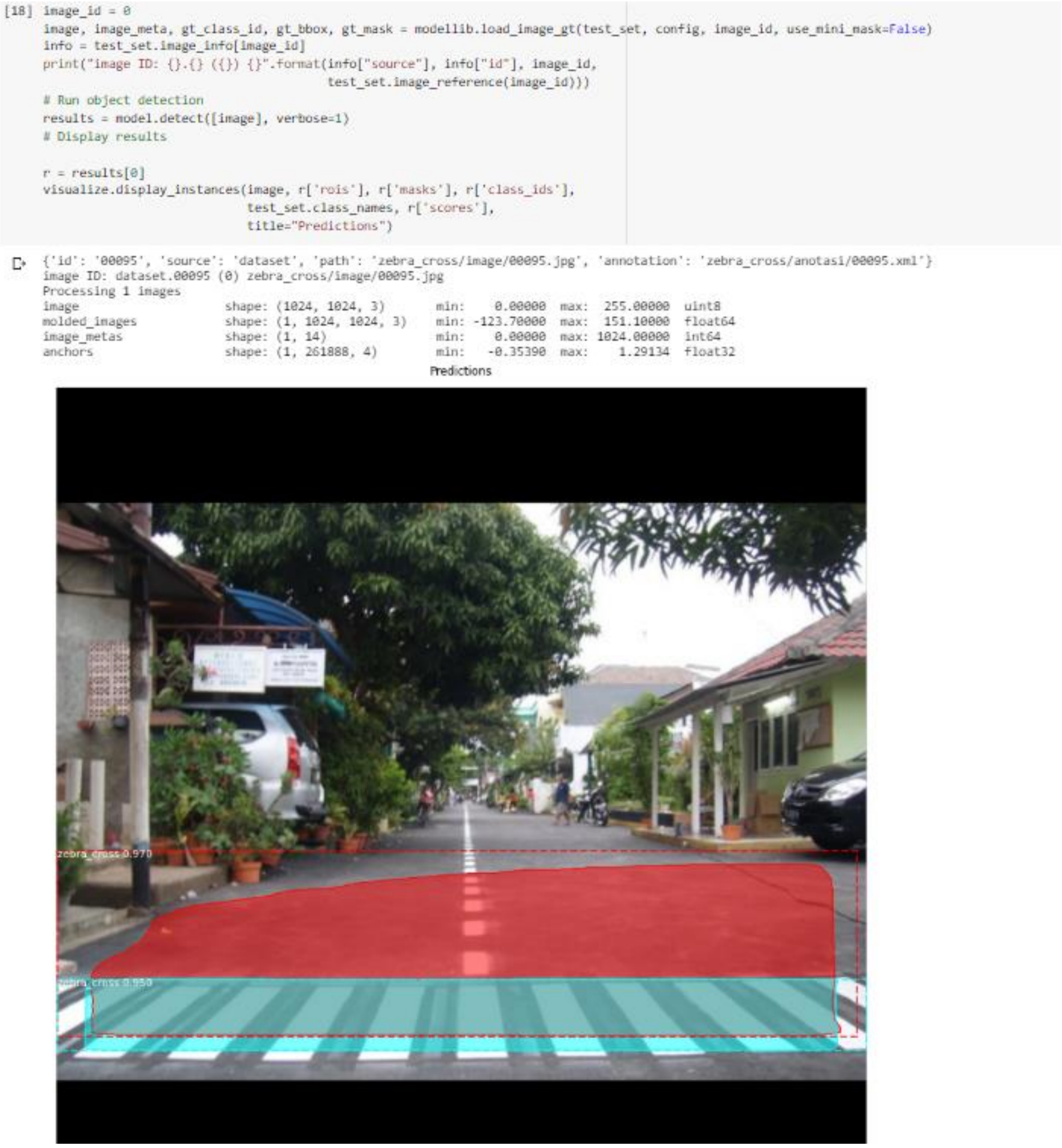
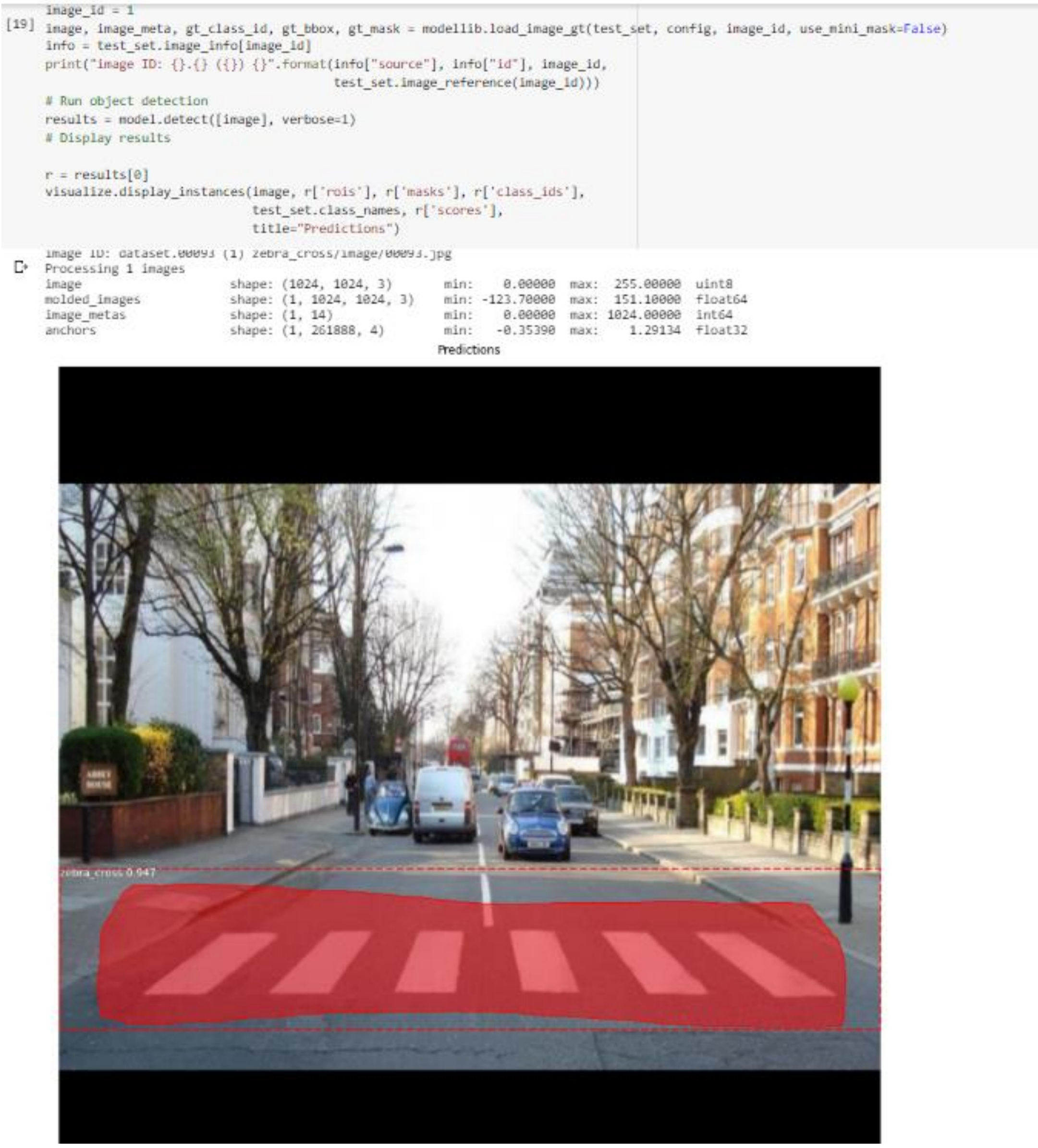


```
[18] image_id = 0
     image, image_meta, gt_class_id, gt_bbox, gt_mask = modellib.load_image_gt(test_set, config, image_id, use_mini_mask=False)
     info = test_set.image_info[image_id]
     print("image ID: {}.{} ({}) {}".format(info["source"], info["id"], image_id,
                                            test_set.image_reference(image_id)))
     # Run object detection
     results = model.detect([image], verbose=1)
     # Display results

     r = results[0]
     visualize.display_instances(image, r['rois'], r['masks'], r['class_ids'],
                                 test_set.class_names, r['scores'],
                                 title="Predictions")
```

```
{'id': '00095', 'source': 'dataset', 'path': 'zebra_cross/image/00095.jpg', 'annotation': 'zebra_cross/anotasi/00095.xml'}
image ID: dataset.00095 (0) zebra_cross/image/00095.jpg
Processing 1 images
image                    shape: (1024, 1024, 3)       min:    0.00000  max:  255.00000  uint8
molded_images            shape: (1, 1024, 1024, 3)    min: -123.70000  max:  151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max: 1024.00000  int64
anchors                  shape: (1, 261888, 4)        min:   -0.35390  max:    1.29134  float32
```

Predictions

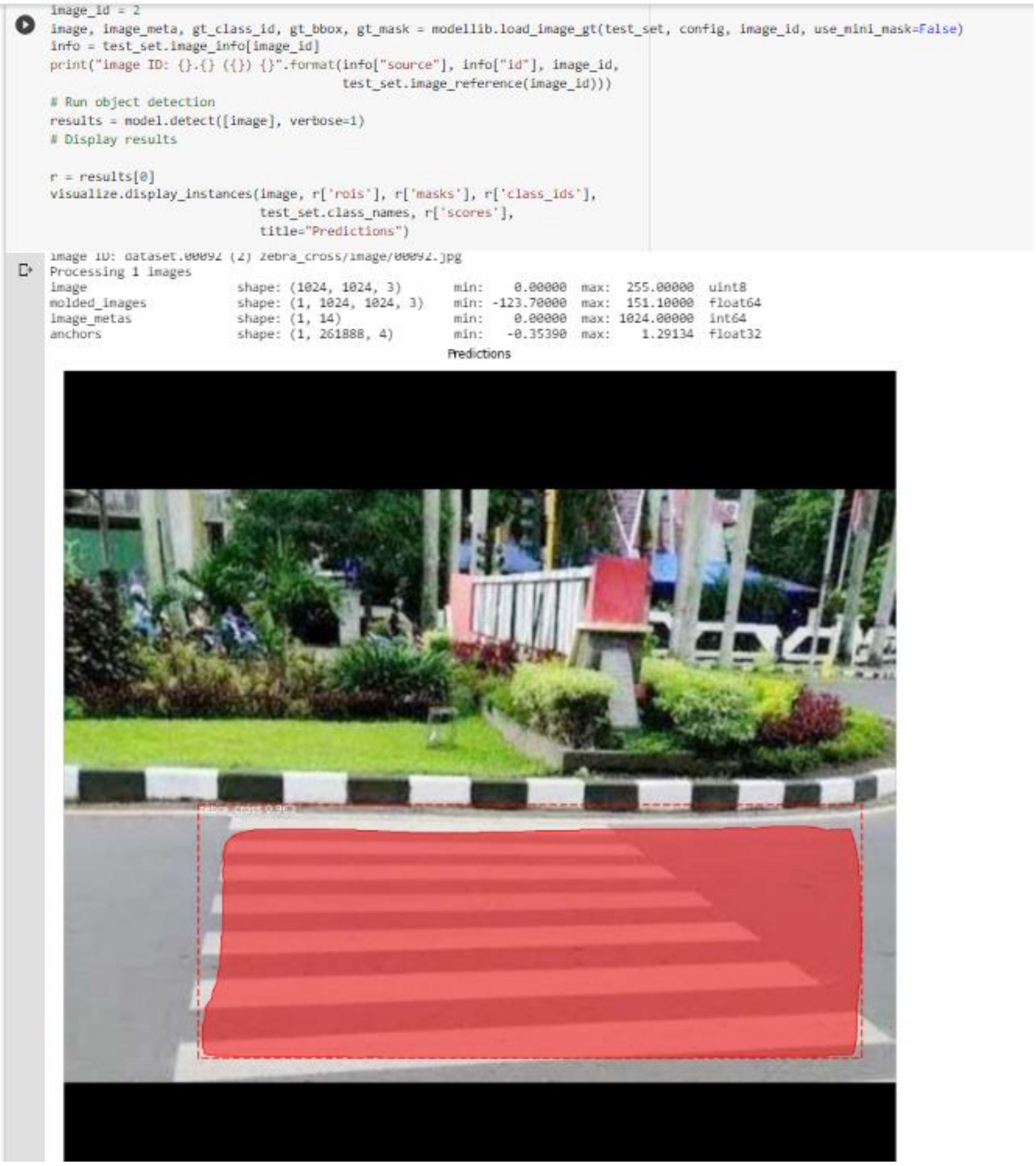


```
[19] image_id = 1
     image, image_meta, gt_class_id, gt_bbox, gt_mask = modellib.load_image_gt(test_set, config, image_id, use_mini_mask=False)
     info = test_set.image_info[image_id]
     print("image ID: {}.{} ({}) {}".format(info["source"], info["id"], image_id,
