

 Navigation

# Machine Learning Mastery

Making Developers Awesome at Machine Learning

[Click to Take the FREE Computer Vision Crash-Course](#)



# How to Train an Object Detection Model with Keras

by Jason Brownlee on [May 29, 2019](#) in [Deep Learning for Computer Vision](#)

[Tweet](#)[Share](#)[Share](#)

Last Updated on October 3, 2019

Object detection is a challenging computer vision task that involves predicting both where the objects are in the image and what type of objects were detected.

The Mask Region-based Convolutional Neural Network, or Mask R-CNN, model is one of the state-of-the-art approaches for object recognition tasks. The Matterport Mask R-CNN project provides a library that allows you to develop and train Mask R-CNN Keras models for your own object detection tasks. Using the library can be tricky for beginners and requires the careful preparation of the dataset, although it allows fast training via transfer learning with top performing models trained on challenging object detection tasks, such as MS COCO.

In this tutorial, you will discover how to develop a Mask R-CNN model for kangaroo object detection in photographs.

After completing this tutorial, you will know:

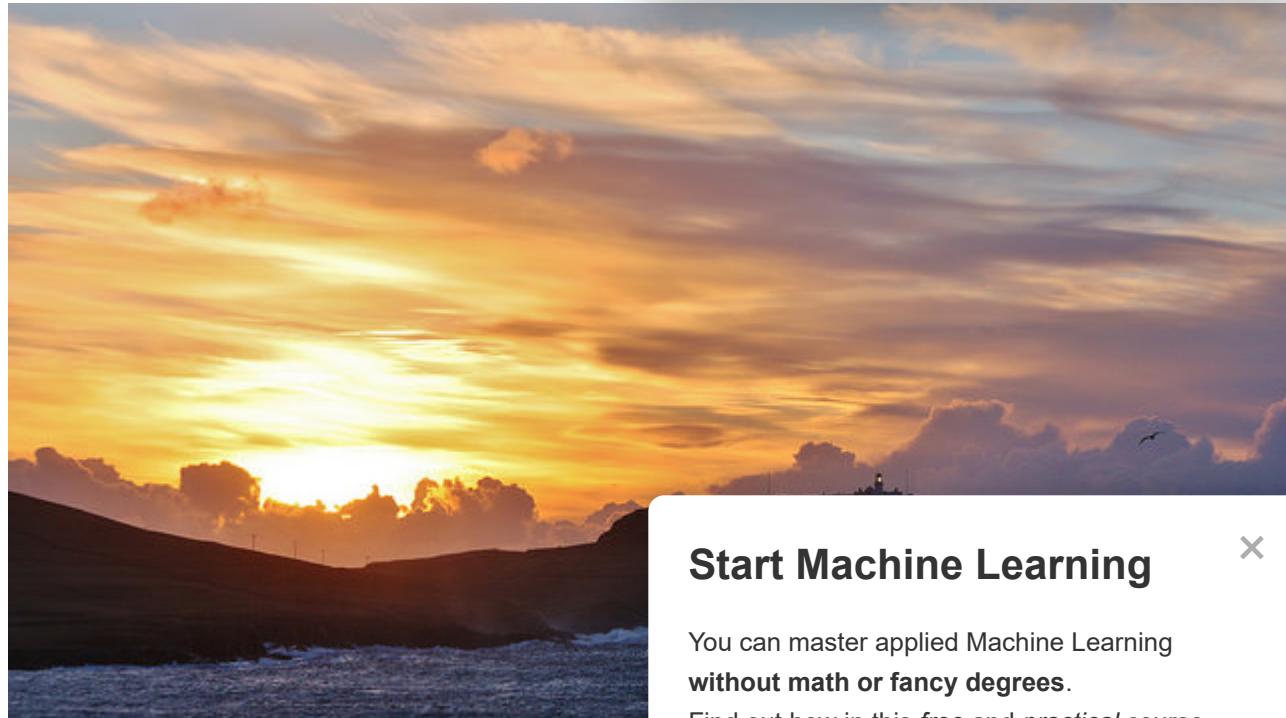
- How to prepare an object detection dataset ready for modeling with an R-CNN.
- How to use transfer learning to train an object detection model on a new dataset.
- How to evaluate a fit Mask R-CNN model on a test dataset and make predictions on new photos.

Discover how to build models for photo classification, object detection, face recognition, and more [in my new computer vision book](#), with 30 step-by-step tutorials and full source code.

Let's get started.

- **Note:** This tutorial requires TensorFlow version 1.14 or higher. It currently does not work with TensorFlow 2 because some third-party libraries have not been updated at the time of writing.

[Start Machine Learning](#)



How to Train an Object Detection Model to Find  
Photo by [Ronnie Roberts](#)

## Start Machine Learning X

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE

## Tutorial Overview

This tutorial is divided into five parts; they are:

1. How to Install Mask R-CNN for Keras
2. How to Prepare a Dataset for Object Detection
3. How to Train Mask R-CNN Model for Kangaroo Detection
4. How to Evaluate a Mask R-CNN Model
5. How to Detect Kangaroos in New Photos

## How to Install Mask R-CNN for Keras

Object detection is a task in computer vision that involves identifying the presence, location, and type of one or more objects in a given image.

It is a challenging problem that involves building upon methods for object recognition (e.g. where are they), object localization (e.g. what are their extent), and object classification (e.g. what are they).

The Region-Based Convolutional Neural Network, or R-CNN, is a family of convolutional neural network models designed for object detection, developed by [Ross Girshick](#), et al. There are perhaps four main variations of the approach, resulting in the current pinnacle called Mask R-CNN. The Mask R-CNN introduced in the 2018 paper titled “[Mask R-CNN](#)” is the most recent variation of the family of models and supports both object detection and object segmentation. Object segmentation not only involves localizing objects in the image but also specifies a mask for the image, indicating exactly which pixels in the image belong to the object.

[Start Machine Learning](#)

Mask R-CNN is a sophisticated model to implement, especially as compared to a simple or even state-of-the-art deep convolutional neural network model. Instead of developing an implementation of the R-CNN or Mask R-CNN model from scratch, we can use a reliable third-party implementation built on top of the Keras deep learning framework.

The best-of-breed third-party implementations of Mask R-CNN is the [Mask R-CNN Project](#) developed by [Matterport](#). The project is open source released under a permissive license (e.g. MIT license) and the code has been widely used on a variety of projects and Kaggle competitions.

The first step is to install the library.

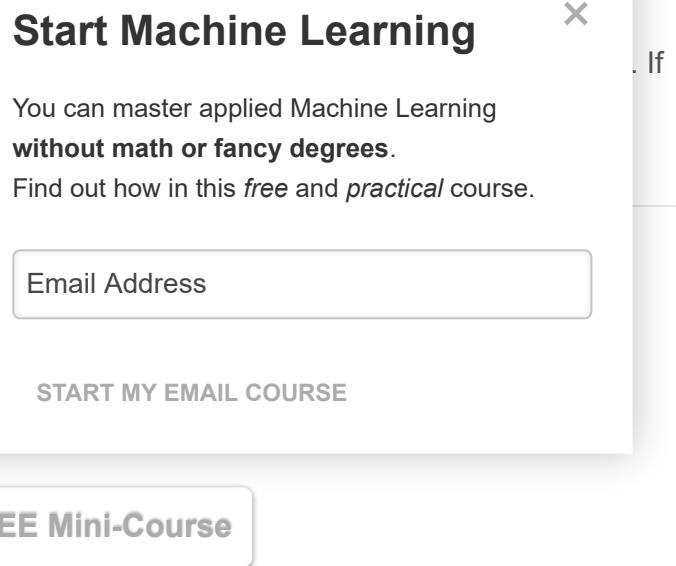
At the time of writing, there is no distributed version of the library, so we have to install it manually. The good news is that this is very easy.

Installation involves cloning the GitHub repository and you are having trouble, see the [installation instructions](#).

## Want Results with Deep Learning?

Take my free 7-day email crash course now

Click to sign-up and also get a free



## Step 1. Clone the Mask R-CNN GitHub Repository

This is as simple as running the following command from your command line:

```
1 git clone https://github.com/matterport/Mask_RCNN.git
```

This will create a new local directory with the name `Mask_RCNN` that looks as follows:

```
1 Mask_RCNN
2   ├── assets
3   ├── build
4   │   ├── bdist.macosx-10.13-x86_64
5   │   └── lib
6   |       └── mrcnn
7   ├── dist
8   ├── images
9   ├── mask_rcnn.egg-info
10  └── mrcnn
11    └── samples
12      ├── balloon
13      ├── coco
14      └── nucleus
```

[Start Machine Learning](#)

15 └─ shapes

## Step 2. Install the Mask R-CNN Library

The library can be installed directly via pip.

Change directory into the *Mask\_RCNN* directory and run the installation script.

From the command line, type the following:

```
1 cd Mask_RCNN
2 python setup.py install
```

On Linux or MacOS, you may need to install the software with `sudo` permissions; for example, you may see an error such as:

```
1 error: can't create or remove files in install
```

In that case, install the software with `sudo`:

```
1 sudo python setup.py install
```

If you are using a Python virtual environment ([virtualenv](#)) (recommended for this tutorial), you can install Mask\_

```
1 sudo ~/anaconda3/envs/tensorflow_p36/bin/python
```

The library will then install directly and you will see a lot of successful installation messages ending with the following:

```
1 ...
2 Finished processing dependencies for mask-rcnn==2.1
```

This confirms that you installed the library successfully and that you have the latest version, which at the time of writing is version 2.1.

## Step 3: Confirm the Library Was Installed

It is always a good idea to confirm that the library was installed correctly.

You can confirm that the library was installed correctly by querying it via the `pip` command; for example:

```
1 pip show mask-rcnn
```

You should see output informing you of the version and installation location; for example:

```
1 Name: mask-rcnn
2 Version: 2.1
3 Summary: Mask R-CNN for object detection and instance segmentation
4 Home-page: https://github.com/matterport/Mask_RCNN
5 Author: Matterport
6 Author-email: waleed.abdulla@gmail.com
7 License: MIT
8 Location: ...
```

[Start Machine Learning](#)

- 9 Requires:
- 10 Required-by:

We are now ready to use the library.

## How to Prepare a Dataset for Object Detection

Next, we need a dataset to model.

In this tutorial, we will use the [kangaroo dataset](#), made available by [Huynh Ngoc Anh](#) (experiencor). The dataset is comprised of 183 photographs that contain kangaroos, and XML annotation files that provide bounding boxes for the kangaroos in each photograph.

The Mask R-CNN is designed to learn to predict both those detected objects, and the kangaroo dataset does dataset to learn a kangaroo object detection task, and segmentation capabilities of the model.

There are a few steps required in order to prepare this in turn in this section, including downloading the dataset `KangarooDataset` object that can be used by the `MaskRCNN` class to confirm that we are loading images and annotations correctly.

### Install Dataset

The first step is to download the dataset into your current working directory.

This can be achieved by cloning the GitHub repository directly, as follows:

```
1 git clone https://github.com/experiencor/kangaroo.git
```

This will create a new directory called “*kangaroo*” with a subdirectory called ‘*images/*’ that contains all of the JPEG photos of kangaroos and a subdirectory called ‘*annotes/*’ that contains all of the XML files that describe the locations of kangaroos in each photo.

```
1 kangaroo
2   ├── annots
3   └── images
```

Looking in each subdirectory, you can see that the photos and annotation files use a consistent naming convention, with filenames using a 5-digit zero-padded numbering system; for example:

```
1 images/00001.jpg
2 images/00002.jpg
3 images/00003.jpg
4 ...
5 annots/00001.xml
6 annots/00002.xml
7 annots/00003.xml
8 ...
```

This makes matching photographs and annotation file

### Start Machine Learning

You can master applied Machine Learning without **math or fancy degrees**.

Find out how in this *free* and *practical* course.

[START MY EMAIL COURSE](#)

We can also see that the numbering system is not contiguous, that there are some photos missing, e.g. there is no ‘00007’ JPG or XML.

This means that we should focus on loading the list of actual files in the directory rather than using a numbering system.

## Parse Annotation File

The next step is to figure out how to load the annotation files.

First, open the first annotation file (*annots/00001.xml*) and take a look; you should see:

```

1 <annotation>
2   <folder>Kangaroo</folder>
3   <filename>00001.jpg</filename>
4   <path>...</path>
5   <source>
6     <database>Unknown</database>
7   </source>
8   <size>
9     <width>450</width>
10    <height>319</height>
11    <depth>3</depth>
12  </size>
13  <segmented>0</segmented>
14  <object>
15    <name>kangaroo</name>
16    <pose>Unspecified</pose>
17    <truncated>0</truncated>
18    <difficult>0</difficult>
19    <bndbox>
20      <xmin>233</xmin>
21      <ymin>89</ymin>
22      <xmax>386</xmax>
23      <ymax>262</ymax>
24    </bndbox>
25  </object>
26  <object>
27    <name>kangaroo</name>
28    <pose>Unspecified</pose>
29    <truncated>0</truncated>
30    <difficult>0</difficult>
31    <bndbox>
32      <xmin>134</xmin>
33      <ymin>105</ymin>
34      <xmax>341</xmax>
35      <ymax>253</ymax>
36    </bndbox>
37  </object>
38 </annotation>
```

## Start Machine Learning

You can master applied Machine Learning  
without math or fancy degrees.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE

We can see that the annotation file contains a “size” element that describes the shape of the photograph, and one or more “object” elements that describe the bounding boxes for the kangaroo objects in the photograph.

The size and the bounding boxes are the minimum information that we require from each annotation file.

We could write some careful XML parsing code to process this idea for a production system. Instead, we will short-cut

[Start Machine Learning](#)

extract the data that we need from each file, e.g. a `//size` query to extract the size element and a `//object` or a `//bndbox` query to extract the bounding box elements.

Python provides the [ElementTree API](#) that can be used to load and parse an XML file and we can use the `find()` and `findall()` functions to perform the XPath queries on a loaded document.

First, the annotation file must be loaded and parsed as an `ElementTree` object.

```
1 # load and parse the file
2 tree = ElementTree.parse(filename)
```

Once loaded, we can retrieve the root element of the document from which we can perform our XPath queries.

```
1 # get the root of the document
2 root = tree.getroot()
```

We can use the `findall()` function with a query for `'//bndbox'` to extract all of the bounding boxes, and then loop through each to extract the `x` and `y`, `min` and `max` values that define the box.

The element text can also be parsed to integer values.

```
1 # extract each bounding box
2 for box in root.findall('.//bndbox'):
3     xmin = int(box.find('xmin').text)
4     ymin = int(box.find('ymin').text)
5     xmax = int(box.find('xmax').text)
6     ymax = int(box.find('ymax').text)
7     coors = [xmin, ymin, xmax, ymax]
```

We can then collect the definition of each bounding box into a list.

The dimensions of the image may also be helpful, which can be queried directly.

```
1 # extract image dimensions
2 width = int(root.find('.//size/width').text)
3 height = int(root.find('.//size/height').text)
```

We can tie all of this together into a function that will take the annotation filename as an argument, extract the bounding box and image dimension details, and return them for use.

The `extract_boxes()` function below implements this behavior.

```
1 # function to extract bounding boxes from an annotation file
2 def extract_boxes(filename):
3     # load and parse the file
4     tree = ElementTree.parse(filename)
5     # get the root of the document
6     root = tree.getroot()
7     # extract each bounding box
8     boxes = list()
9     for box in root.findall('.//bndbox'):
10         xmin = int(box.find('xmin').text)
11         ymin = int(box.find('ymin').text)
12         xmax = int(box.find('xmax').text)
13         ymax = int(box.find('ymax').text)
```

## Start Machine Learning

You can master applied Machine Learning  
without math or fancy degrees.

Find out how in this *free* and *practical* course.

```

14     coors = [xmin, ymin, xmax, ymax]
15     boxes.append(coors)
16 # extract image dimensions
17 width = int(root.find('.//size/width').text)
18 height = int(root.find('.//size/height').text)
19 return boxes, width, height

```

We can test out this function on our annotation files, for example, on the first annotation file in the directory.

The complete example is listed below.

```

1 # example of extracting bounding boxes from an annotation file
2 from xml.etree import ElementTree
3
4 # function to extract bounding boxes from an annotation file
5 def extract_boxes(filename):
6     # load and parse the file
7     tree = ElementTree.parse(filename)
8     # get the root of the document
9     root = tree.getroot()
10    # extract each bounding box
11    boxes = list()
12    for box in root.findall('.//bndbox'):
13        xmin = int(box.find('xmin').text)
14        ymin = int(box.find('ymin').text)
15        xmax = int(box.find('xmax').text)
16        ymax = int(box.find('ymax').text)
17        coors = [xmin, ymin, xmax, ymax]
18        boxes.append(coors)
19    # extract image dimensions
20    width = int(root.find('.//size/width').text)
21    height = int(root.find('.//size/height').text)
22    return boxes, width, height
23
24 # extract details form annotation file
25 boxes, w, h = extract_boxes('kangaroo/annots/00001.xml')
26 # summarize extracted details
27 print(boxes, w, h)

```

Running the example returns a list that contains the details of each bounding box in the annotation file, as well as two integers for the width and height of the photograph.

```
1 [[233, 89, 386, 262], [134, 105, 341, 253]] 450 319
```

Now that we know how to load the annotation file, we can look at using this functionality to develop a Dataset object.

## Develop KangarooDataset Object

The mask-rcnn library requires that train, validation, and test datasets be managed by a `mrcnn.utils.Dataset` object.

This means that a new class must be defined that extends the `mrcnn.utils.Dataset` class and defines a function to load the dataset, with any name you like such as `load_dataset()`, and override two functions, one for loading a mask called `load_mask()` and one for loading an image reference (path or URL) called `image_reference()`.

[Start Machine Learning](#)

```

2 class KangarooDataset(Dataset):
3     # load the dataset definitions
4     def load_dataset(self, dataset_dir, is_train=True):
5         # ...
6
7     # load the masks for an image
8     def load_mask(self, image_id):
9         # ...
10
11    # load an image reference
12    def image_reference(self, image_id):
13        # ...

```

To use a *Dataset* object, it is instantiated, then your custom load function must be called, then finally the built-in *prepare()* function is called.

For example, we will create a new class called *KangarooDataset*:

```

1 # prepare the dataset
2 train_set = KangarooDataset()
3 train_set.load_dataset(...)
4 train_set.prepare()

```

The custom load function, e.g. *load\_dataset()* is responsible for loading the images in the dataset.

Classes are defined by calling the built-in *add\_class()* function on the dataset, the ‘*class\_id*’ or integer for the class (e.g. 1 for the kangaroo class), and the ‘*class\_name*’ (e.g. ‘kangaroo’).

```

1 # define one class
2 self.add_class("dataset", 1, "kangaroo")

```

Objects are defined by a call to the built-in *add\_image()* function and specifying the ‘*source*’ (the name of the dataset), a unique ‘*image\_id*’ (e.g. the filename without the file extension like ‘00001’), and the path for where the image can be loaded (e.g. ‘kangaroo/images/00001.jpg’).

This will define an “*image info*” dictionary for the image that can be retrieved later via the index or order in which the image was added to the dataset. You can also specify other arguments that will be added to the image info dictionary, such as an ‘*annotation*’ to define the annotation path.

```

1 # add to dataset
2 self.add_image('dataset', image_id='00001', path='kangaroo/images/00001.jpg', annotation='kangaroo')

```

For example, we can implement a *load\_dataset()* function that takes the path to the dataset directory and loads all images in the dataset.

Note, testing revealed that there is an issue with image number ‘00090’, so we will exclude it from the dataset.

```

1 # load the dataset definitions
2 def load_dataset(self, dataset_dir):
3     # define one class
4     self.add_class("dataset", 1, "kangaroo")
5     # define data locations

```

## Start Machine Learning

You can master applied Machine Learning  
without math or fancy degrees.

Find out how in this *free* and *practical* course.

[START MY EMAIL COURSE](#)

```

6  images_dir = dataset_dir + '/images/'
7  annotations_dir = dataset_dir + '/annots/'
8  # find all images
9  for filename in.listdir(images_dir):
10     # extract image id
11     image_id = filename[:-4]
12     # skip bad images
13     if image_id in ['00090']:
14         continue
15     img_path = images_dir + filename
16     ann_path = annotations_dir + image_id + '.xml'
17     # add to dataset
18     self.add_image('dataset', image_id=image_id, path=img_path, annotation=ann_path)

```

We can go one step further and add one more argument to the function to define whether the *Dataset* instance is for training or test/validation. We have about 160 photos, so we can use about 20%, or the last 32 photos, as a test or validation dataset and the first

This division can be made using the integer in the file to be train and equal or after 150 used for test. The update datasets is provided below.

```

1 # load the dataset definitions
2 def load_dataset(self, dataset_dir, is_train=True):
3     # define one class
4     self.add_class("dataset", 1, "kangaroo")
5     # define data locations
6     images_dir = dataset_dir + '/images/'
7     annotations_dir = dataset_dir + '/annots/'
8     # find all images
9     for filename in.listdir(images_dir):
10        # extract image id
11        image_id = filename[:-4]
12        # skip bad images
13        if image_id in ['00090']:
14            continue
15        # skip all images after 150 if we are building the train set
16        if is_train and int(image_id) >= 150:
17            continue
18        # skip all images before 150 if we are building the test/val set
19        if not is_train and int(image_id) < 150:
20            continue
21        img_path = images_dir + filename
22        ann_path = annotations_dir + image_id + '.xml'
23        # add to dataset
24        self.add_image('dataset', image_id=image_id, path=img_path, annotation=ann_path)

```

## Start Machine Learning

You can master applied Machine Learning  
without math or fancy degrees.

Find out how in this *free* and *practical* course.

[START MY EMAIL COURSE](#)

Next, we need to define the *load\_mask()* function for loading the mask for a given '*image\_id*'.

In this case, the '*image\_id*' is the integer index for an image in the dataset, assigned based on the order that the image was added via a call to *add\_image()* when loading the dataset. The function must return an array of one or more masks for the photo associated with the *image\_id*, and the classes for each mask.

We don't have masks, but we do have bounding boxes. We can load the bounding boxes for a given photo and return them as masks. The library will then infer bounding boxes from our "*masks*" which will be the same size.

[Start Machine Learning](#)

First, we must load the annotation file for the *image\_id*. This involves first retrieving the ‘*image info*’ dict for the *image\_id*, then retrieving the annotations path that we stored for the image via our prior call to *add\_image()*. We can then use the path in our call to *extract\_boxes()* developed in the previous section to get the list of bounding boxes and the dimensions of the image.

```
1 # get details of image
2 info = self.image_info[image_id]
3 # define box file location
4 path = info['annotation']
5 # load XML
6 boxes, w, h = self.extract_boxes(path)
```

We can now define a mask for each bounding box, and an associated class.

A mask is a two-dimensional array with the same dimensions as the image. It has all zero values where the object isn’t and all one values where the object is.

We can achieve this by creating a NumPy array with a channel dimension for each bounding box, and a one channel for each bounding box.

```
1 # create one array for all masks, each on a different channel
2 masks = zeros([h, w, len(boxes)], dtype='uint8')
```

Each bounding box is defined as *min* and *max*, *x* and *y* coordinates.

These can be used directly to define row and column indices.

```
1 # create masks
2 for i in range(len(boxes)):
3     box = boxes[i]
4     row_s, row_e = box[1], box[3]
5     col_s, col_e = box[0], box[2]
6     masks[row_s:row_e, col_s:col_e, i] = 1
```

All objects have the same class in this dataset. We can retrieve the class index via the ‘*class\_names*’ dictionary, then add it to a list to be returned alongside the masks.

```
1 self.class_names.index('kangaroo')
```

Tying this together, the complete *load\_mask()* function is listed below.

```
1 # load the masks for an image
2 def load_mask(self, image_id):
3     # get details of image
4     info = self.image_info[image_id]
5     # define box file location
6     path = info['annotation']
7     # load XML
8     boxes, w, h = self.extract_boxes(path)
9     # create one array for all masks, each on a different channel
10    masks = zeros([h, w, len(boxes)], dtype='uint8')
11    # create masks
12    class_ids = list()
13    for i in range(len(boxes)):
14        box = boxes[i]
15        row_s, row_e = box[1], box[3]
16        col_s, col_e = box[0], box[2]
```

## Start Machine Learning

You can master applied Machine Learning  
without math or fancy degrees.

Find out how in this *free* and *practical* course.

[START MY EMAIL COURSE](#)

```

17     masks[row_s:row_e, col_s:col_e, i] = 1
18     class_ids.append(self.class_names.index('kangaroo'))
19 return masks, asarray(class_ids, dtype='int32')

```

Finally, we must implement the *image\_reference()* function.

This function is responsible for returning the path or URL for a given '*image\_id*', which we know is just the '*path*' property on the '*image info*' dict.

```

1 # load an image reference
2 def image_reference(self, image_id):
3     info = self.image_info[image_id]
4     return info['path']

```

And that's it. We have successfully defined a *Dataset* object for the *mask-rcnn* library for our Kangaroo dataset.

The complete listing of the class and creating a train a

```

1 # split into train and test set
2 from os import listdir
3 from xml.etree import ElementTree
4 from numpy import zeros
5 from numpy import asarray
6 from mrcnn.utils import Dataset
7
8 # class that defines and loads the kangaroo dataset
9 class KangarooDataset(Dataset):
10     # load the dataset definitions
11     def load_dataset(self, dataset_dir, is_train=True):
12         # define one class
13         self.add_class("dataset", 1, "kangaroo")
14         # define data locations
15         images_dir = dataset_dir + '/images/'
16         annotations_dir = dataset_dir + '/annots/'
17         # find all images
18         for filename in listdir(images_dir):
19             # extract image id
20             image_id = filename[:-4]
21             # skip bad images
22             if image_id in ['00090']:
23                 continue
24             # skip all images after 150 if we are building the train set
25             if is_train and int(image_id) >= 150:
26                 continue
27             # skip all images before 150 if we are building the test/val set
28             if not is_train and int(image_id) < 150:
29                 continue
30             img_path = images_dir + filename
31             ann_path = annotations_dir + image_id + '.xml'
32             # add to dataset
33             self.add_image('dataset', image_id=image_id, path=img_path, annotation=ann_path)
34
35         # extract bounding boxes from an annotation file
36         def extract_boxes(self, filename):
37             # load and parse the file
38             tree = ElementTree.parse(filename)
39             # get the root of the document
40             root = tree.getroot()
41             # extract each bounding box
42             boxes = list()

```

## Start Machine Learning

You can master applied Machine Learning  
without math or fancy degrees.

