Application example: Photo OCR

Problem description and pipeline

Machine Learning

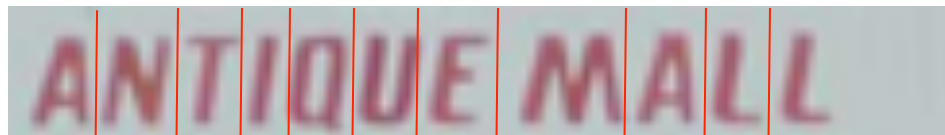# The Photo OCR problem

Photo OCR pipeline

→ 1. Text detection
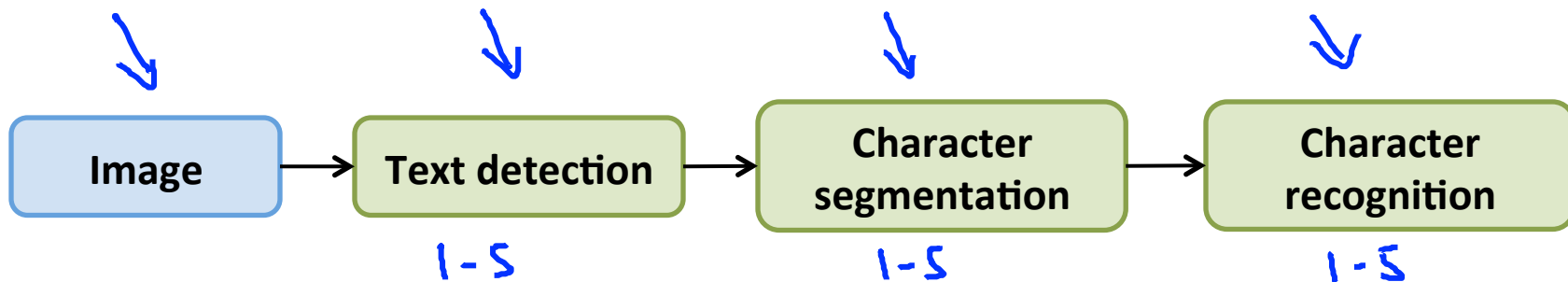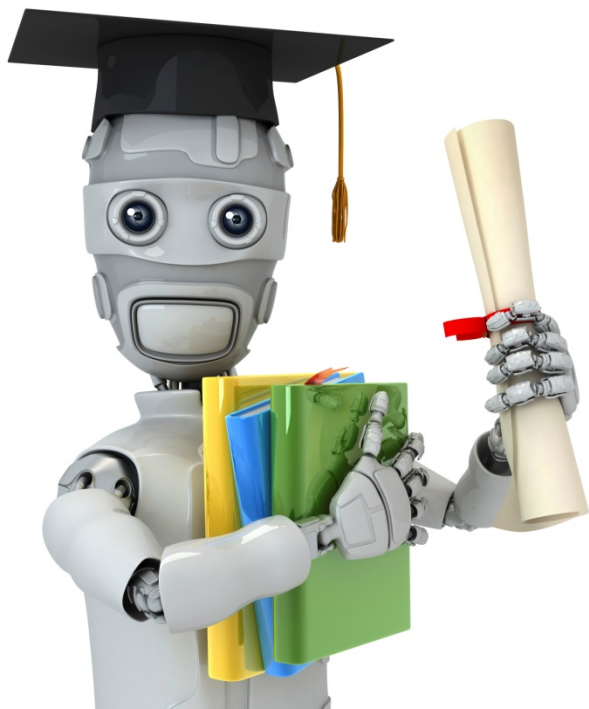