                                            test_set.image_reference(image_id)))
     # Run object detection
     results = model.detect([image], verbose=1)
     # Display results

     r = results[0]
     visualize.display_instances(image, r['rois'], r['masks'], r['class_ids'],
                                 test_set.class_names, r['scores'],
                                 title="Predictions")
```

```
image ID: dataset.00093 (1) zebra_cross/image/00093.jpg
Processing 1 images
image                    shape: (1024, 1024, 3)       min:    0.00000  max:  255.00000  uint8
molded_images            shape: (1, 1024, 1024, 3)    min: -123.70000  max:  151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max: 1024.00000  int64
anchors                  shape: (1, 261888, 4)        min:   -0.35390  max:    1.29134  float32
```

Predictions

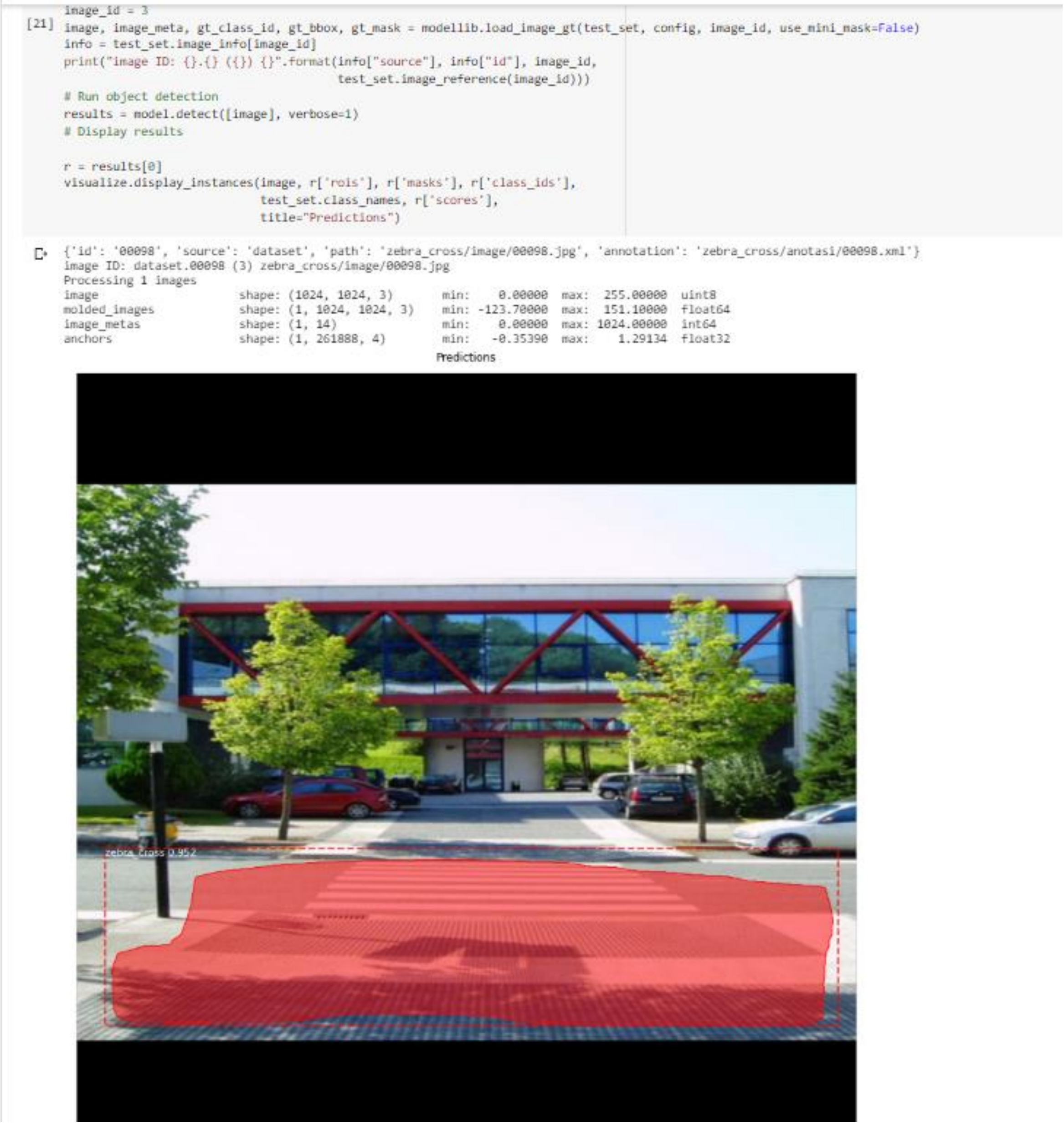


```
image_id = 2
image, image_meta, gt_class_id, gt_bbox, gt_mask = modellib.load_image_gt(test_set, config, image_id, use_mini_mask=False)
info = test_set.image_info[image_id]
print("image ID: {}.{} ({}) {}".format(info["source"], info["id"], image_id,
                                       test_set.image_reference(image_id)))
# Run object detection
results = model.detect([image], verbose=1)
# Display results

r = results[0]
visualize.display_instances(image, r['rois'], r['masks'], r['class_ids'],
                            test_set.class_names, r['scores'],
                            title="Predictions")
```

```
image ID: dataset.00092 (2) zebra_cross/image/00092.jpg
Processing 1 images
image                    shape: (1024, 1024, 3)       min:    0.00000  max:  255.00000  uint8