Find out how in this *free* and *practical* course.

[START MY EMAIL COURSE](#)

[Start Machine Learning](#)

```

43     for box in root.findall('.//bndbox'):
44         xmin = int(box.find('xmin').text)
45         ymin = int(box.find('ymin').text)
46         xmax = int(box.find('xmax').text)
47         ymax = int(box.find('ymax').text)
48         coors = [xmin, ymin, xmax, ymax]
49         boxes.append(coors)
50     # extract image dimensions
51     width = int(root.find('.//size/width').text)
52     height = int(root.find('.//size/height').text)
53     return boxes, width, height
54
55 # load the masks for an image
56 def load_mask(self, image_id):
57     # get details of image
58     info = self.image_info[image_id]
59     # define box file location
60     path = info['annotation']
61     # load XML
62     boxes, w, h = self.extract_boxes(path)
63     # create one array for all masks, each
64     # masks = zeros([h, w, len(boxes)], dtype
65     # create masks
66     class_ids = list()
67     for i in range(len(boxes)):
68         box = boxes[i]
69         row_s, row_e = box[1], box[3]
70         col_s, col_e = box[0], box[2]
71         masks[row_s:row_e, col_s:col_e, i] = 1
72         class_ids.append(self.class_names[i])
73     return masks, asarray(class_ids, dtype
74
75 # load an image reference
76 def image_reference(self, image_id):
77     info = self.image_info[image_id]
78     return info['path']
79
80 # train set
81 train_set = KangarooDataset()
82 train_set.load_dataset('kangaroo', is_train=True)
83 train_set.prepare()
84 print('Train: %d' % len(train_set.image_ids))
85
86 # test/val set
87 test_set = KangarooDataset()
88 test_set.load_dataset('kangaroo', is_train=False)
89 test_set.prepare()
90 print('Test: %d' % len(test_set.image_ids))

```

Running the example successfully loads and prepares the train and test dataset and prints the number of images in each.

```

1 Train: 131
2 Test: 32

```

Now that we have defined the dataset, let's confirm that the images, masks, and bounding boxes are handled correctly.

## Test KangarooDataset Object

The first useful test is to confirm that the images and r

[Start Machine Learning](#)

We can test this by creating a dataset and loading an image via a call to the `load_image()` function with an `image_id`, then load the mask for the image via a call to the `load_mask()` function with the same `image_id`.

```

1 # load an image
2 image_id = 0
3 image = train_set.load_image(image_id)
4 print(image.shape)
5 # load image mask
6 mask, class_ids = train_set.load_mask(image_id)
7 print(mask.shape)
```

Next, we can plot the photograph using the Matplotlib API, then plot the first mask over the top with an alpha value so that the photograph underneath can still be seen

```

1 # plot image
2 pyplot.imshow(image)
3 # plot mask
4 pyplot.imshow(mask[:, :, 0], cmap='gray', alpha=0.5)
5 pyplot.show()
```

The complete example is listed below.

```

1 # plot one photograph and mask
2 from os import listdir
3 from xml.etree import ElementTree
4 from numpy import zeros
5 from numpy import asarray
6 from mrcnn.utils import Dataset
7 from matplotlib import pyplot
8
9 # class that defines and loads the kangaroo dataset
10 class KangarooDataset(Dataset):
11     # load the dataset definitions
12     def load_dataset(self, dataset_dir, is_train=True):
13         # define one class
14         self.add_class("dataset", 1, "kangaroo")
15         # define data locations
16         images_dir = dataset_dir + '/images/'
17         annotations_dir = dataset_dir + '/annots/'
18         # find all images
19         for filename in listdir(images_dir):
20             # extract image id
21             image_id = filename[:-4]
22             # skip bad images
23             if image_id in ['00090']:
24                 continue
25             # skip all images after 150 if we are building the train set
26             if is_train and int(image_id) >= 150:
27                 continue
28             # skip all images before 150 if we are building the test/val set
29             if not is_train and int(image_id) < 150:
30                 continue
31             img_path = images_dir + filename
32             ann_path = annotations_dir + image_id + '.xml'
33             # add to dataset
34             self.add_image('dataset', image_id=image_id, path=img_path, annotation=ann_path)
35
36     # extract bounding boxes from an annotation file
37     def extract_boxes(self, filename):
38         # load and parse the file
39         tree = ElementTree.parse(filename)
40         # get the root of the document
```

## Start Machine Learning

You can master applied Machine Learning  
without math or fancy degrees.

Find out how in this *free* and *practical* course.

[START MY EMAIL COURSE](#)

[Start Machine Learning](#)

```

41     root = tree.getroot()
42     # extract each bounding box
43     boxes = list()
44     for box in root.findall('.//bndbox'):
45         xmin = int(box.find('xmin').text)
46         ymin = int(box.find('ymin').text)
47         xmax = int(box.find('xmax').text)
48         ymax = int(box.find('ymax').text)
49         coors = [xmin, ymin, xmax, ymax]
50         boxes.append(coors)
51     # extract image dimensions
52     width = int(root.find('.//size/width').text)
53     height = int(root.find('.//size/height').text)
54     return boxes, width, height
55
56     # load the masks for an image
57 def load_mask(self, image_id):
58     # get details of image
59     info = self.image_info[image_id]
60     # define box file location
61     path = info['annotation']
62     # load XML
63     boxes, w, h = self.extract_boxes(path)
64     # create one array for all masks, each
65     masks = zeros([h, w, len(boxes)], dtype=uint8)
66     # create masks
67     class_ids = list()
68     for i in range(len(boxes)):
69         box = boxes[i]
70         row_s, row_e = box[1], box[3]
71         col_s, col_e = box[0], box[2]
72         masks[row_s:row_e, col_s:col_e, i] = 1
73         class_ids.append(self.class_names[i])
74     return masks, asarray(class_ids, dtype='int32')
75
76     # load an image reference
77 def image_reference(self, image_id):
78     info = self.image_info[image_id]
79     return info['path']
80
81 # train set
82 train_set = KangarooDataset()
83 train_set.load_dataset('kangaroo', is_train=True)
84 train_set.prepare()
85 # load an image
86 image_id = 0
87 image = train_set.load_image(image_id)
88 print(image.shape)
89 # load image mask
90 mask, class_ids = train_set.load_mask(image_id)
91 print(mask.shape)
92 # plot image
93 pyplot.imshow(image)
94 # plot mask
95 pyplot.imshow(mask[:, :, 0], cmap='gray', alpha=0.5)
96 pyplot.show()

```

## Start Machine Learning

You can master applied Machine Learning  
without math or fancy degrees.

Find out how in this *free* and *practical* course.

[START MY EMAIL COURSE](#)

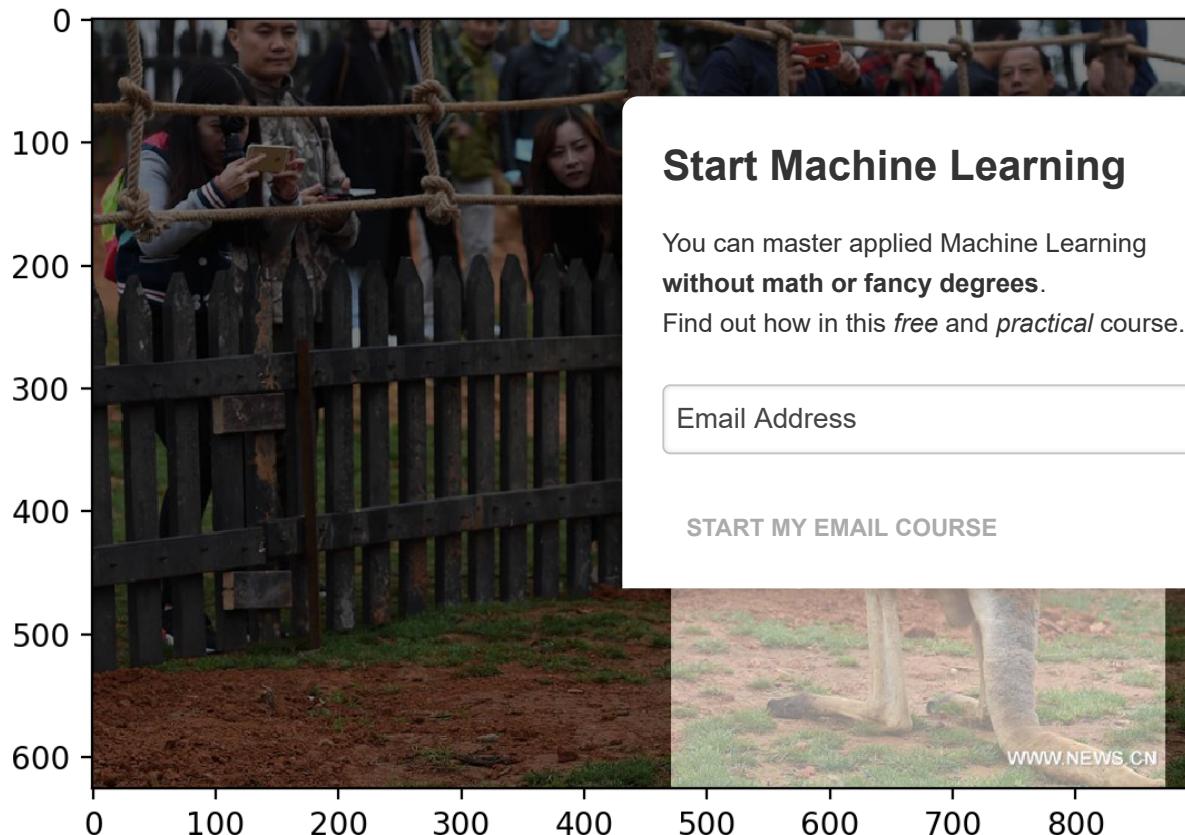
Running the example first prints the shape of the photograph and mask NumPy arrays.

We can confirm that both arrays have the same width and height and only differ in terms of the number of channels. We can also see that the first photograph (e.g. *image\_id=0*) in this case only has one mask.

[2 \(626, 899, 1\)](#)

A plot of the photograph is also created with the first mask overlaid.

In this case, we can see that one kangaroo is present in the photo and that the mask correctly bounds the kangaroo.



Photograph of Kangaroo With Object Detection Mask Overlaid

We could repeat this for the first nine photos in the dataset, plotting each photo in one figure as a subplot and plotting all masks for each photo.

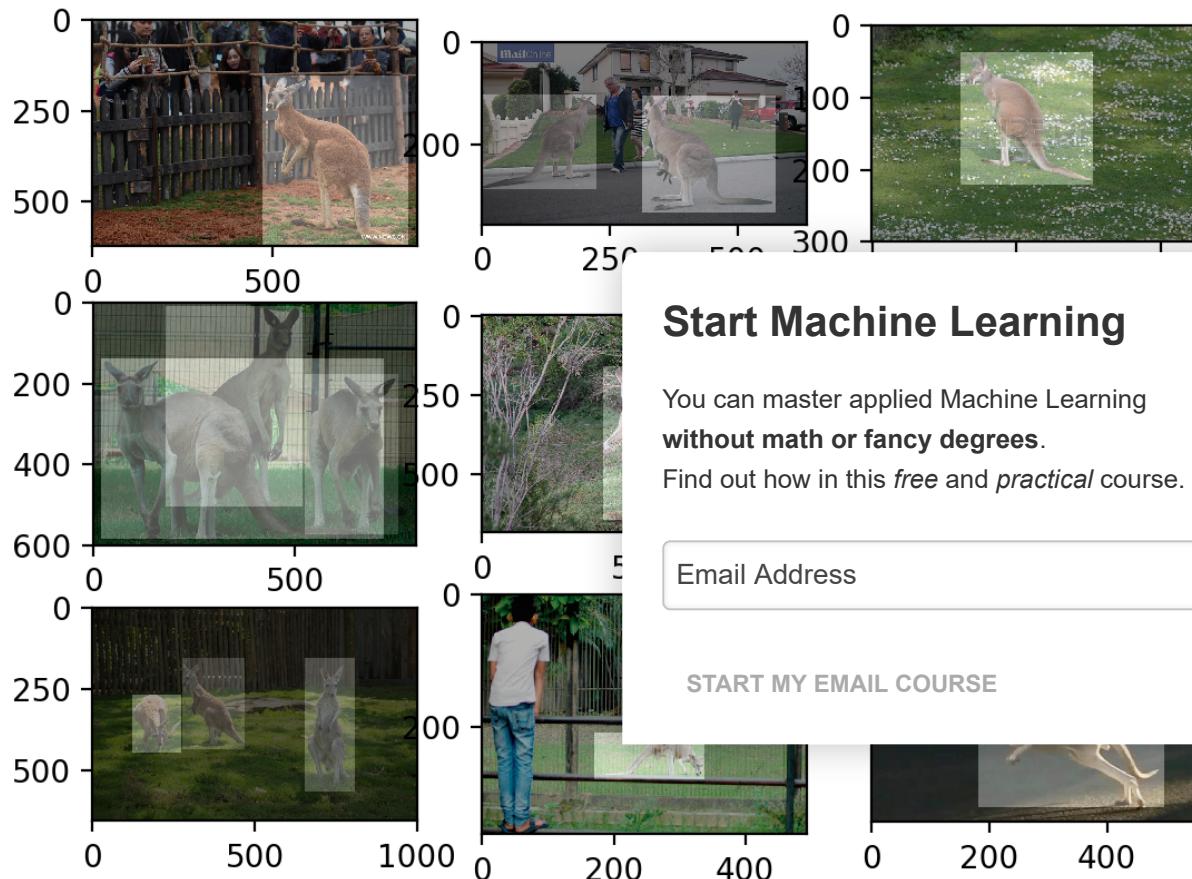
```

1 # plot first few images
2 for i in range(9):
3     # define subplot
4     pyplot.subplot(330 + 1 + i)
5     # plot raw pixel data
6     image = train_set.load_image(i)
7     pyplot.imshow(image)
8     # plot all masks
9     mask, _ = train_set.load_mask(i)
10    for j in range(mask.shape[2]):
11        pyplot.imshow(mask[:, :, j], cmap='gray', alpha=0.3)
12    # show the figure
13    pyplot.show()

```

[Start Machine Learning](#)

Running the example shows that photos are loaded correctly and that those photos with multiple objects correctly have separate masks defined.



Plot of First Nine Photos of Kangaroos in the Training Dataset With Object Detection Masks

Another useful debugging step might be to load all of the ‘*image info*’ objects in the dataset and print them to the console.

This can help to confirm that all of the calls to the `add_image()` function in the `load_dataset()` function worked as expected.

```

1 # enumerate all images in the dataset
2 for image_id in train_set.image_ids:
3     # load image info
4     info = train_set.image_info[image_id]
5     # display on the console
6     print(info)

```

Running this code on the loaded training dataset will then show all of the ‘*image info*’ dictionaries, showing the paths and ids for each image in the dataset.

```

1 {'id': '00132', 'source': 'dataset', 'path': '.../kangaroo_00132.jpg'}
2 {'id': '00046', 'source': 'dataset', 'path': '.../kangaroo_00046.jpg'}      Start Machine Learning

```

```
3 {'id': '00052', 'source': 'dataset', 'path': 'kangaroo/images/00052.jpg', 'annotation': 'kangaro...  
4 ...
```

Finally, the *mask-rcnn* library provides utilities for displaying images and masks. We can use some of these built-in functions to confirm that the Dataset is operating correctly.

For example, the *mask-rcnn* library provides the *mrcnn.visualize.display\_instances()* function that will show a photograph with bounding boxes, masks, and class labels. This requires that the bounding boxes are extracted from the masks via the *extract\_bboxes()* function.

```
1 # define image id  
2 image_id = 1  
3 # load the image  
4 image = train_set.load_image(image_id)  
5 # load the masks and the class ids  
6 mask, class_ids = train_set.load_mask(image_id)  
7 # extract bounding boxes from the masks  
8 bbox = extract_bboxes(mask)  
9 # display image with masks and bounding boxes  
10 display_instances(image, bbox, mask, class_ids)
```

For completeness, the full code listing is provided below.

```
# display image with masks and bounding boxes  
from os import listdir  
from xml.etree import ElementTree  
from numpy import zeros  
from numpy import asarray  
from mrcnn.utils import Dataset  
from mrcnn.visualize import display_instances  
from mrcnn.utils import extract_bboxes  
  
# class that defines and loads the kangaroo dataset  
class KangarooDataset(Dataset):  
    # load the dataset definitions  
    def load_dataset(self, dataset_dir, is_train=True):  
        # define one class  
        self.add_class("dataset", 1, "kangaroo")  
        # define data locations  
        images_dir = dataset_dir + '/images/'  
        annotations_dir = dataset_dir + '/annots/'  
        # find all images  
        for filename in listdir(images_dir):  
            # extract image id  
            image_id = filename[:-4]  
            # skip bad images  
            if image_id in ['00090']:  
                continue  
            # skip all images after 150 if we are building the train set  
            if is_train and int(image_id) >= 150:  
                continue  
            # skip all images before 150 if we are building the test/val set  
            if not is_train and int(image_id) < 150:  
                continue  
            img_path = images_dir + filename  
            ann_path = annotations_dir + image_id + '.xml'  
            # add to dataset  
            self.add_image('dataset', image_id=image_id, path=img_path, annotation=ann_path)  
  
        # extract bounding boxes from an annotation file  
        def extract_boxes(self, filename):  
            # load and parse the file
```

## Start Machine Learning

You can master applied Machine Learning without math or fancy degrees.

Find out how in this *free* and *practical* course.

[START MY EMAIL COURSE](#)

```

tree = ElementTree.parse(filename)
# get the root of the document
root = tree.getroot()
# extract each bounding box
boxes = list()
for box in root.findall('.//bndbox'):
    xmin = int(box.find('xmin').text)
    ymin = int(box.find('ymin').text)
    xmax = int(box.find('xmax').text)
    ymax = int(box.find('ymax').text)
    coors = [xmin, ymin, xmax, ymax]
    boxes.append(coors)
# extract image dimensions
width = int(root.find('.//size/width').text)
height = int(root.find('.//size/height').text)
return boxes, width, height

# load the masks for an image
def load_mask(self, image_id):
    # get details of image
    info = self.image_info[image_id]
    # define box file location
    path = info['annotation']
    # load XML
    boxes, w, h = self.extract_boxes(path)
    # create one array for all masks, each or
    masks = zeros([h, w, len(boxes)], dtype='uint8')
    # create masks
    class_ids = list()
    for i in range(len(boxes)):
        box = boxes[i]
        row_s, row_e = box[1], box[3]
        col_s, col_e = box[0], box[2]
        masks[row_s:row_e, col_s:col_e, i] = 1
        class_ids.append(self.class_names.index('kangaroo'))
    return masks, asarray(class_ids, dtype='int32')

# load an image reference
def image_reference(self, image_id):
    info = self.image_info[image_id]
    return info['path']

# train set
train_set = KangarooDataset()
train_set.load_dataset('kangaroo', is_train=True)
train_set.prepare()
# define image id
image_id = 1
# load the image
image = train_set.load_image(image_id)
# load the masks and the class ids
mask, class_ids = train_set.load_mask(image_id)
# extract bounding boxes from the masks

```

## Start Machine Learning



You can master applied Machine Learning  
without math or fancy degrees.

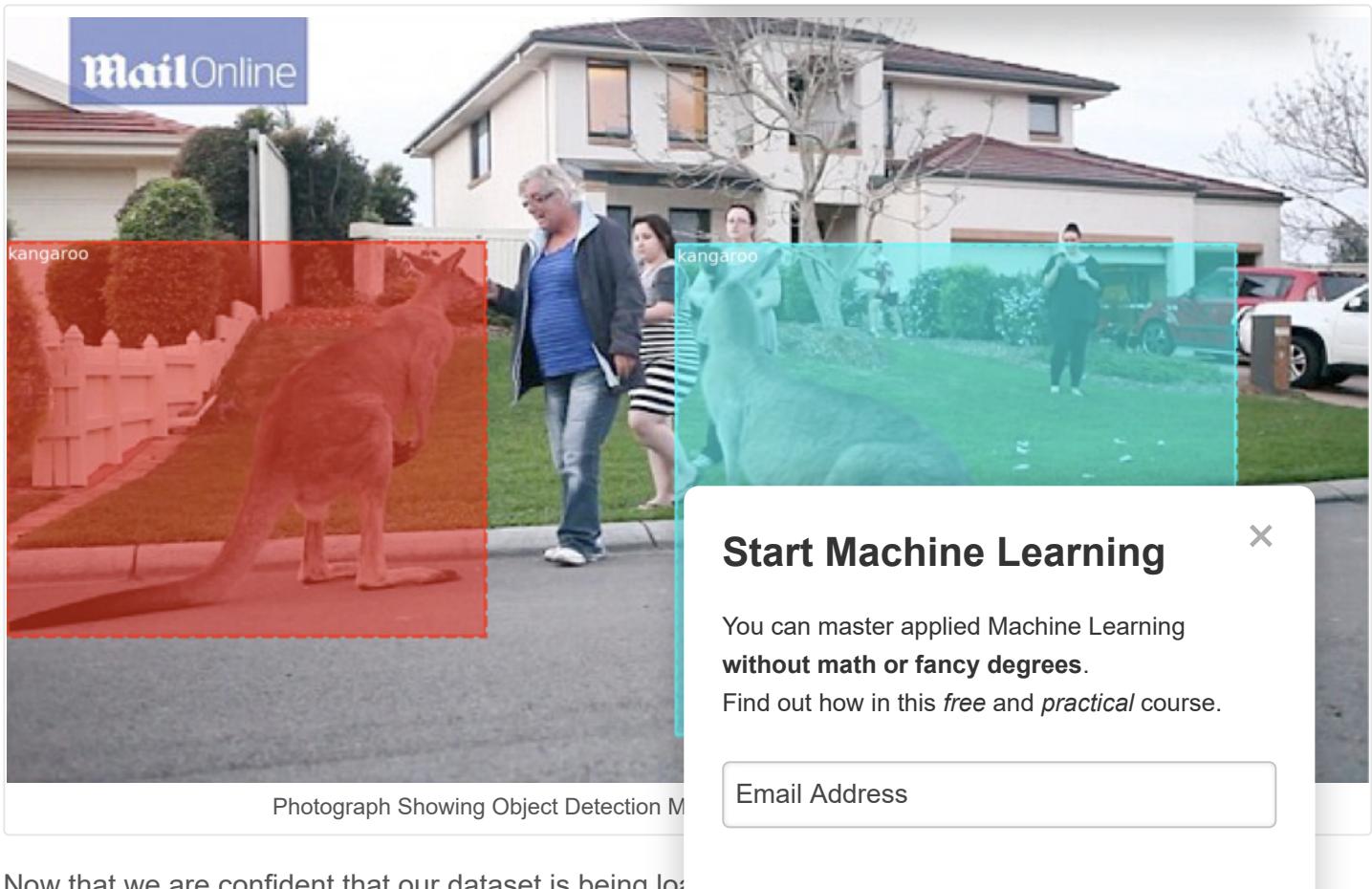
Find out how in this *free* and *practical* course.

[START MY EMAIL COURSE](#)

Running the example creates a plot showing the photograph with the mask for each object in a separate color.

The bounding boxes match the masks exactly, by design, and are shown with dotted outlines. Finally, each object is marked with the class label, which in this case

[Start Machine Learning](#)



Now that we are confident that our dataset is being loaded correctly, we can move on to training our model.

## How to Train Mask R-CNN Model for Kangaroo Detection

A Mask R-CNN model can be fit from scratch, although like other computer vision applications, time can be saved and performance can be improved by using transfer learning.

The Mask R-CNN model pre-fit on the MS COCO object detection dataset can be used as a starting point and then tailored to the specific dataset, in this case, the kangaroo dataset.

The first step is to download the model file (architecture and weights) for the pre-fit Mask R-CNN model. The weights are available from the GitHub project and the file is about 250 megabytes.

Download the model weights to a file with the name '`mask_rcnn_coco.h5`' in your current working directory.

- [Download Weights \(mask\\_rcnn\\_coco.h5\) 246M](#)

Next, a configuration object for the model must be defined.

This is a new class that extends the `mrcnn.config.Config` class and defines properties of both the prediction problem (such as name and the number of classes) and the algorithm for training the model (such as the learning rate).

[Start Machine Learning](#)

The configuration must define the name of the configuration via the ‘NAME’ attribute, e.g. ‘*kangaroo\_cfg*’, that will be used to save details and models to file during the run. The configuration must also define the number of classes in the prediction problem via the ‘NUM\_CLASSES’ attribute. In this case, we only have one object type of kangaroo, although there is always an additional class for the background.

Finally, we must define the number of samples (photos) used in each training epoch. This will be the number of photos in the training dataset, in this case, 131.

Tying this together, our custom *KangarooConfig* class is defined below.

```

1 # define a configuration for the model
2 class KangarooConfig(Config):
3     # Give the configuration a recognizable name
4     NAME = "kangaroo_cfg"
5     # Number of classes (background + kangaroos)
6     NUM_CLASSES = 1 + 1
7     # Number of training steps per epoch
8     STEPS_PER_EPOCH = 131
9
10 # prepare config
11 config = KangarooConfig()

```

Next, we can define our model.

This is achieved by creating an instance of the *mrcnn*. This will be used for training via setting the ‘mode’ argument.

The ‘config’ argument must also be specified with an instance of our *KangarooConfig* class.

