Week : 11

Photo OCR (optical character recognition) pipeline :

In machine learning, the problems or tasks, which can be divided into various steps are called as pipeline. Photo OCR is a technique by which a machine can read the texts remaining in a photo. To do so, we can divide our algorithm into three parts : text detection, character segmentation and character classification.

Photo OCR pipeline

→ 1. Text detection

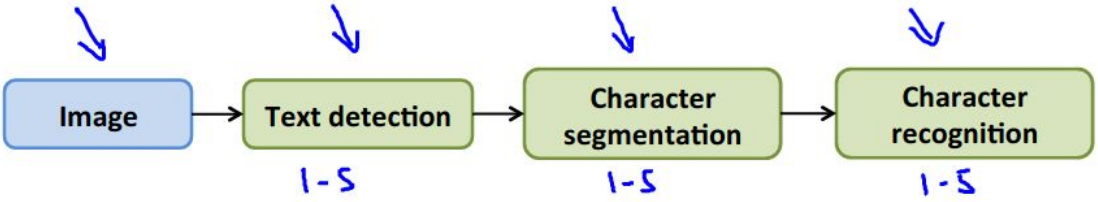
→ 2. Character segmentation

→ 3. Character classification

Cleaning → Cleaning

Andrew Ng

Photo OCR pipeline