molded_images            shape: (1, 1024, 1024, 3)    min: -123.70000  max:  151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max: 1024.00000  int64
anchors                  shape: (1, 261888, 4)        min:   -0.35390  max:    1.29134  float32
```

Predictions



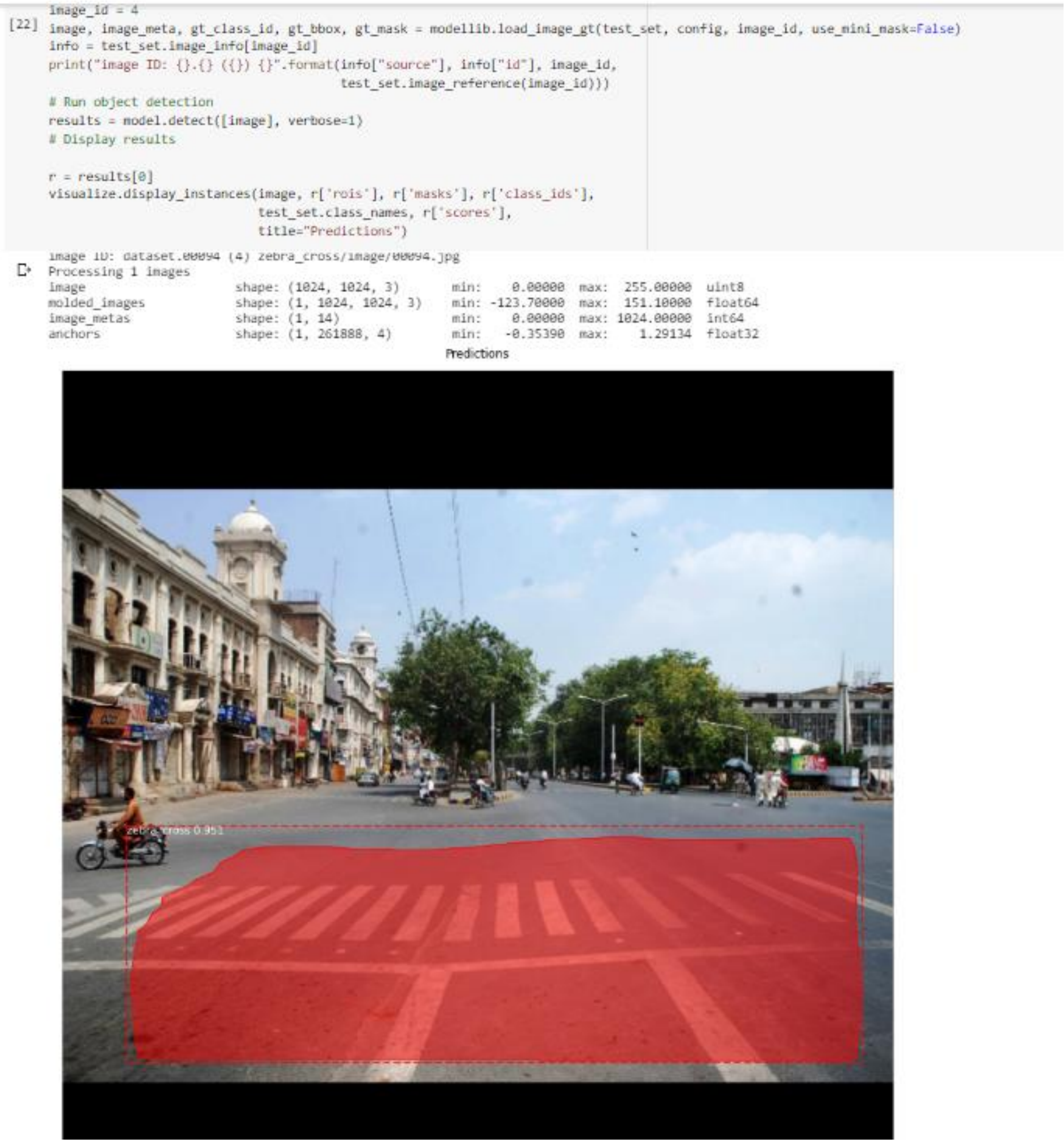
[21]

```
image_id = 3
image, image_meta, gt_class_id, gt_bbox, gt_mask = modellib.load_image_gt(test_set, config, image_id, use_mini_mask=False)
info = test_set.image_info[image_id]
print("image ID: {}.{} ({}) {}".format(info["source"], info["id"], image_id,
                                       test_set.image_reference(image_id)))
# Run object detection
results = model.detect([image], verbose=1)
# Display results

r = results[0]
visualize.display_instances(image, r['rois'], r['masks'], r['class_ids'],
                            test_set.class_names, r['scores'],
                            title="Predictions")
```

```
{'id': '00098', 'source': 'dataset', 'path': 'zebra_cross/image/00098.jpg', 'annotation': 'zebra_cross/anotasi/00098.xml'}
image ID: dataset.00098 (3) zebra_cross/image/00098.jpg
Processing 1 images
image                    shape: (1024, 1024, 3)       min:    0.00000  max:  255.00000  uint8
molded_images            shape: (1, 1024, 1024, 3)    min: -123.70000  max:  151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max: 1024.00000  int64
anchors                  shape: (1, 261888, 4)        min:   -0.35390  max:    1.29134  float32
```

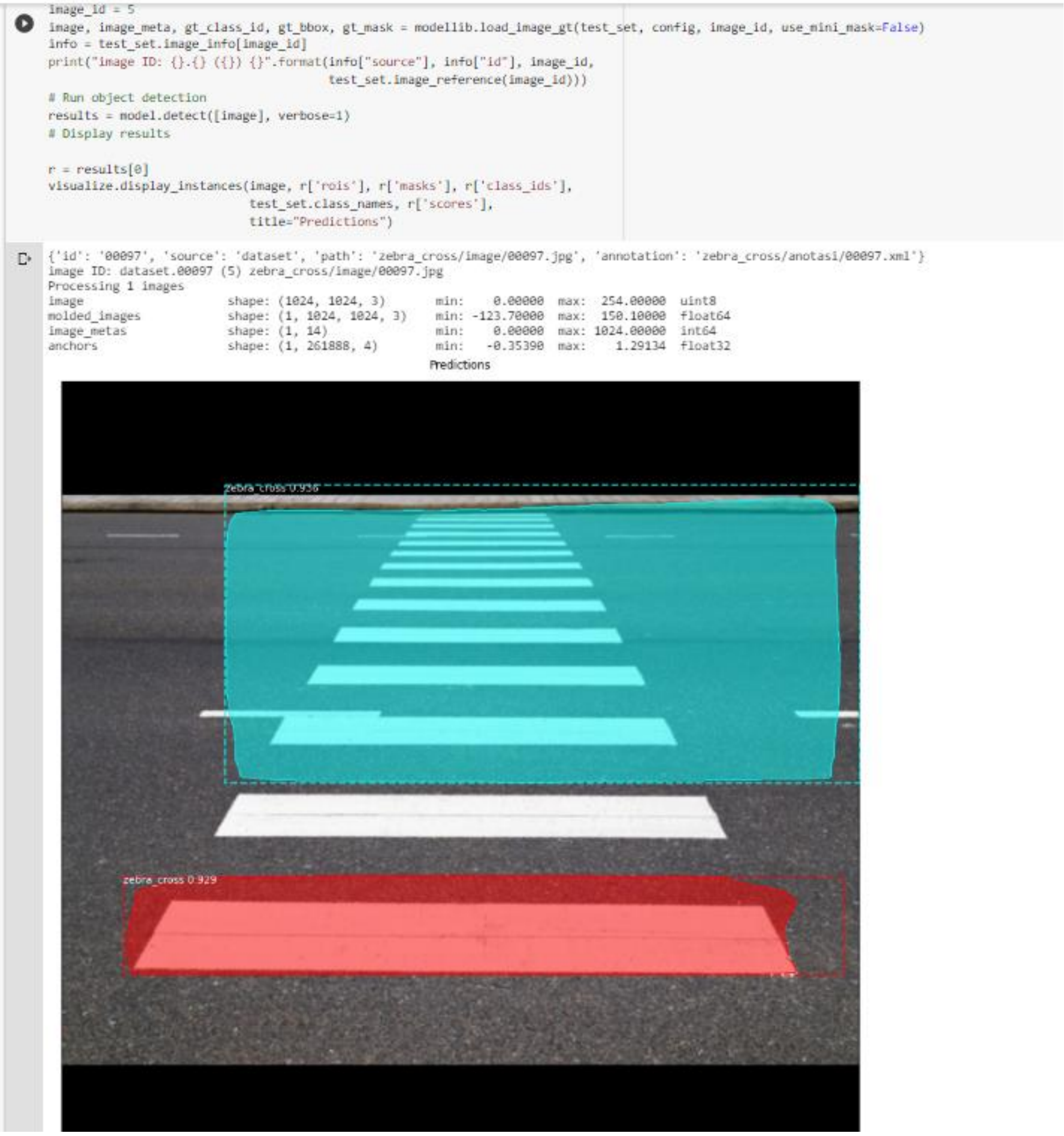
Predictions



```
    image_id = 4
