

Ceiling Analysis: What Part of the Pipeline to Work on Next ¶

Throughout the course repeatedly said one of the most valuable resources is **developer time**

- Pick the right thing for us and our team to work on
- **Avoid spending a lot of time to realize the work was pointless in terms of enhancing performance**

Estimating the errors due to each component (ceiling analysis): Photo OCR pipeline



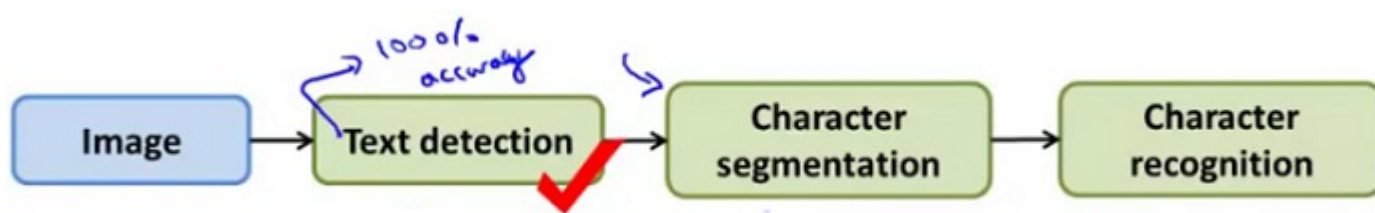
We can have three modules in our pipeline where each one could have a small team on it:

- **Where should we allocate resources?**
 - What part of the pipeline should we spend the most time trying to improve?
- **Good to have a single real number as an evaluation metric**
- So, character accuracy for this example
 - Find that our test set has 72% accuracy

Concretely, we have some set of test set images and from each test set images, we run it through **text detection**, then **character segmentation**, then **character recognition** and we find that on our test set the **overall accuracy** of the entire system was 72% on whatever metric we chose.

Ceiling analysis on our pipeline of Photo OCR pipeline

- We're going to the first module:
 - Mess around with the test set - **manually tell the algorithm where the text is**
 - **Simulate if our text detection system was 100% accurate**
 - So we're feeding the character segmentation module with 100% accurate data now
 - How does this change the accuracy of the overall system



So, we're going to go to our test set and just give it the correct answers, **give it the correct labels** for the text detection part of the pipeline, so that as if we have a **perfect test detection system on our test set**.

- Accuracy goes up to 89%

Next we do the same for the character segmentation:

- Again, we're gonna go to our test set, and now we're going to give it **the correct text detection output**
 - and give it **the correct character segmentation output**.
 - So go to the test set and manually **label the correct segmentations of the text into individual characters**
 - see how much that helps.
- Accuracy goes up to 90% now

Finally we do the same for character recognition

- Accuracy goes up to 100%

Having done this we can qualitatively show what the upside to improving each module would be

- Perfect text detection improves accuracy by 17%
- Would bring the biggest gain if we could improve

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

- Perfect character segmentation would improve it by 1%
 - **Not worth working on**
- Perfect character recognition would improve it by 10%
 - **Might be worth working on, depends if it looks easy or not**

The "ceiling" is that each module has a ceiling by which making it perfect would improve the system overall

- how much could we possibly gain if one of these components became absolutely perfect?

Video Question: Suppose you perform ceiling analysis on a pipelined machine learning system, and when we plug in the ground-truth labels for one of the components, the performance of the overall system improves very little. This probably means: (check all that apply)

- We should dedicate significant effort to collecting more data for that component.

It is probably not worth dedicating engineering resources to improving that component of the system.

If that component is a classifier training using gradient descent, it is probably not worth running gradient descent for 10x as long to see if it converges to better classifier parameters.

- Choosing more features for that component may help (reducing bias), and reducing the number of features for that component (reducing variance) is unlikely to do so.