Finally, a directory is needed where configuration files can be saved and where checkpoint models can be saved at the end of each epoch. We will use the current working directory.

```

1 # define the model
2 model = MaskRCNN(mode='training', model_dir='./', config=config)

```

Next, the pre-defined model architecture and weights can be loaded. This can be achieved by calling the *load\_weights()* function on the model and specifying the path to the downloaded ‘*mask\_rcnn\_coco.h5*’ file.

The model will be used as-is, although the class-specific output layers will be removed so that new output layers can be defined and trained. This can be done by specifying the ‘exclude’ argument and listing all of the output layers to exclude or remove from the model after it is loaded. This includes the output layers for the classification label, bounding boxes, and masks.

```

1 # load weights (mscoco)
2 model.load_weights('mask_rcnn_coco.h5', by_name=True, exclude=['mrcnn_class_logits', "mrcnn_bbx"]

```

Next, the model can be fit on the training dataset by calling the *train()* function and passing in both the training dataset and the validation dataset. We can also specify the learning rate as the default learning rate in the configuration (0.001).

We can also specify what layers to train. In this case, we will only train the output layers of the model.

[Start Machine Learning](#)

```
1 # train weights (output layers or 'heads')
2 model.train(train_set, test_set, learning_rate=config.LEARNING_RATE, epochs=5, layers='heads')
```

We could follow this training with further epochs that fine-tune all of the weights in the model. This could be achieved by using a smaller learning rate and changing the 'layer' argument from 'heads' to 'all'.

The complete example of training a Mask R-CNN on the kangaroo dataset is listed below.

This may take some time to execute on the CPU, even with modern hardware. I recommend running the code with a GPU, such as on [Amazon EC2](#), where it will finish in about five minutes on a P3 type hardware.

```
# fit a mask rcnn on the kangaroo dataset
from os import listdir
from xml.etree import ElementTree
from numpy import zeros
from numpy import asarray
from mrcnn.utils import Dataset
from mrcnn.config import Config
from mrcnn.model import MaskRCNN

# class that defines and loads the kangaroo dataset
class KangarooDataset(Dataset):
    # load the dataset definitions
    def load_dataset(self, dataset_dir, is_train=True):
        # define one class
        self.add_class("dataset", 1, "kangaroo")
        # define data locations
        images_dir = dataset_dir + '/images/'
        annotations_dir = dataset_dir + '/annots/'
        # find all images
        for filename in listdir(images_dir):
            # extract image id
            image_id = filename[:-4]
            # skip bad images
            if image_id in ['00090']:
                continue
            # skip all images after 150 if we are building the train set
            if is_train and int(image_id) >= 150:
                continue
            # skip all images before 150 if we are building the test/val set
            if not is_train and int(image_id) < 150:
                continue
            img_path = images_dir + filename
            ann_path = annotations_dir + image_id + '.xml'
            # add to dataset
            self.add_image('dataset', image_id=image_id, path=img_path, annotation=ann_path)

    # extract bounding boxes from an annotation file
    def extract_boxes(self, filename):
        # load and parse the file
        tree = ElementTree.parse(filename)
        # get the root of the document
        root = tree.getroot()
        # extract each bounding box
        boxes = list()
        for box in root.findall('.//bndbox'):
            xmin = int(box.find('xmin').text)
            ymin = int(box.find('ymin').text)
            xmax = int(box.find('xmax').text)
            ymax = int(box.find('ymax').text)
            coors = [xmin, ymin, xmax, ymax]
```

## Start Machine Learning

You can master applied Machine Learning **without math or fancy degrees**. Find out how in this *free* and *practical* course.

Email Address

[START MY EMAIL COURSE](#)

[Start Machine Learning](#)

```

        boxes.append(coors)
    # extract image dimensions
    width = int(root.find('.//size/width').text)
    height = int(root.find('.//size/height').text)
    return boxes, width, height

# load the masks for an image
def load_mask(self, image_id):
    # get details of image
    info = self.image_info[image_id]
    # define box file location
    path = info['annotation']
    # load XML
    boxes, w, h = self.extract_boxes(path)
    # create one array for all masks, each on a different channel
    masks = zeros([h, w, len(boxes)], dtype='uint8')
    # create masks
    class_ids = list()
    for i in range(len(boxes)):
        box = boxes[i]
        row_s, row_e = box[1], box[3]
        col_s, col_e = box[0], box[2]
        masks[row_s:row_e, col_s:col_e, i] =
            class_ids.append(self.class_names.in
    return masks, asarray(class_ids, dtype='i

# load an image reference
def image_reference(self, image_id):
    info = self.image_info[image_id]
    return info['path']

# define a configuration for the model
class KangarooConfig(Config):
    # define the name of the configuration
    NAME = "kangaroo_cfg"
    # number of classes (background + kangaroo)
    NUM_CLASSES = 1 + 1
    # number of training steps per epoch
    STEPS_PER_EPOCH = 131

# prepare train set
train_set = KangarooDataset()
train_set.load_dataset('kangaroo', is_train=True)
train_set.prepare()
print('Train: %d' % len(train_set.image_ids))
# prepare test/val set
test_set = KangarooDataset()
test_set.load_dataset('kangaroo', is_train=False)
test_set.prepare()
print('Test: %d' % len(test_set.image_ids))
# prepare config
config = KangarooConfig()
config.display()
# define the model
model = MaskRCNN(mode='training', model_dir='./', config=config)

```

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

[START MY EMAIL COURSE](#)

Running the example will report progress using the standard Keras progress bars.

[Start Machine Learning](#)

We can see that there are many different train and test loss scores reported for each of the output heads of the network. It can be quite confusing as to which loss to pay attention to.

In this example where we are interested in object detection instead of object segmentation, I recommend paying attention to the loss for the classification output on the train and validation datasets (e.g. `mrcnn_class_loss` and `val_mrcnn_class_loss`), as well as the loss for the bounding box output for the train and validation datasets (`mrcnn_bbox_loss` and `val_mrcnn_bbox_loss`).

```

1 Epoch 1/5
2 131/131 [=====] - 106s 811ms/step - loss: 0.8491 - rpn_class_loss: 0.0
3 Epoch 2/5
4 131/131 [=====] - 69s 526ms/step - loss: 0.4774 - rpn_class_loss: 0.00
5 Epoch 3/5
6 131/131 [=====] - 67s 515ms/step - loss: 0.4171 - rpn_class_loss: 0.00
7 Epoch 4/5
8 131/131 [=====] - 66s 515ms/step - loss: 0.4171 - rpn_class_loss: 0.00
9 Epoch 5/5
10 131/131 [=====] - 65s 515ms/step - loss: 0.4171 - rpn_class_loss: 0.00

```

A model file is created and saved at the end of each epoch, followed by random characters.

A model must be selected for use; in this case, the loss is printed at the end of each epoch, so we will use the final model at the end of training.

Copy the model file from the config directory into your project and follow the following sections to evaluate the model and make predictions.

## Start Machine Learning

You can master applied Machine Learning  
without math or fancy degrees.

Find out how in this *free* and *practical* course.

Email Address

[START MY EMAIL COURSE](#)

The results suggest that perhaps more training epochs could be useful, perhaps fine-tuning all of the layers in the model; this might make an interesting extension to the tutorial.

Next, let's look at evaluating the performance of this model.

## How to Evaluate a Mask R-CNN Model

The performance of a model for an object recognition task is often evaluated using the mean absolute precision, or mAP.

We are predicting bounding boxes so we can determine whether a bounding box prediction is good or not based on how well the predicted and actual bounding boxes overlap. This can be calculated by dividing the area of the overlap by the total area of both bounding boxes, or the intersection divided by the union, referred to as "*intersection over union*," or IoU. A perfect bounding box prediction will have an IoU of 1.

It is standard to assume a positive prediction of a bounding box if the IoU is greater than 0.5, e.g. they overlap by 50% or more.

Precision refers to the percentage of the correctly predicted bounding boxes ( $\text{IoU} > 0.5$ ) out of all bounding boxes predicted. Recall is the percentage of the correctly predicted bounding boxes ( $\text{IoU} > 0.5$ ) out of all objects in the photo.

[Start Machine Learning](#)

As we make more predictions, the recall percentage will increase, but precision will drop or become erratic as we start making false positive predictions. The recall ( $x$ ) can be plotted against the precision ( $y$ ) for each number of predictions to create a curve or line. We can maximize the value of each point on this line and calculate the average value of the precision or AP for each value of recall.

**Note:** there are variations on how AP is calculated, e.g. the way it is calculated for the widely used PASCAL VOC dataset and the MS COCO dataset differ.

The average or mean of the average precision (AP) across all of the images in a dataset is called the mean average precision, or mAP.

The mask-rcnn library provides a `mrcnn.utils.compute_ap` to calculate the AP and other metrics for a given images. These AP scores can be collected across a dataset to calculate the mAP, which is a good metric for how good the model is at detecting objects in a dataset.

First, we must define a new `Config` object to use for making predictions. We can reuse our previously defined `KangarooConfig` to reuse the parameters and values. We can use the same values to keep the code compact. The configuration parameters for the GPU for inference that are different from how they are defined in the training code are running on the GPU or CPU).

```
1 # define the prediction configuration
2 class PredictionConfig(Config):
3     # define the name of the configuration
4     NAME = "kangaroo_cfg"
5     # number of classes (background + kangaroo)
6     NUM_CLASSES = 1 + 1
7     # simplify GPU config
8     GPU_COUNT = 1
9     IMAGES_PER_GPU = 1
```

Next, we can define the model with the config and set the ‘`mode`’ argument to ‘`inference`’ instead of ‘`training`’.

```
1 # create config
2 cfg = PredictionConfig()
3 # define the model
4 model = MaskRCNN(mode='inference', model_dir='./', config=cfg)
```

Next, we can load the weights from our saved model.

We can do that by specifying the path to the model file. In this case, the model file is ‘`mask_rcnn_kangaroo_cfg_0005.h5`’ in the current working directory.

```
1 # load model weights
2 model.load_weights('mask_rcnn_kangaroo_cfg_0005.h5', by_name=True)
```

Next, we can evaluate the model. This involves enumerating the images in a dataset, making a prediction, and calculating the AP for the prediction before predicting a mean AP across all images.

First, the image and ground truth mask can be loaded from the dataset. This can be achieved using the `load_image_gt()` convenience function.

## Start Machine Learning X

You can master applied Machine Learning  
**without math or fancy degrees.**  
Find out how in this *free* and *practical* course.

Email Address

[START MY EMAIL COURSE](#)

```
1 # load image, bounding boxes and masks for the image id
2 image, image_meta, gt_class_id, gt_bbox, gt_mask = load_image_gt(dataset, cfg, image_id, use_mir
```

Next, the pixel values of the loaded image must be scaled in the same way as was performed on the training data, e.g. centered. This can be achieved using the `mold_image()` convenience function.

```
1 # convert pixel values (e.g. center)
2 scaled_image = mold_image(image, cfg)
```

The dimensions of the image then need to be expanded one sample in a dataset and used as input to make a prediction with the model.

```
1 sample = expand_dims(scaled_image, 0)
2 # make prediction
3 yhat = model.detect(sample, verbose=0)
4 # extract results for first sample
5 r = yhat[0]
```

Next, the prediction can be compared to the ground truth using the `compute_ap()` function.

```
1 # calculate statistics, including AP
2 AP, _, _, _ = compute_ap(gt_bbox, gt_class_id,
```

The AP values can be added to a list, then the mean value calculated.

Tying this together, the `evaluate_model()` function below takes a dataset, model and configuration.

```
1 # calculate the mAP for a model on a given dataset
2 def evaluate_model(dataset, model, cfg):
3     APs = list()
4     for image_id in dataset.image_ids:
5         # load image, bounding boxes and masks for the image id
6         image, image_meta, gt_class_id, gt_bbox, gt_mask = load_image_gt(dataset, cfg, image_id)
7         # convert pixel values (e.g. center)
8         scaled_image = mold_image(image, cfg)
9         # convert image into one sample
10        sample = expand_dims(scaled_image, 0)
11        # make prediction
12        yhat = model.detect(sample, verbose=0)
13        # extract results for first sample
14        r = yhat[0]
15        # calculate statistics, including AP
16        AP, _, _, _ = compute_ap(gt_bbox, gt_class_id, gt_mask, r["rois"], r["class_ids"], r["scores"])
17        # store
18        APs.append(AP)
19    # calculate the mean AP across all images
20    mAP = mean(APs)
21    return mAP
```

We can now calculate the mAP for the model on the train and test datasets.

```
1 # evaluate model on training dataset
2 train_mAP = evaluate_model(train_set, model, cfg)
3 print("Train mAP: %.3f" % train_mAP)
4 # evaluate model on test dataset
5 test_mAP = evaluate_model(test_set, model, cfg)
6 print("Test mAP: %.3f" % test_mAP)
```

## Start Machine Learning

You can master applied Machine Learning without math or fancy degrees.

Find out how in this *free* and *practical* course.

[START MY EMAIL COURSE](#)

The full code listing is provided below for completeness.

```
# evaluate the mask rcnn model on the kangaroo dataset
from os import listdir
from xml.etree import ElementTree
from numpy import zeros
from numpy import asarray
from numpy import expand_dims
from numpy import mean
from mrcnn.config import Config
from mrcnn.model import MaskRCNN
from mrcnn.utils import Dataset
from mrcnn.utils import compute_ap
from mrcnn.model import load_image_gt
from mrcnn.model import mold_image

# class that defines and loads the kangaroo dataset
class KangarooDataset(Dataset):
    # load the dataset definitions
    def load_dataset(self, dataset_dir, is_train=True):
        # define one class
        self.add_class("dataset", 1, "kangaroo")
        # define data locations
        images_dir = dataset_dir + '/images/'
        annotations_dir = dataset_dir + '/annots/'
        # find all images
        for filename in listdir(images_dir):
            # extract image id
            image_id = filename[:-4]
            # skip bad images
            if image_id in ['00090']:
                continue
            # skip all images after 150 if we are building the train set
            if is_train and int(image_id) >= 150:
                continue
            # skip all images before 150 if we are building the test/val set
            if not is_train and int(image_id) < 150:
                continue
            img_path = images_dir + filename
            ann_path = annotations_dir + image_id + '.xml'
            # add to dataset
            self.add_image('dataset', image_id=image_id, path=img_path, annotation=ann_path)

    # extract bounding boxes from an annotation file
    def extract_boxes(self, filename):
        # load and parse the file
        tree = ElementTree.parse(filename)
        # get the root of the document
        root = tree.getroot()
        # extract each bounding box
        boxes = list()
        for box in root.findall('.//bndbox'):
            xmin = int(box.find('xmin').text)
            ymin = int(box.find('ymin').text)
            xmax = int(box.find('xmax').text)
            ymax = int(box.find('ymax').text)
            coors = [xmin, ymin, xmax, ymax]
            boxes.append(coors)
        # extract image dimensions
        width = int(root.find('.//size/width').text)
        height = int(root.find('.//size/height').text)
        return boxes, width, height

    # load the masks for an image
    def load_mask(self, image_id):
```

## Start Machine Learning X

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

[START MY EMAIL COURSE](#)

```

def load_masks(self, image_id):
    # get details of image
    info = self.image_info[image_id]
    # define box file location
    path = info['annotation']
    # load XML
    boxes, w, h = self.extract_boxes(path)
    # create one array for all masks, each on a different channel
    masks = zeros([h, w, len(boxes)], dtype='uint8')
    # create masks
    class_ids = list()
    for i in range(len(boxes)):
        box = boxes[i]
        row_s, row_e = box[1], box[3]
        col_s, col_e = box[0], box[2]
        masks[row_s:row_e, col_s:col_e, i] = 1
        class_ids.append(self.class_names.index('kangaroo'))
    return masks, asarray(class_ids, dtype='int32')

# load an image reference
def image_reference(self, image_id):
    info = self.image_info[image_id]
    return info['path']

# define the prediction configuration
class PredictionConfig(Config):
    # define the name of the configuration
    NAME = "kangaroo_cfg"
    # number of classes (background + kangaroo)
    NUM_CLASSES = 1 + 1
    # simplify GPU config
    GPU_COUNT = 1
    IMAGES_PER_GPU = 1

# calculate the mAP for a model on a given dataset
def evaluate_model(dataset, model, cfg):
    APs = list()
    for image_id in dataset.image_ids:
        # load image, bounding boxes and masks for the image id
        image, image_meta, gt_class_id, gt_bbox, gt_mask = load_image_gt(dataset, cfg, image_id, u
        # convert pixel values (e.g. center)
        scaled_image = mold_image(image, cfg)
        # convert image into one sample
        sample = expand_dims(scaled_image, 0)
        # make prediction
        yhat = model.detect(sample, verbose=0)
        # extract results for first sample
        r = yhat[0]
        # calculate statistics, including AP
        AP, _, _, _ = compute_ap(gt_bbox, gt_class_id, gt_mask, r["rois"], r["class_ids"], r["scor
        # store
        APs.append(AP)
    # calculate the mean AP across all images
    mAP = mean(APs)
    return mAP

# load the train dataset
train_set = KangarooDataset()
train_set.load_dataset('kangaroo', is_train=True)
train_set.prepare()
print('Train: %d' % len(train_set.image_ids))
# load the test dataset
test_set = KangarooDataset()
test_set.load_dataset('kangaroo', is_train=False)
test_set.prepare()
print('Test: %d' % len(test_set.image_ids))

```

## Start Machine Learning

You can master applied Machine Learning  
without math or fancy degrees.

Find out how in this *free* and *practical* course.

[START MY EMAIL COURSE](#)

```
# create config
cfg = PredictionConfig()
# define the model
model = MaskRCNN(mode='inference', model_dir='./', config=cfg)
# load model weights
model.load_weights('mask_rcnn_kangaroo_cfg_0005.h5', by_name=True)
# evaluate model on training dataset
```

Running the example will make a prediction for each image in the train and test datasets and calculate the mAP for each.

A mAP above 90% or 95% is a good score. We can see perhaps slightly better on the test dataset, instead of the training.

This may be because the dataset is very small, and/or the training.

[1 Train mAP: 0.929](#)  
[2 Test mAP: 0.958](#)

Now that we have some confidence that the model is working, let's move on to making predictions.

## Start Machine Learning

You can master applied Machine Learning **without math or fancy degrees**.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE

## How to Detect Kangaroos in New Photos

We can use the trained model to detect kangaroos in new photographs, specifically, in photos that we expect to have kangaroos.

First, we need a new photo of a kangaroo.

We could go to Flickr and find a random photo of a kangaroo. Alternately, we can use any of the photos in the test dataset that were not used to train the model.

We have already seen in the previous section how to make a prediction with an image. Specifically, scaling the pixel values and calling *model.detect()*. For example:

```
1 # example of making a prediction
2 ...
3 # load image
4 image = ...
5 # convert pixel values (e.g. center)
6 scaled_image = mold_image(image, cfg)
7 # convert image into one sample
8 sample = expand_dims(scaled_image, 0)
9 # make prediction
10 yhat = model.detect(sample, verbose=0)
11 ...
```

Let's take it one step further and make predictions for multiple images at once. This will allow us to make predictions on a photo with bounding boxes side-by-side with the photo and the original photo.

[Start Machine Learning](#)

visual guide to how good the model is at making predictions.

The first step is to load the image and mask from the dataset.

```
1 # load the image and mask
2 image = dataset.load_image(image_id)
3 mask, _ = dataset.load_mask(image_id)
```

Next, we can make a prediction for the image.

```
1 # convert pixel values (e.g. center)
2 scaled_image = mold_image(image, cfg)
3 # convert image into one sample
4 sample = expand_dims(scaled_image, 0)
5 # make prediction
6 yhat = model.detect(sample, verbose=0)[0]
```

Next, we can create a subplot for the ground truth and

```
1 # define subplot
2 pyplot.subplot(n_images, 2, i*2+1)
3 # plot raw pixel data
4 pyplot.imshow(image)
5 pyplot.title('Actual')
6 # plot masks
7 for j in range(mask.shape[2]):
8     pyplot.imshow(mask[:, :, j], cmap='gray',
```

We can then create a second subplot beside the first and draw the predicted bounding boxes in red.

```
1 # get the context for drawing boxes
2 pyplot.subplot(n_images, 2, i*2+2)
3 # plot raw pixel data
4 pyplot.imshow(image)
5 pyplot.title('Predicted')
6 ax = pyplot.gca()
7 # plot each box
8 for box in yhat['rois']:
9     # get coordinates
10    y1, x1, y2, x2 = box
11    # calculate width and height of the box
12    width, height = x2 - x1, y2 - y1
13    # create the shape
14    rect = Rectangle((x1, y1), width, height, fill=False, color='red')
15    # draw the box
16    ax.add_patch(rect)
```

We can tie all of this together into a function that takes a dataset, model, and config and creates a plot of the first five photos in the dataset with ground truth and predicted bound boxes.

```
1 # plot a number of photos with ground truth and predictions
2 def plot_actual_vs_predicted(dataset, model, cfg, n_images=5):
3     # load image and mask
4     for i in range(n_images):
5         # load the image and mask
6         image = dataset.load_image(i)
7         mask, _ = dataset.load_mask(i)
8         # convert pixel values (e.g. center)
9         scaled_image = mold_image(image, cfg)
```

## Start Machine Learning

You can master applied Machine Learning  
without math or fancy degrees.

Find out how in this *free* and *practical* course.

[START MY EMAIL COURSE](#)

```

10 # convert image into one sample
11 sample = expand_dims(scaled_image, 0)
12 # make prediction
13 yhat = model.detect(sample, verbose=0)[0]
14 # define subplot
15 pyplot.subplot(n_images, 2, i*2+1)
16 # plot raw pixel data
17 pyplot.imshow(image)
18 pyplot.title('Actual')
19 # plot masks
20 for j in range(mask.shape[2]):
21     pyplot.imshow(mask[:, :, j], cmap='gray', alpha=0.3)
22 # get the context for drawing boxes
23 pyplot.subplot(n_images, 2, i*2+2)
24 # plot raw pixel data
25 pyplot.imshow(image)
26 pyplot.title('Predicted')
27 ax = pyplot.gca()
28 # plot each box
29 for box in yhat['rois']:
30     # get coordinates
31     y1, x1, y2, x2 = box
32     # calculate width and height of t
33     width, height = x2 - x1, y2 - y1
34     # create the shape
35     rect = Rectangle((x1, y1), width,
36     # draw the box
37     ax.add_patch(rect)
38 # show the figure
39 pyplot.show()

```

The complete example of loading the trained model and train and test datasets is listed below.

```

# detect kangaroos in photos with mask rcnn model
from os import listdir
from xml.etree import ElementTree
from numpy import zeros
from numpy import asarray
from numpy import expand_dims
from matplotlib import pyplot
from matplotlib.patches import Rectangle
from mrcnn.config import Config
from mrcnn.model import MaskRCNN
from mrcnn.model import mold_image
from mrcnn.utils import Dataset

# class that defines and loads the kangaroo dataset
class KangarooDataset(Dataset):
    # load the dataset definitions
    def load_dataset(self, dataset_dir, is_train=True):
        # define one class
        self.add_class("dataset", 1, "kangaroo")
        # define data locations
        images_dir = dataset_dir + '/images/'
        annotations_dir = dataset_dir + '/annots/'
        # find all images
        for filename in listdir(images_dir):
            # extract image id
            image_id = filename[:-4]
            # skip bad images
            if image_id in ['00090']:
                continue
            # skip all images after 150 if we are

```

## Start Machine Learning

You can master applied Machine Learning  
without math or fancy degrees.

Find out how in this *free* and *practical* course.

[START MY EMAIL COURSE](#)

```

        if is_train and int(image_id) >= 150:
            continue
        # skip all images before 150 if we are building the test/val set
        if not is_train and int(image_id) < 150:
            continue
        img_path = images_dir + filename
        ann_path = annotations_dir + image_id + '.xml'
        # add to dataset
        self.add_image('dataset', image_id=image_id, path=img_path, annotation=ann_path)

# load all bounding boxes for an image
def extract_boxes(self, filename):
    # load and parse the file
    root = ElementTree.parse(filename)
    boxes = list()
    # extract each bounding box
    for box in root.findall('.//bndbox'):
        xmin = int(box.find('xmin').text)
        ymin = int(box.find('ymin').text)
        xmax = int(box.find('xmax').text)
        ymax = int(box.find('ymax').text)
        coors = [xmin, ymin, xmax, ymax]
        boxes.append(coors)
    # extract image dimensions
    width = int(root.find('.//size/width').text)
    height = int(root.find('.//size/height').text)
    return boxes, width, height

# load the masks for an image
def load_mask(self, image_id):
    # get details of image
    info = self.image_info[image_id]
    # define box file location
    path = info['annotation']
    # load XML
    boxes, w, h = self.extract_boxes(path)
    # create one array for all masks, each on a different channel
    masks = zeros([h, w, len(boxes)], dtype='uint8')
    # create masks
    class_ids = list()
    for i in range(len(boxes)):
        box = boxes[i]
        row_s, row_e = box[1], box[3]
        col_s, col_e = box[0], box[2]
        masks[row_s:row_e, col_s:col_e, i] = 1
        class_ids.append(self.class_names.index('kangaroo'))
    return masks, asarray(class_ids, dtype='int32')

# load an image reference
def image_reference(self, image_id):
    info = self.image_info[image_id]
    return info['path']

# define the prediction configuration
class PredictionConfig(Config):
    # define the name of the configuration
    NAME = "kangaroo_cfg"
    # number of classes (background + kangaroo)
    NUM_CLASSES = 1 + 1
    # simplify GPU config
    GPU_COUNT = 1
    IMAGES_PER_GPU = 1

# plot a number of photos with ground truth and predicted
def plot_actual_vs_predicted(dataset, model, cfg,
    # Load image and mask

```

## Start Machine Learning

You can master applied Machine Learning  
without math or fancy degrees.

Find out how in this *free* and *practical* course.