[22] image, image_meta, gt_class_id, gt_bbox, gt_mask = modellib.load_image_gt(test_set, config, image_id, use_mini_mask=False)
    info = test_set.image_info[image_id]
    print("image ID: {}.{} ({}) {}".format(info["source"], info["id"], image_id,
                                           test_set.image_reference(image_id)))
    # Run object detection
    results = model.detect([image], verbose=1)
    # Display results

    r = results[0]
    visualize.display_instances(image, r['rois'], r['masks'], r['class_ids'],
                                test_set.class_names, r['scores'],
                                title="Predictions")
```

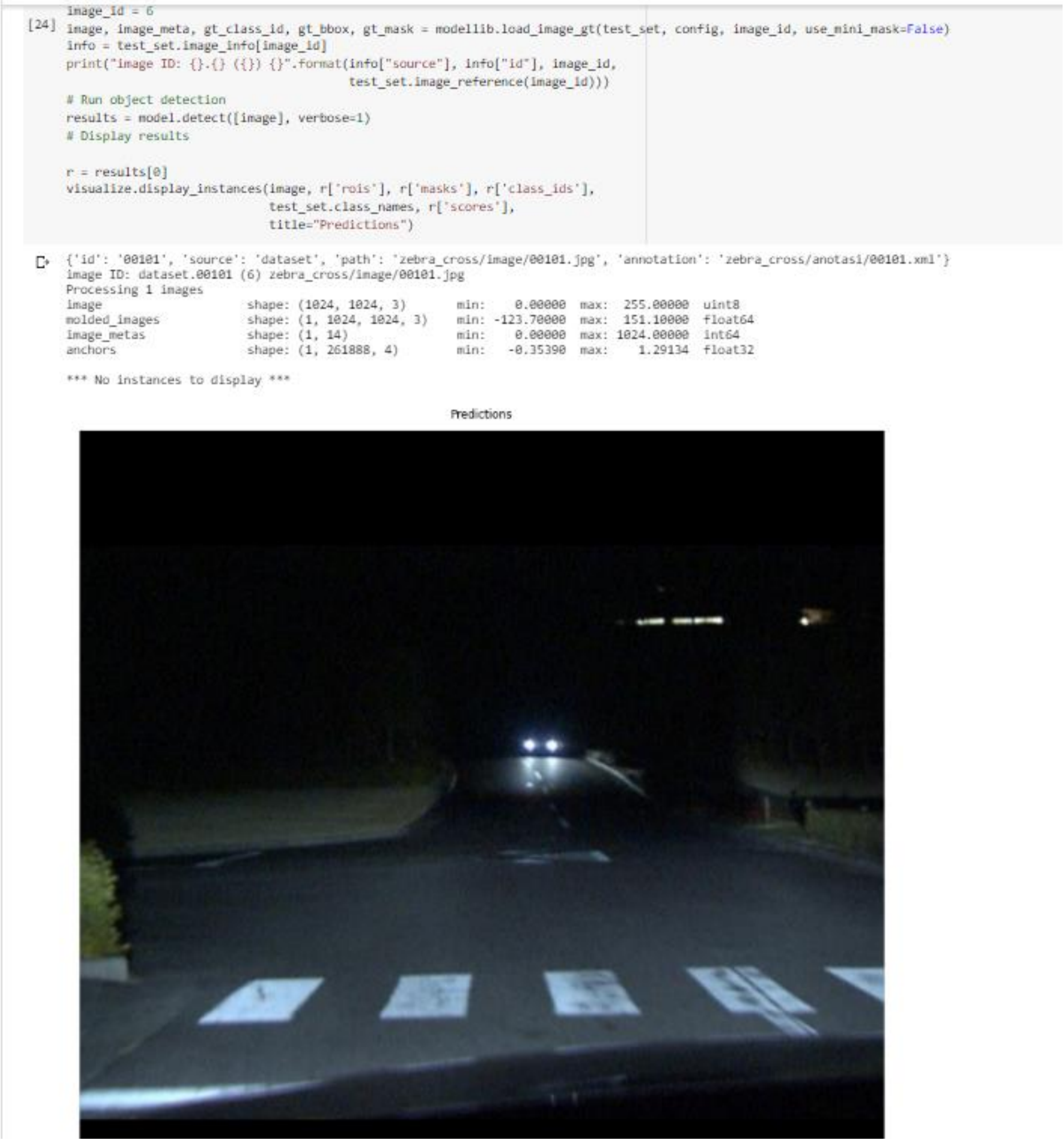
```
image ID: dataset.00094 (4) zebra_cross/image/00094.jpg
Processing 1 images
image                    shape: (1024, 1024, 3)       min:    0.00000  max:  255.00000  uint8
molded_images            shape: (1, 1024, 1024, 3)    min: -123.70000  max:  151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max: 1024.00000  int64
anchors                  shape: (1, 261888, 4)        min:   -0.35390  max:    1.29134  float32
```



```
image_id = 5
image, image_meta, gt_class_id, gt_bbox, gt_mask = modellib.load_image_gt(test_set, config, image_id, use_mini_mask=False)
info = test_set.image_info[image_id]
print("image ID: {}.{} ({}) {}".format(info["source"], info["id"], image_id,
                                       test_set.image_reference(image_id)))
# Run object detection
results = model.detect([image], verbose=1)
# Display results

r = results[0]
visualize.display_instances(image, r['rois'], r['masks'], r['class_ids'],
                            test_set.class_names, r['scores'],
