

## Application Example: Photo OCR

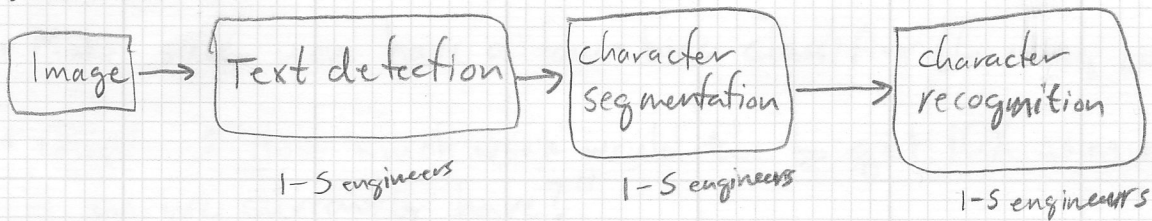
①

- Detect/locate text in an image
- Transcribe text

### Pipeline:

1. detect text
2. character segmentation
3. character classification

### Photo OCR Pipeline



## Sliding Windows

### Text detection (computer vision)

- train a classifier on a test set of small positive and negative examples.
- take patches from the larger image and use classifier to find item in image
- Vary the size of the patch by rescaling them to match the classifier input size.
- draw rectangles around text regions
  - "expansion operator" - similar to a cellular automaton update rule to fill in gaps between characters
  - select regions with appropriate aspect ratio

Segmentation - Use a classifier trained to identify the gaps between characters.

Character Classification -

## Getting Lots of Data - Artificial Data Synthesis

(2)

Use font library to paste characters in a random background.

- Use distortions (blurring, warping) to multiply the number of training examples. Distortions should resemble the target channel. ([www.pdsounds.org](http://www.pdsounds.org) - audio examples) Purely random noise usually does not help unless present in channel.
  - Text: Warping, blurring
  - Audio: Background noise, processing distortion

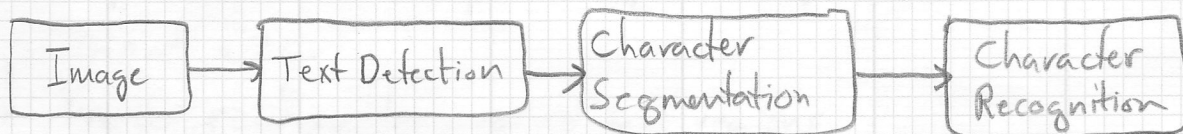
• Make sure you have a low-bias classifier (learning curves)

• Question: "How much work would it be to get 10X as much data as we currently have?"

- time to label one new datum X # desired

Another Method: crowd sourcing (Amazon "Mechanical Turk")

## Ceiling Analysis - how to assign development effort



Starting from the beginning, replace the output of a module with a labeled output set. Record the improvement in the evaluation metric. Repeat for each module. Assign priority to the module which resulted in the largest performance gain when replaced.

Component	Accuracy	Improvement	Priority
Full System	72%	17%	1
Replace Text Detection	89%		
Replace Character Segmentation	90%	1%	3
Replace Character Recognition	100%	10%	2