```
[19] 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': '00149', 'source': 'dataset', 'path': 'zebra_cross/images/00149.jpg', 'annotation': 'zebra_cross/annotation/00149.xml'}
image ID: dataset.00149 (0) zebra_cross/images/00149.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
```

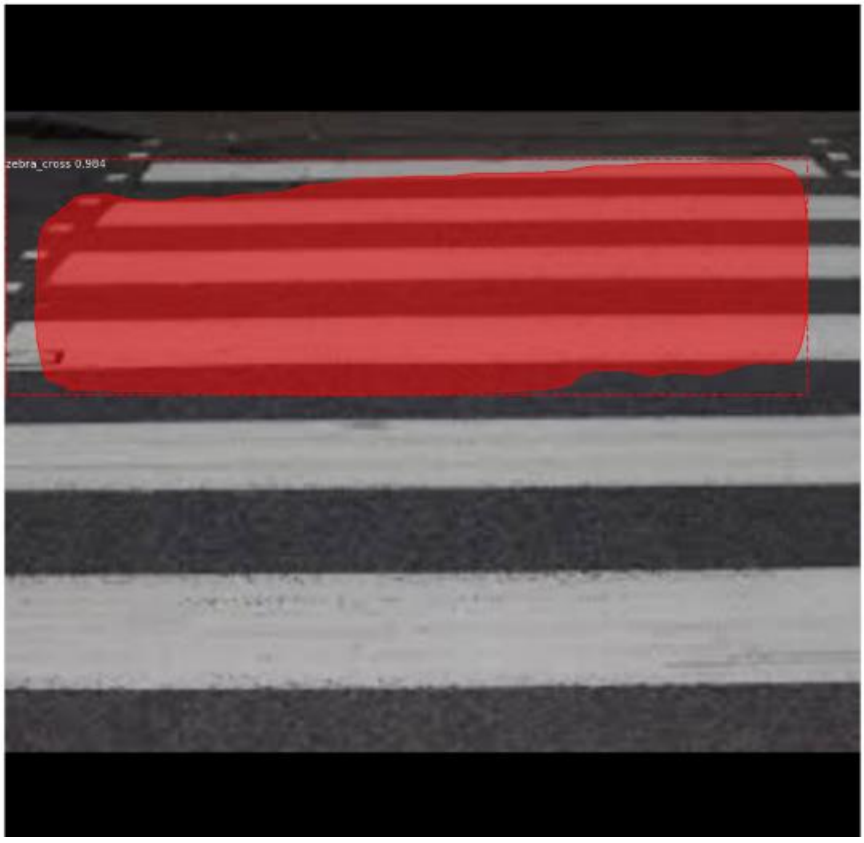
Predictions

```
image_id = 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.00142 (1) zebra_cross/images/00142.jpg
Processing 1 images
image                    shape: (1024, 1024, 3)       min:    0.00000  max:  187.00000  uint8
molded_images            shape: (1, 1024, 1024, 3)    min: -123.70000  max:   83.10000  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 = 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")
```

```
{'id': '00146', 'source': 'dataset', 'path': 'zebra_cross/images/00146.jpg', 'annotation': 'zebra_cross/annotation/00146.xml'}
image ID: dataset.00146 (2) zebra_cross/images/00146.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
```