                            title="Predictions")
```

```
{'id': '00097', 'source': 'dataset', 'path': 'zebra_cross/image/00097.jpg', 'annotation': 'zebra_cross/anotasi/00097.xml'}
image ID: dataset.00097 (5) zebra_cross/image/00097.jpg
Processing 1 images
image                    shape: (1024, 1024, 3)       min:    0.00000  max:  254.00000  uint8
molded_images            shape: (1, 1024, 1024, 3)    min: -123.70000  max:  150.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max: 1024.00000  int64
anchors                  shape: (1, 261888, 4)        min:   -0.35390  max:    1.29134  float32
```



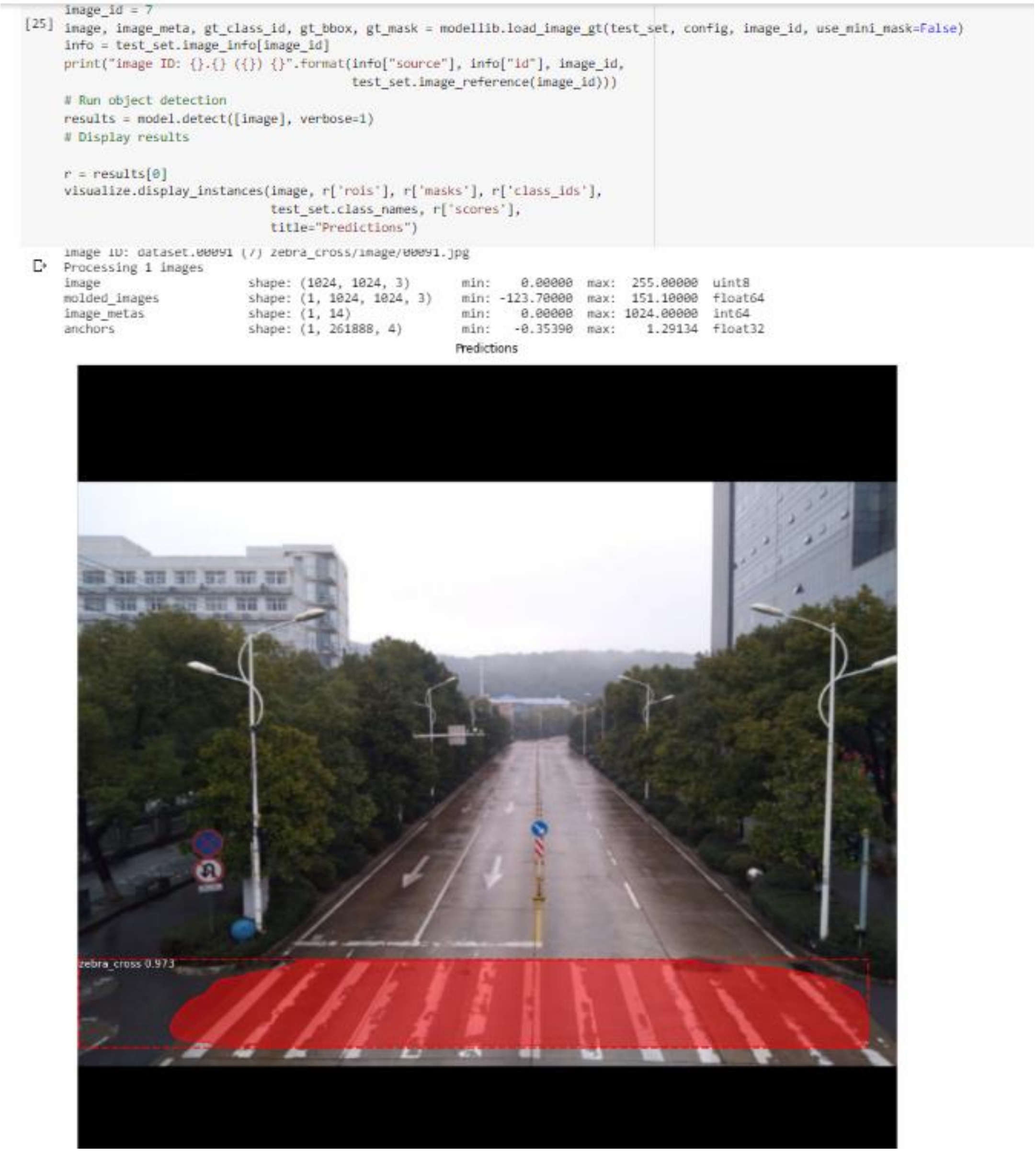
```
[24] image_id = 6
     image, image_meta, gt_class_id, gt_bbox, gt_mask = modellib.load_image_gt(test_set, config, image_id, use_mini_mask=False)
     info = test_set.image_info[image_id]
     print("image ID: {}.{} ({}) {}".format(info["source"], info["id"], image_id,
                                            test_set.image_reference(image_id)))
     # Run object detection
     results = model.detect([image], verbose=1)
     # Display results

     r = results[0]
     visualize.display_instances(image, r['rois'], r['masks'], r['class_ids'],
                                 test_set.class_names, r['scores'],
                                 title="Predictions")
```

```
{'id': '00101', 'source': 'dataset', 'path': 'zebra_cross/image/00101.jpg', 'annotation': 'zebra_cross/anotasi/00101.xml'}
image ID: dataset.00101 (6) zebra_cross/image/00101.jpg
Processing 1 images
image                    shape: (1024, 1024, 3)       min:    0.00000  max:  255.00000  uint8
