Tips and tricks for better image models

So our bike detector model could be a little bit better. Let me show you some tricks to make a model even better at detecting things that you're looking for.

So now we're going to try to find ways to make the model even better. And one of the ways we can do that is we can kind of randomize what happens to the training images. Let me show you what I mean. These are this is called data augmentation or transforms. And basically what it does, it takes the input data. So in our case, the images of bikes that I took on the street and kind of messes with the imagery a little bit. For example, you could randomly change the brightness. So then the model gets better at detecting bikes in darker shadows and in sunlight. But without having to go find bikes in shadow and in sunlight, it just can randomly change the brightness. Same can be true for contrast. Right. So this is going to be not that big a difference to our eyes, but to a computer. The values of the pixels are changing a lot. This is still a cat, right? It's in fact the same cat. But to a computer, it has to learn those variations a little bit differently because the computer is looking at groups of pixels to try to identify the characteristics of a cat.

Similarly, we might apply some random cropping again as we load in our data. Maybe we crop it randomly so that even all of these images get registered as cat and not dog or car or something like that. And again, sort of making more data out of the images we have. There are lots of transformations. Some of them involve rotation. Some of them involve flipping. This is very common. Flipping left and right because you want to be able to recognize a cat. In both cases. Right. And that's a very easy way to do it.

So then you can actually see here there are some we can zoom in on this a little bit. We can zoom in on this a little bit here and see here this is actually kind of blurring and gittering the image to try to make it more generalizable, right? So if the computer can recognize this as a cat, it's probably going to be better at recognizing this as a cat as well.

And you can add padding to the border. There's a couple of technical reason for doing that. You can even do something which is called prospective work, which warps the images. So it looks like almost like the cat or whatever you're taking a picture of. The bike is shot from different angles, even though it's the same image.

OK. So we may you may remember that we were, if I scroll all the way back up. We said that we were using something called get transforms. Let's see. Where was I?

Here's our data block. Right here. We're using get transforms and get transforms is actually a collection of transforms. It's a it's a pretty good set. And it has shown to work pretty well in fast A.I.. And I can go through here. So get transforms is actually doing that horizontal flip. It's not doing the vertical flip. I think somehow I got some upside down images in there and I think that was my fault. It rotates the image. Plus or minus 10 degrees, zooms in a little bit, changes the lighting about 20 percent, warps it 20 percent. And all of this happens with a percent chance of 75. So this is just this is these are the transforms that come out of the box when you use that get transforms command, which we did before.

We can change those a little bit. And I mean, I'm actually going to do this. Here we could say things like vertical flip equals true or rotate more. What I'm going to do is I'm actually going to for the next round, I'm going to do a little bit more transformation and I'm going to say, yeah, use the get transforms, but also don't rotate it as much. Bikes are usually, you know, kind of horizontal. I don't think we need to rotate them too much. And I do want to zoom in more. I happen to know that a lot of the pictures I took have the bike sort of in the center. So let's try that. We'll, we'll zoom in more. So we'll do some data augmentation just by adding some more transformations.

So that's one trick to try to make your model what's called more generalizable.

Another way to improve the model using the same set of images is to actually change the size of the images. This is really fascinating. And here's why this works. So originally, remember, we had images that were we had shrunk them down to 2x24 by 2x24. And if you think about it, that constitutes 50,000 pixels per image. Right. So there are little images, but there's still lots of pixels. And the computer is looking at little groups of pixels to try to identify patterns of bikes, you know, like part of the wheel or the handlebars or the crossbar. So if we load those same images, but instead of shrinking them down to 2x24 by 2x24, we shrink them down to 5x12 by 5x12. Well, that's those that that constitutes 262,000 pixels per image.

So to the computer, those are completely different images. And still, we're asking it to look for bikes inside those images. So if you think about all of the little groups, there's many different kinds of groups of pixels in there. And we can actually basically use those 100 images as 100 to the computer. Fresh images that it can learn from. So that's what we're going to do here. Remember, we have our learner already and we have our data already.

So we're going to load in some new data. It's actually slight. It's pretty much the same. But we're going to change it a little bit. But we're going to call this data 2. And everything else is really the same. We're still going to get it from our data path. We're going to split by a random percentage. We're going to do labels from the folder. Here we're going to use my transforms, which remember, we've altered them a little bit. Now we're going to do less rotation and more zoom. But here's the key thing. We're going to bring in the size at 5x12 by 5x12. This will be seen as brand new images to the model. We're going to lower our batch size because those images are bigger, so we don't want to overwhelm the GPU. But that's OK. We're good 12.

And then let's run that cell. So we've loaded in our in our data. Here it goes. And now we're going to add that into the learner, so we're going to say, OK, now the data is going to be this data 2.

So we're going to load that in and we're going to freeze what we've trained before just to make sure that it's all set. And then we're going to train some more using this new data. So I'm going to start this training cycle and I'll see you in just a couple of minutes.

OK, so a couple of minutes later. And look at our error rate. Oh, that is really nice. That's only getting it wrong about 9 percent of the time. So that's a basically ninety one percent accuracy. That's pretty gorgeous.

So remember, we trained additionally trained this learner. This learned variable. We can actually go back up and see how we do. So learn has been adjusted and we can actually use it right in here. So this was our bike detection section where if we go here and play, so learn is here, we're going to predict images. But now this is the new version of learn, right?

Because we have trained it additionally on those larger images. And if I play here. Let's see what happens.

All right. So still getting, still seeing, let's see, well, actually, is their bike in 038?

There is. So actually really, really smartly picking up this bike right here as it leaves the frame. That's pretty impressive. Let's see what 0021 is. And there's definitely bikes in that frame. So, yes, it looks like it's getting better. Of course, we don't know which ones it's missing, but the ones that it's identifying seem to be much better. And we have a smaller list to look at here as well. So that's probably more useful to us. Interestingly, we can go back up into even this area and see what we can run this interpretations again.

So if we run this cell and.

And then here we can plot the top losses on the heat map and see how it's doing. And you can actually see this is super fascinating. You can see where it's highlighted the bikes. It really can spot the bikes. Although here it zoomed in on a bike, although it guessed no bike. I think it was distracted by this van here. But here it's definitely spotting a bike. It knows it's a bike. And look at you can see where the hotspots are. These are the indicators to the model that this is a bike. I think that's super fascinating and much more specific than what we saw before.

Up next, how do you use machine learning for text documents?