→ 2. Character segmentation

→ 3. Character classification

Cleaning → Cleaning

Andrew Ng

Photo OCR pipeline

Image	→	Text detection	→	Character segmentation	→	Character recognition

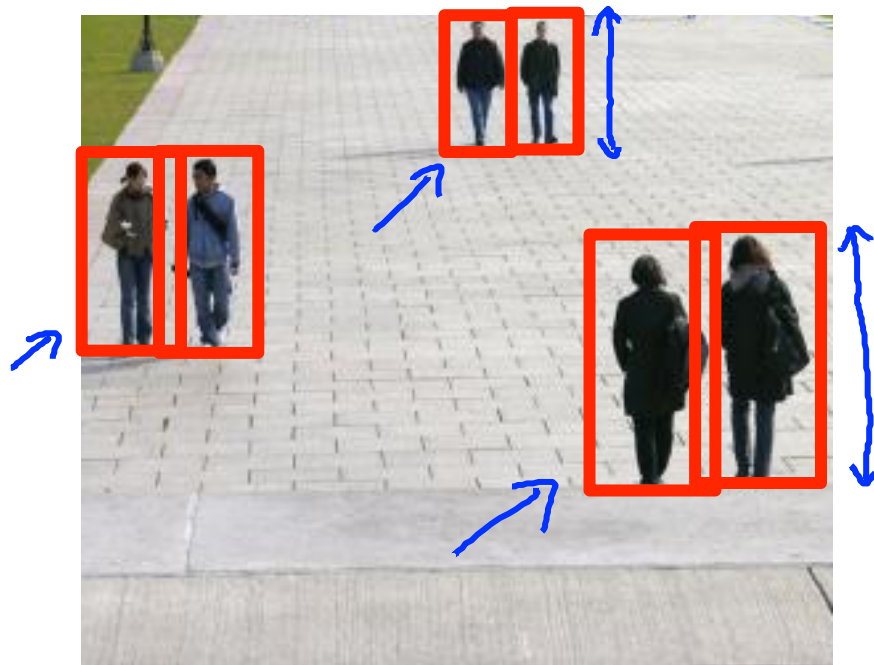Application example: Photo OCR

Sliding windows

Machine Learning

Text detection

Pedestrian detection

Supervised learning for pedestrian detection

1000
10,000
⋮

$x =$ pixels in 82x36 image patches

Positive examples $(y = 1)$

Negative examples $(y = 0)$

Andrew Ng

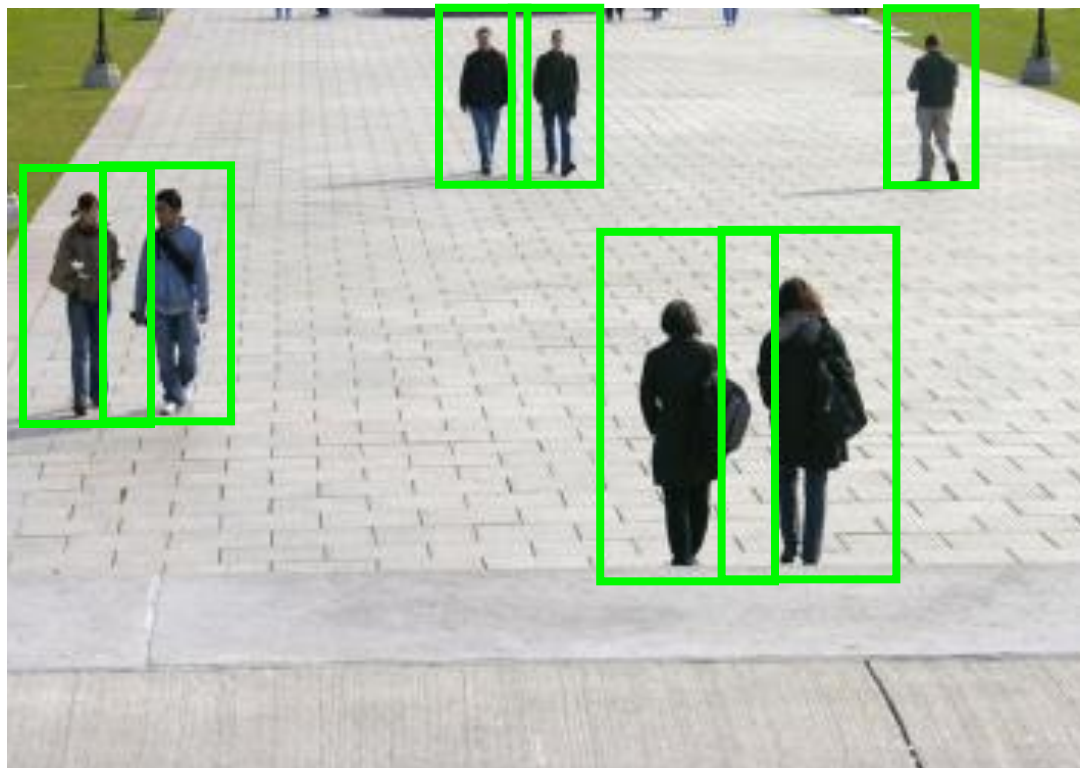# Sliding window detection

step-size /stride