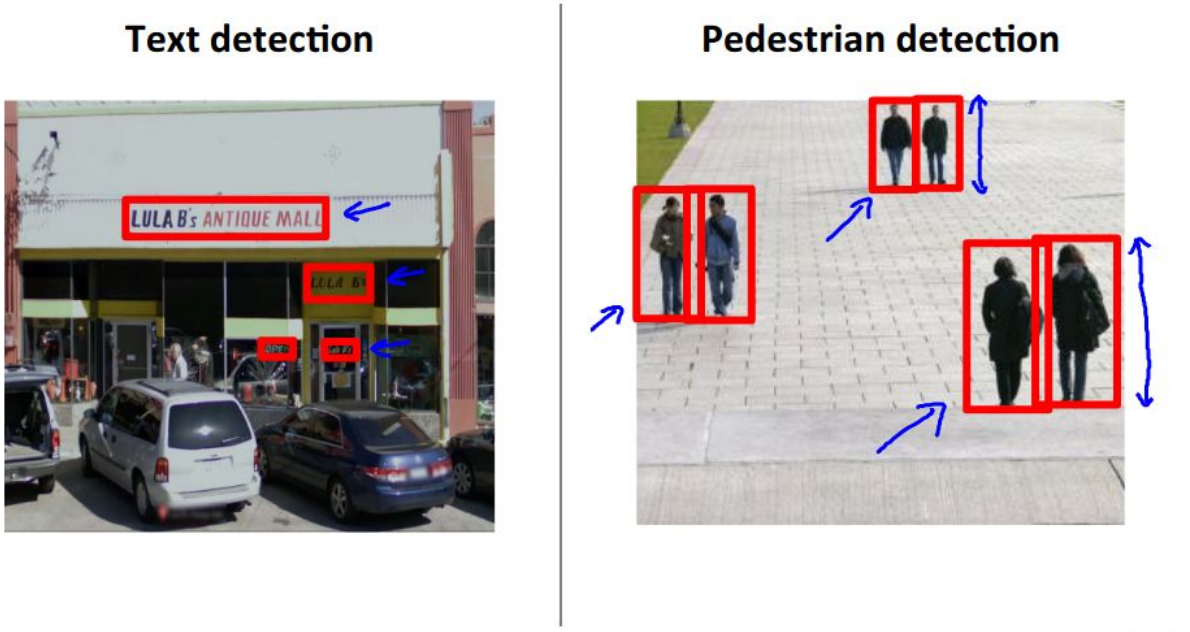


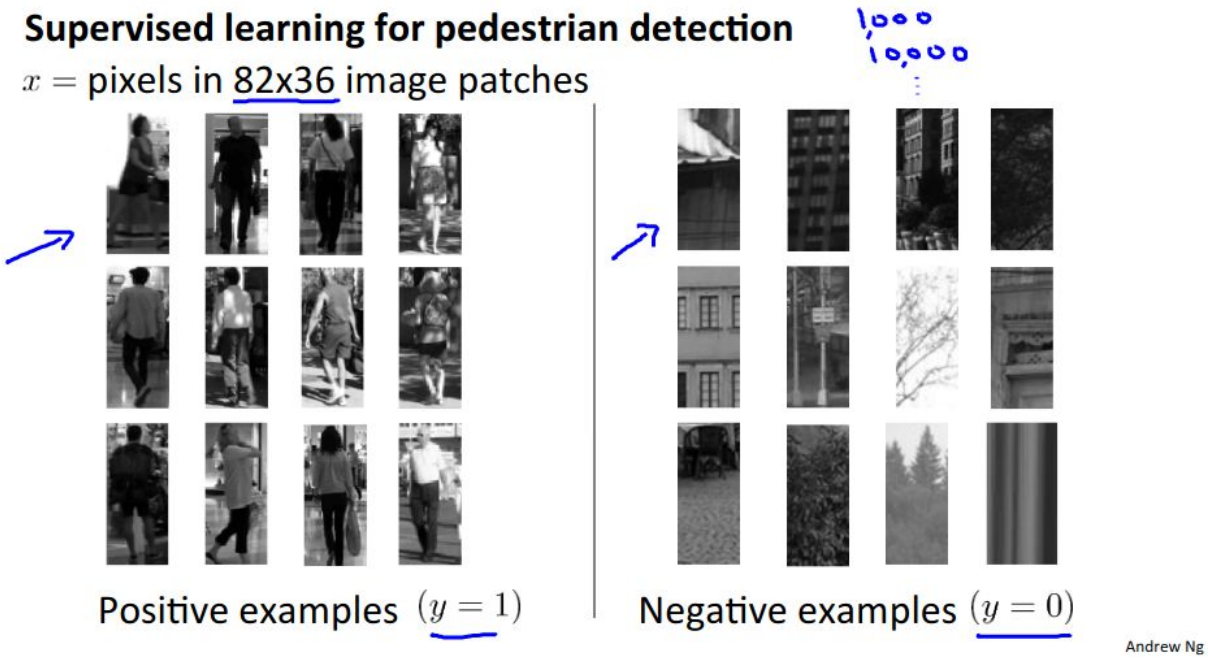
Sliding window :

Sliding window is an object detection technique in machine learning. In this technique, at first by neural network or logistic regression we train our algorithm and then we make

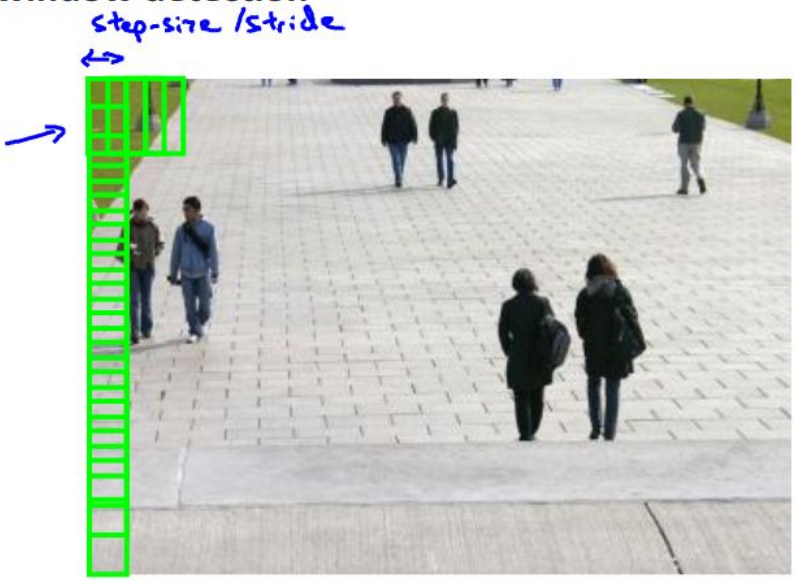
a window which we will slide in x and y direction. By doing this, we can detect our desired object.

We use this in OCR to detect text areas. If you can slide the window by 1 pixel each time, it will give the best result. But 3-4 pixels are mostly used.