molded_images            shape: (1, 1024, 1024, 3)    min: -123.70000  max:  151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max: 1024.00000  int64
anchors                  shape: (1, 261888, 4)        min:   -0.35390  max:    1.29134  float32

*** No instances to display ***
```

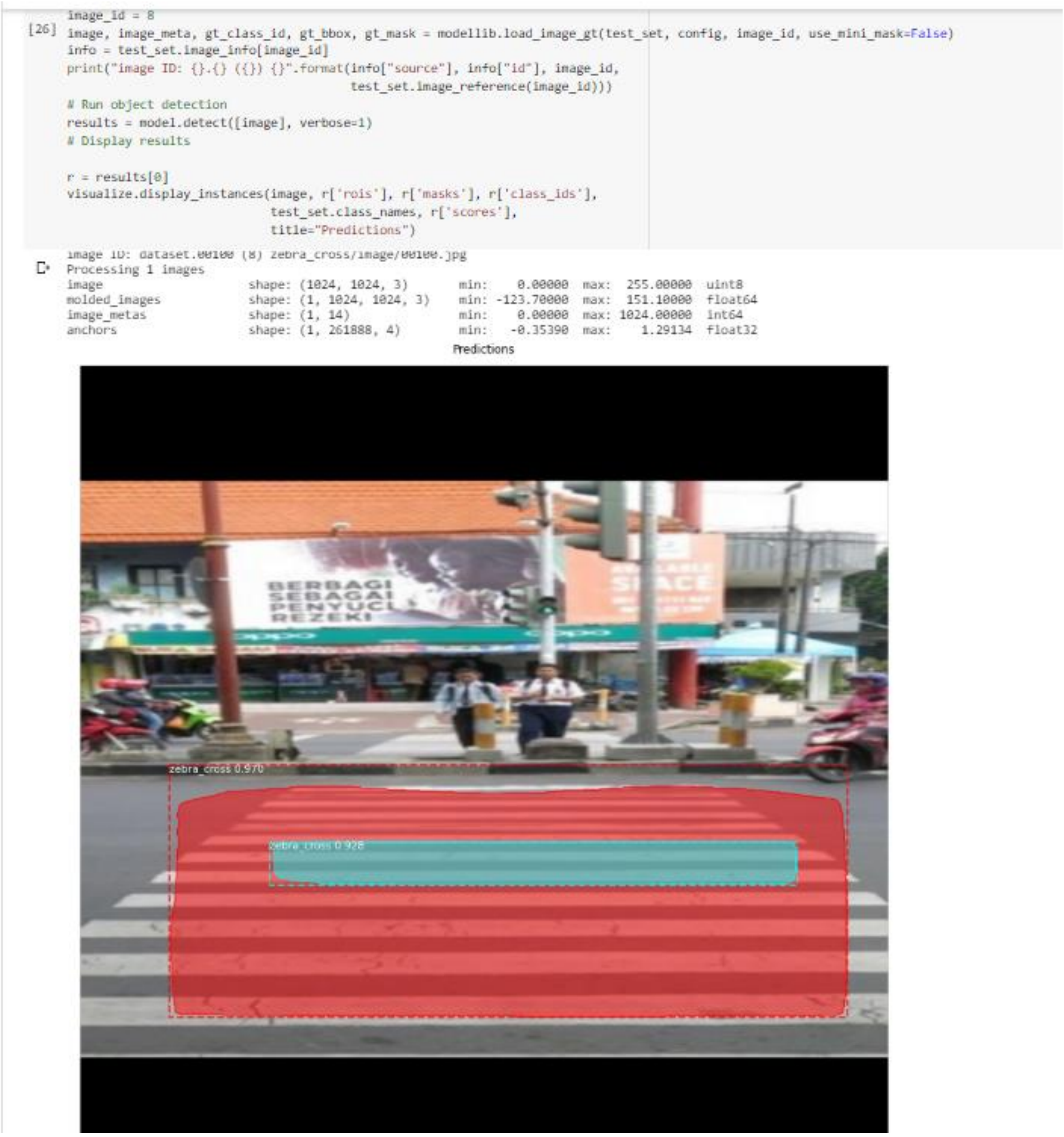
Predictions



```
[25] image_id = 7
     image, image_meta, gt_class_id, gt_bbox, gt_mask = modellib.load_image_gt(test_set, config, image_id, use_mini_mask=False)
     info = test_set.image_info[image_id]
     print("image ID: {}.{} ({}) {}".format(info["source"], info["id"], image_id,
                                            test_set.image_reference(image_id)))
     # Run object detection
     results = model.detect([image], verbose=1)
     # Display results

     r = results[0]
     visualize.display_instances(image, r['rois'], r['masks'], r['class_ids'],
                                 test_set.class_names, r['scores'],
                                 title="Predictions")
```

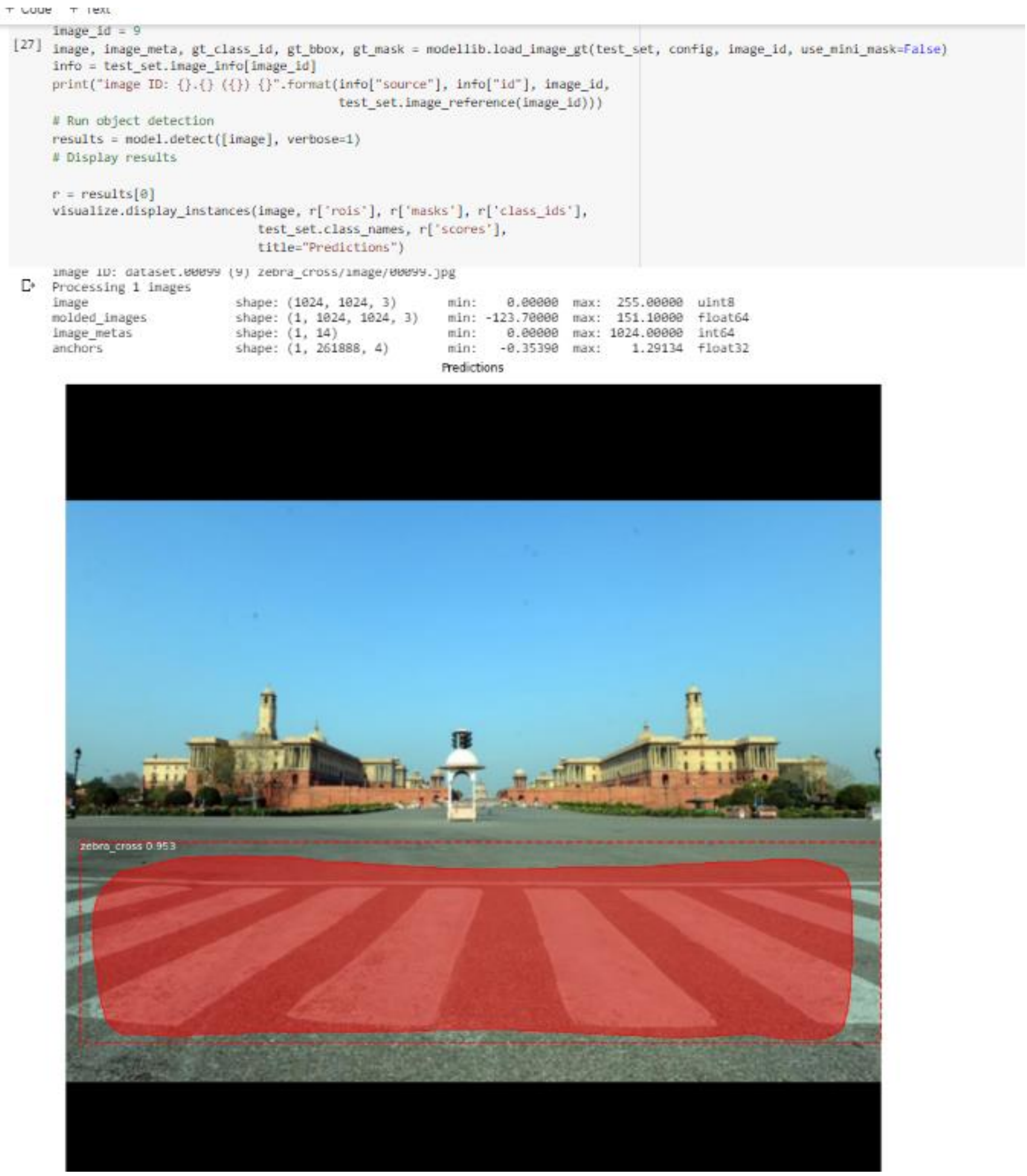
```
image ID: dataset.00091 (7) zebra_cross/image/00091.jpg
Processing 1 images
image                    shape: (1024, 1024, 3)       min:    0.00000  max:  255.00000  uint8
molded_images            shape: (1, 1024, 1024, 3)    min: -123.70000  max:  151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max: 1024.00000  int64
anchors                  shape: (1, 261888, 4)        min:   -0.35390  max:    1.29134  float32
```



```
    image_id = 8
[26] image, image_meta, gt_class_id, gt_bbox, gt_mask = modellib.load_image_gt(test_set, config, image_id, use_mini_mask=False)