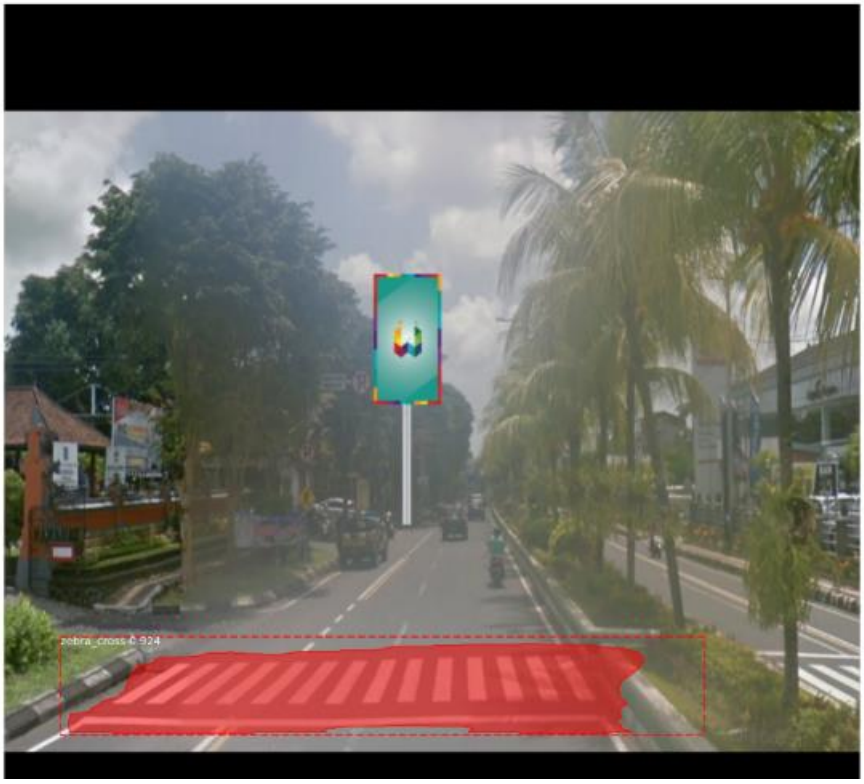
Predictions

```
image_id = 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': '00151', 'source': 'dataset', 'path': 'zebra_cross/images/00151.jpg', 'annotation': 'zebra_cross/annotation/00151.xml'
image ID: dataset.00151 (3) zebra_cross/images/00151.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:  149.10000  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



```
[23] image_id = 4
     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': '00150', 'source': 'dataset', 'path': 'zebra_cross/images/00150.jpg', 'annotation': 'zebra_cross/annotation/00150.xml'}
image ID: dataset.00150 (4) zebra_cross/images/00150.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

```
