How Does Faster-RCNN Accepts Various Image Sizes?

Faster-RCNN accepts various image sizes as the input. This can be seen in the screenshot below. However, as noted in the `config.py` file from `SCALES` and `MAX_SIZE` variables, the variation of acceptance image sizes is constrained within a specified range: a minimum of 600 pixels on one side and a maximum of 1000 of one side. In the case of an aspect ratio of an image not conforming to both standards, the maximum size of 1000 is observed and the minimum size of 600 pixels is ignored.

The ability of a parametric machine learning model allowing for such flexbility surprised me. After all, a parametric machine learning model is a *function* meaning the domain is fixed. For example, a function $f : \mathcal{R}^2 \rightarrow \mathcal{R}$ only accepts pairs of real numbers and gives out one real number. The function f can only be given two real numbers. Yet the Faster RCNN function accepts inputs with different sizes. So what is going on?

How does a parametric machine learning model, such as Faster-RCNN, allow for such flexibility? The answer is Region Proposal Network (RPN) layer.

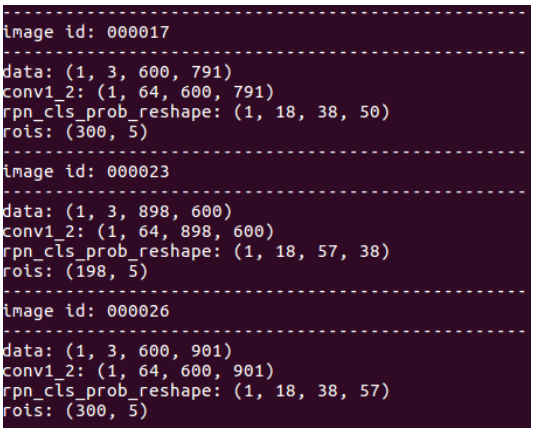

Figure 1. Different input image sizes are used as input in the Faster-RCNN model. The screen is the output from a program running the Faster-RCNN model on images from PASCAL VOC. The "image id" is the name of the image. The first image's name is 000017. The names "data", "conv1_2", "rpn_cls_prob_reshape", and "rois" are different layers of the Faster RCNN model. The "data" layer is the size of the input image.

Before we dive into the details of the solution, we need to first understand the basics of how Faster RCNN operates.

1 Summary of Model

The input in the deep learning model is an image. The image's size, as noted above, can vary. The shortest side is at least 600 pixels. The longest side is at most 1000 pixels. If an image can not meet both size criteria, then only the constraint of 1000 pixels is satisfied. This decision seems arbitrary.

Preprint. Work in progress.

The first set of layers of the deep learning model are a set of convolution layers with rectified linear unit activations and max pooling in-between the convolutions. The precise details are not important to this discussion. For more details, reference the Faster RCNN paper or the VGG16 model architecture [1], [2]. The important point is that the operations are strictly convolutional. We demonstrate in Section 2 that convolutional can be applied to images of any size given some mild constraints. This allows the various image input sizes to be appled through the first several layers of the deep learning model.

The Region Proposal Network (RPN) layer is the next layer of the model. This is the layer where inputs of many sizes are funneled into a set output size. The input size can vary, as shown by the "rpn_cls_prob_reshape" variable in Figure 1.

2 Convolution Accepts Many Sizes

We assume the reader has a general understanding of the convolution operation. Here we provide a brief review to establish a common vocabulary. What might not be obvious at first is that convolution can be applied to any size image, given some mild constraints on the image size. These are given in the next paragraph.

As mentioned previously, there are some mild constraints on the image size for convolution to be applied. The contraints are given as follows, and then after we argue these contraints are easily met in many practical settings.

1. The image must not be "too small"
2. **what about stride? and padding?**

3 Region Proposal Network

The RPN layer converts a feature map (the activations of a layer in a deep learning model) into a set of proposed object regions with associated *objectness* scores. An *objectness* score is a measure of how likely the associated region is to contain an object.

Different input image sizes are used as input in the Faster-RCNN model. The screen is the output from a program running the Faster-RCNN model on images from PASCAL VOC. The "image id" is the name of the image. The first image's name is 000017. The names "data", "conv1_2", "rpn_cls_prob_reshape", and "rois" are different layers of the Faster RCNN model. The "data" layer is the size of the input image.