Sliding window detection

82x36

Andrew Ng

Sliding window detection

Sliding window detection
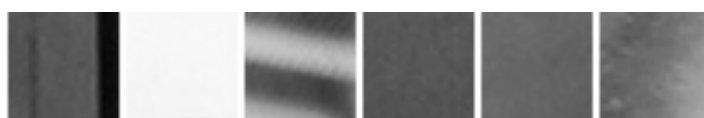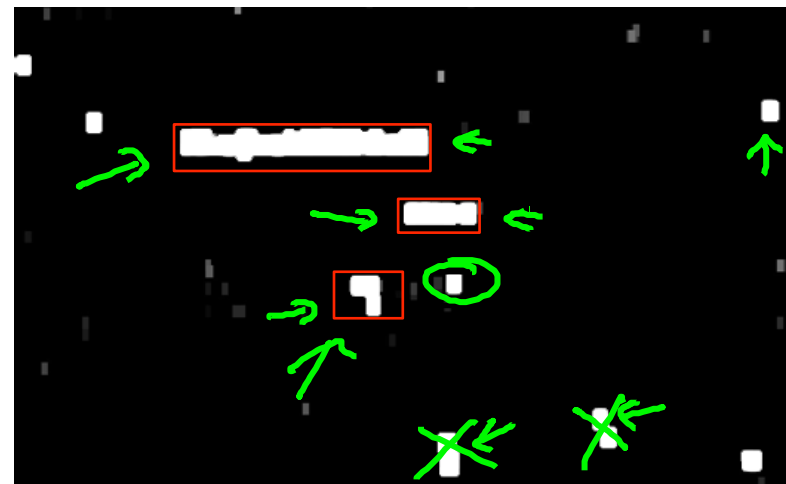
Text detection

Text detection

Positive examples $(y = 1)$

Negative examples $(y = 0)$

Text detection

"expansion"

Andrew Ng
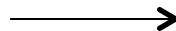
1D Sliding window for character segmentation
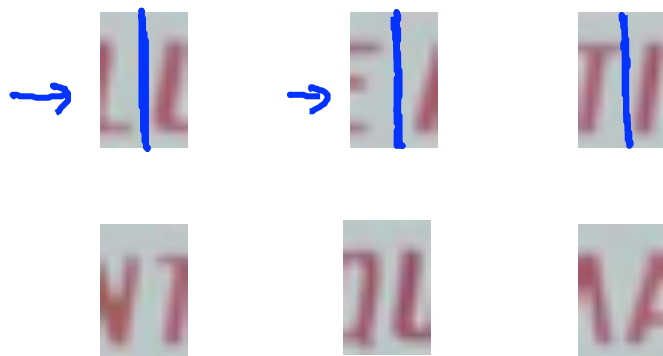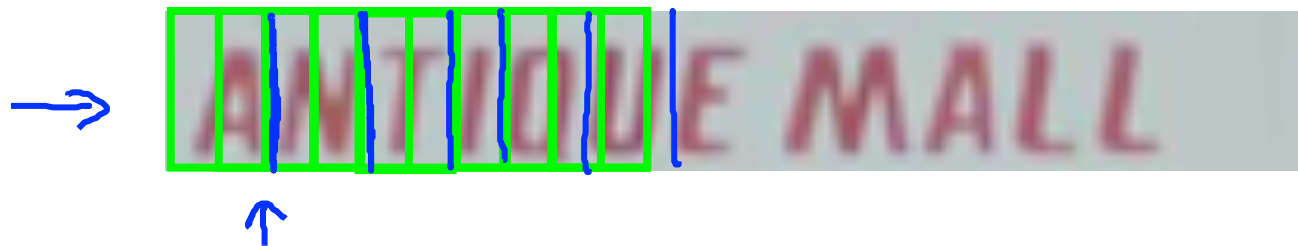
Positive examples $(y = 1)$
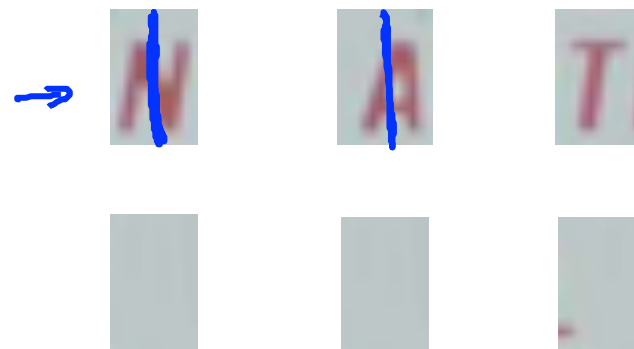
Negative examples $(y = 0)$

