Application Case Study:
Often times, a Machine Learning application has a pipeline stream of applications that each contribute to the final output of a model.
An classical example:
Photo Optical Character Recognition:

⇨ To get a computer to read the text on a image

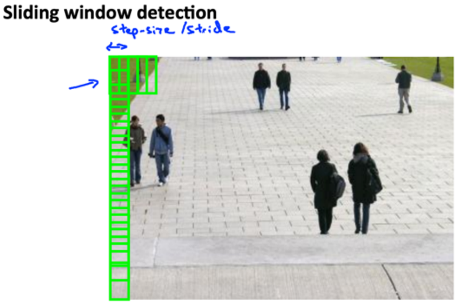
This is streamlined into 3 subsystems:

1) Text Detection
2) Character Segmentation
3) Character Classification

Each subsystem could be considered as a Machine Learning Project of its own, and be thusly split into work divided amongst a team of data engineers.

Although, not all sections necessarily using Machine Learning:

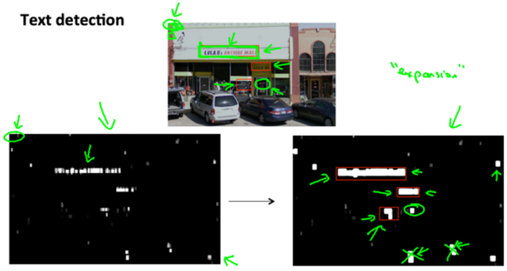
For Computer Vision:



Sliding Window Detection is how text could be detected in the image.
Perhaps we train a model to detect characters when it is 28x28 scale. So on the image, we run a scan of a square of 28x28 over the picture, with a stride of 2. Meaning that we shift 2 pixels across each time.

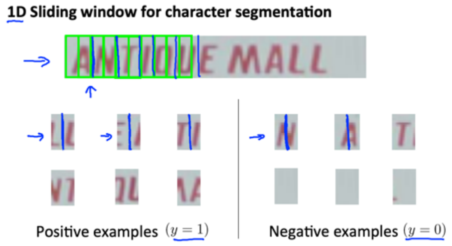
We can also do the same with a larger subsection, and then scale that image down before fitting it into our model.

Then perhaps we get the output 2nd picture:



The light spots indicating regions which the model has some confidence, has characters.
We then draw bounding boxes around the area and “expand” the area to a consistant shape.

2 Character Segmentation:
Same idea as above, but on these identified regions instead.


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

3: Once such features have been detected and segmented, we feed the resulting output into a character classifier.

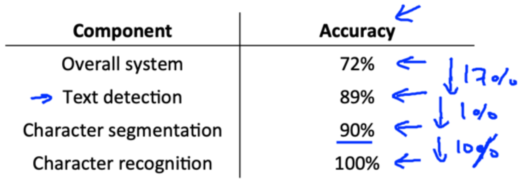
Large Scale Data Collection:
Different ML projects have different kinds of data collection required.
There are generally 2 ways to make new data.

1) Artificial Datasets
    - Synthesising a new dataset from information online
    - Eg: Using Computer Font letters and sticking them on random backgrounds.
2) Synthesis of more data from the current datasets
    - Dependent on the dataset, eg introducing distortions or noise into the dataset.
    - However, does not help to add purely random or meaningless noise to the dataset.

However, before engaging on such endevours:

1. Ensure a low bias classifier before expending the effort.
    - Eg Increase number of features / hidden units in the NN 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, est number of hours?
    - “Crowd Source” (Amazon Mechanical Turk)

Celing Analysis => What part of the pipeline to work on next?
Eg: Overall accuracy of OCR => 72%
In the pipeline, simulate each part as having a 100% accuracy, and checking which portion gets the largest gain in final accuracy.



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

↓17%
↓1%
↓10%

Hence, Text Detection appears to be the part that most impacts the accuracy of the entire sequence, followed by Character Recognition System.
Hence Character Segmentation is likely not worth much more effort to improve.