Week 11 - Lecture 1

Photo OCR

Problem Description and Pipeline

Photo OCR stands for Photo Ocular Character Recognition. The Photo OCR problem focus on how to make computers get text content from images.

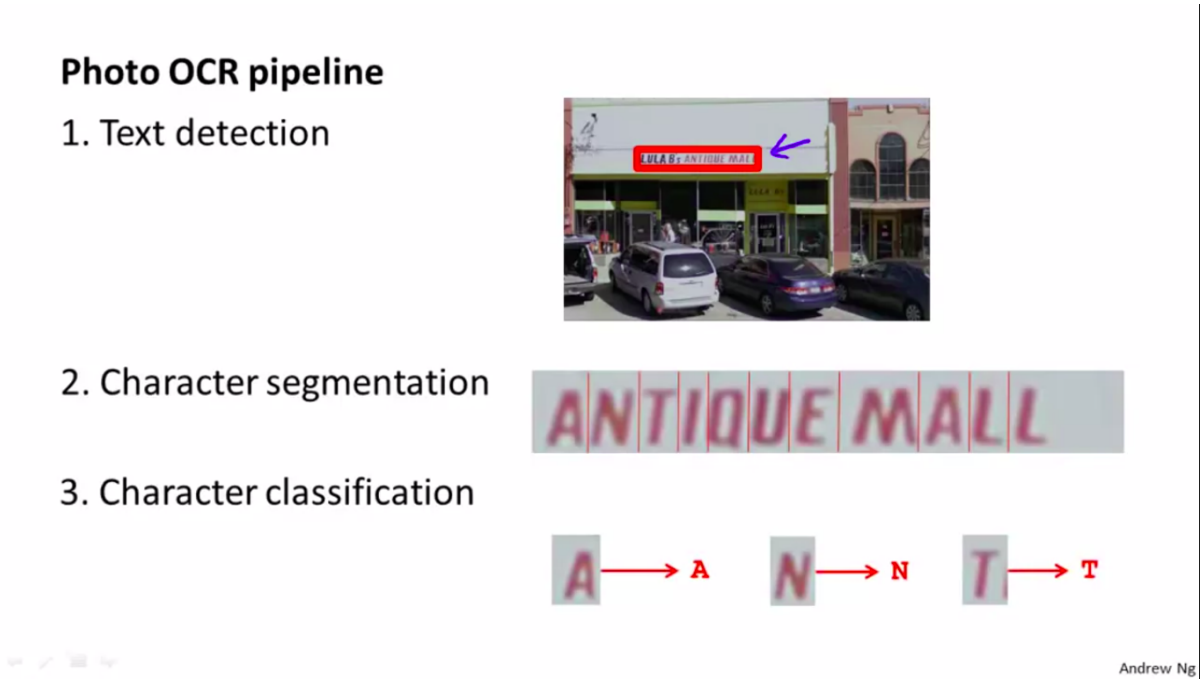
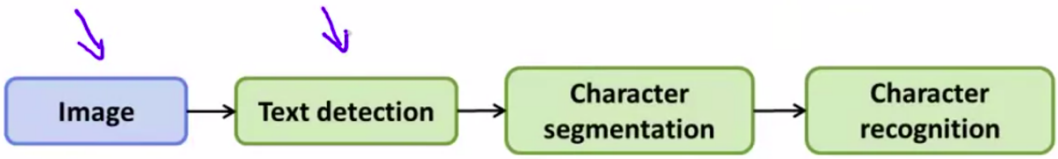


