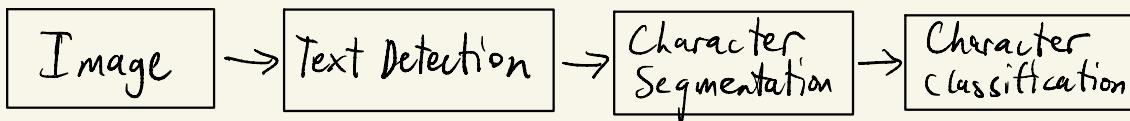



Jun 1, 2021

Photo optical character recognition focuses on how to get computers to read the text in the images we take.

Photo OCR Pipeline:

1. Text detection → finding regions of text on an image
2. Character segmentation → separating individual letters
3. Character classification → converting letter imgs to strings/chars



Supervised Learning For Pedestrian Detection:

* Slightly easier example of object detection
 x = pixels in 82×36 image patches



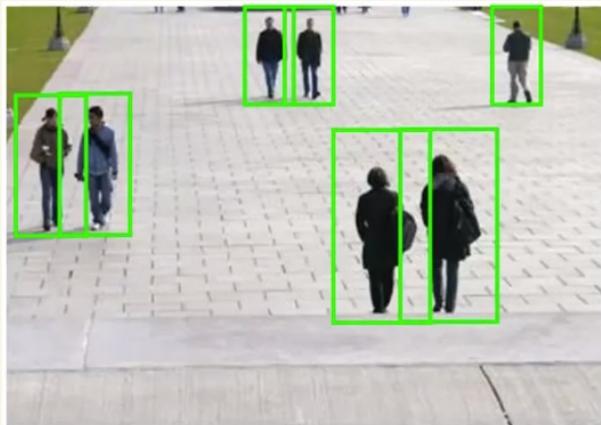
Sliding Window Detection:

resized to 82x36, check if there are pedestrians



↳ Increase window size, slide again

Result:



Text Detection!

PATRIDNF

OUONIKU

Positive examples ($y = 1$)



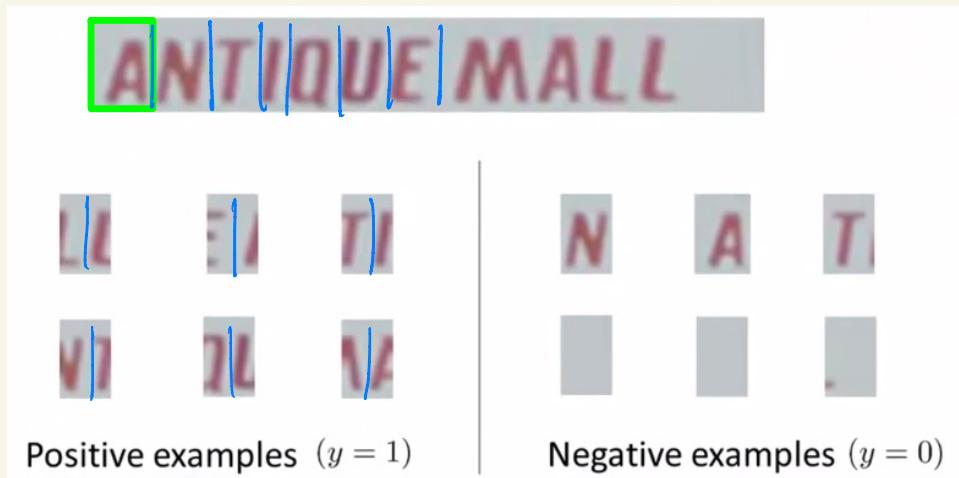
Negative examples ($y = 0$)

Text detection



Discard white rectangles whose aspect ratios are more tall than wide.

1D Sliding Window for Character Segmentation!



Train classifier to detect midpoint of two characters.

↳ Use to split characters.

Finally, use a regular classifier to classify the individual letters.

Artificial Data Synthesis



Real data



Synthetic data

Basically, take characters from a font found online, place it against some random background image, add some distortions/scaling/rotations, and you have realistic-looking synthetic data.

Data Augmentation! #Also part of data synthesis

Alternatively, you can introduce artificial transformations to your original images to amplify your dataset.

#But the augmentations should be representative of the types seen in the test set!

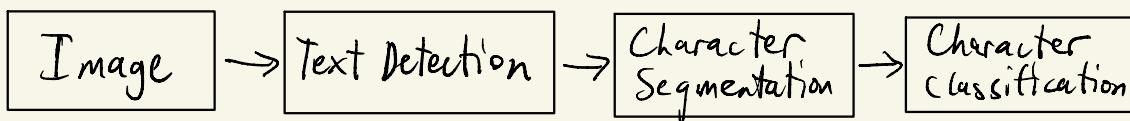
↳ random augmentations are less likely to be useful

Additional Notes:

1. Make sure you have a low bias classifier before augmenting data.
 - ↳ Or increase # features / hidden units until it is low bias
2. Is the amount of work needed to get 10X more data a lot, or not that much?

Ceiling Analysis:

What part of the ML pipeline should we spend the most time on?



Component	Accuracy	\$ of overall system
Overall System	72%	
100% ← Text Detection	89%	+17% ↳ Most benefit from spending time on text detection.
100% ← Character Segmentation	90%	+1%
100% ← Character Classification	100%	+10%

How much does the system improve if you make each component 100% accurate?

↳ Plug in ground-truths and evaluate.