Photo OCR pipeline

→ 1. Text detection

→ 2. Character segmentation
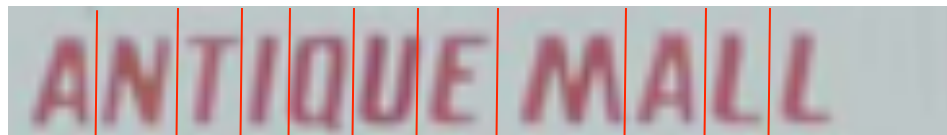
→ 3. Character classification
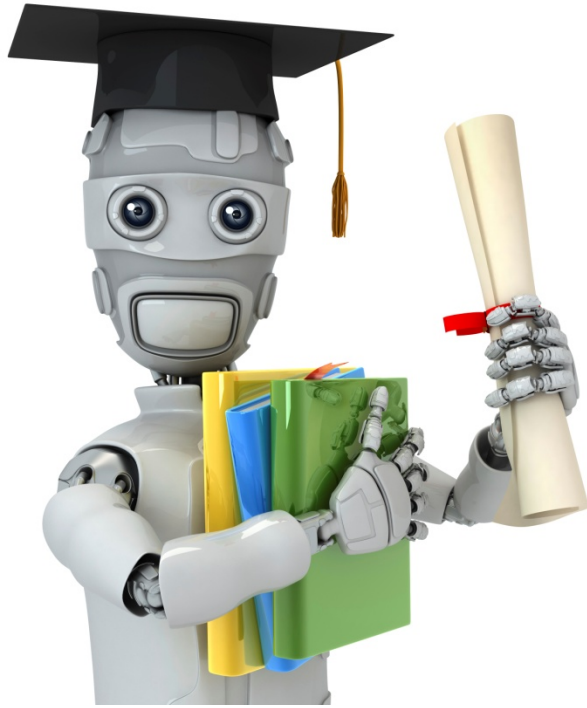
Machine Learning

Application example: Photo OCR

Getting lots of data: Artificial data synthesis

Character recognition
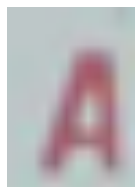
Artificial data synthesis for photo OCR