Photo OCR pipeline



Sliding Windows

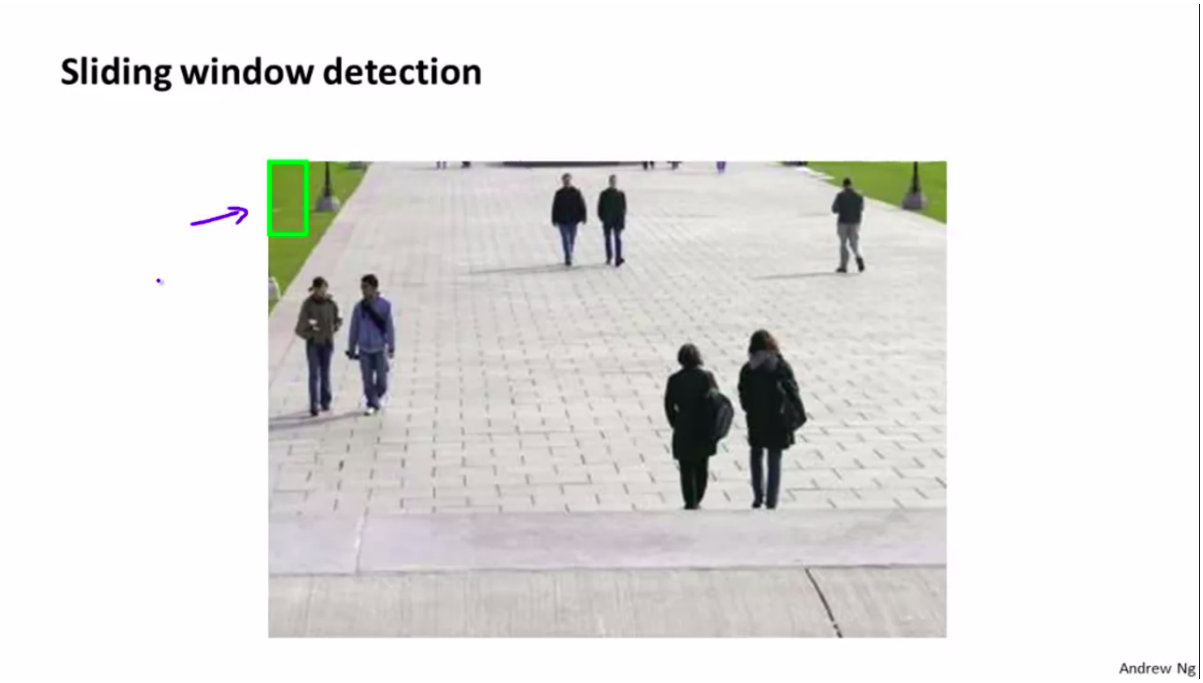
Supervised learning for pedestrian detection