[24] 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': '00145', 'source': 'dataset', 'path': 'zebra_cross/images/00145.jpg', 'annotation': 'zebra_cross/annotation/00145.xml'}
image ID: dataset.00145 (5) zebra_cross/images/00145.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
```

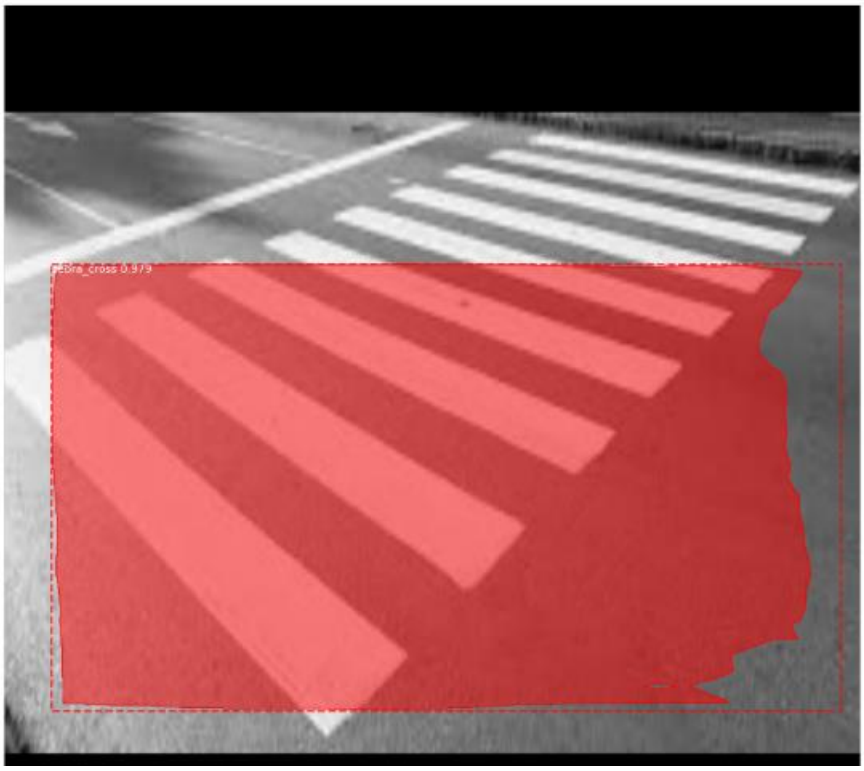
Predictions

```
[25] 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': '00144', 'source': 'dataset', 'path': 'zebra_cross/images/00144.jpg', 'annotation': 'zebra_cross/annotation/00144.xml'}
image ID: dataset.00144 (6) zebra_cross/images/00144.jpg
Processing 1 images
image                    shape: (1024, 1024, 3)       min:    0.00000  max:  253.00000  uint8
molded_images            shape: (1, 1024, 1024, 3)    min: -123.70000  max:  149.10000  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

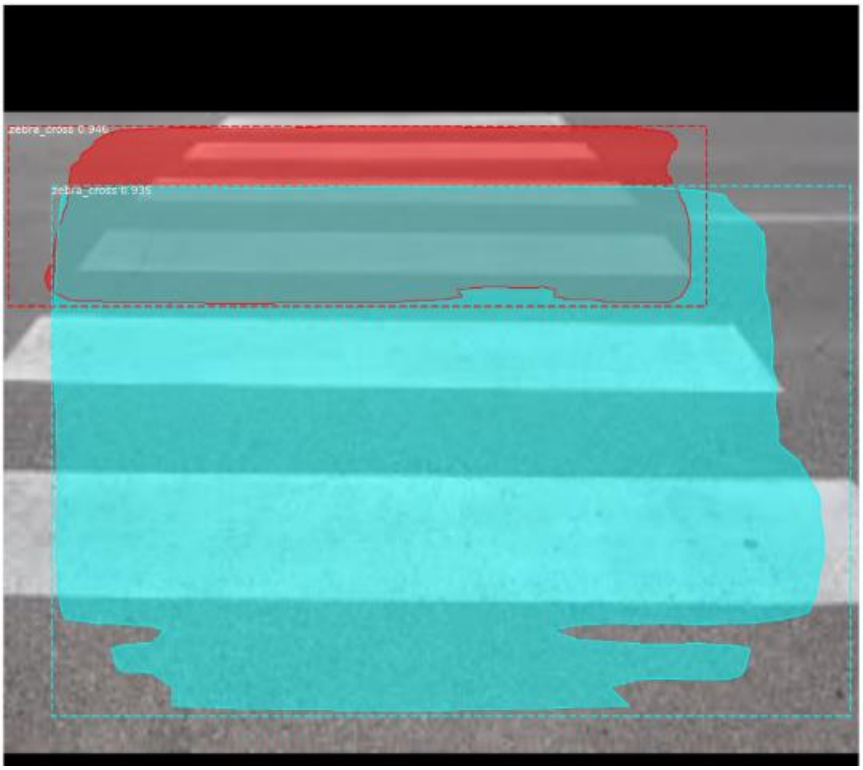


