**4.1 Understanding ImageGenerator**

But what happens when you use larger images and where the feature might be in different locations? For example, how about these images of horses and humans? They have different sizes and different aspect ratios. The subject can be in different locations. In some cases, there may even be multiple subjects.

In addition to that, the earlier examples with a fashion data used a built-in dataset. All of the data was handily split into training and test sets for you and labels were available. In many scenarios, that's not going to be the case and you'll have to do it for yourself.

图片包含 屏幕截图

描述已自动生成

In particular, the **image generator** in TensorFlow.

So for example, consider this directory structure. You have an images directory and in that, you have sub-directories for training and validation. When you put sub-directories in these for horses and humans and store the requisite images in there, the image generator can create a feeder for those images and auto label them for you.



图片包含 屏幕截图

描述已自动生成

It's a common mistake that people point the generator at the sub-directory. It will fail in that circumstance. **You should always point it at the directory that contains sub-directories that contain your images.**

The nice thing about this code is that the images are resized for you as they're loaded. So you don't need to preprocess thousands of images on your file system.

图片包含 屏幕截图

描述已自动生成

**4.2 Defining a ConvNet to use complex images**

图片包含 屏幕截图, 文字

描述已自动生成

First of all, you'll notice that there are three sets of convolution pooling layers at the top. This reflects the higher complexity and size of the images.

Another thing to pay attention to is the input shape. We resize their images to be 300 by 300 as they were loaded, but they're also color images. So there are three bytes per pixel.

Remember before when you created the output layer, you had one neuron per class, but now there's only one neuron for two classes. That's because we're using a different activation function where sigmoid is great for binary classification, where one class will tend towards zero and the other class tending towards one.

图片包含 屏幕截图

描述已自动生成

4.3 **Training the ConvNet with fit\_generator**