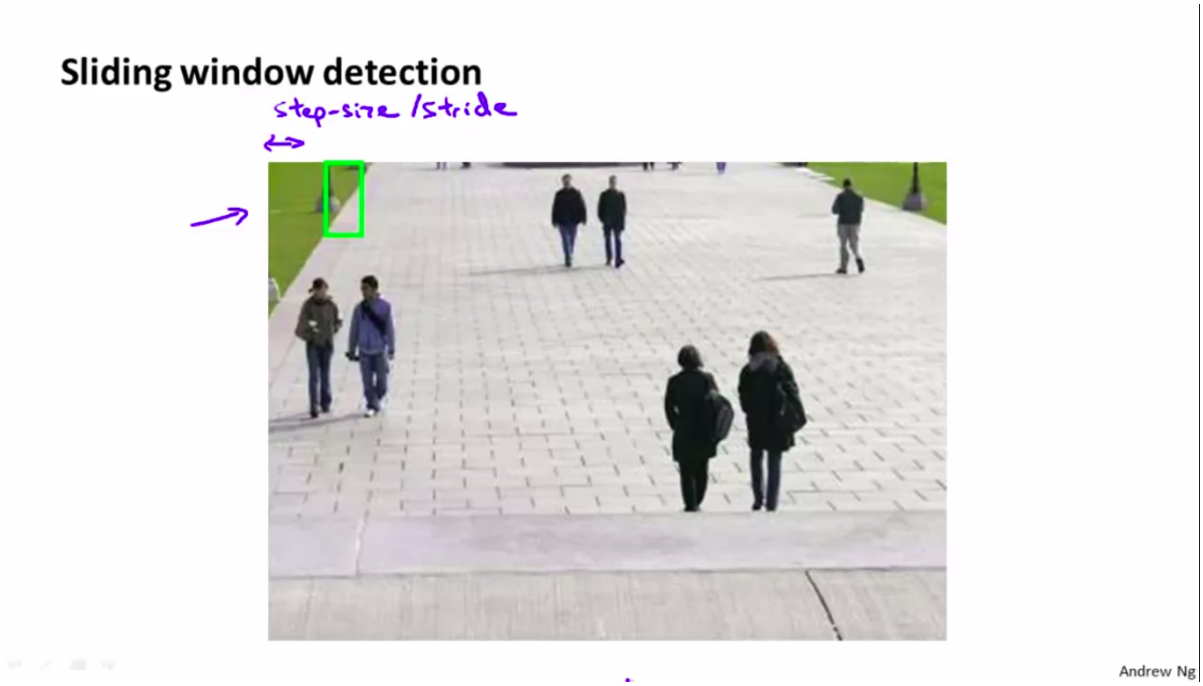
x = pixels in 82x36 image patches

Positive examples $(y = 1)$

Negative examples $(y = 0)$

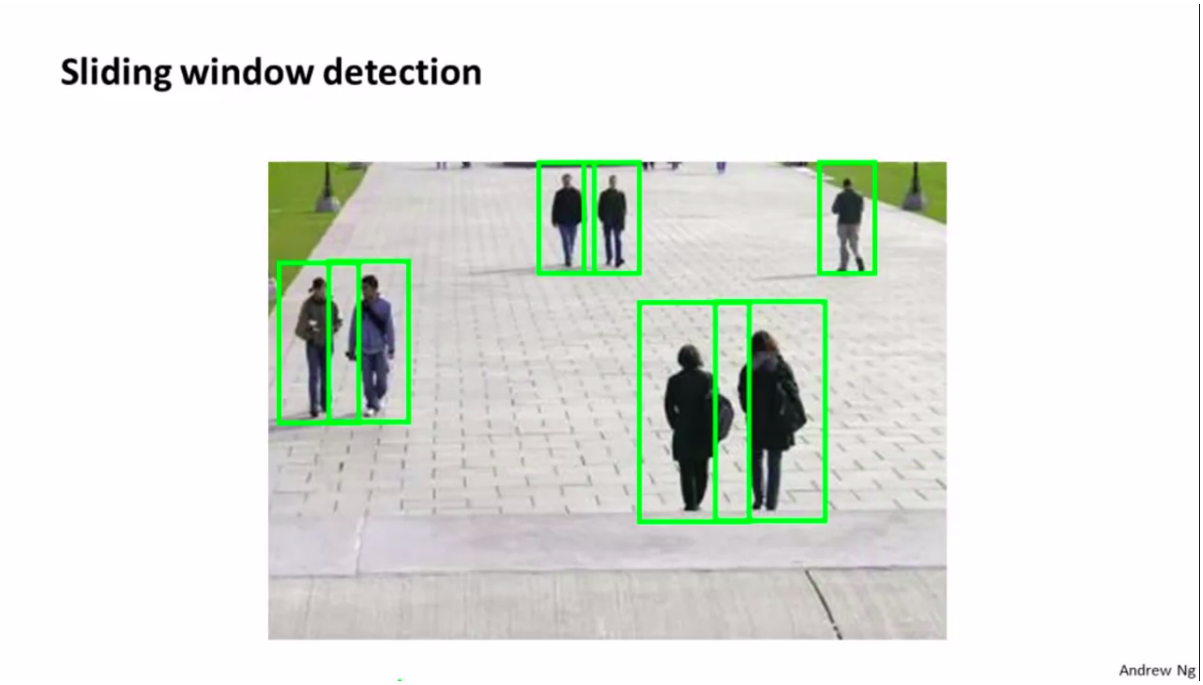
Andrew Ng



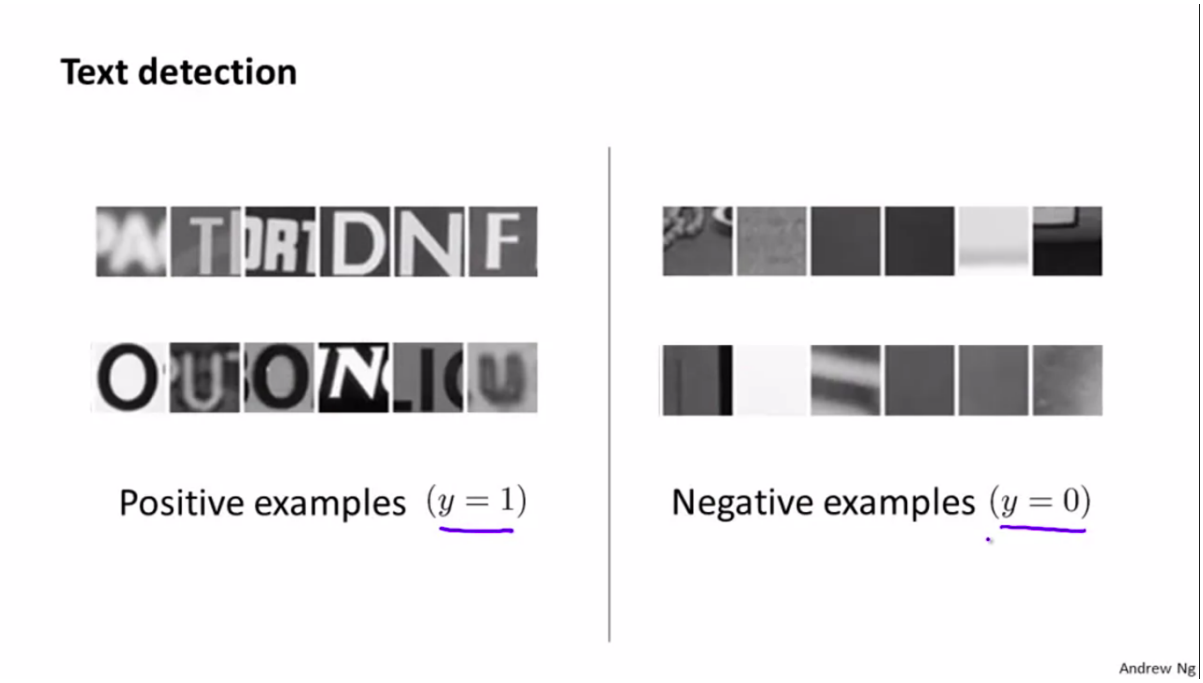


If the window (rectangle) is too small, we can try to increase it and then resize each cropped image to the original rectangle size (to keep the same dimmensions we used to train the model).