[START MY EMAIL COURSE](#)

```

# load image and mask
for i in range(n_images):
    # load the image and mask
    image = dataset.load_image(i)
    mask, _ = dataset.load_mask(i)
    # convert pixel values (e.g. center)
    scaled_image = mold_image(image, cfg)
    # convert image into one sample
    sample = expand_dims(scaled_image, 0)
    # make prediction
    yhat = model.detect(sample, verbose=0)[0]
    # define subplot
    pyplot.subplot(n_images, 2, i*2+1)
    # plot raw pixel data
    pyplot.imshow(image)
    pyplot.title('Actual')
    # plot masks
    for j in range(mask.shape[2]):
        pyplot.imshow(mask[:, :, j], cmap='gray')
    # get the context for drawing boxes
    pyplot.subplot(n_images, 2, i*2+2)
    # plot raw pixel data
    pyplot.imshow(image)
    pyplot.title('Predicted')
    ax = pyplot.gca()
    # plot each box
    for box in yhat['rois']:
        # get coordinates
        y1, x1, y2, x2 = box
        # calculate width and height of the box
        width, height = x2 - x1, y2 - y1
        # create the shape
        rect = Rectangle((x1, y1), width, height)
        # draw the box
        ax.add_patch(rect)
    # show the figure
    pyplot.show()

# load the train dataset
train_set = KangarooDataset()
train_set.load_dataset('kangaroo', is_train=True)
train_set.prepare()
print('Train: %d' % len(train_set.image_ids))
# load the test dataset
test_set = KangarooDataset()
test_set.load_dataset('kangaroo', is_train=False)
test_set.prepare()
print('Test: %d' % len(test_set.image_ids))
# create config
cfg = PredictionConfig()
# define the model
model = MaskRCNN(mode='inference', model_dir='./', config=cfg)
# load model weights
-----
```

## Start Machine Learning

You can master applied Machine Learning  
without math or fancy degrees.

Find out how in this *free* and *practical* course.

[START MY EMAIL COURSE](#)

Running the example first creates a figure showing five photos from the training dataset with the ground truth bounding boxes, with the same photo and the predicted bounding boxes alongside.

[Start Machine Learning](#)

We can see that the model has done well on these examples, finding all of the kangaroos, even in the case where there are two or three in one photo. The second photo down (in the right column) does show a slip-up where the model has predicted a bounding box around the same kangaroo twice.



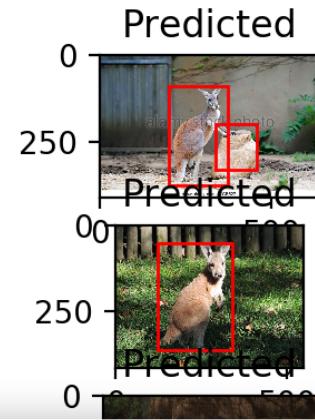
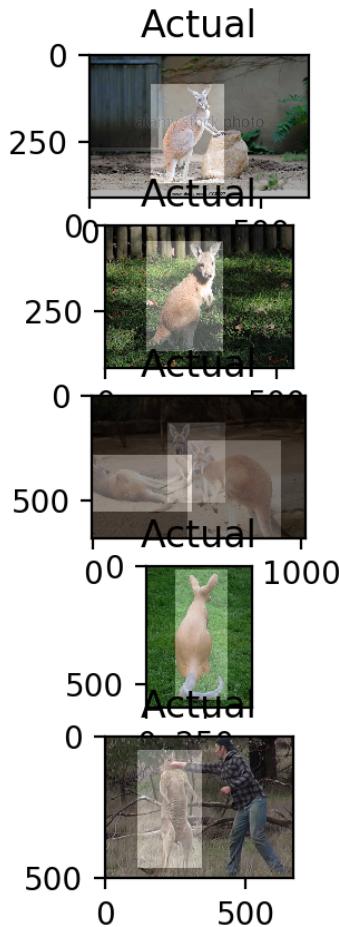
Plot of Photos of Kangaroos From the Training Dataset With Ground Truth and Predicted Bounding Boxes

A second figure is created showing five photos from the test dataset with ground truth bounding boxes and predicted bounding boxes.

These are images not seen during training, and again, in each photo, the model has detected the kangaroo. We can see that in the case of the second last photo that a minor mistake was made. Specifically, the same kangaroo was detected multiple times.

No doubt these differences can be ironed out with more training, perhaps with a larger dataset and/or data augmentation, to encourage the model to detect people as background and to detect a given kangaroo once only.

[Start Machine Learning](#)



Plot of Photos of Kangaroos From the Training Dataset With Ground Truth and Predicted Bounding Boxes

## Start Machine Learning

You can master applied Machine Learning  
without math or fancy degrees.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE

## Further Reading

This section provides more resources on the topic if you are looking to go deeper.

## Papers

- Mask R-CNN, 2017.

## Projects

- Kangaroo Dataset, GitHub.
- Mask RCNN Project, GitHub.

## APIs

- `xml.etree.ElementTree API`
- `matplotlib.patches.Rectangle API`
- `matplotlib.pyplot.subplot API`
- `matplotlib.pyplot.imshow API`

[Start Machine Learning](#)

## Articles

- Splash of Color: Instance Segmentation with Mask R-CNN and TensorFlow, 2018.
- Mask R-CNN – Inspect Ballon Trained Model, Notebook.
- Mask R-CNN – Train on Shapes Dataset, Notebook.
- mAP (mean Average Precision) for Object Detection, 2018.

## Summary

In this tutorial, you discovered how to develop a Mask R-CNN model for kangaroo object detection in photographs.

Specifically, you learned:

- How to prepare an object detection dataset ready
- How to use transfer learning to train an object det
- How to evaluate a fit Mask R-CNN model on a tes

Do you have any questions?

Ask your questions in the comments below and I will c

## Start Machine Learning

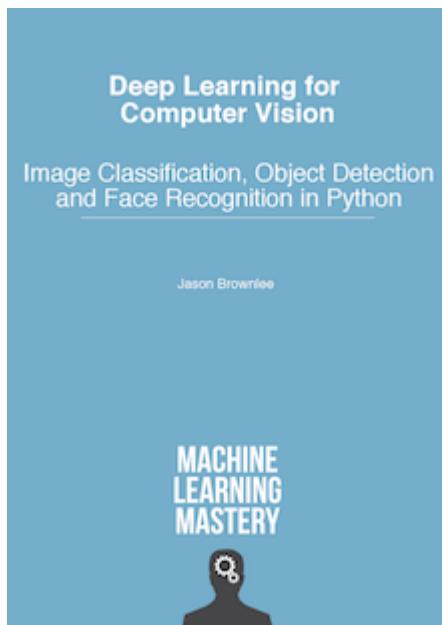
You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE

## Develop Deep Learning Models for Vision Today.



### Develop Your Own Vision Models in Minutes

...with just a few lines of python code

Discover how in my new Ebook:  
*Deep Learning for Computer Vision*

It provides **self-study tutorials** on topics like:  
*classification, object detection (yolo and rcnn), face recognition (vggface and facenet), data preparation* and much more...

### Finally Bring Deep Learning to your Vision Projects

Skip the Academics. Just Results.

[SEE WHAT'S INSIDE](#)

Tweet

Share

Share

## About Jason Brownlee

[Start Machine Learning](#)



Jason Brownlee, PhD is a machine learning specialist who teaches developers how to get results with modern machine learning methods via hands-on tutorials.

[View all posts by Jason Brownlee →](#)

< How to Perform Object Detection With YOLOv3 in Keras

A Gentle Introduction to Deep Learning for Face Recognition >

## 385 Responses to *How to Train an Object Detection Model with Keras*



**Milemi** June 1, 2019 at 5:38 am #

Great tutorial !

Could you give us advice how to annotate images, please?  
What is the best practice ?  
How many images per object is enough ?  
How to annotate when there are several objects in the same image ?  
Thank you.

### Start Machine Learning

You can master applied Machine Learning **without math or fancy degrees**.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE

X



**Jason Brownlee** June 1, 2019 at 6:17 am #

REPLY ↗

Great questions, thanks!

I hope to cover the topic in the future.



**Usama Ahmed** October 7, 2019 at 8:40 pm #

REPLY ↗

Here is the image annotation tool.

<https://github.com/tzutalin/labellmg>



**Jason Brownlee** October 8, 2019 at 7:59 am #

REPLY ↗

Thanks for sharing.

**simonYU** June 4, 2019 at 6:05 pm #

### Start Machine Learning



hi, Jason, while display\_instances:

running : display\_instances(image, bbox, mask, class\_ids, train\_set.class\_names)

An error occurred while starting the kernel ,

home/user/anaconda3/bin/python: symbol lookup error:

/home/user/anaconda3/lib/python3.6/site-packages/numpy/core/../../../../libmkl\_intel\_thread.so: undefined symbol: \_\_kmpc\_global\_thread\_num

Pls find the solution .

thanks



**Jason Brownlee** June 5, 2019 at 8:33 am #

This looks like it might be an issue with your development environment.

Perhaps this post will help you to setup your development environment:

<https://machinelearningmastery.com/setup-python-environment-for-machine-learning-development/>



**simonYU** June 5, 2019 at 8:12 pm #

yes!thanks for your guide !and now , it works well on Win10 plateform although still running for much much more time waiting .

The issue is focused on the MKL lib .

But for Ubuntu issue remained .....

## Start Machine Learning

You can master applied Machine Learning **without math or fancy degrees**.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** June 6, 2019 at 6:23 am #

REPLY ↗

Nice work!



**SimonYu** June 10, 2019 at 12:16 pm #

hi,Jason, thank for your kind tutorials , and for this case-study,  
what is the function of train\_set.prepare()?

Please provide much more of HowTo about it ? thanks !



**Jason Brownlee** June 10, 2019 at 12:16 pm #

Start Machine Learning

Good question, you can see what prepare() does here:

[https://github.com/matterport/Mask\\_RCNN/blob/master/mrcnn/utils.py#L294](https://github.com/matterport/Mask_RCNN/blob/master/mrcnn/utils.py#L294)



**Amine** December 23, 2019 at 1:02 am #

REPLY ↗

Hi any developments?



**roopesh** June 4, 2019 at 6:19 pm #

REPLY ↗

very nice steps !! How to predict with real time



**Jason Brownlee** June 5, 2019 at 8:33 am #

Great suggestion, I hope to cover it in the



**Usama Ahmed** September 26, 2019 at 11:59 pm #

Use OpenCV to capture video from attached camera.

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** September 27, 2019 at 8:02 am #

REPLY ↗

Agreed!



**maryam** June 8, 2019 at 5:02 am #

REPLY ↗

Hi Jason,

Thank you very much for the precious tutorial. I face a problem in people counting project when I am going to track people though detecting them is not hard.

would you please give me a tutorial about the best tracking methods such as “deep tracking” or other else?  
Best

Maryam



**Jason Brownlee** June 8, 2019 at 7:05 am #

REPLY ↗

Start Machine Learning

Thanks for the suggestion.



**gary** June 19, 2019 at 8:51 pm #

REPLY ↗

Thank you very much for such a beautiful yet detailed tutorial. Its been great learning from you.



**Jason Brownlee** June 20, 2019 at 8:31 am #

REPLY ↗

Thanks, I'm glad it helped.



**gary** June 24, 2019 at 10:27 pm #

Hi jason, i am trying to train multiple classes?

Do i use multiple lines of:

```
self.add_class("dataset", 1, "kangaroo")
self.add_class("dataset", 2, "tiger")?
```

## Start Machine Learning

You can master applied Machine Learning **without math or fancy degrees**.

Find out how in this *free* and *practical* course.

START MY EMAIL COURSE



**Jason Brownlee** June 25, 2019 at 6:20 am #

REPLY ↗

You can specify all of your classes with a unique integer.



**Romell Domínguez** August 28, 2019 at 6:47 am #

Hi Jason, and then just add each image using:

```
self.add_class("dataset", 1, "kangaroo")
self.add_class("dataset", 2, "tiger")?
```

```
self.add_image('dataset', ... )
```

what parameter do i need set for identify 'the class'



**Romell Domínguez** August 30, 2019 at 8:25 pm #

i solve that problem for polygons shapes

## Start Machine Learning



**Jason Brownlee** August 31, 2019 at 6:04 am #

I'm happy to hear that, well done!



**Biki** November 8, 2019 at 12:28 am #

Please let me know how to do this



**Akshay** February 12, 2020 at 3:01 pm #

If we have both kangaroo and tiger mask?

```
self.add_class("dataset", 1, "kangaroo")
self.add_class("dataset", 2, "tiger")
```

I meant this part!!!

```
1  def load_mask(self, image_id):
2      # get details of image
3      info = self.image_info[image_id]
4      # define box file location
5      path = info['annotation']
6      # load XML
7      boxes, w, h = self.extract_boxes(path)
8      # create one array for all masks, each on a different channel
9      masks = zeros([h, w, len(boxes)], dtype='uint8')
10     # create masks
11     class_ids = list()
12     for i in range(len(boxes)):
13         box = boxes[i]
14         row_s, row_e = box[1], box[3]
15         col_s, col_e = box[0], box[2]
16         masks[row_s:row_e, col_s:col_e, i] = 1
17         class_ids.append(self.class_names.index('kangaroo'))
18     return masks, asarray(class_ids, dtype='int32')
```

## Start Machine Learning

You can master applied Machine Learning **without math or fancy degrees**.

Find out how in this *free* and *practical* course.

[START MY EMAIL COURSE](#)



**Jason Brownlee** February 12, 2020 at 5:50 am #

Sorry, I cannot review/debug your code.



**marry** June 20, 2019 at 12:27 pm #

[REPLY ↗](#)

ValueError: Dimension 1 in both shapes must be equal, but are 8 and 16. Shapes are [1024, 8] and [1024,16]. for 'Assign\_682' (op: 'Assign') with input shapes

[Start Machine Learning](#)

hello,Jason,How to solve this error when calculating the mAP value?



**Jason Brownlee** June 20, 2019 at 2:00 pm #

REPLY ↗

Sorry to hear that, I have some suggestions here that might help:

<https://machinelearningmastery.com/faq/single-faq/why-does-the-code-in-the-tutorial-not-work-for-me>



**mahmoud** July 10, 2019 at 6:30 pm #

REPLY ↗

hi jason,

I want to inquire about this file ~mask\_rcnn\_kangaroo.h5  
how i can find it also why you seprate the training and testing datasets,  
,I mean at the last version of file it contains only the pre-trained weights and it does not save the new weights after training so it can be used



**Jason Brownlee** July 11, 2019 at 9:46 am #

The model is fit on the training dataset, save the trained weights and then load them out test dataset.

Does that help?

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**mahmoud** July 11, 2019 at 6:18 pm #

REPLY ↗

```
ya but my question before train on dataset kangaroo i load weights to model
# load weights (mscoco) and exclude the output layers
model.load_weights('mask_rcnn_coco.h5', by_name=True, exclude=['mrcnn_class_logits',
'mrcnn_bbox_fc', "mrcnn_bbox", "mrcnn_mask"])
then after training
# load model weights
model.load_weights('mask_rcnn_kangaroo_cfg_0005.h5', by_name=True)
why we load the weights again
```

i have the file mask\_rcnn\_coco.h5, i think it has any initial weights ,but i do not know what is the file mask\_rcnn\_kangaroo\_cfg\_0005.h5 contains and where i can find this problem



**Jason Brownlee** July 12, 2019 at 8:32 pm #

REPLY ↗

Start Machine Learning

The new set of weights is focused on only detecting kangaroos based on our own dataset.

Does that help?



**mahmoud** July 12, 2019 at 5:58 pm #

ya but i can not find this new set of weights ,i mean when it creates the file mask\_rcnn\_kangaroo\_cfg\_0005.h5



**Jason Brownlee** July 13, 2019

It will be in the same directory



**mahmoud** July 14, 2019 at 7:25 am #

thnx for your response ,another question ,do i need to have the same structure of Kangaroo dataset to

## Start Machine Learning



You can master applied Machine Learning  
without math or fancy degrees.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** July 14, 2019 at 8:18 am #

It is not required, but it might be a helpful start if you are having trouble.



**mahmoud eltaher** July 15, 2019 at 7:27 pm #

ya I need to do this because I want to implement the model on my problem so I have some images with some circles and I want to detect these circles



**Wolverin** July 13, 2019 at 3:29 am #

REPLY ↗

same problem with me, i am using google colab.

this 'mask\_rcnn\_kangaroo\_cfg\_0005.h5' file is created while training as said in the blog. but i cannot find anywhere in my gdrive.

**Jason Brownlee** July 13, 2019 at 7:00 am #

Start Machine Learning



Perhaps try running on your workstation from the command line?



**kevi** October 8, 2019 at 1:03 am #

REPLY ↗

I have ran on Google colab and .h5 saved in (/content/Mask\_RCNN/kangaroo\_cfg\*) folder.  
Check



**zakaria** October 10, 2019 at 3:40 am #

REPLY ↗

Hi plz i am using colab google, i am havin



**Jason Brownlee** October 10, 2019 at 7:00 pm #

X

I don't know about colab, sorry.

Perhaps try posting on stackoverflow?

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jeremy Immanuel Putra Tandjung** July 16, 2019 at 2:25 pm #

REPLY ↗

Hello Jason,

First of all, nice tutorial! Having the overall code at the end of each step really helped keep track of where I am in the code! Keep up the good job!

I have a question, I notice that it took you on average a minute per epoch to train. However, I tried doing this with a different dataset and right now i'm on my first epoch and it's ETA 3.5 hours. My desktop is fairly fast with a ryzen 7 cpu and a nvidia 1050Ti gpu.

So is there something that I'm missing? My training dataset consist of 296 pictures of playing cards in different situations with a total file size of 30.4 MB (I'm trying to train a model to detect playing cards)

Or is that a normal? Or is there some setting I'm missing?

Thanks!



**Jason Brownlee** July 17, 2019 at 8:15 am #

REPLY ↗

It may be a factor of the number of images?

It may be hardware?

Perhaps experiment on some p3 EC2 instances or

Start Machine Learning



**Choi** July 16, 2019 at 4:51 pm #

REPLY ↗

Hi Jason.

This post is so helpful to me to learn R-CNN training!

As I do my work, I encounter some problems now.

First I train the model based on 'mask\_rcnn\_coco.h5' weight first.

So i got the model weight : 'mask\_rcnn\_carpk\_cfg\_0010.h5' file

how can i append more training images and train based on above file?

I just tried to append more images by load\_images fun  
load\_weights('mask\_rcnn\_carpk\_cfg\_0010.h5', by\_name=True)  
["mrcnn\_bbox\_fc", "mrcnn\_bbox", "mrcnn\_mask"])

But it did not work..

Is there any other things to set??

Thank you!!

## Start Machine Learning

X

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** July 17, 2019 at 8:18 am #

Good question, I don't have an example of this sorry. You may need to dive into the mask rcnn API.



**Choi** July 18, 2019 at 2:05 am #

REPLY ↗

hmm.. Please could you tell me some recommended papers or blogs about Mask R-CNN API for implementing my task?

Thank you!



**Jason Brownlee** July 18, 2019 at 8:32 am #

REPLY ↗

Perhaps start here:

<https://machinelearningmastery.com/how-to-perform-object-detection-in-photographs-with-mask-r-cnn-in-keras/>



**Nathan Starliper** July 17, 2019 at 5:33 am #

REPLY ↗

Start Machine Learning

Hi Jason,

Great tutorial. However, I am bit confused as to why you used Mask RCNN instead of Faster RCNN? Mask RCNN is essentially Faster RCNN except with segmentation added. Here in this example you basically converted the segmentation into bounding boxes so it seems to me that it would have saved you quite a bit of effort and manual labor to just use Faster RCNN model instead?

Thanks,

Nate



**Jason Brownlee** July 17, 2019 at 8:31 am #

REPLY ↗

Good question.

Optionality. We can do object detection which is useful if needed.



**SATYAM SAREEN** July 22, 2019 at 7:43 pm #

Great Tutorial Sir,

I really learned a lot.

I have a doubt regarding multiclass detection. I have 2

helmet. what changes should I make in the program? Like adding classes through add\_class function.

Huge Respect and Love.

Satyam Sareen

## Start Machine Learning

X

You can master applied Machine Learning without math or fancy degrees.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** July 23, 2019 at 8:00 am #

REPLY ↗

Perhaps this tutorial will help you train your model:

<https://machinelearningmastery.com/how-to-train-an-object-detection-model-with-keras/>



**SATYAM SAREEN** July 23, 2019 at 7:38 pm #

REPLY ↗

Good Afternoon Sir,

You have attached the link to the same blog. Can you suggest the changes to be made in your code so that it runs smoothly for multiclass object detection?

Warm Regards

Satyam Sareen

## Start Machine Learning



**Jason Brownlee** July 24, 2019 at 7:52 am #

REPLY ↗

What do you mean smoothly?



**AutoRoboCulture** November 18, 2019 at 3:05 am #

REPLY ↗

Hello Satyam Sareen,

Check out the code below, I have changed it to your requirement. If any query comment it down.  
Keep it up!

Code:

```
class KangarooDataset(Dataset):
    # load the dataset definitions
    def load_dataset(self, dataset_dir, is_train=True):
        # define two class
        self.add_class("dataset", 1, "personWithHe")
        self.add_class("dataset", 2, "personWithoutHe")
        # define data locations
        images_dir = dataset_dir + '/images/'
        annotations_dir = dataset_dir + '/annots/'
        # find all images

        for filename in.listdir(images_dir):
            # extract image id
            image_id = filename[:-4]
            #print('IMAGE ID: ',image_id)
            # skip all images after 90 if we are building the train set
            if is_train and int(image_id) >= 90: #set limit for your train and test set
                continue
            # skip all images before 90 if we are building the test/val set
            if not is_train and int(image_id) < 90:
                continue
            img_path = images_dir + filename
            ann_path = annotations_dir + image_id + '.xml'
            # add to dataset
            self.add_image('dataset', image_id=image_id, path=img_path, annotation=ann_path, class_ids=[0,1,2]) # for your case it is 0:BG, 1:PerWithHel.., 2:PersonWithoutHel... #Change required

            # extract bounding boxes from an annotation file
            def extract_boxes(self, filename):
                # load and parse the file
                tree = ElementTree.parse(filename)
                # get the root of the document
                root = tree.getroot()
                # extract each bounding box
```

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE

```

boxes = list()
#for box in root.findall('.//bndbox'):
for box in root.findall('.//object'): #Change required
    name = box.find('name').text #Change required
    xmin = int(box.find('./bndbox/xmin').text)
    ymin = int(box.find('./bndbox/ymin').text)
    xmax = int(box.find('./bndbox/xmax').text)
    ymax = int(box.find('./bndbox/ymax').text)
    #coors = [xmin, ymin, xmax, ymax, name]
    coors = [xmin, ymin, xmax, ymax, name] #Change required
    boxes.append(coors)
# extract image dimensions
width = int(root.find('.//size/width').text)
height = int(root.find('.//size/height').text)
return boxes, width, height

# load the masks for an image
def load_mask(self, image_id):
    # get details of image
    info = self.image_info[image_id]
    # define box file location
    path = info['annotation']
    # load XML
    boxes, w, h = self.extract_boxes(path)
    # create one array for all masks, each on a different channel
    masks = zeros([h, w, len(boxes)], dtype='uint8')
    # create masks
    class_ids = list()
    for i in range(len(boxes)):
        box = boxes[i]
        row_s, row_e = box[1], box[3]
        col_s, col_e = box[0], box[2]
        if (box[4] == 'personWithHelmet'): #Change required #change this to your .XML file
            masks[row_s:row_e, col_s:col_e, i] = 2 #Change required #assign number to your class_id
            class_ids.append(self.class_names.index('personWithHelmet')) #Change required
        else:
            masks[row_s:row_e, col_s:col_e, i] = 1 #Change required
            class_ids.append(self.class_names.index('personWithoutHelmet')) #Change required
    return masks, asarray(class_ids, dtype='int32')

# load an image reference
def image_reference(self, image_id):
    info = self.image_info[image_id]
    return info['path']

# define a configuration for the model
class KangarooConfig(Config):

```

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

[START MY EMAIL COURSE](#)

[Start Machine Learning](#)

```
# define the name of the configuration
NAME = "kangaroo_cfg"
# number of classes (background + personWithoutHelmet + personWithHelmet)
NUM_CLASSES = 1 + 2 #Change required
# number of training steps per epoch
STEPS_PER_EPOCH = 90
```



**Jason Brownlee** November 18, 2019 at 6:50 am #

Thanks for sharing.



**Ashutosh Srivastava** February

You are great @AutoRoboCu



**Akshay** February 12, 2020 at 3:22

Halo, I tried to train with multi  
like this

```
File "C:.....\lib\site-packages\keras\engine\training_utils.py", line 145, in
standardize_input_data
str(data_shape))
```

ValueError: Error when checking input: expected input\_image to have shape (None, None, 1) but got array with shape (1024, 1024, 3).

PS: I am working with gray scale images. and 3 classes. Inside single image both classes are present

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

**START MY EMAIL COURSE**



**Ademola Okerinde** February 26, 2020 at 10:47 am #

Thanks



**Nourhan** March 4, 2020 at 9:25 pm #

Thank you so much for sharing these changes.

However, after I followed all of them and adjusted the whole thing to fit my dataset. I keep

**Start Machine Learning**

getting this error:

RuntimeError: generator raised StopIteration

from that training line:

```
model.train(train_set, test_set, learning_rate=config.LEARNING_RATE, epochs=5,
layers='heads')
```

Do you have any suggestions to overcome it?



**hila** March 12, 2020 at 10:14 pm #

hey i was using your code for  
can you please show us your prediction  
lot of errors which i cannot solve.



**mahmoud** July 25, 2019 at 7:25 pm #

is this model also suppose to detect the mask  
will need some modification to segment the images.

## Start Machine Learning X

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

**START MY EMAIL COURSE**

**REPLY ↩**



**Jason Brownlee** July 26, 2019 at 8:19 am #

Yes, if masks are provided.

In the case of kangaroos, we do not provide masks – just bounding boxes, therefore masks cannot be learned.

**REPLY ↩**



**mahmoud** July 29, 2019 at 8:47 pm #

when i try to test image with multiple kangaroos ,it failed to detect them is there are two  
kangaroos interference it detect them as only one ?? any advice

**REPLY ↩**



**Jason Brownlee** July 30, 2019 at 6:11 am #

Perhaps the model requires more training on photos with multiple kangaroos?