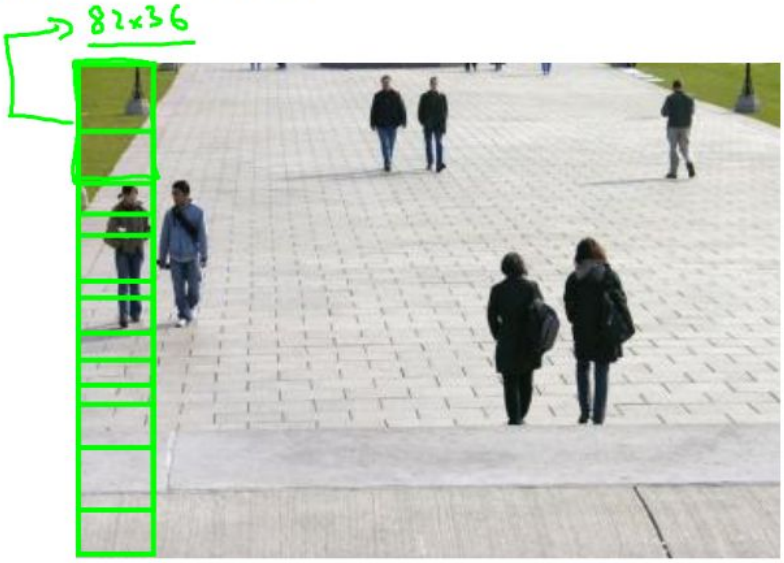




Sliding window detection



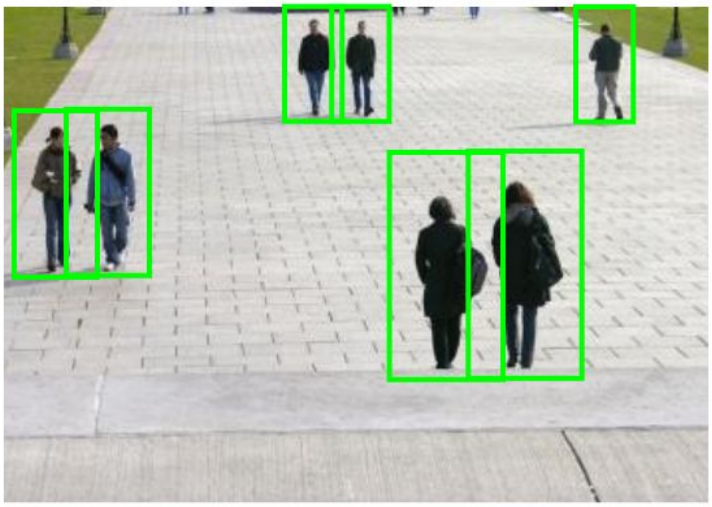
Sliding window detection



Sliding window detection

Andrew Ng

Sliding window detection

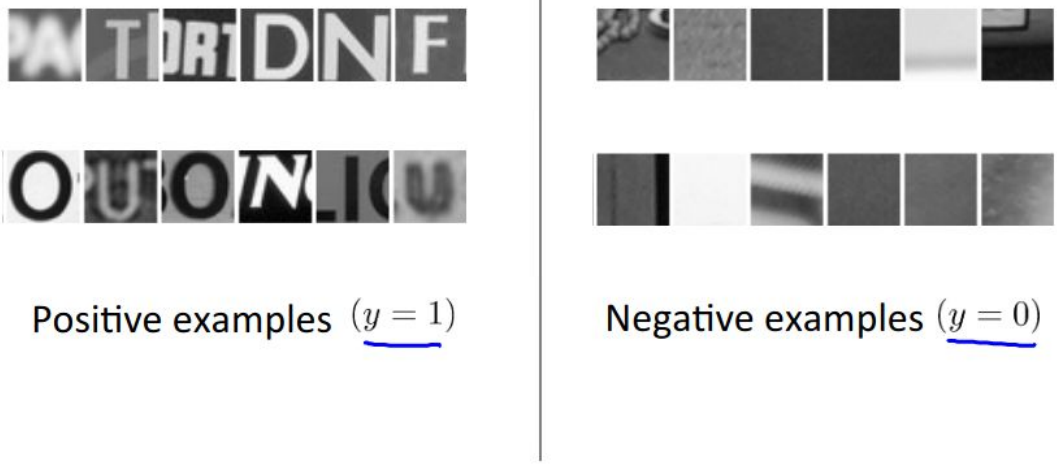


Andrew Ng

Text detection

Andrew Ng

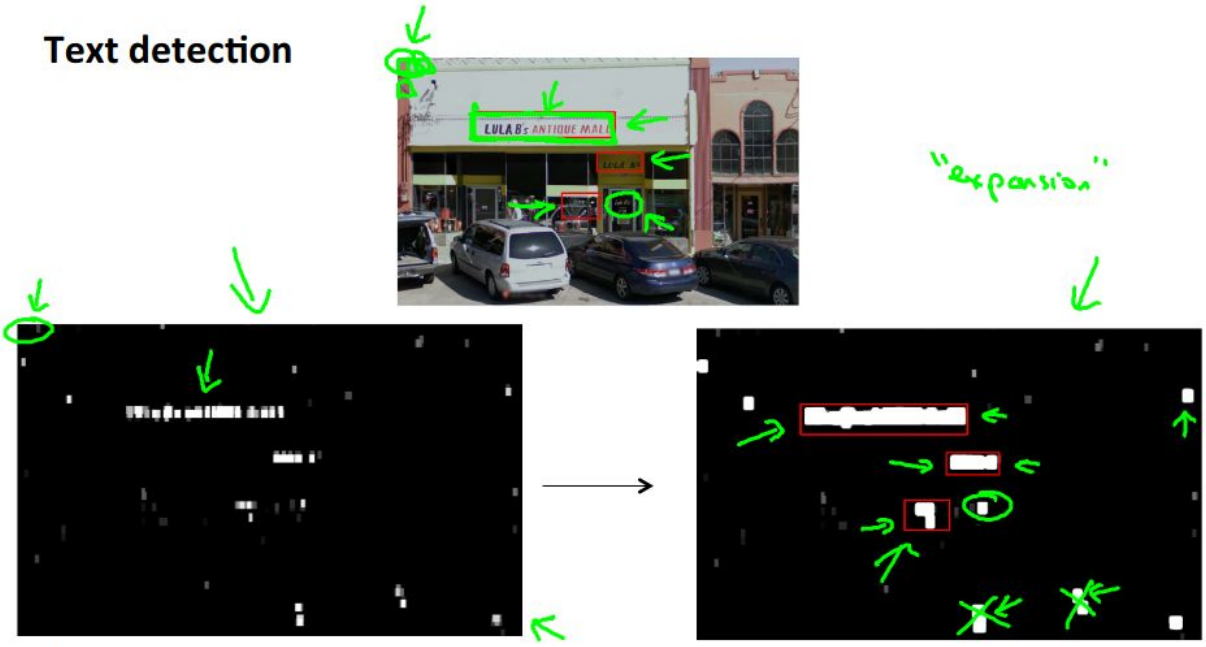
Text detection



Andrew Ng

