Labelling images with fast.ai & ResNet

OK, let's talk about the plan for this time. We're going to take some images that are in a folder and have a machine learning model identify what's inside those images. Is it a plane, a car, what is it? That could be super useful if you have lots and lots of images, and you don't want to have to look through all of them. Right?

OK. So the way we're going to do this is we're going to take a model that's already been trained on lots and lots of images. In fact, this model is called resnet, R-E-S-N-E-T, and it has been trained using a dataset called Image Net. Image Net has 14 million pictures in it and their labels. And so this model called resenet has been trained to try to identify based on the pixels in those images, what the descriptions should be. It's basically designed to identify the contents of images.

So we're going to use resnet, which has been trained on the image net database, to see if we can have it identify the images in our folder. Remember, it's probably best to watch the entire video first to see how we're going about this, and then go back and use the notebook and go along in the notebook and try it yourself. You can even watch the video a second time while you're doing that and stop and start the video as you need.

So back to Colab.Research.Google.com. As always, we're going to click on the GitHub tab here at the top and type Quartz. And this one's BB label images with resnet.

Here's what we're going to do. If we have a folder full of images, we want to get descriptions of each image, and we're going to do this with a pre-trained model. This model has been trained on millions of pictures, and it's a neural network that's been trained to look at images and make a guess from 1,000 possible categories, or labels, or classifications. All of those words mean the same thing. And we're going to use fast A.I., which is a Python library, to do this. A lot of what we're doing in this class will be with fast A.I., so our plan is to load some test images and prepare the list of 1,000 possible labels or classes. We're going to load in that resnet model, so this is the pre-trained model, which has been trained to recognize all of these images. And then we're going to try predicting the class, or the content, of an image, and then we'll try to do that for several images. Alright. So here are all the credits from where I got a lot of this code and help to build this. And we've done this part. We do need to run the cell, as usual, and say, "Yes, we know it wasn't written by Google." And then we need to load a whole bunch of programs and libraries, and so that happens right here. OK.

And we're also going to load the data. I will show you later how to load your own data, but for now, this cell will go and get just a few images that we're going to use to classify. So here, I'm just going to set the data path. That is the place where all of the data can be found, and we can see LS is list. And it's like directory. And so I can see that my data folder now has two things in it. There's a JSON file that has the list of 1,000 classifications in order. We also have an images folder. I can look at that by listing the data, slash images, and you can see there's just eight images in here. That's what we're going to be be looking at. If we look at one of these images, you can see these are just pictures I've taken. There is a little river in upstate New York. So I'm going to also load in that JSON file of image classifications, so I'm going to do. And let's take a look at those.

OK, so this is a list of image classifications. It's actually in order. These are just the first 10 image classifications, and basically the model will spit out a number. This would be class 0. This is class 1. This is class 2. And so we get to turn those numbers into actual English, which is helpful. Alright. So next, we're going to actually go out to the Internet and get the resnet model. We're going to use one called resnet 50. And the first time you do this, it'll actually go download it. It's loading that into our notebook. Remember, this is a model. It doesn't actually have 14 million images in it. What it has is the math, the numbers, and the weights that it used to calculate which images belong to which categories, which classes. And so it's the model. It's basically the smarts without having to download all of the images.

We're going to talk about transforms later. So for now, just hit this button. Alright. And then we're going to hit play, here, and that actually loads our data into a thing called the data bunch. In almost every case with fast A.I., we load our data into this thing called a data bunch. We're going to go into more of that later, I promise.

So for now, we're going to make this thing called a learned variable. A learner is basically the model that we're going to use. In this case, though, we're using this resnet model. So here we're saying use our data and use it with this model. Now that we've made our learner, basically, this is the prediction engine, this is the predictor, what I can do is I can open a new image and call it image. I can show you that image here. There it is. That's the image we had before. And then I can use this line to predict what it thinks this thing is. And you can see what happens here is we get a prediction class, get something called the prediction index, and we get a prediction list. And that all comes from saying learn.predict on our image. So this is what actually generates values that get put into these three things. In fact, if I say print the pred class, the prediction class, look, it thinks that it's a valley. That's pretty good, actually. And then using these other two values, here, I can actually find out how confident it was that that's a valley. So here's the code for that. 88 percent is what it thinks, 88 percent confidence that this is a valley. Pretty good.

If we wanted to look at all eight images, what I can do is I can use this code where I assign image files to the list of images. That's what's going on here. And then just print out that list of image files. There they are. Recognize them from before. Bunch of JPEGS. And then this is actually a loop written in Python. If you don't know Python, that's OK. Basically what's happening is that it's saying that in image files we're going to go through each file, and we're going to open it, and we're going to call it image. IMG. Then we're going to do our prediction. This is the same line from before. Learn.Predict on the image, whichever one we're on at the moment, and we're going to get are variables here. And we can calculate our confidence, and then we can print each one: the file, the class, that's the thing we think it is, and its confidence. So if I run this, you can see it's going through the list of eight images we have in our folder, and it tells us what it thinks is inside.

Now, imagine you have 8,000 images instead of just eight. This could be really useful, right? You might be able to even use this and search for the things you're looking for. You could do a text search of all these images that you have in your folder. That's pretty cool. But here we're not actually looking. There's our valley. That worked. But what about these others? How good was it?

This code will do the same thing, except it will also show the image. So let's take a look here. It's going through. It's finding them all. Alright, let's see how we did. OK. This, it said, was an umbrella confidence of 61 percent. Pretty good. This says it's an aircraft carrier.

OK, I see why. It's actually the Staten Island Ferry. So close, but not quite. But I mean, within reason. Right? It's a boat. Here's our valley. This, it said, was a stopwatch. I think I would have called that just a watch, but okay. This says it's a beer glass. Well, if you were looking for beer glasses, this just doesn't work. This is a wine glass with wine in it. Here it says this is a liner, sort of like an ocean liner, I guess. OK. I can buy that. And it said this one was a barn, not quite a barn. But if it doesn't have gazebo in it's thousand options, I can get that. And here it says this is a marmot. That's not a marmot at all. It is completely wrong. And you can tell it wasn't entirely confident on that. And so, no, this is a seal.

OK. Well, that's how you could do this. But you can see that with a machine learning model that we can just load right on this computer here for free, we get a pretty good sense of what's inside our folder.

Be sure to check out the readings for this week. There's a really good one about some of the problems with the image net database. Now image net is that 14 million image data set with all the labels. Some of those labels are kind of problematic. So it works OK for our particular situation, but you should understand, and this is true for any model, you should understand the data upon which that model was trained. Because if there is problems or biases, you're going to want to know that for your work.

So for next time, we'll actually do something very similar, but we'll use the power of Google to actually identify our images.