## Start Machine Learning



thanks for your response, another question is there a new version of Mask RCNN available on github .

also what i need to have mask on my model how i can provide the model and make my model learn it also



**Jason Brownlee** July 31, 2019 at 6:52 am #

The model can learn the mask, if you provide a dataset that has masks on the images.



**mahmoud** August 22, 2019 at 7:11 pm #

thanks for your response i can't understand what is the mask loss on this



**Jason Brownlee** August 23, 2019 at 12:01 pm #

I don't follow sorry, what do you mean?



**Nishant Gaurav** July 29, 2019 at 6:39 pm #

REPLY ↗

I am getting this error. Please help  
 OSError Traceback (most recent call last)  
 in ()  
 —> 1 model.load\_weights('mask\_rcnn\_kangaroo\_cfg\_0005.h5', by\_name=True)  
 2 frames  
 /usr/local/lib/python3.6/dist-packages/h5py/\_hl/files.py in make\_fid(name, mode, userblock\_size, fapl, fcpl, swmr)  
 140 if swmr and swmr\_support:  
 141 flags |= h5f.ACC\_SWMR\_READ  
 —> 142 fid = h5f.open(name, flags, fapl=fapl)  
 143 elif mode == 'r+':  
 144 fid = h5f.open(name, h5f.ACC\_RDWR, fapl=fapl)  
 h5py/\_objects.pyx in h5py.\_objects.with\_phil.wrapper()  
 h5py/\_objects.pyx in h5py.\_objects.with\_phil.wrapper()  
 h5py/h5f.pyx in h5py.h5f.open()

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE

OSError: Unable to open file (unable to open file: name = 'mask\_rcnn\_kangaroo\_cfg\_0005.h5', errno = 2, error message = 'No such file or directory', flags = 0, o\_flags = 0)



**Jason Brownlee** July 30, 2019 at 6:06 am #

REPLY ↗

The error suggests that the path to your data file is incorrect or the file is corrupted in some way?



**Nishant Gaurav** July 31, 2019 at 9:45 pm #

REPLY ↗

Thanks for the suggestion. The problem is how do we resolve the problem with the multiclass characters given in the same image and want to know how we apply the multiclass label.

## Start Machine Learning

You can master applied Machine Learning **without math or fancy degrees**.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** August 1, 2019 at 10:00 pm #

Perhaps extract the images of digits from each segmented image.



**Dicko** July 29, 2019 at 8:13 pm #

REPLY ↗

Hi there, when I copied the example exactly, I am getting a train mAP of 0.000 and a test mAP of 0.000 also. Clearly something is wrong, I was wondering if anyone knew what the issue could be and how to resolve it. Thank you.



**Jason Brownlee** July 30, 2019 at 6:09 am #

REPLY ↗

Sorry to hear that, I have some suggestions here that might help:

<https://machinelearningmastery.com/faq/single-faq/why-does-the-code-in-the-tutorial-not-work-for-me>



**saka** July 30, 2019 at 2:26 pm #

REPLY ↗

Dear Jason, Thanks! I really learned a lot.

I am getting this error for the coding line "from mrcnn.u

Start Machine Learning

```
" from mrcnn.utils import Dataset
```

ModuleNotFoundError: No module named 'mrcnn' .

However, I checked if the library was installed by typing "show mask-rcnn" and got the results below,

Name: mask-rcnn

Version: 2.1

Summary: Mask R-CNN for object detection and instance segmentation

Home-page: [https://github.com/matterport/Mask\\_RCNN](https://github.com/matterport/Mask_RCNN)

Author: Matterport

Author-email: [waleed.abdulla@gmail.com](mailto:waleed.abdulla@gmail.com)

License: MIT

Location: c:\users\sakal\appdata\local\continuum\anaconda3\lib\site-packages\mask\_rcnn-2.1-nv3.7.egg

According the information above, It seems no problem me about this. Thanks!!



**Jason Brownlee** July 31, 2019 at 6:44 am #

Sorry to hear that.

Are you running the code from the command line i

## Start Machine Learning

You can master applied Machine Learning **without math or fancy degrees**.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**saka** July 31, 2019 at 2:20 pm #

REPLY ↗

I run your example code on Spyder IDE



**Jason Brownlee** August 1, 2019 at 6:41 am #

REPLY ↗

I recommend not using an IDE and instead running the code from the command line:  
<https://machinelearningmastery.com/faq/single-faq/how-do-i-run-a-script-from-the-command-line>



**saka** August 1, 2019 at 9:47 am #

It works now by running the code from the command line. Thanks!

just curious the reason why it is different from running from an IDE



**Jason Brownlee** August 1, 2019

Start Machine Learning

Happy to hear that.

It is a very common problem, I explain more here:

<https://machinelearningmastery.com/faq/single-faq/why-dont-use-or-recommend-notebooks>



**saka** July 31, 2019 at 2:31 pm #

REPLY ↗

By the way, I got the Messages below when I installed by typing “python setup.py install”

WARNING:root:Fail load requirements file, so using default ones.  
running install

.

.

Processing dependencies for mask-rcnn==2.1  
Finished processing dependencies for mask-rc

Do you think the warning above matter? Thank

## Start Machine Learning

X

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** August 1, 2019 at

Probably not.



**Dicko** July 30, 2019 at 7:13 pm #

REPLY ↗

Thanks for that I'll have a look through the code and see if I've made a mistake somewhere when copying.

Is there a file which has the complete code written so that i can just copy and past the whole lot rather than bits at a time?

Thank you 😊



**Jason Brownlee** July 31, 2019 at 6:48 am #

REPLY ↗

Each of my tutorials has the complete file embedding, you can copy-paste it directly.



**Nishant Gaurav** July 31, 2019 at 9:52 pm #

REPLY ↗

File "", line 21  
self.add\_class("dataset", 2, "1")

Start Machine Learning

^

IndentationError: unindent does not match any outer indentation level

Hi

I am getting this error when I added just two new lines, in the code.

```
def load_dataset(self, dataset_dir, is_train=True):
    # define one class
    self.add_class("dataset", 1, "N")
    self.add_class("dataset", 2, "1") //Added this new line
    # define data locations
    images_dir = dataset_dir + '/images/'
    annotations_dir = dataset_dir + '/annots/'

    for i in range(len(boxes)):
        box = boxes[i]
        row_s, row_e = box[1], box[3]
        col_s, col_e = box[0], box[2]
        masks[row_s:row_e, col_s:col_e, i] = 1
        class_ids.append(self.class_names.index('N'))
        class_ids.append(self.class_names.index('1')) //Added
    return masks, asarray(class_ids, dtype='int32')
```

## Start Machine Learning

X

You can master applied Machine Learning  
without math or fancy degrees.

Find out how in this *free* and *practical* course.

[START MY EMAIL COURSE](#)


**Jason Brownlee** August 1, 2019 at 6:50 am #

Sorry to hear that, it looks like you did not copy the code with white space.

I show how to copy the code correctly here:

<https://machinelearningmastery.com/faq/single-faq/how-do-i-copy-code-from-a-tutorial>



**Nishant Gaurav** July 31, 2019 at 11:14 pm #

REPLY ↗

IndexError Traceback (most recent call last)

```
in
2 plt.imshow(image)
3 # plot mask
--> 4 plt.imshow(mask[:, :, 0], cmap='gray', alpha=0.1)
5 plt.show()
```

IndexError: index 0 is out of bounds for axis 2 with size 0

I am getting this error after I added those extra two lines.



**Jason Brownlee** August 1, 2019 at 6:51 am #

[Start Machine Learning](#)

Sorry to hear that, I have some suggestions here:

<https://machinelearningmastery.com/faq/single-faq/why-does-the-code-in-the-tutorial-not-work-for-me>



**ahmadreza** August 1, 2019 at 2:44 am #

REPLY ↗

hi Sir

I am getting this error. Please help

if is\_train and int(image\_id) >= 150:

ValueError: invalid literal for int() with base 10: 'Thumb'



**Jason Brownlee** August 1, 2019 at 6:55 am #

X

Sorry to hear that, I have some suggestions here:  
<https://machinelearningmastery.com/faq/single-faq/why-does-the-code-in-the-tutorial-not-work-for-me>

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Julio César Álvarez Iglesias** September 14, 2019 at 6:17 am #

REPLY ↗

I am facing the same problem. Did you manage to resolve this issue?



**Jason Brownlee** September 14, 2019 at 6:17 am #

Perhaps you have a thumb nail file in the folder?

If so, perhaps try deleting it?



**Nishant Gaurav** August 1, 2019 at 2:16 pm #

REPLY ↗

Hi Sir,

Could you please give some insight where do I need to make changes for the multi-class label in the code so that I could identify the different characters and numbers in a single image?

Please give some insight with examples so that it is easier to understand.

Thanks so much for helping.



**Jason Brownlee** August 2, 2019 at 6:40 am #

REPLY ↗

## Start Machine Learning

This is a common question that I answer here:

[https://machinelearningmastery.com/faq/single-faq/can-you-change-the-code-in-the-tutorial-to-\\_\\_/](https://machinelearningmastery.com/faq/single-faq/can-you-change-the-code-in-the-tutorial-to-__/)



**mh** August 6, 2019 at 6:58 pm #

REPLY ↗

Thanks for your tutorial.

But i want to ask is there any model can deal with the objects which have similar color on the back ground.



**Jason Brownlee** August 7, 2019 at 7:45 am #

Perhaps. You may have to do some testing  
model.



**Tal** August 9, 2019 at 12:33 am #

Thank you very much for this great and clear  
If I may ask:  
Is there a way to evaluate the model while training? For

Thanks a million,

Tal

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** August 9, 2019 at 8:15 am #

REPLY ↗

Yes, you can use a hold out validation dataset:

<https://machinelearningmastery.com/difference-test-validation-datasets/>



**Tal** August 14, 2019 at 9:37 pm #

REPLY ↗

Thank you for the quick response!



**Jason Brownlee** August 15, 2019 at 8:09 am #

REPLY ↗

No problem.

## Start Machine Learning



**Selman Bozkir** August 14, 2019 at 4:57 am #

REPLY ↗

Hi Jason,

I have a problem. My dataset contains only 872 training images and 15 classes. Meanwhile, my images are rather bigger than kangroo or pascal voc files. They are around 1500 pixel wide and 1000 pixel tall. I have changed the python codes in order to apply multi-class classification. My equipment is 1050 ti on a 24 GB memory system. I have run your code for kangroo data, it was ok. But whenever I have done it for my custom data, the memory requirement is getting higher than 20 GB and makes the ubuntu run on slow swap memory yielding a dead situation.

What is the problem? is it normal? What about the ram consumption in your case. I did not check it for kangroo data. But I remember that, on 5th epoch it acti...“

What could be a walk-around about this problem?



**Jason Brownlee** August 14, 2019 at 6:46 am

Perhaps you can reduce the size of the in



**Selman Bozkir** August 14, 2019 at 6:59 a

Well, for a fair scientific study, I would not reduce it but, the only way I found is to reduce IMAGE\_MIN\_DIM = 400 and IMAGE\_MAX\_DIM = 512. However, it is interesting that, for each epoch, the total memory consumption is getting higher.

Moreover, I need to say that, the training procedure always starts with giving warnings such as “UserWarning: Converting sparse IndexedSlices to a dense Tensor of unknown shape. This may consume a large amount of memory.”

This is the problem actually. Is it possible to solve this? I have googled it but the solutions did not come so clear to me (or it sounds so technical).

Currently, I can train the model for only 4 epochs. More needs more memory. This is for me, a certain bug since, the advancing epochs should not increase the memory consumption.

Btw, I really thank for your reply.

As I told, this memory issue really made me sad. Is this normal?

REPLY ↗



**Jason Brownlee** August 14, 2019 at 2:08 pm #

Perhaps you can use progressive loading and only load/yield one batch of images into memory at a time.

[Start Machine Learning](#)

This can be achieved with the ImageDataGenerator:

<https://machinelearningmastery.com/how-to-load-large-datasets-from-directories-for-deep-learning-with-keras/>



**Selman Bozkir** August 14, 2019 at 7:26 pm #

Dear Jason;

Thanks so much for your advice. Here, I would like to share my experience with you and others. The only solution I have found so far is that setting the `use_multiprocessing=False` in `model.py` and reducing the number of workers to 1. This has helped me. Btw, I am now using 384 and `IMAGE_MIN_DIM =384` . Now me.

I hope this information may help others.

Cheers

## Start Machine Learning

You can master applied Machine Learning **without math or fancy degrees**.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** August 15, 2019 at 1:45 pm #

Nice! Thanks for sharing.



**N. Arvind** August 18, 2019 at 1:58 am #

REPLY ↗

Dear Jason

Good morning!

We have used this model to detect bounding boxes and masks for id cards.

We provided annotations in .csv files as quadrilaterals and modified 'load\_mask' function accordingly. We are looking for quadrilateral shaped masks.

We are able to detect bounding boxes correctly. We are not able to detect masks correctly. Although incorrect masks do show up.

We have used the exact code. Learning rate is 0.00001. We have used 800 images and 65 epochs for training. A higher learning rate gives NaN loss. We have checked the entire dataset for any discrepancy.

Can you guide where we are going wrong ? Can we use this exact code with exactly the same config with four vertices to generate masks ?

Warm regards,

N. Arvind

## Start Machine Learning



**Jason Brownlee** August 18, 2019 at 6:48 am #

REPLY ↗

Well done!

Perhaps look into data preparation?



**Per Nord** August 22, 2019 at 12:34 am #

REPLY ↗

Great tutorial! I've managed to successfully train a model and now I want to use the model in Android and iOS.

I've learned that this requires me to convert my model to TensorFlow's saved model format.

I expected this to be trivial, but alas. The MaskRCNN interface does not seem to support this.

Did you ever try this?

If not, it would be a great continuation to this tutorial.

## Start Machine Learning

You can master applied Machine Learning **without math or fancy degrees**.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** August 22, 2019 at 6:29 am #

REPLY ↗

Sorry, I have not tried this.



**Abilash** August 24, 2019 at 10:03 pm #

REPLY ↗

Hi Jason,

That was nice Tutorial, i have some errors on trying with multiclass.

IndexError: boolean index did not match indexed array along dimension 0; dimension is 2 but corresponding boolean dimension is 1

I have two class ( full glass and empty glass) and have made NUM\_CLASSES = 1 + 2 in config along with self.add\_class("dataset", 1, "Full Glass") and self.add\_class("dataset", 2, "Not Full Glass") also made changes class\_ids.append(self.class\_names.index('Full'))  
class\_ids.append(self.class\_names.index('Not Full')).

Please help me out, i am unable to resolve the error since many attempts.



**Jason Brownlee** August 25, 2019 at 6:37 am #

REPLY ↗

It's hard to debug this for you off the cuff,

Start Machine Learning

Perhaps double check you made all of the required changes?



**Julio César Álvarez Iglesias** September 14, 2019 at 4:42 am #

REPLY ↗

I am facing the same problem. Did you manage to resolve this issue?



**Jason Brownlee** September 14, 2019 at 6:23 am #

REPLY ↗

Try removing “Thumb” files from your folder.



**Shubhangi** November 5, 2019 at 3:51 pm #

Hi

Having same error!! Could you find some solutions

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Alaki** September 4, 2019 at 5:14 pm #

I hope its not a repeated question. I wonder if you have tutorial on training a model for custom multi-object detection ? basically, an image taken where we would like to recognize multiple images in an image. There is no pre-trained model on these objects, and we have labeled a few set of images. (again each image, is labeled with multiple rectangular which are covering each object).

Thank you again for all these nice tutorials.



**Jason Brownlee** September 5, 2019 at 6:49 am #

REPLY ↗

I believe you adapt the above tutorial for this purpose, the model supports multiple objects in one image, and they can be different types.



**Ade** September 6, 2019 at 5:45 pm #

REPLY ↗

Dear Dr Jason,

Good day sir, I am a Machine Learning Engineer. I am currently working on logo detection system. I have tried MobileNet SSD, Faster RCNN and their seemed to be a higher number of false positives when I try the model out. It seems its not too good for logo that is very small in size. I have also created Haar and LBP cascade model and it seemed to perform better than the

Start Machine Learning

question: is there any other technique that can do very well with small logos with different contrast, orientations? Thank you.



**Jason Brownlee** September 7, 2019 at 5:22 am #

REPLY ↗

I'm not sure off hand, sorry. Perhaps check the literature?

I recall some interesting work on test-time augmentation that might be very helpful to you.



**K\_gao** September 9, 2019 at 5:01 pm #

Dear Jason!

You've made a great work again. Thank you for this post.

What if I want to train an add not just one object to my model? I have 100 class, and every class has 500 images, how to load image into the memory! It is possible to make it with less iteration?

Do you have a post about this?

Thanx

## Start Machine Learning

X

You can master applied Machine Learning **without math or fancy degrees**.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** September 10, 2019 at 5:37 am #

REPLY ↗

Yes, you can use progressive loading with a data generator, see this post:

<https://machinelearningmastery.com/how-to-load-large-datasets-from-directories-for-deep-learning-with-keras/>



**Ankit** September 13, 2019 at 4:58 am #

REPLY ↗

Hi Jason,

The notebook is very helpful and full of knowledge but I am having problems while training the model on a different dataset(fruits -apple, banana, orange).

After loading the images,annots and masks when I try to train the model i am getting the following error:  
RemoteTraceback Traceback (most recent call last)

RemoteTraceback:

“””

Traceback (most recent call last):

File “/usr/lib/python3.6/multiprocessing/pool.py”, line 11

Start Machine Learning

```

result = (True, func(*args, **kwds))
File "/usr/local/lib/python3.6/dist-packages/keras/utils/data_utils.py", line 641, in next_sample
return six.next(_SHARED_SEQUENCES[uid])
File "/content/Mask_RCNN/mrcnn/model.py", line 1709, in data_generator
use_mini_mask=config.USE_MINI_MASK)
File "/content/Mask_RCNN/mrcnn/model.py", line 1265, in load_image_gt
class_ids = class_ids[_idx]
IndexError: boolean index did not match indexed array along dimension 0; dimension is 6 but corresponding
boolean dimension is 2
"""

```

The above exception was the direct cause of the following exception:

IndexError Traceback (most recent call last)

```

in ()
3 learning_rate = config.LEARNING_RATE,
4 epochs = 10,
—> 5 layers = 'all' )

7 frames

```

```

/usr/lib/python3.6/multiprocessing/pool.py in get(self, t
642 return self._value
643 else:
-> 644 raise self._value
645
646 def __set__(self, i, obj):

```

IndexError: boolean index did not match indexed array along dimension 0; dimension is 6 but corresponding  
boolean dimension is 2

Please provide a hint about the same.

Also, I am using multiclass for 3 fruits.

## Start Machine Learning X

You can master applied Machine Learning **without math or fancy degrees**.  
Find out how in this *free* and *practical* course.

START MY EMAIL COURSE



**Jason Brownlee** September 13, 2019 at 5:45 am #

REPLY ↗

Perhaps double check that you are loading the data correctly or as you expect?



**bella** October 22, 2019 at 4:03 am #

REPLY ↗

Hey, How did you solve this issue of IndexError?



**Shubhangi** September 17, 2019 at 4:33 pm #

[Start Machine Learning](#)

I do train model on my own dataset but the prediction of model is not getting right. can you pls help me ?

Actually i have train for kangaroo class name but in prediction i am getting person class tag



**Jason Brownlee** September 18, 2019 at 5:57 am #

REPLY ↗

Perhaps start with the example in the tutorial and adapt it for your specific dataset?



**Shubhangi** September 20, 2019 at 3:17 pm #

REPLY ↗

Thank You i solved it ....But i have total from images but i am not getting correct output not good at all ....i have done 50 epochs at 25

## Start Machine Learning

You can master applied Machine Learning **without math or fancy degrees**.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** September 21, 2019 at 10:15 pm #

I have some general suggestions performance here that may help:  
<https://machinelearningmastery.com/start-here/#better>



**Shubhangi** October 7, 2019 at 8:50 pm #

Hello. I found some issues regarding accuracy of model. I dont know what issue is there which effect accuracy of model. Same cnfiguration as described above is used in my model but accuracy is no good. The ROI getting from prediction of model is not correct. Can some one Please help me out



**Jason Brownlee** October 8, 2019 at 8:00 am #

Is this on your own dataset or the dataset used in the above tutorial?

I have some general suggestions here that might help to diagnose and address performance issues:

<https://machinelearningmastery.com/start-here/#better>

**Jeorge** September 21, 2019 at 4:32 pm #

Start Machine Learning



Hello. I Have this error. I dont know how to solve it:

```
~/ve/main/lib/python3.7/site-packages/mask_rcnn-2.1-py3.7.egg/mrcnn/model.py in compile(self,
learning_rate, momentum)
2197 tf.reduce_mean(layer.output, keepdims=True)
2198 * self.config.LOSS_WEIGHTS.get(name, 1.))
-> 2199 self.keras_model.metrics_tensors.append(loss)
2200
2201 def set_trainable(self, layer_regex, keras_model=None, indent=0, verbose=1):
```

AttributeError: 'Model' object has no attribute 'metrics\_tensors'



**Jason Brownlee** September 22, 2019 at 9:26

Sorry, I have not seen that error before.

Are you able to confirm that your Keras/TensorFlow

Are you able to try Python 3.6 instead, I don't think



**Kay** September 25, 2019 at 8:57 pm #

You can add the line

model.keras\_model.metrics\_tensors = []

right after the model definition to circumvent the error.

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

**START MY EMAIL COURSE**



**Jason Brownlee** September 26, 2019 at 6:34 am #

**REPLY ↗**

Thanks for sharing.



**ahasan** September 29, 2019 at 2:17 am #

**REPLY ↗**

exactly where should I change in the model.py?



**Mikael** October 2, 2019 at 7:32 am #

**REPLY ↗**

See [https://github.com/matterport/Mask\\_RCNN](https://github.com/matterport/Mask_RCNN)

**Start Machine Learning**



**Jason Brownlee** October 2, 2019 at 8:07 am #

REPLY ↲

Why did you share this Mikael?



**Kevin** October 8, 2019 at 1:20 am #

REPLY ↵

Hi Jason Brownlee,

Great tutorial for object detection. This is the first time, I visited this site and I loved the way to document your post. I have walk-through each line of code and so far I am able to run the code successfully. You have developed well documented code guide for us. Based on your guide, I have developed a model on Weed detection problem. And yes, I am able to detect the weeds in the images.

By the way, I have one question:

-> How to save full keras model (architecture + weights) model.

I have tried:

```
1) self.keras model.save("model name.h5")
```

2) save weights only = False

but It gives error:

[TypeError: can't pickle thread.RLock objects]

If possible please help me on this.

Thanks.

Kevin



• Jason Brownlee October 8, 2019 at 8:07 am #

REPLY ↗

Well done Kevin!

`model.save()` should be sufficient. Perhaps there is an issue with your development environment?

Perhaps try AWS:

<https://machinelearningmastery.com/develop-evaluate-large-deep-learning-models-keras-amazon-web-services/>

## Start Machine Learning



**Reem** October 8, 2019 at 2:28 am #

REPLY ↗

How can I create the annotated xml file? The VGG tool only creates a csv file or json file. Could you please assist in the way of creating the xml file or the conversion from csv/json to xml?

Thanks



**Jason Brownlee** October 8, 2019 at 8:08 am #

REPLY ↗

I believe there are a ton of image annotation tools available that can create the annotations with/for you.



**Kevin** October 8, 2019 at 5:39 pm #

Hi Reem,

Check out this annotation tool, this will create .xml [<https://github.com/tzutalin/labellImg>]

## Start Machine Learning

You can master applied Machine Learning **without math or fancy degrees**.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Akash Nakarmi** October 8, 2019 at 3:18 am #

REPLY ↗

Jason,

Thanks for the very nice tutorial. I was able to train the model and get `mask_rcnn_kangaroo_cfg_0005.h5` created. However, when I ran the model evaluation code, I got the following error. Could you help me resolve this?

`AssertionError: Create model in inference mode, and it is complaining on line  
yhat=model.detect(sample, verbose=0) saying that len(images) must be equal to BATCH_SIZE.`

Thanks.

Akash



**Jason Brownlee** October 8, 2019 at 8:08 am #

REPLY ↗

Sorry to hear that, I have some suggestions here:

<https://machinelearningmastery.com/faq/single-faq/why-does-the-code-in-the-tutorial-not-work-for-me>

**Shubhangi** October 9, 2019 at 6:48 pm #

Start Machine Learning



I have my own data set ....Thank you for general suggestions this is helpful for me but i don't understand that why accuracy of model is not good even using same structure and configuration of model as suggested above.

And I have also tried on different data set for all issues is same by that I conclude that there is some minor issue in the script which is not detected by me so please help me out ....

If u want my source code i will that also

ThankYou



**Jason Brownlee** October 10, 2019 at 6:55 am #

REPLY ↗

What problem are you having exactly?



**Shubhangi** October 10, 2019 at 9:24 pm #

With the accuracy of model



**Jason Brownlee** October 11, 2019

You can discover general advice on diagnosing issues and improving performance with neural nets here:

<https://machinelearningmastery.com/start-here/#better>

## Start Machine Learning

You can master applied Machine Learning **without math or fancy degrees**.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Dinesh Kumar** October 10, 2019 at 4:58 pm #

REPLY ↗

Hello Jason,

While trying to train the model I got the following message.

File "C:\Users\userid\AppData\Local\Continuum\anaconda3\lib\site-packages\tensorflow\_core\python\framework\ops.py", line 523, in \_disallow\_in\_graph\_mode  
" this function with @tf.function.".format(task))

OperatorNotAllowedInGraphError: using a tf.Tensor as a Python bool is not allowed in Graph execution.  
Use Eager execution or decorate this function with @tf.function.

Could you please suggest on this

**Jason Brownlee** October 11, 2019 at 6:15 am

Start Machine Learning



Sorry to hear that, are you able to confirm that you are using Python 3.6, TensorFlow 1.14, and Keras 2.3 or better?