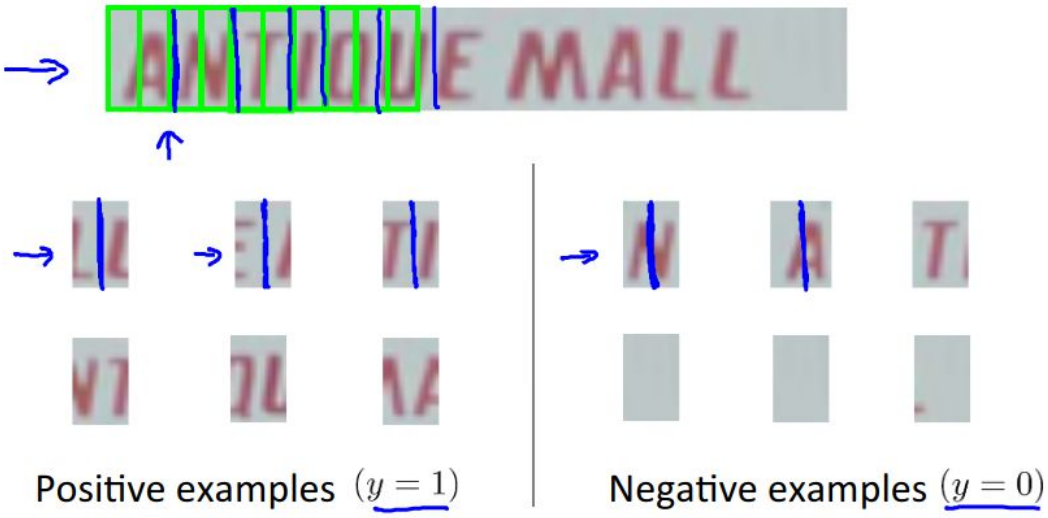
After directing the text areas, we will mark them as white color in black background. It is not mandatory but it will help. Then we will extend the width of the white color by 1-2 pixels. We will do so, so that we can distinguish between text true areas and false text areas. Normally text areas are larger in width than height.