Hopefully, at the end of this process, you algorithm will have correctly identified where are the pedestrians on the image.

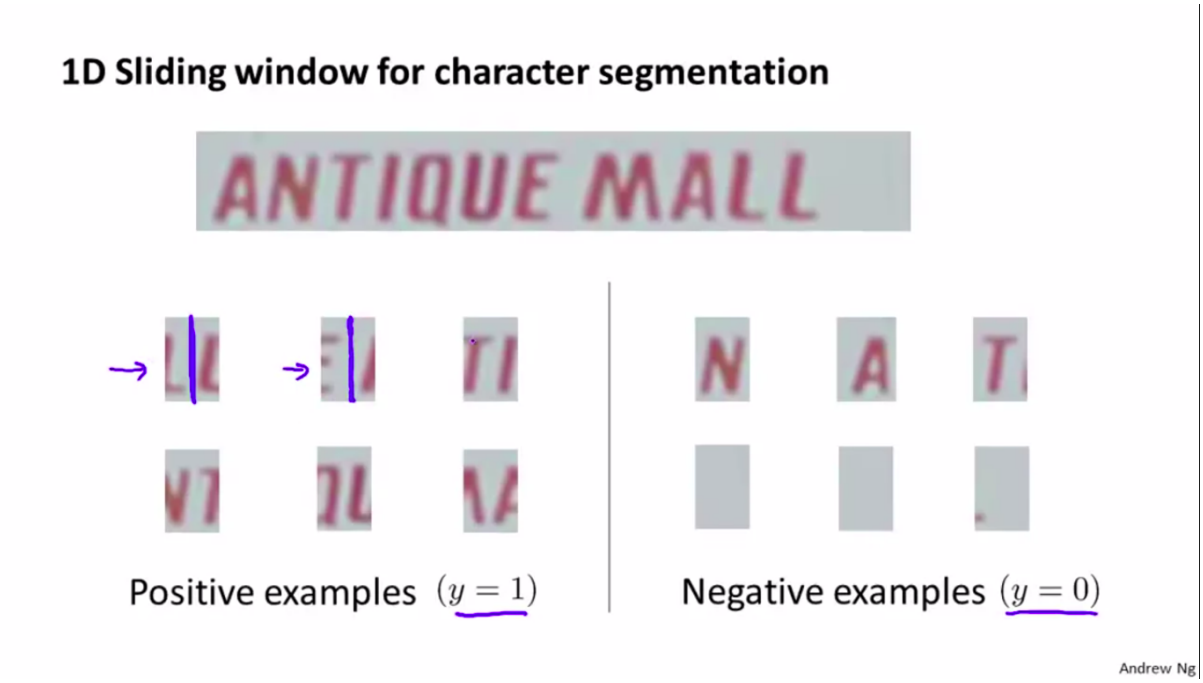


The text detection problem can be posed similarly to the pedestrian detection problem.



If we use the sliding window technique, we'll get as result that the white regions show where we have texts in our image.