**Dinesh Kumar** October 11, 2019 at 7:09 pm #

REPLY ↗

Hello Jason,

Thank you for your reply.

I am using python 3.7, TensorFlow 2.0.0 and keras 2.3.1

Regards,

Dinesh



**Jason Brownlee** October 12, 2019

This example will not work with TensorFlow 2.0.0. Please scroll right at the top of the page:

Note: This tutorial requires TensorFlow version 1.14 or higher. TensorFlow 2 because some third-party libraries have not yet been updated.

## Start Machine Learning



You can master applied Machine Learning without math or fancy degrees.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Dinesh Kumar** October 15, 2019 at 3:49 pm #

Hello Jason,

Thanks for your reply,

I will use TF 1.14.

Regards,

Dinesh



**mark** October 18, 2019 at 6:59 am #

REPLY ↗

hello jason,

i just wanted to know how much time it takes to make a prediction on a new image.

so basically how long does it take to run

yhat = model.detect(sample, verbose=0)[0]

thank you for your time.

## Start Machine Learning



**Jason Brownlee** October 18, 2019 at 8:18 am #

REPLY ↗

Fractions of a second, although depends on hardware of course.



**mark** October 19, 2019 at 3:52 pm #

REPLY ↗

well i need to know how many times it can be run in 1 second.if run on your computer can you give me an estimate of how many times it would run in 1 second. (5,10,20,30,40, 50, 60, 60+)

thanks,

mark



**Jason Brownlee** October 20, 2019

Perhaps you can calculate those data – that way they will be meaningful/useful.



**Jerico** October 18, 2019 at 12:31 pm #

×

## Start Machine Learning

You can master applied Machine Learning **without math or fancy degrees**.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** October 18, 2019 at 2:52 pm #

REPLY ↗

Thanks.

In real life from a photo? Not using these models, sorry.



**Jerico** October 20, 2019 at 6:24 pm #

REPLY ↗

yep! from a photo. Assuming i took a pix of a kangaroo and test it on your model . definitely your model will recognize it as kangaroo. what i'm opt is, the dimension of kangaroo, i'm sure you have technique on how to determine its size using the model that you had created.



**Jason Brownlee** October 21, 2019 at 8:16 pm #

REPLY ↗

Start Machine Learning

No idea off the cuff, sorry.

It does not sound tractable as each photo has a different scale.



**Yaroslav** October 20, 2019 at 4:19 am #

REPLY ↗

Hi.

I found out, that we can't assign image id randomly (not from 0). Perhaps class Dataset creates list, not a numpy array. I checked myself and realized that I can't access image with id, for example, 317 while i have only 100 images.

Thus, I don't know why this field "image id" exists, when it numbered anyway from 0, increasing by 1.



**Jason Brownlee** October 20, 2019 at 6:25 am #

Thanks for sharing.



**Yaroslav** October 20, 2019 at 10:33 pm #

Thanks for your great article. It's the k

## Start Machine Learning

You can master applied Machine Learning  
without math or fancy degrees.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** October 21, 2019 at 6:18 am #

REPLY ↗

Thanks!



**JuanM** October 22, 2019 at 3:19 am #

REPLY ↗

Good afternoon, I have a problema with the code. When I start the training the proces is stack in the first Epoch. What can i do to ?



**JuanM** October 22, 2019 at 3:23 am #

REPLY ↗

WARNING:tensorflow:From C:\Users\Juan\AppData\Local\Programs\Python\Python36\lib\site-packages\keras\callbacks.py:708: The name tf.summary.FileWriter is deprecated. Please use tf.compat.v1.summary.FileWriter instead.

Epoch 1/5

in this momento I have the problem

Start Machine Learning



**Jason Brownlee** October 22, 2019 at 5:58 am #

REPLY ↗

TensorFlow 2.0 is not support for this tutorial at the moment, try TensorFlow 1.14 instead.



**Jason Brownlee** October 22, 2019 at 5:57 am #

REPLY ↗

Sorry to hear that, I have some suggestions here:

<https://machinelearningmastery.com/faq/single-faq/why-does-the-code-in-the-tutorial-not-work-for-me>



**JuanM** October 23, 2019 at 2:51 am #

In this momment my tensorflow is 1.1



**Jason Brownlee** October 23, 2019

Try down-grading tip 1.14? Or per

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Akash Joshi** October 29, 2019 at 4:47 am #

REPLY ↗

Hi Jason,

it was a very great article and thoroughly explained code.

I have a question for you regarding this tutorial. I am trying out this tutorial on my laptop and I have limited processing power. When I tried with the full data set of kangaroos the first epoch took around 8 hrs approx. I stopped it in between then I tried to reduce the data set to about 10 images and started training process but it still showed 7 hours as the ETA and each epoch had 131 steps.

As per my thinking if I reduce the number of images in the data set the training time should reduce and instead of 131 steps it should have 10 steps in each epoch as the data set has only 10 images. I am currently willing to have a lower accuracy.

Can you let me know if my understanding is wrong?



**Jason Brownlee** October 29, 2019 at 5:32 am #

REPLY ↗

Less images might impact model performance generally.

Start Machine Learning

Perhaps try running on EC2?



**Akash Joshi** November 2, 2019 at 12:16 am #

REPLY ↗

Hi Jason,

I tried using less no. of images but i cannot complete the training process as i am getting the following message

2019-11-01 18:37:31.547297: W T:\src\github\tensorflow\tensorflow\core\framework\allocator.cc:108] Allocation of 603979776 exceeds 10% of system memory.

can you tell me why do i get this message?



**Jason Brownlee** November 2, 2019

- Try even fewer images?
- Try EC2 with more RAM?
- Try a smaller model?
- Try progressive loading?

## Start Machine Learning



You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Akash** November 2, 2019 at 7:39 am #

I tried with 3 images also but got the same issue.can you explain or give links to the the last two options you mentioned.



**Asjad Murtaza** October 29, 2019 at 9:54 am #

REPLY ↗

Hi Jason, I plan on following this tutorial for skin segmentation on compaq dataset. The labels are in PBM(Portable Bitmap) format. Is it fine or do I need to do somethings differently ?  
Regards



**Jason Brownlee** October 29, 2019 at 1:49 pm #

REPLY ↗

I don't think it matters as long as the images can be loaded to numpy arrays.

**Juan Pablo** November 1, 2019 at 8:34 am #

Start Machine Learning



Hi Jason,

Thanks for this great article!

One question:

I already have my model trained and my weights (mask\_rcnn\_kangaroo\_cfg\_0019.h5).

How can I validate this with new images?

I mean not to call the test or train datasets

```
plot_actual_vs_predicted('MY PHOTO', model, cfg)
```



**Jason Brownlee** November 1, 2019 at 1:40 pm #

Load the model and use it to make predictions on the expected values.

The section "How to Evaluate a Mask R-CNN Model" in this tutorial covers this topic.



**Juan Pablo** November 2, 2019 at 5:01 am #

Thanks Jason,

But why would I need annotations if I want to validate the model with a new image.

## Start Machine Learning



You can master applied Machine Learning **without math or fancy degrees**.

Find out how in this *free* and *practical* course.

**START MY EMAIL COURSE**



**Jason Brownlee** November 2, 2019 at 6:52 am #

**REPLY** ↗

To confirm the predictions match the expectations and calculate an evaluation score.



**osteocyte** November 5, 2019 at 8:59 pm #

**REPLY** ↗

Hi Jason,

thanks a lot for this great tutorial! Could you please give me a quick hint how one can extract the total number of detected objects in each image?

Thanks a lot, osteocyte



**Jason Brownlee** November 6, 2019 at 6:33 am #

**REPLY** ↗

It will be the number of bounding boxes returned from a call to predict.

**Start Machine Learning**



**Felipe Correa** November 5, 2019 at 11:12 pm #

REPLY ↗

Hi, I already have my trained model (generated with this tutorial). Is it possible to use this model for video live detection?

Do you have a script example or something that you could help me out with.?

Best regards!



**Jason Brownlee** November 6, 2019 at 6:33 am #

REPLY ↗

Yes, perhaps apply to each frame of the video.

I don't have an example at this stage.



**Florian Garrigues** November 6, 2019 at 1:18 am #

REPLY ↗

Hello

first thanks for this amazing tutorial!

Second i have an question how can we modifie you're

Thank you

## Start Machine Learning

You can master applied Machine Learning  
without math or fancy degrees.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** November 6, 2019 at 6:36 am #

REPLY ↗

You can define a mask and a box and then fit the model on it. In my example I treat them as the same.



**tejas** November 6, 2019 at 4:13 pm #

REPLY ↗

```
self.add_class("dataset", 1, "kangaroo")
self.add_class("dataset", 2, "tiger")
self.add_class("dataset", 3, "dog")

class_ids.append(self.class_names.index('kangaroo'))
class_ids.append(self.class_names.index('tiger'))
class_ids.append(self.class_names.index('dog'))
```

for multi class classification is this changes are enough anything more needed?

## Start Machine Learning



**Jason Brownlee** November 7, 2019 at 6:34 am #

REPLY ↗

Looks good to me, off the cuff at least.



**tejas** November 8, 2019 at 12:00 am #

REPLY ↗

Please tell how to do for multi class classification?



**Suave** December 20, 2019 at 5:08 am #

Take a look on your xml file and then and boundig boxes to the right class name. Then creat corelation

## Start Machine Learning



You can master applied Machine Learning without math or fancy degrees.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** December 20, 2019 at

Great tip!



**Shubhangi** November 6, 2019 at 11:35 pm #

REPLY ↗

Hello

first thanks for this amazing tutorial!

Second i have an question how many epochs and time steps are required for 2 lakh dataset



**Jason Brownlee** November 7, 2019 at 6:42 am #

REPLY ↗

Perhaps test different configurations and see what works best for your specific dataset?



**shriya** November 8, 2019 at 12:32 am #

REPLY ↗

boolean index did not match indexed array along dimension 0; dimension is 4 but corresponding boolean dimension is 2

## Start Machine Learning



**Jason Brownlee** November 8, 2019 at 6:44 am #

REPLY ↗

Sorry to hear that, I have some suggestions here:

<https://machinelearningmastery.com/faq/single-faq/why-does-the-code-in-the-tutorial-not-work-for-me>



**shriya** November 8, 2019 at 3:01 pm #

REPLY ↗

i can understand your point but help me .I am enable to figure out.



**yamuna** November 12, 2019 at 4:40 pm #

based on colour object is detecting.How can i



**Jason Brownlee** November 13, 2019 at 5:34 am #

Sorry, I don't understand your question, can you

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Dave** November 13, 2019 at 12:23 am #

REPLY ↗

Thanks for these tutorials, I'm making good progress on my projects.

Can I please ask: Is it solely tagged content that contributes to the training/prediction, or is it the whole image?

If I create a dataset of 100 photos (as an example), and tag the easiest elements (say people) in these photos, will untagged people in these photos work to "untrain" the model? Would I be better off creating a smaller dataset that is more thoroughly tagged, or do untagged elements not matter? Thanks.



**Jason Brownlee** November 13, 2019 at 5:46 am #

REPLY ↗

It is the localized object within the image. Both.

Good question. Test both and compare.



**Saurabh** November 13, 2019 at 12:24 am #

REPLY ↗

Hello Jason,

## Start Machine Learning

Thanks for the interesting technical blog.

I am looking for "How to train SSD based object detection on the custom dataset?". Could you please provide a pointer?

Thanking you!



**Jason Brownlee** November 13, 2019 at 5:46 am #

REPLY ↗

I don't have an example, I hope to have one in the future.



**Saurabh** November 13, 2019 at 6:20 pm #

Thank you!



**Jason Brownlee** November 14, 2019 at 7:59 am #

You're welcome.

## Start Machine Learning

You can master applied Machine Learning  
without math or fancy degrees.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**yamuna** November 13, 2019 at 7:18 pm #

REPLY ↗

i have done object detection to detect gloves.

the gloves are white in colour.

but if the person where white colour shirt then also it is detecting as gloves



**Jason Brownlee** November 14, 2019 at 7:59 am #

REPLY ↗

Well done!

Perhaps expand the training dataset or try data augmentation during training?



**yamuna** November 15, 2019 at 4:58 am #

REPLY ↗

i have augmented the images then i need to do annotations separately? or is there any other way?

## Start Machine Learning



**Jason Brownlee** November 15, 2019 at 7:56 am #

REPLY ↗

You can use augmentation that is “annotation-aware”, e.g. apply augmentation in a consistent way to images and annotations.

Big labs might have code for this, e.g. facebook. Otherwise, custom code will be required.



**bhandavi** November 15, 2019 at 4:55 am #

REPLY ↗

how to retrain the already trained weights with more images?



**Jason Brownlee** November 15, 2019 at 7:56 am #

That is exactly what we do in this tutorial.



**bhandavi** November 15, 2019 at 10:27 pm #

i am asking already trained kangaroo

Replacing coco file with kangaroo .h5 file?

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** November 16, 2019 at 7:24 am #

REPLY ↗

Yes, follow this tutorial and adapt the coco weights with your own dataset.



**Dave** November 15, 2019 at 10:07 am #

REPLY ↗

Hi Jason,

Thanks for a great tutorial. My trained model gives many bbox predictions of different sizes for the same kangaroo, and also for random background objects. This was after training for 2 epochs. After training for further epochs, the losses all flatlined to NaN or 0. Just wondering if you've ever experienced this.

Thanks again,

Dave

**Jason Brownlee** November 16, 2019 at 7:16 am #

Start Machine Learning



Not really.

Perhaps try fitting the model a few times and compare results?



**Aqiff** November 19, 2019 at 2:45 pm #

REPLY ↗

This is a very great tutorial. For the training, I am stuck with this line model = MaskRCNN(mode='training', model\_dir='./', config = config)  
The error is: 'NoneType' object has no attribute 'lower'. How can I fix this?



**Jason Brownlee** November 20, 2019 at 6:07 pm #

Sorry to hear that, I have some suggestions:  
<https://machinelearningmastery.com/faq/single-faq/>



**yamuna** November 21, 2019 at 6:34 pm #

I am trying to predict hand gloves and spects  
1.the people who are not wearing gloves also it is taking them as gloves.  
2.It is completely getting biased on color.where ever it finds white color it is predicting as gloves.

Please help me. I have 1000 images as by training .I have done for nearly 50 epochs

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** November 22, 2019 at 5:59 am #

REPLY ↗

Perhaps include training examples with hands and gloves in the same image to help the model tell the difference?



**Mursyideen** November 24, 2019 at 7:27 pm #

REPLY ↗

Hello there, I am trying to execute this code using my own GPU, however, I have this error  
ResourceExhaustedError: 2 root error(s) found.  
(0) Resource exhausted: OOM when allocating tensor with shape[2,512,128,128] and type float on /job:localhost/replica:0/task:0/device:GPU:0 by allocator GPU\_0\_bfc  
[[{{node rpn\_model\_11/rpn\_class\_raw/convolution-0-TransposeNHWCToNCHW-LayoutOptimizer}}]]  
Hint: If you want to see a list of allocated tensors when OOM happens, add  
report\_tensor\_allocations\_upon\_oom to RunOptions for current allocation info.

Start Machine Learning

[[Mean\_23/\_13623]]

Hint: If you want to see a list of allocated tensors when OOM happens, add report\_tensor\_allocations\_upon\_oom to RunOptions for current allocation info.

(1) Resource exhausted: OOM when allocating tensor with shape[2,512,128,128] and type float on /job:localhost/replica:0/task:0/device:GPU:0 by allocator GPU\_0\_bfc

[[{{node rpn\_model\_11/rpn\_class\_raw/convolution-0-TransposeNHWCToNCHW-LayoutOptimizer}}]]

Hint: If you want to see a list of allocated tensors when OOM happens, add report\_tensor\_allocations\_upon\_oom to RunOptions for current allocation info.

0 successful operations.

0 derived errors ignored.



**Jason Brownlee** November 25, 2019 at 6:27

Sorry, I don't know about this error, perha



**Maged** December 4, 2019 at 7:33 am #

Hey Mursyideen, how did you solve this is

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Ozan Veranyurt** November 25, 2019 at 6:29 am #

REPLY ↗

Hi Jason, I used your tutorial to prepare a Pistol detector. The data is properly loaded and when I try to train the epochs are frozen. It sometimes freezes on images randomly. Here is output

Epoch 1/5

```
25/150 [=====>.....] – ETA: 3:13 – loss: 3.2542 – rpn_class_loss: 0.0182 – rpn_bbox_loss: 0.6457 – mrcnn_class_loss: 0.5098 – mrcnn_bbox_loss: 0.8810 – mrcnn_mask_loss: 1.1995
```

It stops on different images. I checked all images and annexes are oke. I followed the suggestions here : [https://github.com/matterport/Mask\\_RCNN/issues/287](https://github.com/matterport/Mask_RCNN/issues/287) (Made modifications in the model.py under mrcnn )

My tensorflow is 1.15

and keras : 2.2.4

Any suggestions? I am working on different approachs for pistol detection and mrcnn is one of them. It is critical for my thesis. So I will appreciate any suggestions. Maybe a working combination of keras – tensorflow with mrcnn.



**Jason Brownlee** November 25, 2019 at 6:35 am #

REPLY ↗

I wonder if you are running out of memory

Start Machine Learning

Perhaps try running on an AWS EC2 instance?



**Yussi Eikelman** November 28, 2019 at 7:13 pm #

REPLY ↗

Jason Hi,

I have a set of grayscale images of shape(192,384,3) with none/one/multiple masks in each of size (5,5). I'm able to train my model, but unable to receive any result – the tuple from the detect() appears to be empty. In rare cases there is a prediction, which is not good enough. Please help, thanks!



**Jason Brownlee** November 29, 2019 at 6:47 am #

Perhaps the model is not detecting anything.



**Yussi Eikelman** December 1, 2019 at 1:24 pm #

A different question:

```
1 for i in range(len(boxes)):
2     box = boxes[i]
3     row_s, row_e = box[1], box[3]
4     col_s, col_e = box[0], box[2]
5     masks[row_s:row_e, col_s:col_e, i] = 1
6     class_ids.append(self.class_names.index('kangaroo'))
7 return masks, asarray(class_ids, dtype='int32')
```

if the masks = zeros([h, w, len(boxes)], dtype='uint8'),  
in my case each mask is (h = 5,w = 5, i) and the bounding box, for example, is (5, 5, 10, 10). How masks[row\_s:row\_e, col\_s:col\_e, i] = 1, where the indexes are not in the original mask range is (5,5), are affected by the bounding box indexes?

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** December 1, 2019 at 5:43 am #

REPLY ↗

Sorry, I don't follow your question, are you able to elaborate?



**bts** December 1, 2019 at 6:25 pm #

REPLY ↗

Hello Jason,

I am running this code on my mac and I get this error the running epoch 1 and the program gets stuck here.

Start Machine Learning

Epoch 1/5

/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/keras/utils/data\_utils.py:709: UserWarning: An input could not be retrieved. It could be because a worker has died. We do not have any information on the lost sample.

Any idea about this?

Also, should this code be run only on GPU machines?



**Jason Brownlee** December 2, 2019 at 6:00 am #

REPLY ↗

I have not seen that before.

No, the code works fine on the GPU or CPU.

Perhaps try re-installing your development environment?

Perhaps try running either on the cpu or gpu?

Perhaps try posting/searching on stackoverflow?

Perhaps try running other examples and see if they work?



**shankar** December 2, 2019 at 7:19 am #

Hi Jason, what an amazing post..well done on

For my application, in addition to the predicted bounding box+mask+class, I also need to extract the last fully-connected layer of the mask\_rcnn model (that is, the feature vector representation of the input image).

In keras, we can save a model's json and weight files. And then load them again. And extract the output of any intermediate layer as:

1. `model.summary()`
2. `feature_extractor = tf.keras.models.Model(inputs=model.input, outputs=model.get_layer('avg_pool').output)`
3. `features = feature_extractor.predict(my_image)`

In mask\_rcnn, we load the pre-trained model `mask_rcnn_coco.h5`. Do you know how we can access and extract the last fully-connected weights?

My research is stuck because I am unable to complete this step. I shall be grateful if you can guide me (either via email, or on this forum).

Regards-Shankar



**Jason Brownlee** December 2, 2019 at 1:53 pm #

REPLY ↗

Thanks!

Start Machine Learning

Great question.

Hmmm, not off hand, sorry. Some experimentation will be required.



**Maged** December 4, 2019 at 7:45 am #

REPLY ↩

Hey @Jason thank you for a fantastic tutorial. Please keep it up :)!

Two questions if you may,

- How can we reduce the batch\_size ?
- How can we reduce the image\_dimensions given to the model?

Both of these are attempts to fix the “..Resource exhaust



**Jason Brownlee** December 4, 2019 at 1:56 pm #

Thanks.

Good question about the batch size, I'm not sure of the function?

I believe you have control over the images sizes –

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Maged** December 5, 2019 at 9:49 pm #

REPLY ↩

Thanks Jason for your reply, Here is how I fixed it by modifying the KangarooConfig class

```
class KangarooConfig(Config):
    # define the name of the configuration
    NAME = "kangaroo_cfg"
    # number of classes (background + kangaroo)
    NUM_CLASSES = 1 + 1
    STEPS_PER_EPOCH = 131

    GPU_COUNT = 1
    IMAGES_PER_GPU = 1

    IMAGE_MIN_DIM = 400
    IMAGE_MAX_DIM = 512
```



**Jason Brownlee** December 6, 2019 at 5:15 am #

REPLY ↩

Well done, thanks for sharing!

Start Machine Learning



**João Granzotti** March 12, 2020 at 4:44 am #

REPLY ↗

what this parameters means?

GPU\_COUNT = 1

IMAGES\_PER\_GPU = 1

they are the number of GPUs that i have and the batch size ? I want to reduce the batch size to.

Thanks for you help.



**Tim** December 8, 2019 at 2:05 am #

Hey Jason, When I plot the graph "Actual" vs there a way to tweak it so it appears similar to the "Pre Thank you,



**Jason Brownlee** December 8, 2019 at 6:16 am #

Yes, I intentionally darken the photo to highlight the detection.

You can remove the code to do that. Just plot the photo and use the box to drop a colored rectangle.

## Start Machine Learning



You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**amine** December 21, 2019 at 6:24 am #

REPLY ↗

hi trying to train on my dataset , however i get this error when trying to load the data  
help please

```
FileNotFoundException Traceback (most recent call last)
```

```
in ()
```

```
1 image_id = 1
```

```
—> 2 image = train_set.load_image(image_id)
```

```
3 print(image.shape)
```

```
4 # load image mask
```

```
5 mask, class_ids = train_set.load_mask(image_id)
```

```
6 frames
```

```
/usr/local/lib/python3.6/dist-packages/imageio/core/request.py in _parse_uri(self, uri)
```

```
271 # Reading: check that the file exists (but is allowed a dir)
```

```
272 if not os.path.exists(fn):
```

```
—> 273 raise FileNotFoundError("No such file:
```

Start Machine Learning

274 else:

275 # Writing: check that the directory to write to does exist

FileNotFoundException: No such file: '/content/Mask\_RCNN/Amine/imagesacdf21.JPG'



**Jason Brownlee** December 21, 2019 at 7:18 am #

REPLY ↗

Looks like the image you are trying to load does not exist on your workstation.



**Narottam** December 12, 2019 at 9:40 pm #

Hi Jason, please confirm for mask RCNN model we need to create .xml file) ? If no, then please suggest changes in code to better output the way we got after using 'display\_instances(dataset.class\_names)' under evaluate\_model function.



**Jason Brownlee** December 13, 2019 at 6:01 pm #

You, you can just work with object boxes and labels in the way you wish.

## Start Machine Learning

You can master applied Machine Learning **without math or fancy degrees**.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Sally Jac** December 14, 2019 at 4:32 am #

REPLY ↗

Hi Jason, when I am creating the model, I keep getting this error

```
/anaconda3/lib/python3.7/site-packages/mask_rcnn-2.1-py3.7.egg/mrcnn/model.py in
detection_targets_graph(proposals, gt_class_ids, gt_boxes, gt_masks, config)
551 positive_count = int(config.TRAIN_ROIS_PER_IMAGE *
552 config.ROI_POSITIVE_RATIO)
-> 553 positive_indices = tf.random_shuffle(positive_indices)[:positive_count]
554 positive_count = tf.shape(positive_indices)[0]
555 # Negative ROIs. Add enough to maintain positive:negative ratio.
```

AttributeError: module 'tensorflow' has no attribute 'random\_shuffle'

I am unsure of how to debug this. I tried changing random\_shuffle to random.shuffle in model.py but it does not work. Or have I downloaded the wrong MaskRCNN? What is the link to download the MaskRCNN? Thank you for your help.

Start Machine Learning



**Jason Brownlee** December 14, 2019 at 6:26 am #

REPLY ↗

It looks like you are using tensorflow version 2, and the maskrcnn model requires tensorflow 1.14 or 1.15.

This is mentioned right at the top of the tutorial.



**Twayne Jeremy** December 15, 2019 at 5:40 pm #

REPLY ↗

Hello Jason, your tutorial is really helpful. However, I've seen some errors while trying it.

When I am evaluating the model, I received this

ValueError: shapes (1,1048576) and (1050624

Thank you for your help.

## Start Machine Learning

X

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** December 16, 2019 at 1:18 pm #

I'm sorry to hear that, I have some information on this:  
<https://machinelearningmastery.com/faq-section/dealing-with-grayscale-images/>  
 for-me



**Ekrem Fatih Yilmazer** December 16, 2019 at 2:18 am #

REPLY ↗

I have a data set of liver CT which is grayscale.

Is it possible for me to apply the same model (also transfer learning) for grayscale images . Since the pretrained models are for RGB images, I am curious about whether I can convert them for my application purpose?



**Jason Brownlee** December 16, 2019 at 6:18 am #

REPLY ↗

Perhaps try it and compare to fitting a new model from scratch?



**Zain** December 16, 2019 at 3:44 am #

REPLY ↗

Thank you very much for such an informative article.

