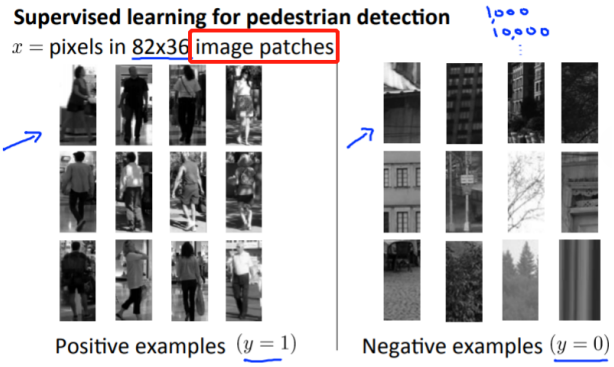
example project: photo OCR

- concept: **pipeline** in machine learning (**modularization**)

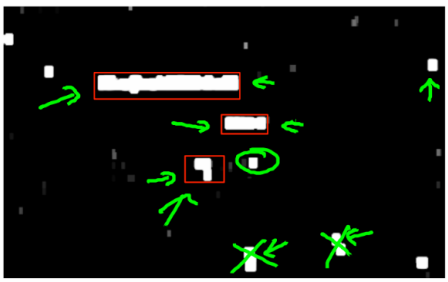
- implement OCR:
  1. using **slide windows** (for text detection)
     1. make image patches



     2. slide through the photo to detect text..(if found👉mark by **bound**)

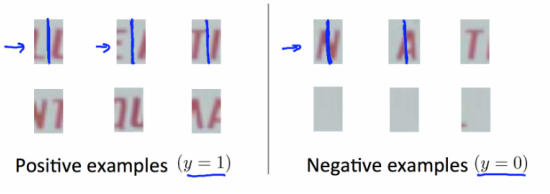
  2. optimize the bounds by **expansion operator.**

     - before



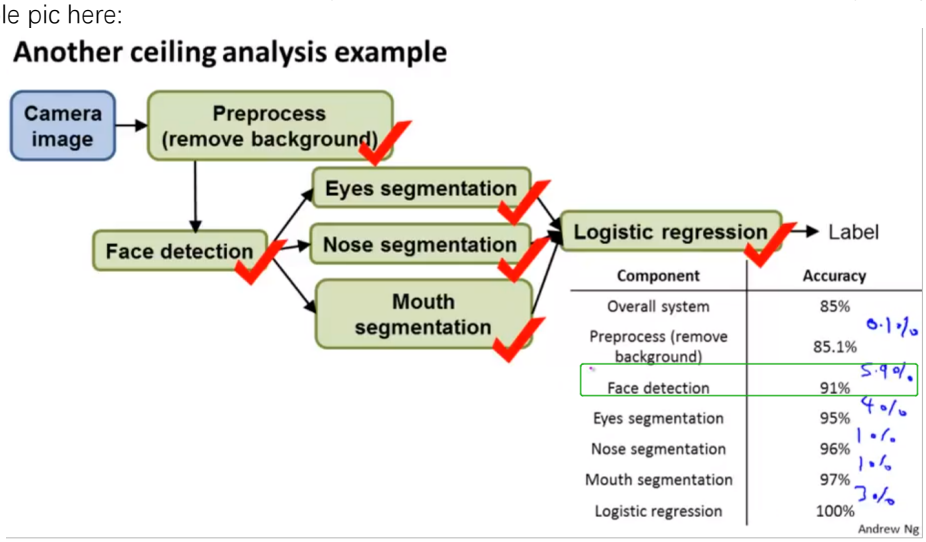
     - after
     - which could realize by Mathematical morphology。
  3. using slide windows (for characters segmentation)

     - define features to judge whether a positive examples or not



Useful skills:

- **ADS** (artificial data synthesis)
  - example : for OCR
    - create synthetic data by **characters from different fonts + random backgrounds.**
    - then, **introducing distortions.**
  - generally step:
    1. **make sure we need more data**. In case that poor performance may caused by high bias, which wouldn't solved by feeding more data.
    2. evaluate **how much work it be** to get a larger size of data.
    3. fetch data!
       1. ADS
       2. collect and label data by ourselves
       3. using crowd source to collect.
- **ceiling analysis**
  - what it is: estimating errors due to each component
  - why use it:**save time** to judge which component should focus on
  - how to use:
    - **use ground-truth to test model accuracy.** (it means that suppose you build up a perfect component which complete its work with 100% accuracy, then we want to see its influence on overall system)
    - example pic here:



    - chart shows that we should focus on improving **face detection** component accuracy, rather than waste time to optimize **pre-process** component's performance.

---

finally

Summary: Main topics

Supervised Learning
- Linear regression, logistic regression, neural networks, SVMs

Unsupervised Learning
- K-means, PCA, Anomaly detection

Special applications/special topics
- Recommender systems, large scale machine learning.

Advice on building a machine learning system
- Bias/variance, regularization; deciding what to work on next: evaluation of learning algorithms, learning curves, error analysis, ceiling analysis.