Text detection



[David Wu]

1D Sliding window for character segmentation



Positive examples $(y = 1)$

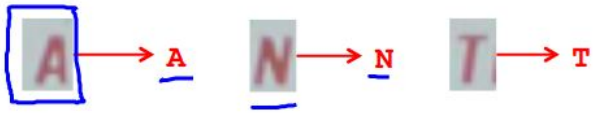
Negative examples $(y = 0)$

Photo OCR pipeline

→ 1. Text detection

→ 2. Character segmentation

→ 3. Character classification

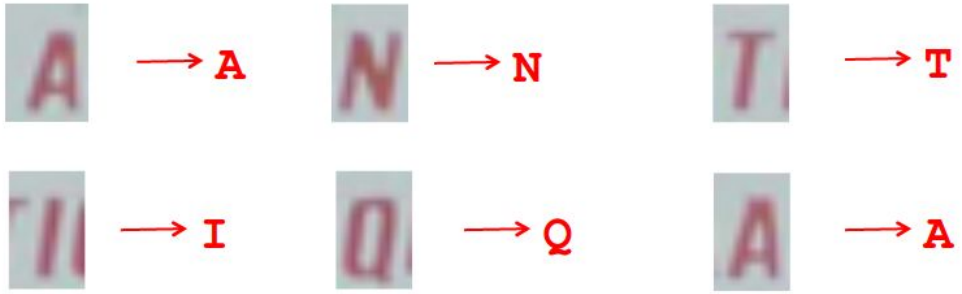


Andrew Ng

Getting lots of data : artificial data synthesis :

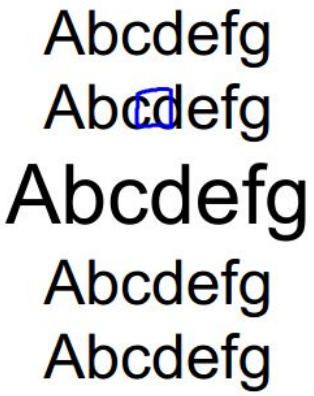
Sometimes it is not much work to do to get more data. If you can synthesys your data with different kinds of noise then from original data, you can get so much more new data. There, by synthesis means modify data by adding some gaussian noise. It also helps us to detect objects from pictures and so on.

Character recognition



Artificial data synthesis for photo OCR

Real data



[Adam Coates and Tao Wang]

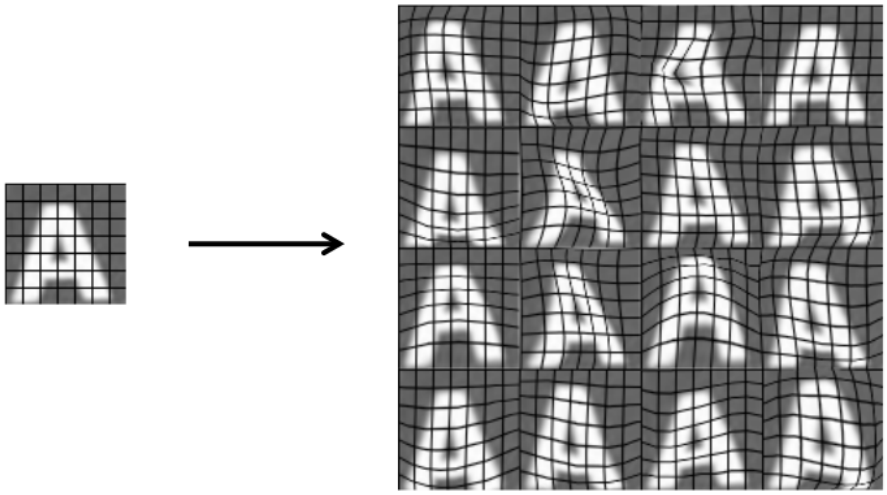
Artificial data synthesis for photo OCR

Real data

Synthetic data

[Adam Coates and Tao Wang]

Synthesizing data by introducing distortions



[Adam Coates and Tao Wang]

Synthesizing data by introducing distortions: Speech recognition

Original audio:

Audio on bad cellphone connection

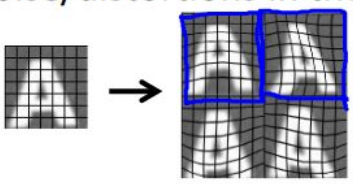
Noisy background: Crowd

Noisy background: Machinery

[www.pdsounds.org]

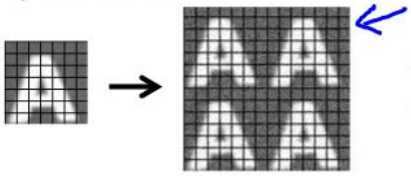
Synthesizing data by introducing distortions

- 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.

