Data preprocessing

You will now do some data preprocessing so it is formatted properly before it is fed to the model:

* Convert the class labels to one-hot representations
* convert everything (i.e. train images, gt boxes and class labels) to tensors.

# The `label\_id\_offset` here shifts all classes by a certain number of indices;

# we do this here so that the model receives one-hot labels where non-background

# classes start counting at the zeroth index.  This is ordinarily just handled

# automatically in our training binaries, but we need to reproduce it here.

label\_id\_offset = 1

train\_image\_tensors = []

# lists containing the one-hot encoded classes and ground truth boxes

gt\_classes\_one\_hot\_tensors = []

gt\_box\_tensors = []

for (train\_image\_np, gt\_box\_np) in zip(train\_images\_np, gt\_boxes):

    # convert training image to tensor, add batch dimension, and add to list

    train\_image\_tensors.append(tf.expand\_dims(tf.convert\_to\_tensor(

        train\_image\_np, dtype=tf.float32), axis=0))

    # convert numpy array to tensor, then add to list

    gt\_box\_tensors.append(tf.convert\_to\_tensor(gt\_box\_np, dtype=tf.float32))

    # apply offset to to have zero-indexed ground truth classes

    zero\_indexed\_groundtruth\_classes = tf.convert\_to\_tensor(

        np.ones(shape=[gt\_box\_np.shape[0]], dtype=np.int32) - label\_id\_offset)

    # do one-hot encoding to ground truth classes

    gt\_classes\_one\_hot\_tensors.append(tf.one\_hot(

        zero\_indexed\_groundtruth\_classes, num\_classes))

print('Done prepping data.')