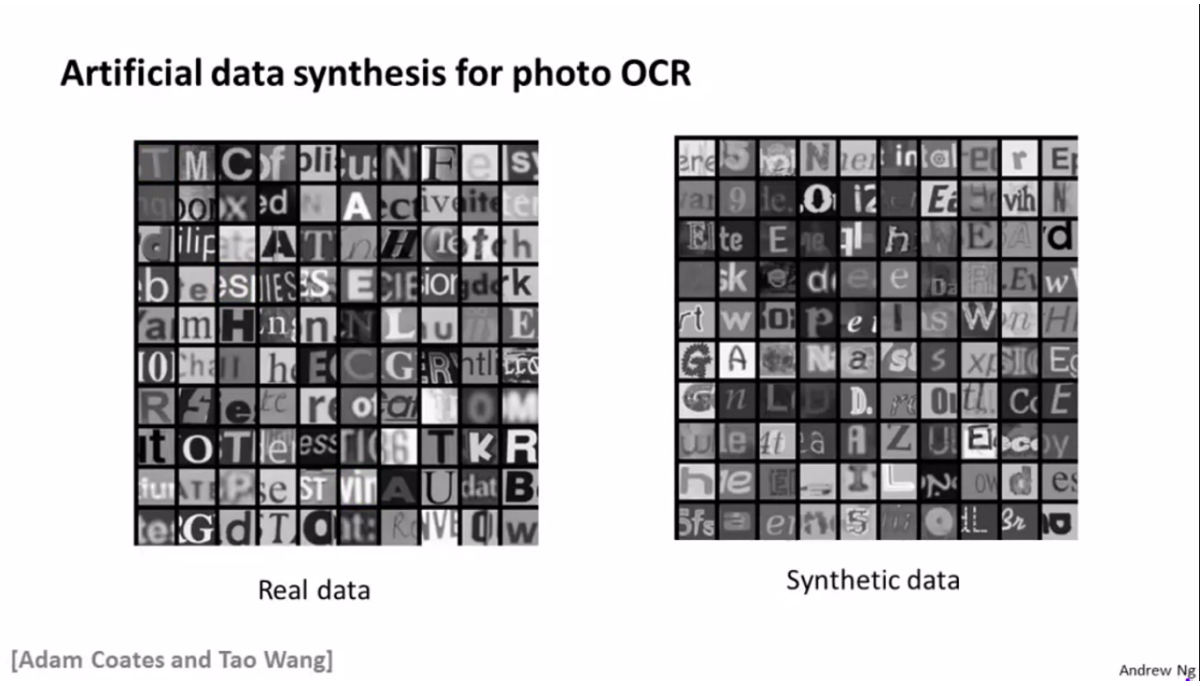
After detecting the regions were there is text on the image, te other step is to identify text segmentations.



At this point, the next step is to use a supervised learning algorithm to classify each of the characters identified in the image.

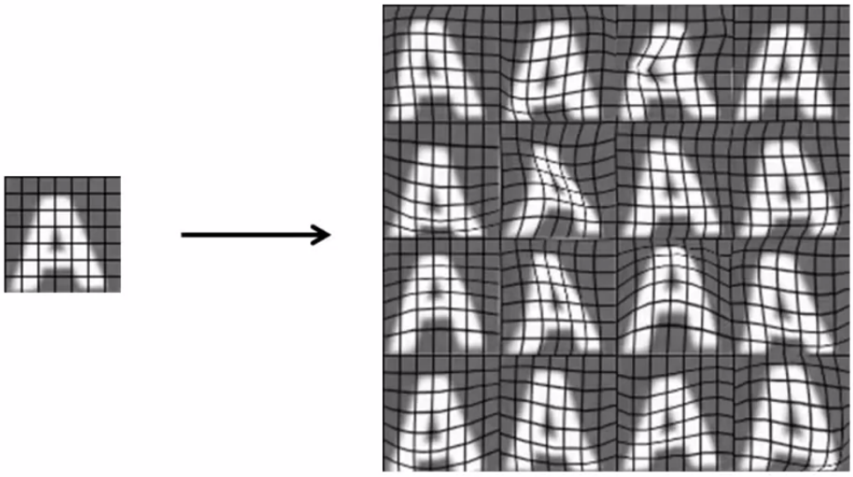
Getting Lots of Data and Artificial Data

Taking the image character classification (last step of Photo OCR problem) problem, suppose that we need to get more data in order to train our model. One way to to this is to get different font libraries on the internet and plot lots of characters on random backgrounds. The result is exemplified on the next picture.



This way, we can construct a synthetic training dataset. Another way to do this is to get a real image (training example) and introduce distortions (like exemplified next).

