**Photo OCR**

# Problem Description and Pipeline

Ideas of how to apply machine learning to computer vision problems, and second is the idea of artificial data synthesis.

Photo OCR stands for Photo Optical Character Recognition.

And one of the things that has interested many developers is how to get our computers to understand the content of these pictures a little bit better. The photo OCR problem focuses on how to get computers to read the text to the purest in images that we take.

Steps: -

First we can go through the image and find the regions where there's text and image.

Second, given the rectangle around that text region, we can then do character segmentation.

And finally, having segmented out into individual characters, we can then run a classifier.

A system like this is what we call a **machine learning pipeline**.

We have an image, which then fed to the text detection system text regions, we then segment out the characters--the individual characters in the text--and then finally we recognize the individual characters.

If you're designing a machine learning system one of the most important decisions will often be what exactly is the pipeline that you want to put together.

# Sliding Windows

Sliding Windows Classifier

Text detection is an unusual problem in computer vision.

Because depending on the length of the text you're trying to find,

these rectangles that you're trying to find can have different aspect ratio.

# Getting Lots of Data and Artificial Data

One of the most reliable ways to get a high performance machine learning system is to take a low bias learning algorithm and to train it on a massive training set.

The idea of artificial data synthesis comprises of two variations, main the first is if we are essentially creating data from, creating new data from scratch. And the second is if we already have a small label training set and we somehow have amplify that training set or use a small training set to turn that into a larger training set.

The distortions you introduce should be representative the source of noises, or distortions, that you might see in the test set. In contrast, usually it does not help perhaps you actually a meaning as noise to your data. But if you're trying to decide what sorts of distortions to add, you know, do think about what

other meaningful distortions you might add that will cause you to generate additional training examples that are at least somewhat representative.

As always, before expending a lot of effort, you know, figuring out how to create artificial training of the sorts of images you expect to see in your test sets examples, it's often a good practice is to make sure that you really have a low biased crossfire, and having a lot more training data will be of help. And standard way to do this is to plot the learning curves, and make sure that you only have a low as well, high variance classifier.

Or if you don't have a low bias classifier, you know, one other thing that's worth trying is to keep increasing the number of features that your classifier has, increasing the number of hidden units in your network, saying, until you actually have a low bias classifier, and only then, should you put the effort into creating a large, artificial training set, so what you really want to avoid is to, you know, spend a whole week or spend a few months figuring out how to get a great artificially synthesized data set. Only to realize afterward, that, you know, your learning algorithm, performance doesn't improve that much, even when you're given a huge training set.

# Ceiling Analysis: What Part of the Pipeline to Work on Next

What parts of the pipeline might be the best use of your time to work on?

As in the development process for other machine learning systems as well,

in order to make decisions on what to do for developing the system is going to be

very helpful to have a single rolled number evaluation metric for this learning system.

Ground Truth Labels.