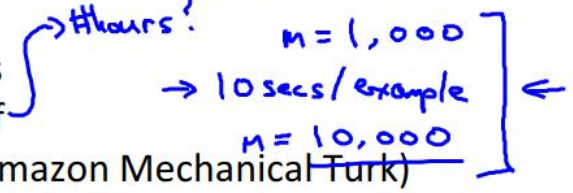


- x_i = intensity (brightness) of pixel i
- $x_i \leftarrow x_i +$ random noise

[Adam Coates and Tao Wang]

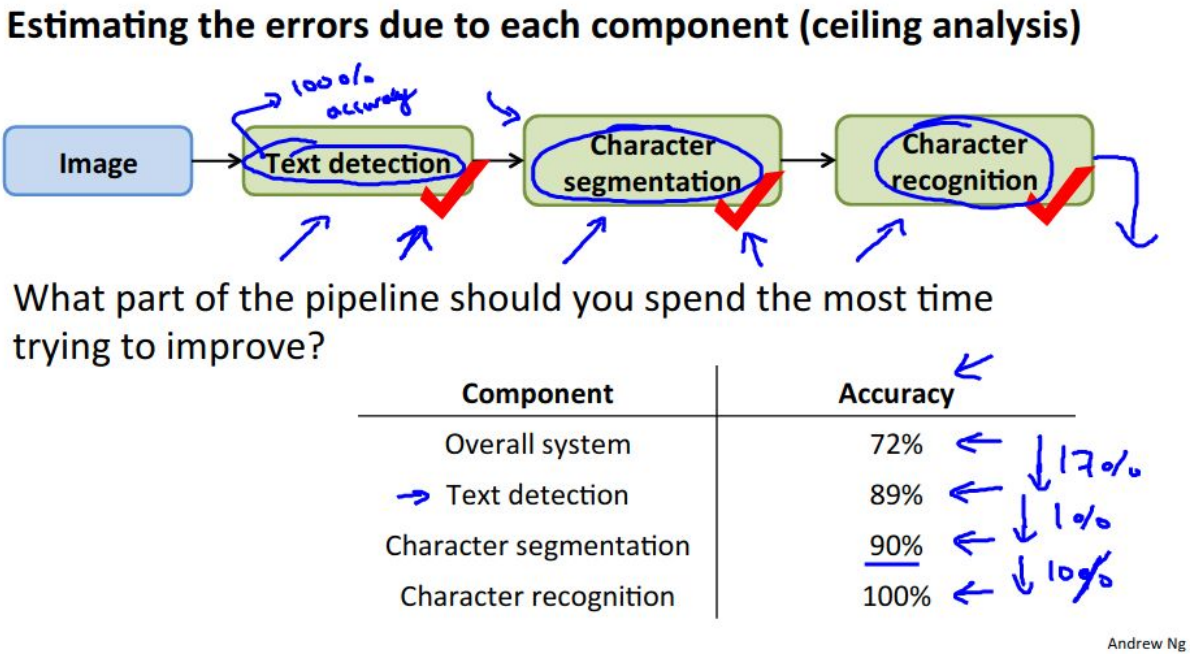
Discussion on getting more data

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
   - "Crowd source" (E.g. Amazon Mechanical Turk)



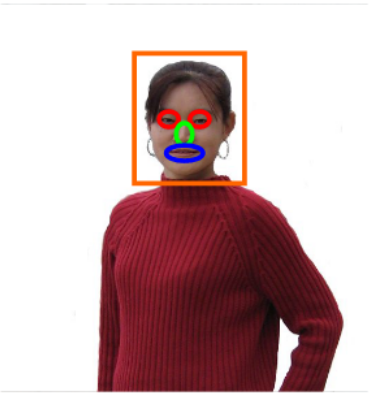
Ceiling analysis : what part of the pipeline to work on next :