Real data

Abcdefg
Abcdefg
Abcdefg
Abcdefg
Abcdefg

Andrew Ng

Artificial data synthesis for photo OCR

Real data

Synthetic data

Synthesizing data by introducing distortions
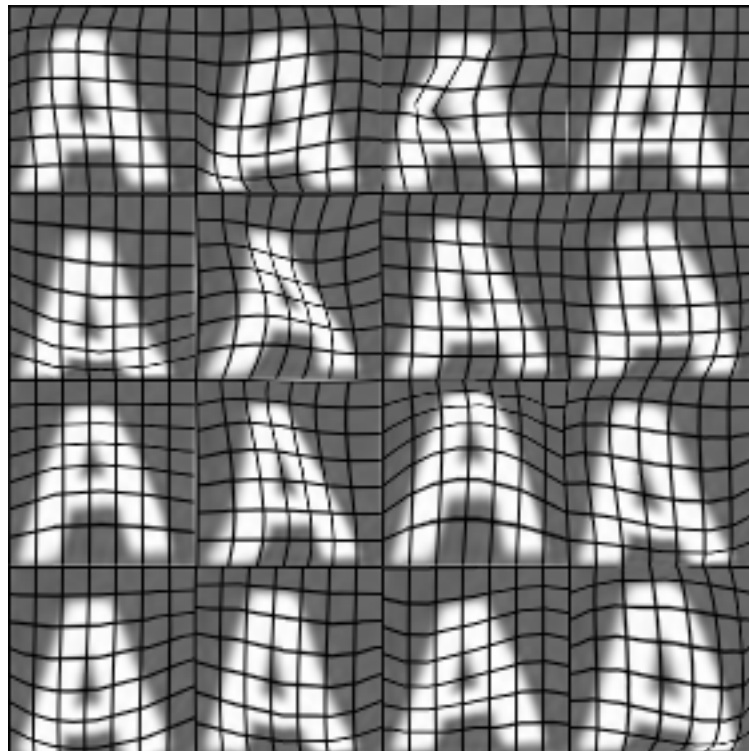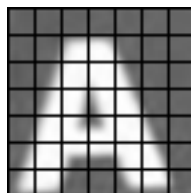
Synthesizing data by introducing distortions: Speech recognition

🔊 Original audio: ←

🔊 Audio on bad cellphone connection

🔊 Noisy background: Crowd

🔊 Noisy background: Machinery

Andrew Ng

Synthesizing data by introducing distortions

→ Distortion introduced should be representation of the type of noise/distortions in the test set.
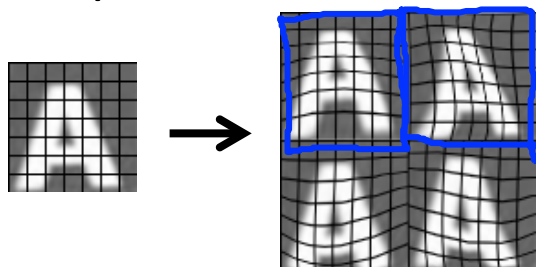
→ Audio:
Background noise,
bad cellphone connection

→ Usually does not help to add purely random/meaningless noise to your data.
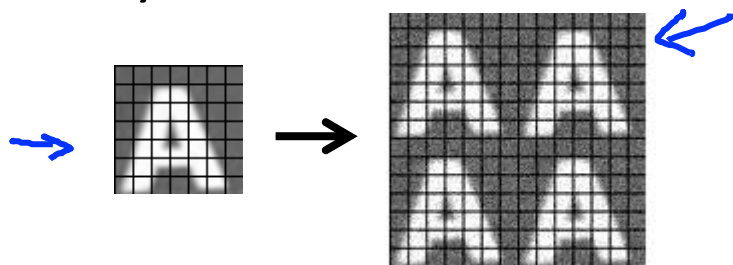
→ x_i = intensity (brightness) of pixel i
→ $x_i \leftarrow x_i +$ random noise

[Adam Coates and Tao Wang]

Andrew Ng

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)

$\rightarrow$ #hours?