I have created a colab notebook which walks through t

Start Machine Learning



**Yansen** December 18, 2019 at 7:07 am #

REPLY ↗

Hi Jason, thanks for the tutorial. Following your instruction I fitted a custom dataset of 200 photos with one label. I got a Train mAP of 0.986 and a Test mAP of 1.000. The detection results are great and even see things I would miss if I do labeling. My question is that: is 1.000 too good to be true?



**Jason Brownlee** December 18, 2019 at 1:26 pm #

REPLY ↗

Wow, well done.

Perhaps think of ways that you could have a misle  
e.g. more/less data? Different measures? Inspect |



**amine** December 20, 2019 at 8:31 am #

hi Mr Brownlee  
thanks for this awsome tutorial however when i tried to  
i get this message:

```
ValueError Traceback (most recent call last)
in ()
79 # train set
80 train_set = KangarooDataset()
--> 81 train_set.load_dataset('kangaroo', is_train=True)
82 train_set.prepare()
83 print('Train: %d' % len(train_set.image_ids))

in load_dataset(self, dataset_dir, is_train)
22 continue
23 # skip all images after 150 if we are building the train set
--> 24 if is_train and int(image_id) >= 10:
25 continue
26 # skip all images before 150 if we are building the test/val set

ValueError: invalid literal for int() with base 10: 'sacdf21'
gratefully yours
```

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** December 20, 2019 at 1:05 pm #

REPLY ↗

Sorry to hear that. Perhaps start with the [working tutorial](#) and slowly adapt it to your needs?

[Start Machine Learning](#)



**amine** December 21, 2019 at 7:46 am #

REPLY ↗

absolutely , the tutorial just did awsome but the crafting part keeps bugging over and over again is there any other way to skip this bug, it's just the splitting part train/test datasets that does not work, I m running on collab if helps?

please I m stuck for hours now

very grateful



**Jason Brownlee** December 21, 2019 at 8:16 am #

REPLY ↗

Yes, don't split into train and test file names.



**residence les jardins** Decem

to code does not work without lose the val dataset?

## Start Machine Learning

You can master applied Machine Learning **without math or fancy degrees**.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**João Vitor Granzotti Machado** December 22, 2019 at 3:05 am #

REPLY ↗

Hello, I am a student from Brazil and I am having a problem executing the code. on the line:

```
model.train (train_set, test_set, learning_rate = config.LEARNING_RATE, epochs = 5, layers = 'heads')
```

When running the program I get the following error:

```
raise StopIteration ()  
StopIteration
```

What could be the cause of this ??



**Jason Brownlee** December 22, 2019 at 6:15 am #

REPLY ↗

Sorry, I have not seen this error before. I have some suggestions here that might help:

<https://machinelearningmastery.com/faq/single-faq/why-does-the-code-in-the-tutorial-not-work-for-me>



**João Vitor Granzotti Machado** December 24, 2019 at 3:51 am #

REPLY ↗

Start Machine Learning

I checked the versions of python, tensorflow and even numpy and they are all correct. The output when executing the code is as follows:

```
C:\Users\João Vitor\trabalho>python object_detection.py
```

Using TensorFlow backend.

Train: 131

Test: 32

Configurations:

BACKBONE resnet101

BACKBONE\_STRIDES [4, 8, 16, 32, 64]

BATCH\_SIZE 2

BBOX\_STD\_DEV [0.1 0.1 0.2 0.2]

COMPUTE\_BACKBONE\_SHAPE None

DETECTION\_MAX\_INSTANCES 100

DETECTION\_MIN\_CONFIDENCE 0.7

DETECTION\_NMS\_THRESHOLD 0.3

FPN\_CLASSIF\_FC\_LAYERS\_SIZE 1024

GPU\_COUNT 1

GRADIENT\_CLIP\_NORM 5.0

IMAGES\_PER\_GPU 2

IMAGE\_CHANNEL\_COUNT 3

IMAGE\_MAX\_DIM 1024

IMAGE\_META\_SIZE 14

IMAGE\_MIN\_DIM 800

IMAGE\_MIN\_SCALE 0

IMAGE\_RESIZE\_MODE square

IMAGE\_SHAPE [1024 1024 3]

LEARNING\_MOMENTUM 0.9

LEARNING\_RATE 0.001

LOSS\_WEIGHTS {'rpn\_class\_loss': 1.0, 'rpn\_bbox\_loss': 1.0, 'mrcnn\_class\_loss': 1.0,

'mrcnn\_bbox\_loss': 1.0, 'mrcnn\_mask\_loss': 1.0}

MASK\_POOL\_SIZE 14

MASK\_SHAPE [28, 28]

MAX\_GT\_INSTANCES 100

MEAN\_PIXEL [123.7 116.8 103.9]

MINI\_MASK\_SHAPE (56, 56)

NAME kangaroo\_cfg

NUM\_CLASSES 2

POOL\_SIZE 7

POST\_NMS\_ROIS\_INFERENCE 1000

POST\_NMS\_ROIS\_TRAINING 2000

PRE\_NMS\_LIMIT 6000

ROI\_POSITIVE\_RATIO 0.33

RPN\_ANCHOR RATIOS [0.5, 1, 2]

RPN\_ANCHOR\_SCALES (32, 64, 128, 256, 512)

RPN\_ANCHOR\_STRIDE 1

## Start Machine Learning

You can master applied Machine Learning  
without math or fancy degrees.

Find out how in this *free* and *practical* course.

[START MY EMAIL COURSE](#)

```
RPN_BBOX_STD_DEV [0.1 0.1 0.2 0.2]
RPN_NMS_THRESHOLD 0.7
RPN_TRAIN_ANCHORS_PER_IMAGE 256
STEPS_PER_EPOCH 131
TOP_DOWN_PYRAMID_SIZE 256
TRAIN_BN False
TRAIN_ROIS_PER_IMAGE 200
USE_MINI_MASK True
USE_RPN_ROIS True
VALIDATION_STEPS 50
WEIGHT_DECAY 0.0001
```

WARNING:tensorflow:From C:\Users\João Vitor\AppData\Local\Programs\Python\Python36\lib\site-packages\keras\backend\tensorflow\_backend.py:1822: The name tf.image.resize\_nearest\_neighbor is deprecated. Please use tf.compat.v1.placeholder instead.

WARNING:tensorflow:From C:\Users\João Vitor\AppData\Local\Programs\Python\Python36\lib\site-packages\keras\backend\tensorflow\_backend.py:1208: calling reduce\_max\_v1 (from tensorflow.python.ops.math\_ops) with keep\_dims is deprecated and will be removed in a future version.

WARNING:tensorflow:From C:\Users\João Vitor\AppData\Local\Programs\Python\Python36\lib\site-packages\keras\backend\tensorflow\_backend.py:1242: calling reduce\_sum\_v1 (from tensorflow.python.ops.math\_ops) with keep\_dims is deprecated and will be removed in a future version.

WARNING:tensorflow:From C:\Users\João Vitor\AppData\Local\Programs\Python\Python36\lib\site-packages\keras\backend\tensorflow\_backend.py:1354: add\_dispatch\_support..wrapper (from tensorflow.python.ops.array\_ops) is deprecated and will be removed in a future version.

Instructions for updating:  
keep\_dims is deprecated, use keepdims instead

WARNING:tensorflow:From C:\Users\João Vitor\AppData\Local\Programs\Python\Python36\lib\site-packages\keras\backend\tensorflow\_backend.py:1354: add\_dispatch\_support..wrapper (from tensorflow.python.ops.array\_ops) is deprecated and will be removed in a future version.

Instructions for updating:  
Use tf.where in 2.0, which has the same broadcast semantics as np.where

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

[START MY EMAIL COURSE](#)

WARNING:tensorflow:From C:\Users\João Vitor\AppData\Local\Programs\Python\Python36\lib\site-packages\mask\_rcnn-2.1-py3.6.egg\mrcnn\model.py:553: The name tf.random\_shuffle is deprecated. Please use tf.random.shuffle instead.

WARNING:tensorflow:From C:\Users\João Vitor\AppData\Local\Programs\Python\Python36\lib\site-packages\mask\_rcnn-2.1-py3.6.egg\mrcnn\utils.py:202: The name tf.log is deprecated. Please use tf.math.log instead.

WARNING:tensorflow:From C:\Users\João Vitor\AppData\Local\Programs\Python\Python36\lib\site-packages\mask\_rcnn-2.1-py3.6.egg\mrcnn\model.py:600: calling crop\_and\_resize\_v1 (from tensorflow.python.ops.image\_ops\_impl) with box\_ind is deprecated and will be removed in a future version.

Instructions for updating:

box\_ind is deprecated, use box\_indices instead

2019-12-23 13:45:34.251579: I tensorflow/core/framework/op\_kernel.cc:1572] supports instructions that this TensorFlow binary

Starting at epoch 0. LR=0.001

Checkpoint Path: ./kangaroo\_cfg20191223T134534

Selecting layers to train

fpn\_c5p5 (Conv2D)

fpn\_c4p4 (Conv2D)

fpn\_c3p3 (Conv2D)

fpn\_c2p2 (Conv2D)

fpn\_p5 (Conv2D)

fpn\_p2 (Conv2D)

fpn\_p3 (Conv2D)

fpn\_p4 (Conv2D)

In model: rpn\_model

rpn\_conv\_shared (Conv2D)

rpn\_class\_raw (Conv2D)

rpn\_bbox\_pred (Conv2D)

mrcnn\_mask\_conv1 (TimeDistributed)

mrcnn\_mask\_bn1 (TimeDistributed)

mrcnn\_mask\_conv2 (TimeDistributed)

mrcnn\_mask\_bn2 (TimeDistributed)

mrcnn\_class\_conv1 (TimeDistributed)

mrcnn\_class\_bn1 (TimeDistributed)

mrcnn\_mask\_conv3 (TimeDistributed)

mrcnn\_mask\_bn3 (TimeDistributed)

mrcnn\_class\_conv2 (TimeDistributed)

mrcnn\_class\_bn2 (TimeDistributed)

mrcnn\_mask\_conv4 (TimeDistributed)

mrcnn\_mask\_bn4 (TimeDistributed)

mrcnn\_bbox\_fc (TimeDistributed)

mrcnn\_mask\_deconv (TimeDistributed)

mrcnn\_class\_logits (TimeDistributed)

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE

Start Machine Learning

mrcnn\_mask (TimeDistributed)

WARNING:tensorflow:From C:\Users\João Vitor\AppData\Local\Programs\Python\Python36\lib\site-packages\keras\optimizers.py:711: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

C:\Users\João Vitor\AppData\Local\Programs\Python\Python36\lib\site-packages\tensorflow\python\ops\gradients\_util.py:93: UserWarning: Converting sparse IndexedSlices to a dense Tensor of unknown shape. This may consume a large amount of memory.  
"Converting sparse IndexedSlices to a dense Tensor of unknown shape. "

C:\Users\João Vitor\AppData\Local\Programs\Python\Python36\lib\site-packages\tensorflow\python\ops\gradients\_util.py:93: UserWarning: Converting sparse IndexedSlices to a dense Tensor of unknown shape. This may consume a large amount of memory.  
"Converting sparse IndexedSlices to a dense Tensor of unknown shape. "

C:\Users\João Vitor\AppData\Local\Programs\packages\tensorflow\python\ops\gradients\_util  
IndexedSlices to a dense Tensor of unknown s  
"Converting sparse IndexedSlices to a dense T  
WARNING:tensorflow:From C:\Users\João Vitor\p  
ackages\keras\backend\tensorflow\_backend.  
tensorflow.python.ops.init\_ops) with dtype is d  
Instructions for updating:

Call initializer instance with the dtype argument  
WARNING:tensorflow:From C:\Users\João Vitor\p  
ackages\keras\callbacks.py:705: The name tf.  
tf.compat.v1.summary.merge\_all instead.

WARNING:tensorflow:From C:\Users\João Vitor\AppData\Local\Programs\Python\Python36\lib\site-packages\keras\callbacks.py:708: The name tf.summary.FileWriter is deprecated. Please use tf.compat.v1.summary.FileWriter instead.

Epoch 1/5

Traceback (most recent call last):

```
File "object_detection.py", line 109, in
    model.train(train_set, test_set, learning_rate=config.LEARNING_RATE, epochs=5, layers='heads')
File "C:\Users\João Vitor\AppData\Local\Programs\Python\Python36\lib\site-packages\mask_rcnn-2.1-py3.6.egg\mrcnn\model.py", line 2374, in train
    File "C:\Users\João Vitor\AppData\Local\Programs\Python\Python36\lib\site-packages\keras\legacy\interfaces.py", line 87, in wrapper
        return func(*args, **kwargs)
    File "C:\Users\João Vitor\AppData\Local\Programs\Python\Python36\lib\site-packages\keras\engine\training.py", line 2065, in fit_generator
        generator_output = next(output_generator)
    File "C:\Users\João Vitor\AppData\Local\Programs\Python\Python36\lib\site-packages\keras\utils\data_utils.py", line 710, in get
        raise StopIteration()
StopIteration
```

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

**START MY EMAIL COURSE**

I can't figure out what's wrong. I am very interested in Mask R-CNN and would like to see it working.  
Can you help me plz ?? Thank you for your attention and greetings from Brazil



**Jason Brownlee** December 24, 2019 at 6:43 am #

REPLY ↗

It looks like you are using tensorflow 2.

You must use tensorflow 1.15.



**João Vitor Granzotti Machado**

The versions of the libraries I

Python: 3.6.8  
Tensorflow: 1.15.0  
Numpy: 1.16.0  
Keras: 2.1.0  
Scipy: 1.4.1

So I think the error is not related to the  
the tutorial.

## Start Machine Learning

X

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** December 24, 2019 at 4:57 pm #

I recommend updating to keras 2.2, at least.



**Thanakorn** March 4, 2020 at 4:46 am #

Hi Jason,

My project is on GOOGLE COLAB. Even though the version of my libraries are "Tensorflow: 1.15.0" and "Keras: 2.2.5", it still appears these several lines, so how can fix this out



**Jason Brownlee** March 4, 2020 at 6:01 am #

Perhaps colab is inappropriate.

**amine** December 22, 2019 at 3:56 am #

Start Machine Learning



hi i tried to subset my data to train and val files , the way data is slit in balloon dataset here is the error

FileNotFoundException Traceback (most recent call last)

```
in ()
85 # train set
86 train_set = KangarooDataset()
--> 87 train_set.load_dataset('Amine','train', is_train=True)
88 train_set.prepare()
89 print('train: %d' % len(train_set.image_ids))

in load_dataset(self, dataset_dir, subset, is_train)
26 #annotations_dir = dataset_dir + '/Amine/'
27 # find all images
--> 28 for filename in.listdir(images_dir):
29 # extract image id
30 image_id = filename[:-4]
```

FileNotFoundException: [Errno 2] No such file or directory:

any way to get out of this bug  
thanks



**Jason Brownlee** December 22, 2019 at 6:18 am #

Looks like the data is not in the required location your workstation.

Perhaps put the data in the same directory as your code, and run the code from the command line.



**Amine** December 23, 2019 at 1:18 am #

REPLY ↗

Hi, how could I select the dataset by names rather than I split them by index before and after your breakup point,(150), it seems to me that could be a better fixer for this bug without having to manipulate files?

What would be your code to change the splitting key?.



**Jason Brownlee** December 23, 2019 at 6:55 am #

REPLY ↗

Sorry, I don't have the capacity to prepare custom code.

Perhaps focus on Python basics first?



hi,

i figured out how to split the data to train and val with in each file others sub file (annots and images) to respect your data structure

her is the result

train: 8

test: 4 # seems ok but

NameError Traceback (most recent call last)

in ()

103 mask, class\_ids = train\_set.load\_mask(image\_id)

104 # extract bounding boxes from the masks

-> 105 bbox = extract\_bboxes(mask)

106 # display image with masks and b

107 display\_instances(image, bbox, m

NameError: name 'extract\_bboxes' is

## Start Machine Learning X

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

[START MY EMAIL COURSE](#)



**Jason Brownlee** December 24, 2019 at 10:30 pm #

I have some suggestions here

<https://machinelearningmastery.com/faqs-for-object-detection-with-keras/>



**amine** December 25, 2019 at 7:12 am #

[REPLY ↗](#)

hi, I tried all clues without any success , I know this would not take few minutes to get solved with a professional like you, all I m asking for is some compassion  
thanks



**amine** December 25, 2019 at 9:58 am #

[REPLY ↗](#)

hi, I finally get it screwed away.

thanks



**Jason Brownlee** December 25, 2019 at 10:42 am #

[REPLY ↗](#)

Well done!

[Start Machine Learning](#)



**Jason Brownlee** December 25, 2019 at 10:40 am #

REPLY ↗

Sorry, I don't have the capacity to customize tutorials – I get hundreds of emails/comments per day – lots of people to help.

More here:

[https://machinelearningmastery.com/faq/single-faq/can-you-change-the-code-in-the-tutorial-to-\\_\\_](https://machinelearningmastery.com/faq/single-faq/can-you-change-the-code-in-the-tutorial-to-__)

If adapting the code is challenging, perhaps start with simpler tutorials here and build up to this more advanced tutorial:

<https://machinelearningmastery.com/start-here/#dlfcv>

Or, perhaps hire a contractor.



**raj** January 6, 2020 at 9:07 pm #

(raj) → Mask\_RCNN git:(master) X python start  
Using TensorFlow backend.

Train: 131

Test: 32

WARNING:tensorflow:From /home/debu/.virtualenvs/raj/lib/python3.6/site-packages/keras/backend/tensorflow\_backend.py:514: tf.compat.v1.placeholder instead.

WARNING:tensorflow:From /home/debu/.virtualenvs/raj/lib/python3.6/site-packages/keras/backend/tensorflow\_backend.py:71: The name tf.get\_default\_graph is deprecated. Please use tf.compat.v1.get\_default\_graph instead.

WARNING:tensorflow:From /home/debu/.virtualenvs/raj/lib/python3.6/site-packages/keras/backend/tensorflow\_backend.py:4076: The name tf.random\_uniform is deprecated. Please use tf.random.uniform instead.

WARNING:tensorflow:From /home/debu/.virtualenvs/raj/lib/python3.6/site-packages/keras/backend/tensorflow\_backend.py:3900: The name tf.nn.max\_pool is deprecated. Please use tf.nn.max\_pool2d instead.

WARNING:tensorflow:From /home/debu/.virtualenvs/raj/lib/python3.6/site-packages/keras/backend/tensorflow\_backend.py:1982: The name tf.image.resize\_nearest\_neighbor is deprecated. Please use tf.compat.v1.image.resize\_nearest\_neighbor instead.

WARNING:tensorflow:From /home/debu/raj/Mask\_RCNN/mrcnn/model.py:341: The name tf.log is deprecated. Please use tf.math.log instead.

WARNING:tensorflow:From /home/debu/raj/Mask\_RCNN/mrcnn/model.py:399: add\_dispatch\_support..wrapper (from tensorflow.python.ops.array\_ops) is deprecated and will be removed in a future version.

Instructions for updating:

Use tf.where in 2.0, which has the same broadcast rule

## Start Machine Learning

You can master applied Machine Learning **without math or fancy degrees**.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE

WARNING:tensorflow:From /home/debu/raj/Mask\_RCNN/mrcnn/model.py:423: calling crop\_and\_resize\_v1 (from tensorflow.python.ops.image\_ops\_impl) with box\_ind is deprecated and will be removed in a future version.

Instructions for updating:

box\_ind is deprecated, use box\_indices instead

WARNING:tensorflow:From /home/debu/raj/Mask\_RCNN/mrcnn/model.py:720: The name tf.sets.set\_intersection is deprecated. Please use tf.sets.intersection instead.

WARNING:tensorflow:From /home/debu/raj/Mask\_RCNN/mrcnn/model.py:722: The name tf.sparse\_tensor\_to\_dense is deprecated. Please use tf.sparse.to\_dense instead.

WARNING:tensorflow:From /home/debu/raj/Mask\_RCNN/mrcnn/model.py:772: to\_float (from tensorflow.python.ops.math\_ops) is deprecated and will be removed in a future version.

Instructions for updating:

Use tf.cast instead.

Traceback (most recent call last):

File "start.py", line 150, in

model.load\_weights(model\_path, by\_name=True)

File "/home/debu/raj/Mask\_RCNN/mrcnn/model.py", line

saving.load\_weights\_from\_hdf5\_group\_by\_name(f, lay

File "/home/debu/.virtualenvs/raj/lib/python3.6/site-pac

load\_weights\_from\_hdf5\_group\_by\_name

str(weight\_values[i].shape) + '.)'

ValueError: Layer #389 (named "mrcnn\_bbox\_fc"), wei  
shape (1024, 324).

(raj) → Mask\_RCNN git:(master) X

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** January 7, 2020 at 7:22 am #

REPLY ↗

Looks like a problem with your development environment?

Perhaps confirm TensorFlow 1.15 and Keras 2.2.



**Saurabh** January 9, 2020 at 12:08 am #

REPLY ↗

Hello Jason,

First of all Happy New Year 2020 and looking forward for more exciting blogs from you.

I have one question regarding labeling tool. As I looked into labeling tool but there is no way to rotate bounding box. In custom dataset, object is not straight and I can't rotate images.

Could you please suggest me any other labeling tool which allows to rotate even bounding box?

Thanking you,

Saurabh

Start Machine Learning



**Jason Brownlee** January 9, 2020 at 7:26 am #

REPLY ↗

Sorry, I don't have good advice for image annotation tools.



**Saurabh** January 10, 2020 at 12:53 am #

REPLY ↗

Thank you!



**Siddhartha Pachhai** January 21, 2020

Hi, Jason, the best solution I have found and easy to use.

<https://github.com/ideonate/jupyter-innotater>

\*Also: The output of the tool does not resemble detection, but the tool produces enough so that

\*I remember having some difficulties installing widgets, but I think its fixable), hopefully this w

## Start Machine Learning



You can master applied Machine Learning **without math or fancy degrees**.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** January 21, 2020 at 7:22 am #

REPLY ↗

Thanks for sharing.



**Suman** January 14, 2020 at 7:32 am #

REPLY ↗

Hi Saurab,

May be for object detection you can use labellmg or labelme. For segmentation you can use CVAT tool.



**Suman** January 14, 2020 at 7:36 am #

REPLY ↗

Hi Jason Brownlee,

Great tutorial for beginners like me, thanks.

Here the mask-rcnn is saving weights, but i want to save the model along with the weights like  
model.save('xxxx.h5') . But this function is not working

Start Machine Learning

Thanks



**Jason Brownlee** January 14, 2020 at 1:46 pm #

REPLY ↗

Thanks!

I believe it is using the tensorflow API. Perhaps investigate an appropriate function.



**Niall Delany** January 15, 2020 at 7:53 pm #

REPLY ↗

Thanks for the great tutorial, very helpful in getting me started. I am trying to apply it to a custom dataset.

My question is, suppose I also have binary mask annotations, how do I load them into the model instead of the xml annotations so that they appear as a bounding box?



**Jason Brownlee** January 16, 2020 at 6:12 am #

You're welcome.

Sure, load any custom masks you like.

## Start Machine Learning

X

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**JJ** January 23, 2020 at 10:37 am #

REPLY ↗

Hi.

I'm stuck at "Parse Annotation File" step.

Where could I type the "tree = ElementTree.parse(filename)"?



**Jason Brownlee** January 23, 2020 at 12:56 pm #

REPLY ↗

Sorry to hear that, perhaps try copying the "complete examples" at the end of each section.



**Nelli** January 23, 2020 at 2:27 pm #

REPLY ↗

After training, in the prediction, the displayed image is showing with a bounding box but the label is not there. Please reply to me. Thanks in advance

Start Machine Learning



**Jason Brownlee** January 24, 2020 at 7:42 am #

REPLY ↗

In this case there is only one label, which is kangaroo.

For a more general example with box and label see this tutorial:

<https://machinelearningmastery.com/how-to-perform-object-detection-in-photographs-with-mask-r-cnn-in-keras/>



**Nelli** January 28, 2020 at 6:07 am #

REPLY ↗

Thanks for your quick reply. The suggestion is good.  
Could you please help me in converting the above code?



**Jason Brownlee** January 28, 2020

What is “apk”?



**Suman** January 30, 2020 at 7:56 am #

I want to convert the trained model to apk file to deploy on the mobile devices.  
Please suggest to me.

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** January 30, 2020 at 2:13 pm #

I don't know what apk is sorry, or about putting it on mobile devices.

Perhaps try posting your question to stackoverflow?



**Samrawit** January 26, 2020 at 1:30 pm #

REPLY ↗

Hi

Does Mask-R-cnn only work in annotated image only, can i use normal image? And which annotation approach (automatic,manual or semi automatic) could gives better results?

**Jason Brownlee** January 27, 2020 at 7:01 am

Start Machine Learning



It learns from annotated images.

It is used on normal images.



**Samrawit** January 27, 2020 at 5:39 pm #

REPLY ↗

Thank you. So is there any example on automatically annotating image data-set and how to use them for object detection and mask an object?



**Jason Brownlee** January 28, 2020

No, I believe it is manual at this stage.



**Samrawit** January 28, 2020 at 7:10 pm #

Thank you very much! One more question: Is it possible to do object detection with out using pre-trained weights?

## Start Machine Learning



You can master applied Machine Learning **without math or fancy degrees**.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** January 29, 2020 at 6:31 am #

I don't have such an example.

It makes sense to use pre-trained weights as a starting point for transfer learning.



**mahmoud** January 29, 2020 at 9:12 am #

REPLY ↗

hi Jason, thanks for your illustration

i run the MaskRCNN on my dataset and it gives me horrible result

Train mAP: 0.818

Test mAP: 0.549

can you advice me why it can result in such a big difference on the Train and Test set ???

how i can face this problem.



**Jason Brownlee** January 29, 2020 at 1:46 pm #

REPLY ↗

The model has overfit your training dataset.

Start Machine Learning

This might help:

<https://machinelearningmastery.com/introduction-to-regularization-to-reduce-overfitting-and-improve-generalization-error/>



**Samrawit** March 27, 2020 at 6:48 pm #

REPLY ↗

Hi, i was trying to use Mask\_RCNN and i don't know how to first train the feature map (Backbone) before i return the model to pressed to the next level (Region Proposal Network) because i wanted to see the accuracy of convolution layer.