    info = test_set.image_info[image_id]
    print("image ID: {}.{} ({}) {}".format(info["source"], info["id"], image_id,
                                           test_set.image_reference(image_id)))
    # Run object detection
    results = model.detect([image], verbose=1)
    # Display results

    r = results[0]
    visualize.display_instances(image, r['rois'], r['masks'], r['class_ids'],
                                test_set.class_names, r['scores'],
                                title="Predictions")
```

```
image ID: dataset.00100 (8) zebra_cross/image/00100.jpg
Processing 1 images
image                    shape: (1024, 1024, 3)       min:    0.00000  max:  255.00000  uint8
molded_images            shape: (1, 1024, 1024, 3)    min: -123.70000  max:  151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max: 1024.00000  int64
anchors                  shape: (1, 261888, 4)        min:   -0.35390  max:    1.29134  float32
```

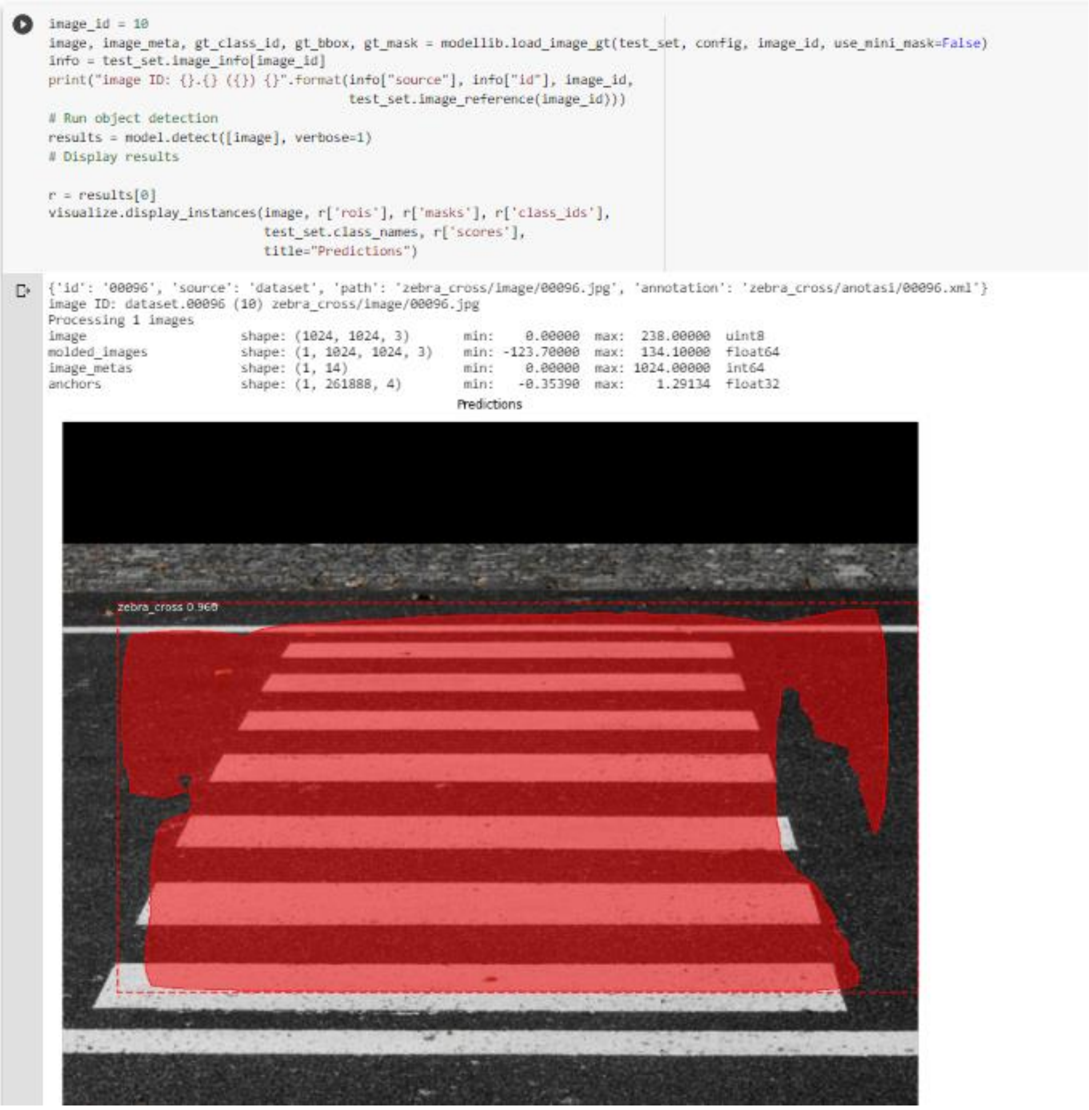


```
[27] image_id = 9
     image, image_meta, gt_class_id, gt_bbox, gt_mask = modellib.load_image_gt(test_set, config, image_id, use_mini_mask=False)
     info = test_set.image_info[image_id]
     print("image ID: {}.{} ({}) {}".format(info["source"], info["id"], image_id,
                                            test_set.image_reference(image_id)))
     # Run object detection
     results = model.detect([image], verbose=1)
     # Display results

     r = results[0]
     visualize.display_instances(image, r['rois'], r['masks'], r['class_ids'],
                                 test_set.class_names, r['scores'],
                                 title="Predictions")