$m = 1,000$

$\rightarrow$ 10 secs / example

$m = 10,000$

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)
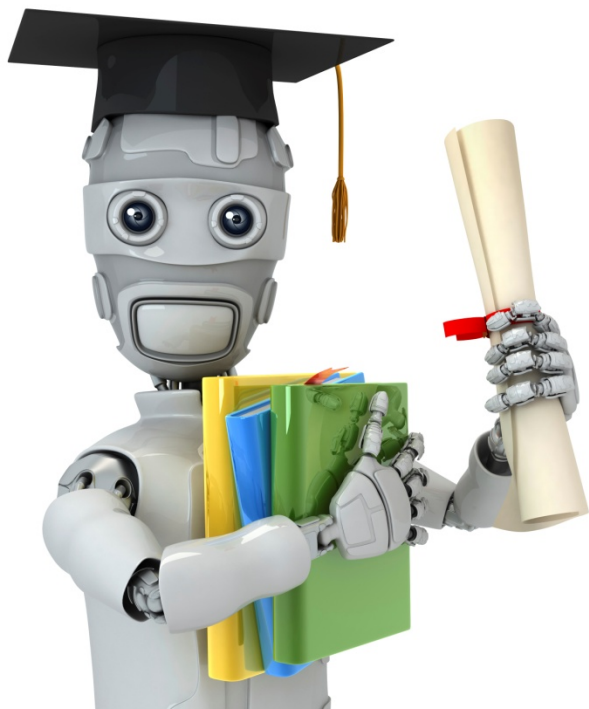
Machine Learning

Application example: Photo OCR

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

Estimating the errors due to each component (ceiling analysis)

What part of the pipeline should you spend the most time trying to improve?

Component	Accuracy
Overall system	72%
Text detection	89%
Character segmentation	90%
Character recognition	100%

Another ceiling analysis example

Face recognition from images
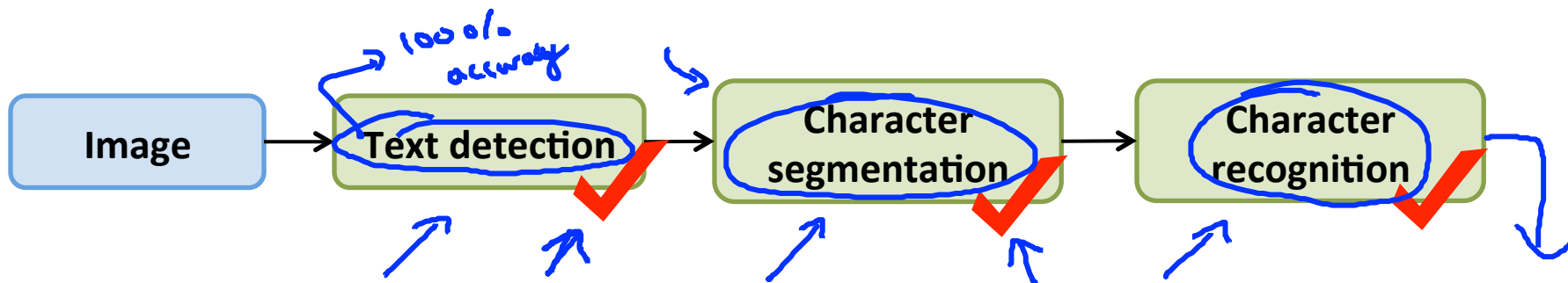(Artificial example)


```
┌─────────┐      ┌──────────────────────┐
│ Camera  │ ───► │     Preprocess       │
│ image   │      │ (remove background)  │
└─────────┘      └──────────────────────┘
                            │
                            ▼
              ┌──────────────────┐        ┌──────────────────────┐
              │                  │ ─────► │  Eyes segmentation   │
              │                  │        └──────────────────────┘         ┌──────────────────────┐
              │  Face detection  │ ─────► │  Nose segmentation   │ ──────► │  Logistic regression │ ──► Label
              │                  │        └──────────────────────┘         └──────────────────────┘
              │                  │ ─────► │        Mouth         │
              └──────────────────┘        │    segmentation      │
                                          └──────────────────────┘
```

Another ceiling analysis example

Component	Accuracy
Overall system	85%
Preprocess (remove background)	85.1%
Face detection	91%
Eyes segmentation	95%
Nose segmentation	96%
Mouth segmentation	97%
Logistic regression	100%

Handwritten annotations:
- Preprocess (remove background): 0.1%
- Face detection: 5.9%
- Eyes segmentation: 4%
- Nose segmentation: 1%
- Mouth segmentation: 1%
- Logistic regression: 3%

Andrew Ng