Thank you!



**Jason Brownlee** March 28, 2020 at 6:16

Not sure that is possible...



**Rohit** January 30, 2020 at 5:12 pm #

Hi Jason, thank you for this wonderful article.

I am working on a case where we have multiple labels for each object in an image.

The task is similar to the one asked in the following problem:

<https://stackoverflow.com/questions/49358088/does-tensorflows-object-detection-api-support-multi-class-multi-label-detection>

Could you suggest how to approach to this problem?

## Start Machine Learning

You can master applied Machine Learning **without math or fancy degrees**.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** January 31, 2020 at 7:39 am #

REPLY ↗

Yes, I believe the mask rcnn can support that.

Perhaps start with this model:

<https://machinelearningmastery.com/how-to-perform-object-detection-in-photographs-with-mask-r-cnn-in-keras/>



**Sai Abinesh** February 5, 2020 at 8:34 pm #

REPLY ↗

Hello Jason,

## Start Machine Learning

Thank you very much for a great tutorial. It's a great resource for anyone trying to get started with object detection and for people who need to check their configurations.

I am retraining just the "heads" layer of a resnet101 backbone, on a 3d synthetic dataset generated using Unreal Engine and python. I have 7 object classes + 1 background, and a total of 591 training images and 60 real images for validation.

Using default training config from the maskrcnn official repo, I suspect there is a case of over-fitting, as the val loss decreases while the training loss decreases.

Here they are pasted below.

<https://imgur.com/a/CgYJxCs>

I also constructed a training curve of my own, by calculating the AP50 (Average Precision at 50% Intersection Over Union) for all the epochs from epoch 1 to epoch 100. It seems like the network is not improving a lot. The curve can be found below.

<https://imgur.com/MWzvWZz>

How should I adjust my learning rates, weight decays? on the size of the dataset, number of classes etc? My

class aerial\_trains\_Config(Config):

"""Configuration for training on the toy shapes dataset.  
Derives from the base Config class and overrides values  
to the toy shapes dataset.

"""

# Give the configuration a recognizable name  
NAME = "Baldonnell\_from\_scratch\_from9m"

# Train on 1 GPU and 8 images per GPU. We can put multiple images on each  
# GPU because the images are small. Batch size is 8 (GPUs \* images/GPU).

GPU\_COUNT = 1

IMAGES\_PER\_GPU = 2

# Number of classes (including background)  
NUM\_CLASSES = 1 + 7 # background + 80 default classes

# Use small images for faster training. Set the limits of the small side  
# the large side, and that determines the image shape.

IMAGE\_MIN\_DIM = 256

IMAGE\_MAX\_DIM = 2048

# Use smaller anchors because our image and objects are small  
RPN\_ANCHOR\_SCALES = (64, 128, 256, 512, 1024) # anchor side in pixels

# Reduce training ROIs per image because the images are small and have  
# few objects. Aim to allow ROI sampling to pick 33% positive ROIs.

TRAIN\_ROIS\_PER\_IMAGE = 32

# Use a small epoch since the data is simple  
STEPS\_PER\_EPOCH = 600

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE

```
LEARNING_RATE = 0.001
LEARNING_MOMENTUM = 0.9

# Weight decay regularization
WEIGHT_DECAY = 0.0001
```



**Jason Brownlee** February 6, 2020 at 8:23 am #

REPLY ↗

Very cool!

This might give you ideas:

<https://machinelearningmastery.com/learning-rate-f/>



**Ofis Taşıma** February 6, 2020 at 6:18 pm #

Thank you ver much your great article about O



**Jason Brownlee** February 7, 2020 at 8:10 am #

I'm happy it helped!



**Ashutosh Srivastava** February 7, 2020 at 8:15 pm #

REPLY ↗

Hi Jason,

This is really a great article. I am trying to solve my multi-object detection problem following your approach, i think there will be a need of just a little tweak into this code but i am stuck.

I have added multiple classes in load\_dataset function:

```
self.add_class("dataset", 1, "list")
self.add_class("dataset", 2, "Menu")
self.add_class("dataset", 3, "Home")
```

but here in load\_mask function you are appending class\_ids statically as 1 "kangaroos", i want to add classes w.r.t objects found.

Kindly check and help.



**Ashutosh Srivastava** February 7, 2020 at 8:19 pm #

REPLY ↗

Start Machine Learning

And as per me, here object detection is implemented but classification is missing as their is only one object. Correct me if i am wrong here.



**Jason Brownlee** February 8, 2020 at 7:08 am #

REPLY ↗

I don't understand, sorry? Can you elaborate?



**Jason Brownlee** February 8, 2020 at 7:08 am #

REPLY ↗

Looks fine to be, perhaps test it?



**Kavilca** February 11, 2020 at 4:06 am #

Awesome article, thank you for this blog



**Jason Brownlee** February 11, 2020 at 5:18 am #

You're welcome.

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Saurabh** February 11, 2020 at 11:50 pm #

REPLY ↗

Hello Jason,

Could you please share your views on "How to label overlapping objects?" What is the best practice with reference to overlapping objects? The problem is most of the labeling tools don't support oriented bounding boxes.

How can I inform my object detector that it should look at only certain part of images without cropping images? Can I edit images and put white/black (constant) color so that object detector will ignore such areas?

Kindly share your views.

Thanking you!



**Jason Brownlee** February 12, 2020 at 5:47 am #

REPLY ↗

I don't have specific advice on the topic, s

Start Machine Learning



**Saurabh** February 12, 2020 at 7:24 am #

REPLY ↗

Thank you!



**Savyasachi** February 12, 2020 at 4:26 am #

REPLY ↗

Hello Dr. Brownlee!

I'm running this matterport/mrcnn code on my custom dataset (to detect comic characters). I'm using a total of 6500 images. My training model saturates with a loss (saturation happens):9th-10th) and it breaks my heart. What's the best way to handle this? (Rest of the config is default)

Thank you so much!



**Jason Brownlee** February 12, 2020 at 5:54 am #

Some of the suggestions here might help:  
<https://machinelearningmastery.com/start-here/#best-practices>

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Savyasachi** February 20, 2020 at 4:35 am #

REPLY ↗

Hi again Dr. Brownlee!

Is there a way to know if my code will perform well/worse in the first epoch (or some time sooner) rather than waiting for 6 long hours to get a loss value?

Every time I make some changes, I have to run it through the whole cycle till I see the saturation (in loss) after which, I have to manually perform a 'Keyboard Interrupt'



**Jason Brownlee** February 20, 2020 at 6:21 am #

REPLY ↗

No.



**Phil** February 13, 2020 at 11:42 am #

REPLY ↗

Hi Jason! This is a great tutorial. This is the exact solution to the problem I'm trying to solve. One quick question. My model gets trained fine but it is not able to detect objects during the training or at the end. So I'm basically left with a trained model that does not work.

I've searched my whole system in the case it was cached at some other location. Could not find it though.  
Could you please tell me if you've come across this kind of a problem before and how to solve it?

I am on windows 10  
With keras==2.2.5  
tensorflow==1.15  
mask-rcnn-12rics==0.2.3



**Jason Brownlee** February 13, 2020 at 1:24 pm #

REPLY ↗

The models are saved in the current working directory I believe, under a subdirectory for the run.



**Juan** February 14, 2020 at 11:58 pm #

Hi Jason, I'm having troubles understandig ho

During the annotation process, do we need to split each class into two separate folders?  
e.g. class1/annots and class1/images, class2/annots and class2/images

I don't know if this is the good approach, since there m

It would be great to know how should the folder structure be and the code for the load\_dataset function.

Thanks!

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** February 15, 2020 at 6:31 am #

REPLY ↗

The choice is yours, as long as it is presented consistently to the model during training.



**Steven** February 18, 2020 at 6:26 pm #

REPLY ↗

Hey Jason,

Thanks, Thanks, Thanks.

This is the best tutorial i found for keras.

I had no hard problems to do this.

You did very well.

**Jason Brownlee** February 19, 2020 at 7:59 am #

Start Machine Learning



Thanks!



**Narottam** February 18, 2020 at 9:26 pm #

REPLY ↗

Hi Jason, i build a single/multi-class classification poc project on different object using your tutorials. Thanks for the neat explanation above.

Now as a part of complete project I require your's suggestion on below points:

1. Ideally in which case model accuracy will be high i.e in single class model or multi-class model (I did single and multi-class on different object) and accuracy on new data seems to be low on traning epoch-100 and learning\_rate = 0.0001
2. What are the different hyper-parameters I can tune to get better accuracy using Mask RCNN
3. I'm working on architecture project, how can i detect how can i detect the line between A & B

Your help will be very much appreciated !!!

## Start Machine Learning

You can master applied Machine Learning **without math or fancy degrees**.

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** February 19, 2020 at 8:03 am #

Well done!

This might give you ideas for improving model performance generally:

<https://machinelearningmastery.com/start-here/#better>

Not sure about detecting lines, sorry. Sounds like classical computer vision might be useful.



**Helmy** February 23, 2020 at 5:22 am #

REPLY ↗

Hey Jason, is it worth it to pass the images through an edge detector like Sobel, prewitt, canny as a pre-processing step before sending them off to Mask RCNN ?

In an attempt to make it "Easier" to increase accuracy ? Any literature or references you recommend reading ?



**Jason Brownlee** February 23, 2020 at 7:32 am #

REPLY ↗

Probably not. Perhaps try it?



**G** February 27, 2020 at 3:57 pm #

Start Machine Learning

Hi! Can you explain what's going on with:

```
pyplot.subplot(330 + 1 + i)
```

Why those numbers?



**Jason Brownlee** February 28, 2020 at 5:56 am #

REPLY ↗

3 rows, 3 columns and the image number from 1 to 9.



**Runist** February 28, 2020 at 6:04 pm #

The code give me a lot of warning such as "Allocated 1.15GiB with freed\_by\_count=0. The caller indicated there could be performance gains if more memory were freed. May be I should change a better computer. But is there



**Jason Brownlee** February 29, 2020 at 7:09 am #

REPLY ↗

Perhaps try running on a machine with more memory, e.g. GPU.

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Runist** February 29, 2020 at 1:03 pm #

REPLY ↗

You mean GPU more RAM or CPU?



**Jason Brownlee** March 1, 2020 at 5:20 am #

REPLY ↗

Perhaps.



**He** March 3, 2020 at 12:29 am #

REPLY ↗

Hi Jason, can you kindly create a tutorial to Estimate the Speed of Object in the detected boxes? Or have any reference to such tutorials?

**Jason Brownlee** March 3, 2020 at 6:00 am #

Start Machine Learning



Thanks for the suggestion.



**Nourhan** March 4, 2020 at 9:21 pm #

REPLY ↗

Hello Mr. Jason, thank you for the very beneficial and informative tutorials you are making. I appreciate your great effort. I would like to suggest having a similar tutorial in multiple classes object detection not only a one, if possible. Thanks again.



**Jason Brownlee** March 5, 2020 at 6:34 am #

REPLY ↗

Thanks.

Great suggestion.



**Dimitrios Politikos** March 6, 2020 at 10:13 pm #

Hi Jason,

When I'm trying to evaluate the PredictionConfig in a c

```
cfg = PredictionConfig()
# define the model
model = MaskRCNN(mode='inference', model_dir='./', config=cfg)
# load model weights
model.load_weights('mask_rcnn_train_config_0005.h5', by_name=True)
# evaluate model on training dataset
train_mAP = evaluate_model(train_set, model, cfg)
```

I get as a message:

"re-start from epoch 5 " and the run stuck there.

Should I wait or there is a bug in my code?

Thanks,

Dimitris

## Start Machine Learning

You can master applied Machine Learning  
without math or fancy degrees.

Find out how in this *free* and *practical* course.

START MY EMAIL COURSE



**Jason Brownlee** March 7, 2020 at 7:17 am #

REPLY ↗

No, you can ignore the warning I think.

## Start Machine Learning



**Dimitrios Politikos** March 7, 2020 at 8:03 am #

REPLY ↗

Thank you for your response. Really appreciated!

D.



**Jason Brownlee** March 8, 2020 at 6:00 am #

REPLY ↗

You're welcome.



**Steven** March 7, 2020 at 12:48 am #

Hey Jason,

I successfully run your project on my cpu.  
Now I want to do this on my gpu.

I installed the latest versions of all libraries.

TF-gpu : 2.1.0

keras-gpu: 2.3.1

cudnn: 7.6.5

cudatoolkit: 10.1.243

the problem is, that model.py throws many Errors like renaming tf.log(x) to tf.math.log(x)...

the question is:

1. Can you publish a project for latest library versions?
  2. Can you say which library versions I have to install for using your project on gpu?
- Like I said it works fine in an environment without gpu usage. But this never happened.

I hope you can help me.

## Start Machine Learning

X

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** March 7, 2020 at 7:19 am #

REPLY ↗

The example will not work with TensorFlow 2 because the Mask RCNN library has not yet been updated to support it.



**Steven** March 7, 2020 at 12:52 pm #

REPLY ↗

could you give me please an example for settings?

Start Machine Learning

which version do you use, or which are available.

Stucked on it for 2 days now...



**Jason Brownlee** March 8, 2020 at 6:01 am #

REPLY ↗

Yes, I mention this at the top of the page.

You can use TensorFlow 1.14 or 1.15.



**Steven** March 11, 2020 at 2:56 am

I tried but don't work how it s

I got now:

tf-gpu: 1.14

keras: 2.2.5

cuda: 10.0

cudnn: 7.4.1.5

The script run until Epoch 1/20:

Image 1/100 [.....]

and it doesn't make progress.

Can you give me please your versions of these 4 things, to get it work?

## Start Machine Learning

X

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Jason Brownlee** March 11, 2020 at 5:28 am #

Perhaps there is something going on with your workstation.

Perhaps try running other code to confirm your libraries can fit a basic model.

Perhaps try running the code on another machine to confirm you have everything you need?



**jackson** March 12, 2020 at 2:43 am #

REPLY ↗

Hello Jason,

I have a question!

Why is it that when training the model, the loss for the classification output on the train set is usually lower than that of the validation datasets (e.g. mrcnn\_class\_loss and val\_mrcnn\_class\_loss), as well as why is the loss for the bounding box output for the train lower than that of the validation datasets (mrcnn\_bbox\_loss and val\_mrcnn\_bbox\_loss)?

Thank you.

## Start Machine Learning



**Jason Brownlee** March 12, 2020 at 8:52 am #

REPLY ↗

Some difference between the two sets is to be expected, see this:

<https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/>



**Steve** March 16, 2020 at 7:46 pm #

REPLY ↗

Hey Jason,

is it possible that my training on gpu (8GB gpu) not work  
is tried to use resnet50 but ist got Allocation problems.

How many gpu memory do you have?



**Jason Brownlee** March 17, 2020 at 8:13 am #

REPLY ↗

Maybe.

I generally recommend training on AWS EC2:

<https://machinelearningmastery.com/develop-evaluate-large-deep-learning-models-keras-amazon-web-services/>



**Steven** March 16, 2020 at 10:50 pm #

REPLY ↗

Hey Jason,

i'm not sure if i can train pictures with the size 1024×1024 with resnet50 on my gpu

I 've got a GeForce RTX 2070 and i can't run it on gpu.

I don't have any Exceptions. The commandprompt just hanging.

Over a monitor for gpu i see that it want to use all, but i think it's not enough.

Can you help me?

Another question is, how can i manipulate the resnet50 to a smaller net (if it's the solution for my problem)



**Jason Brownlee** March 17, 2020 at 8:16 am #

REPLY ↗

Might be too large. Perhaps try smaller im

Start Machine Learning



**Steven** March 18, 2020 at 2:39 am #

REPLY ↗

I tried to use 64×64 images but it still not works.

I also tried just one picture per epoch.

The problem occurred for others too. [https://github.com/matterport/Mask\\_RCNN/issues/287](https://github.com/matterport/Mask_RCNN/issues/287)

Now I'm wondering if it's a problem of the generator.

I don't think it's a problem of storage of gpu because the script stucked without any errors.

Can you pls help with some advices?



**Jason Brownlee** March 18, 2020 at 11:27 pm #

Sorry to hear that, I don't have an

## Start Machine Learning



You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**Steven** March 18, 2020 at 11:27 pm #

did you run your project on gpu  
is it even possible?

and what libraries you use?

pls say me your versions of :

Tensorflow

Keras

Cudnn

Cuda

Python

I really have no other ideas than trying your versions and hope for working



**Jason Brownlee** March 19, 2020 at 6:27 am #

Yes.

Tensorflow 1.14 or 1.15, Python 3.6 and an EC2 instance:

<https://machinelearningmastery.com/develop-evaluate-large-deep-learning-models-keras-amazon-web-services/>

It also works just fine on CPU with the same libraries.

Start Machine Learning

**chiraz** March 18, 2020 at 11:52 pm #

REPLY ↗

Hello Jason

I am working with Faster rcnn for defects detection and i would like that you help me how to detect objects from scratch with my own dataset with of course a pretrained cnn like vgg16 or resnet. How to prepare the data and insert it in Jupyter notebook or even in anaconda virtual environment. I will be very thankful

Thanks

**Jason Brownlee** March 19, 2020 at 6:28 am #

REPLY ↗

This tutorial will help you to setup your de  
<https://machinelearningmastery.com/setup-python-anaconda/>

**Thanakorn** March 20, 2020 at 12:36 am #

Hi Jason,

I would like to ask you that how can I add the name of

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

**START MY EMAIL COURSE**

REPLY ↗

**Jason Brownlee** March 20, 2020 at 8:46 am #

You can draw text directly onto the image. Perhaps review the pillow API or the matplotlib API.

**João Vitor Granzotti Machado** March 24, 2020 at 2:55 am #

REPLY ↗

Hi Jason, I'm trying to make a traffic light detector, I have a very large dataset of images known as DTLD and I would like to use it in this tutorial.

The images of the dataset have dimensions 2048X1024 and the objects to be detected are very small. When performing the training and validation for the first time, the result obtained was very bad. I imagine it is due to the resizing performed on the images.

If I change the IMAGE\_RESIZE\_MODE parameter from "square" to "none" can I continue using transfer learning normally? Or would it be necessary to train the network from scratch?

In the config.py file the following information is provided, however I don't know if I can change this parameter according to my will.

```
# Input image resizing
# Generally, use the "square" resizing mode for training and prediction
# and it should work well in most cases. In this mode, i
```

**Start Machine Learning**

```
# up such that the small side is = IMAGE_MIN_DIM, but ensuring that the
# scaling doesn't make the long side > IMAGE_MAX_DIM. Then the image is
# padded with zeros to make it a square so multiple images can be put
# in one batch.

# Available resizing modes:
# none: No resizing or padding. Return the image unchanged.
# square: Resize and pad with zeros to get a square image
# of size [max_dim, max_dim].
# pad64: Pads width and height with zeros to make them multiples of 64.
# If IMAGE_MIN_DIM or IMAGE_MIN_SCALE are not None, then it scales
# up before padding. IMAGE_MAX_DIM is ignored in this mode.
# The multiple of 64 is needed to ensure smooth scaling of feature
# maps up and down the 6 levels of the FPN pyramid
# crop: Picks random crops from the image. First, scale
# on IMAGE_MIN_DIM and IMAGE_MIN_SCALE, then
# size IMAGE_MIN_DIM x IMAGE_MIN_DIM. Can be
# IMAGE_MAX_DIM is not used in this mode.
IMAGE_RESIZE_MODE = "square"
IMAGE_MIN_DIM = 800
IMAGE_MAX_DIM = 1024
```



**Jason Brownlee** March 24, 2020 at 6:08 am #

I wonder if you can use smaller images.

It might be worth looking in the literature for models that are appropriate for this specific problem or detecting small objects generally.

## Start Machine Learning X

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

**START MY EMAIL COURSE**



**João Vitor Granzotti Machado** March 24, 2020 at 9:38 am #

**REPLY ↗**

In this case an interesting processing would be to change the size of the 2048×1024 to 2048×512 images, cutting the lower half of the image, as it is a known fact that there are no traffic lights below the horizon line.

Using the default values for maximum and minimum size of images (IMAGE\_MIN\_DIM = 800, IMAGE\_MAX\_DIM = 1024) I didn't get a good result, I was wondering if it would be possible to increase the values IMAGE\_MIN\_DIM and IMAGE\_MAX\_DIM and continue using transfer learning.



**Jason Brownlee** March 24, 2020 at 1:44 pm #

**REPLY ↗**

Good question, perhaps try it and compare results?

**Start Machine Learning**



**João Vitor Granzotti Machado** March 25, 2020 at 3:27 am #

anging the values of IMAGE\_MIN\_DIM and IMAGE\_MAX\_DIM I get the following error:

OSError: [Errno 12] Cannot allocate memory

I'm running the code on Google Colab, as I don't have the processing power necessary to train the base in a reasonable time on my computer.

Therefore, there are two possibilities for this error, either it is related to excess size of the images or it is not possible to carry out transfer learning by changing the mentioned parameters.



**Jason Brownlee** March 25, 2020 at 10:30 pm #

Perhaps try and AWS EC2 wi

## Start Machine Learning

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE



**João Vitor Granzotti Machado** March 25, 2020 at 10:45 pm #

The fact that my images are in BGR (sabotaging my training



**Jason Brownlee** March 24, 2020 at 1:44 pm #

REPLY ↗

Perhaps you can convert some and see if it makes a difference?



**Prashanth Mariappan** March 28, 2020 at 2:50 am #

REPLY ↗

Hey this is a great tutorial it is very helpful could you please tell what are all the changes required if we want to train multiple classes. I tried on my own iam getting some errors in load\_mask() function



**Jason Brownlee** March 28, 2020 at 6:26 am #

REPLY ↗

Very few changes, just to the definition of the model – e.g. how the dataset is loaded and classes are defined.

**Prashanth Mariappan** March 28, 2020 at 10:15 pm #

Start Machine Learning



I've made the changes . But I am getting the following error in the part where we check out data set with masks. Please help me out

```
AssertionError Traceback (most recent call last)
in ()
105 bbox = extract_bboxes(mask)
106 # display image with masks and bounding boxes
-> 107 display_instances(image, bbox, mask, class_ids, train_set.class_names)

/content/drive/My Drive/masked rcnn/Mask_RCNN/mrcnn/visualize.py in display_instances(image,
boxes, masks, class_ids, class_names, scores, title, figsize, ax, show_mask, show_bbox, colors,
captions)
103 print("\n*** No instances to display *** \n")
104 else:
-> 105 assert boxes.shape[0] == masks.shape[0]
106
107 # If no axis is passed, create one and auto-
```

AssertionError:

In load data set i have added the 2nd class using class\_ids. I have added it to class\_ids. What else should I do?



**Jason Brownlee** March 29, 2020 at 10:30 pm #

Sorry, I don't know the cause of your fault. Perhaps try posting your code and issue to stackoverflow?



**Rajesh** March 30, 2020 at 6:14 pm #

REPLY ↗

Can we use other weights for training. If so where can we download it.



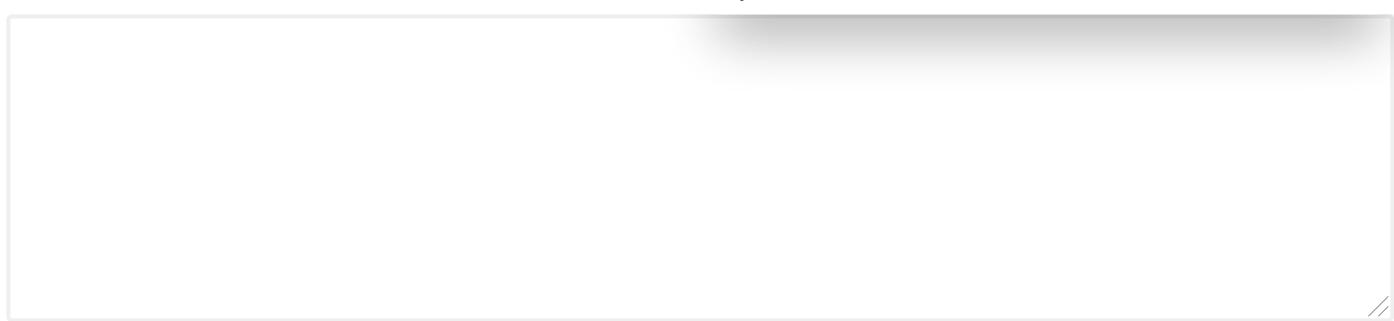
**Jason Brownlee** March 31, 2020 at 7:59 am #

REPLY ↗

I'm not aware of other pretrained weights.

Leave a Reply

[Start Machine Learning](#)

 Name (required) Email (will not be published) (req) Website

SUBMIT COMMENT



### Welcome!

My name is Jason Brownlee PhD, and I help people learn machine learning.

[Read more](#)

## Start Machine Learning



You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

 Email Address

[START MY EMAIL COURSE](#)

### Never miss a tutorial:



### Picked for you:



[How to Train an Object Detection Model with Keras](#)



[How to Develop a Face Recognition System Using FaceNet in Keras](#)



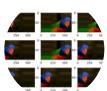
[How to Perform Object Detection With YOLOv3 in Keras](#)

[How to Classify Photos of Dogs and Cats \(with 97% accuracy\)](#)

[Start Machine Learning](#)



How to Configure Image Data Augmentation in Keras



## Loving the Tutorials?

The Deep Learning for Computer Vision EBook is where I keep the **Really Good** stuff.

SEE WHAT'S INSIDE

### Start Machine Learning X

You can master applied Machine Learning  
**without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

START MY EMAIL COURSE

© 2019 Machine Learning Mastery Pty. Ltd. All Rights Reserved.  
Address: PO Box 206, Vermont Victoria 3133, Australia. | ACN 123 123 123  
[LinkedIn](#) | [Twitter](#) | [Facebook](#) | [Newsletter](#) | [RSS](#)

[Privacy](#) | [Disclaimer](#) | [Terms](#) | [Contact](#) | [Sitemap](#) | [Search](#)

Start Machine Learning