```
[26] image_id = 7
     image, image_meta, gt_class_id, gt_bbox, gt_mask = modellib.load_image_gt(test_set, config, image_id, use_mini_mask=False)
     info = test_set.image_info[image_id]
     print("image ID: {}.{} ({}) {}".format(info["source"], info["id"], image_id,
                                           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")
```

```
{'id': '00143', 'source': 'dataset', 'path': 'zebra_cross/images/00143.jpg', 'annotation': 'zebra_cross/annotation/00143.xml'}
image ID: dataset.00143 (7) zebra_cross/images/00143.jpg
Processing 1 images
image                    shape: (1024, 1024, 3)       min:    0.00000  max:  244.00000  uint8
molded_images            shape: (1, 1024, 1024, 3)    min: -123.70000  max:  139.10000  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

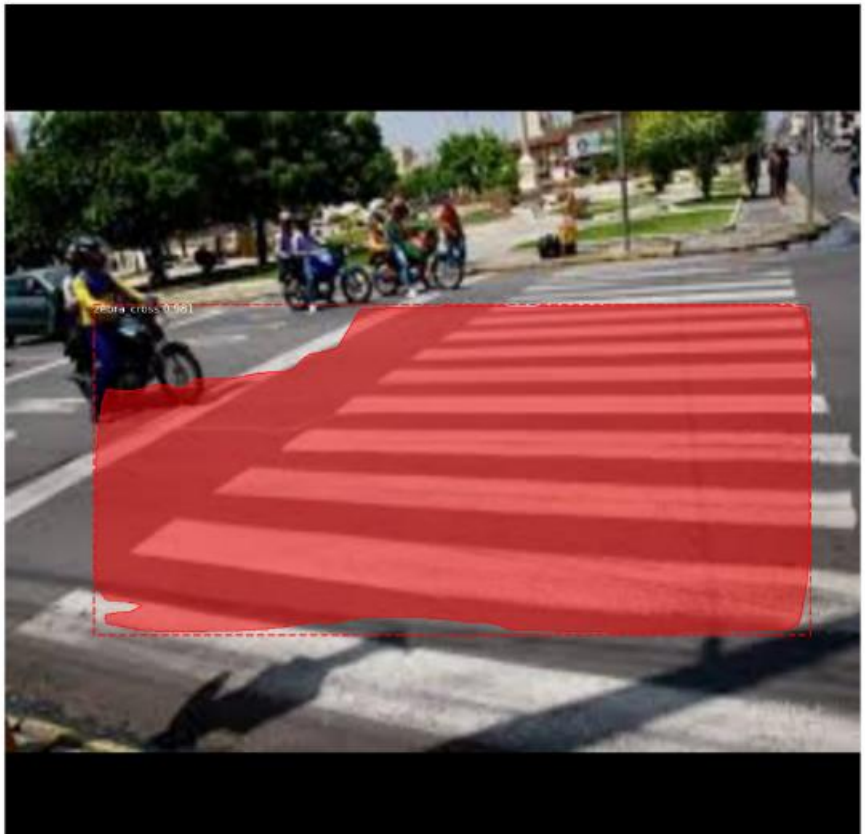


```
image_id = 8
image, image_meta, gt_class_id, gt_bbox, gt_mask = modellib.load_image_gt(test_set, config, image_id, use_mini_mask=False)
info = test_set.image_info[image_id]
print("image ID: {}.{} ({}) {}".format(info["source"], info["id"], image_id,
                                       test_set.image_reference(image_id)))
# 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.00148 (8) zebra_cross/images/00148.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:  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
```

Predictions

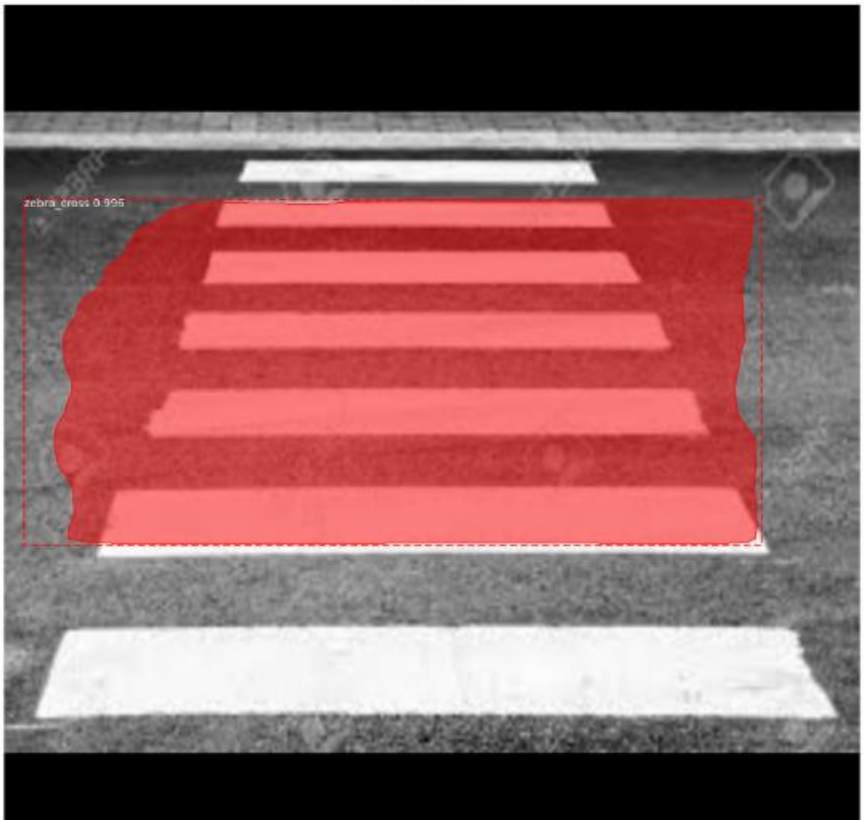


```
[28] image_id = 9
     image, image_meta, gt_class_id, gt_bbox, gt_mask = modellib.load_image_gt(test_set, config, image_id, use_mini_mask=False)
     info = test_set.image_info[image_id]
     print("image ID: {}.{} ({}) {}".format(info["source"], info["id"], image_id,
                                            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")
```

```
{'id': '00141', 'source': 'dataset', 'path': 'zebra_cross/images/00141.jpg', 'annotation': 'zebra_cross/annotation/00141.xml'}
image ID: dataset.00141 (9) zebra_cross/images/00141.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



```
[29] 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")
```

```
image ID: dataset.0014 (10) zebra_cross/images/0014.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