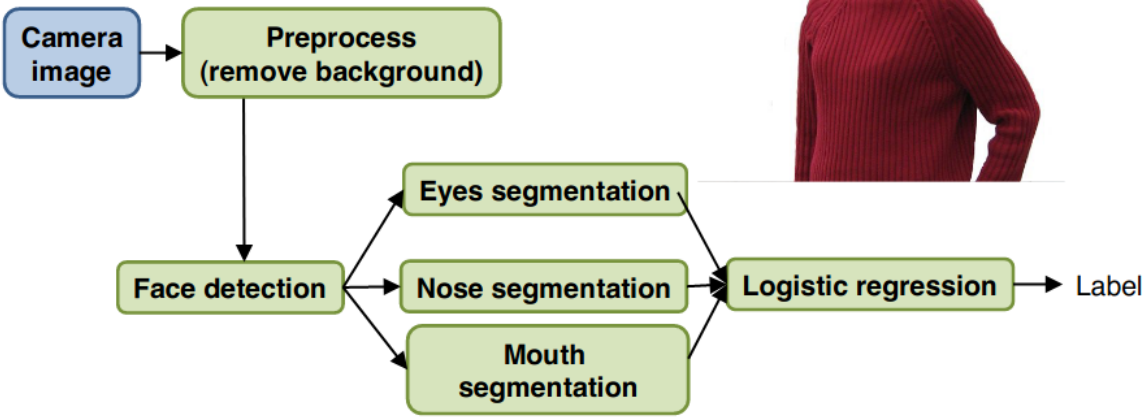
Sometimes when we implement a pipeline on our machine learning problem, we need to make decisions on which part we should spend much time to get better results. In that case ceiling analysis is here to help us. In that technique we improve every part of the pipeline for a small number of training sets and see how much improved our system has. Then we pick the part of the pipeline which contributes most to the improvement of our system. Such as in the photo below, by improving text detection 100% we get 17% better performance and it is the highest. On the contrary, character segmentation contributes just 1% improvement on our system. From this analysis we now can decide that text detection is the hot cake on which part we should spent more time to get better performance from the system.



Another ceiling analysis example

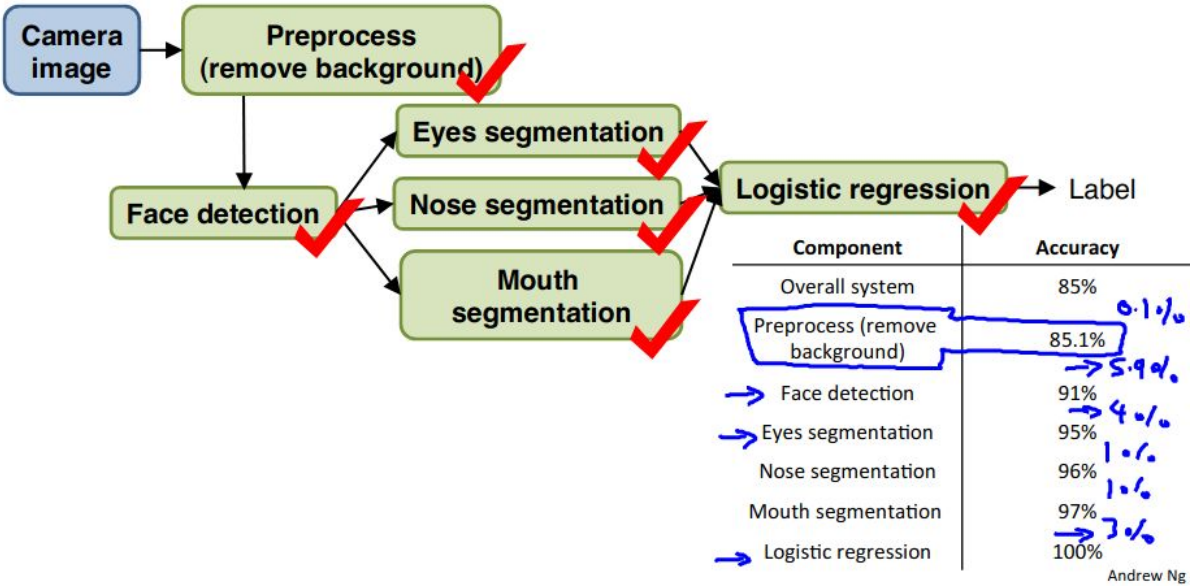
Face recognition from images
(Artificial example)





Andrew Ng

Another ceiling analysis example



Andrew Ng