Searching videos with fast.ai (part 2)

Let's make the model and then use it on the whole video.

OK, so we have our bike data in place. We've got that sorted into bikes and no bikes. Here's bike, there's no bike.

And we're going to train our model. Remember, we're going to use the resnet34 model. We're going to bring that in and combine it with our data that we've loaded into our data bunch. And we're going to use another CNN learner. That's a convolutional noral network. Very useful for image recognition. And so we'll load that all up into this learn variable. You'll see it'll download that resnet34 model because it needs to use that off the Internet. And now we're ready to train. We've got our data, our resident model all loaded into learn. And then we're going to do learn.fit_one.cycle.

We're actually going to do six cycles and that's six cycles of training where it's running through those training and validation sets, trying to learn which images, which set of pixels constitute bikes and which ones do not. This is going to take a couple of minutes. So I'm just going to speed through the video here and see you in a minute.

Okay. Our training is done. We've gone through six cycles and we now have an error rate of just under 20 percent, that means it's getting it incorrect about 20 percent of the time, maybe a fifth of the time. That's not great. It's pretty good, though. I mean, why not? It's only making a mistake. One out of five times. So let's keep going. We're going to learn how to make that even better shortly. So let's see how we're doing. If we run this cell here and then the following one. What it allows us to do is to see where it's making mistakes and actually get a little bit of insight into how the computer is trying to identify bikes.

OK, so here's a bunch of scenes and you can see a little bit it's trying. It's definitely trying to discern the bikes here. It's got its little zoomed in on the on the van a little bit more. Interesting here. It thought that this kiosk was a bike. I'm not quite sure why. Maybe it saw something in the reflection. Here it's looking over here. I thought maybe this was a bike. Here it totally missed this bike. Again, this is an upside down image. Some of them are upside down. These are the worst ones, right? These are the mistakes it made. Here it you can see it's getting to the point where there is no bike. It predicted no bike and there was no bike. That's correct. It just wasn't as confident. So it's really only making mistakes on these four for some reason, maybe because these are upside down and sideways. That might be why. OK. Well, let's keep going. We can see the situations where it was confused and what the balance is. So here's the predicted on this side. Here's the actual. So most of the time when there was a bike, it figured that out. Here there was a case where there was a bike and it missed it. Thought it was it was no bike.

But this is pretty well balanced. Again we could do better and we will in a little bit. Let's try a bike image that it has never seen before. This is just a separate image that I took. There's a picture of somebody on a bike so I can assign that to this IMG image variable. This is where we run our prediction learn.predict on the image and we get some variables as a result of that. One of them is predicted class.

And yes, it predicted there was a bike in here, guessed there was a bike in here. So that's perfect. If we do this in Python, we can say turn that into a string. So a string is just some

letters. That's pretty cool. This outputs a variable here is actually it's confidence that that there was a bike there. And actually you can see. I don't know if you can tell but that, this is actually. It's one hundred percent confident there's a bike in this image, which is kind of cruel. All right. Good on you, computer. You are correct.

Alright. So the next step is that we want to search our video. So in order to do that, we're going to run this command. This uses a special library called F F MPEG, which will actually slice the movie that we have into a bunch of images. So let's go ahead and do that.

If you want details about how this is being done, just take a look in the text here. But the result is that we're going to get a whole bunch of images, one every second of this movie.

And let's take a look.

Here they are. So all of these images in here are all from one second in all the way to forty two seconds in. Alright. So those are images we're going to look for bikes in those images.

I can assign that list to a variable called file_list using this fine command here. So I'm going to do that. And if we take a look. Now, we can see we have a list in a nice computery format. It's called actually a list in Python. And then we can run a loop. So we can say for every file in this file list, open the image file, do a prediction on it, on that image, and then. And then if the for if it equals if the prediction is bike and the confidence is over eighty five percent. That's what I'm saying here. Then print out a little bit of information about what file it was seen in and what the confidence was. So let's go ahead and do that.

Alright.

So this is maybe helpful. It's very confident and you can see it's very confident that there are bikes in all of these frames. Here let's see in myframe0025. Let's see if there is a bike in there. There is indeed. There's a bike right there. It is correct.

What about this one? I'm suspicious of this one. Myframe0039. I'm going to just try that one. I don't see a bike in that picture. So it's not perfect, right? It's. It definitely is getting it definitely is pretty good at finding bikes, but it's also making some mistakes.

This might be just as at this level, this might be good enough to help you sort through a video, but it might not be. There's a couple of the things we can do to make it better. One would just be to have more training images of bikes. Right. We're only looking at about 110 images of bikes to train, so we could probably do better. And we did see that it was making some pretty obvious mistakes. But there's some things that we can do to make it even smarter. And we'll go through that in the next video.