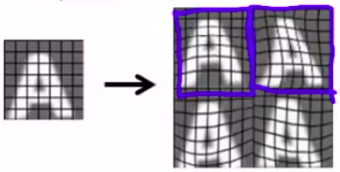
Synthesizing data by introducing distortions



[Adam Coates and Tao Wang]

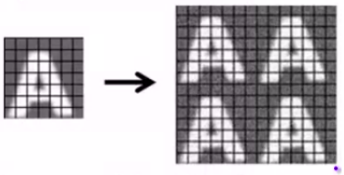
Synthesizing data by introducing distortions

→ Distortion introduced should be representation of the type of noise/distortions in the test set.



→ Audio:
Background noise,
bad cellphone connection

→ Usually does not help to add purely random/meaningless noise to your data.



x_i = intensity (brightness) of pixel i
$x_i \leftarrow x_i +$ random noise

[Adam Coates and Tao Wang]

Discussion on getting more data

1. Make sure you have a low bias classifier before expending the effort. (Plot learning curves). E.g. keep increasing the number of features/number of hidden units in neural network until you have a low bias classifier.
2. "How much work would it be to get 10x as much data as we currently have?"
   - Artificial data synthesis
   - Collect/label it yourself

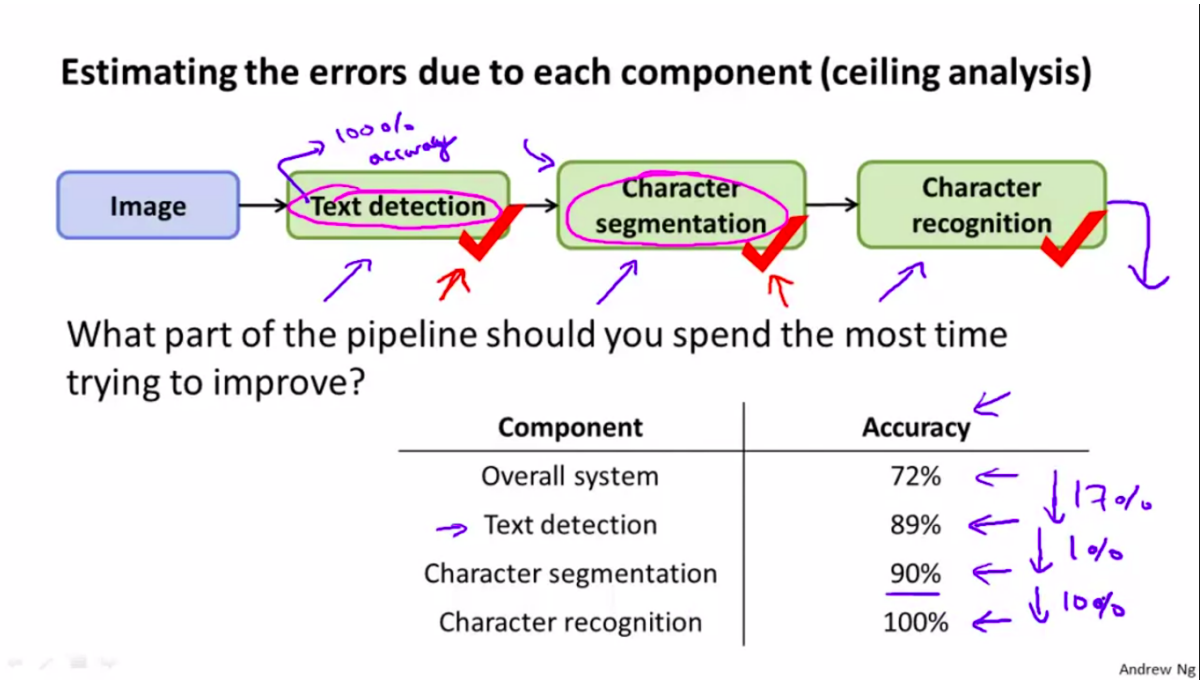
→ #hours?

$m = 1,000$
→ 10 secs/example
$m = 10,000$ ←

Another way of collecting more data would be by "Crowd Source" online, like Amazon Mechanical Turk service.

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

In order to measure how the overall system performance reacts to each of its separate parts performance, we simulate a perfer performance in each stage (provide 100% accuracy by manually preparing the results of each stage) and measure how the overall performance responds to this.



In other words, we evaluate the marginal overall performance gain of expending extra time improving each of the system's parts and focus our efforts on the ones that has higher marginal gains.

References

[1] Machine Learning - Stanford University (https://www.coursera.org/learn/machine-learning)