```

```
image ID: dataset.00099 (9) zebra_cross/image/00099.jpg
Processing 1 images
image                    shape: (1024, 1024, 3)       min:    0.00000  max:  255.00000  uint8
molded_images            shape: (1, 1024, 1024, 3)    min: -123.70000  max:  151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max: 1024.00000  int64
anchors                  shape: (1, 261888, 4)        min:   -0.35390  max:    1.29134  float32
```

Predictions



```
image_id = 10
image, image_meta, gt_class_id, gt_bbox, gt_mask = modellib.load_image_gt(test_set, config, image_id, use_mini_mask=False)
info = test_set.image_info[image_id]
print("image ID: {}.{} ({}) {}".format(info["source"], info["id"], image_id,
                                       test_set.image_reference(image_id)))
# Run object detection
results = model.detect([image], verbose=1)
# Display results

r = results[0]
visualize.display_instances(image, r['rois'], r['masks'], r['class_ids'],
                            test_set.class_names, r['scores'],
                            title="Predictions")
```

```
{'id': '00096', 'source': 'dataset', 'path': 'zebra_cross/image/00096.jpg', 'annotation': 'zebra_cross/anotasi/00096.xml'}
image ID: dataset.00096 (10) zebra_cross/image/00096.jpg
Processing 1 images
image                    shape: (1024, 1024, 3)       min:    0.00000  max:  238.00000  uint8
molded_images            shape: (1, 1024, 1024, 3)    min: -123.70000  max:  134.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max: 1024.00000  int64
anchors                  shape: (1, 261888, 4)        min:   -0.35390  max:    1.29134  float32
```