Comparing Open-source OCR

Michael Renteria
Mrente5@illinois.edu

I'm interested in comparing three of the most popular open source OCR softwares. OCR stands for optical character recognition. In a nutshell, OCR is utilizing computational resources to convert text from an image to actual text and strings that may be processed further by the computer. To find the top contenders I did a search request on GitHub.com to return repos that reference "OCR", and then I ordered the results by the highest number of stars. In this report, I am going to be looking into "tesseract"; "EasyOCR"; and "PaddleOCR", based on their functionality, documentation, and GitHub (version control) repositories. After I take a quick look through their documentation and source code, I will take the same three images and convert them to text with each one of the softwares. After the image is converted to text, I will judge each software based on if it is true to the use case, installation, and expected result.

Tesseract is based on C++, and you are able to interact with it directly with a command line or a C++ api. If you are interested in interacting with it with different languages, you can select from different third party wrappers that make this interaction possible. The official documentation for tesseract is pretty basic and does not have many examples or use cases. I imagine that if you would want to actually build something directly off of tesseract, it would require a lot of code tracing throughout their version control. Tesseract supports over one hundred different languages. The actual installation for Tesseract is pretty standard, with the choice to build from source or via the apt-get package manager. At the time of writing, this software has recent updates, and active community support.

EasyOCR is based on python, you are able to interact with EasyOCR with python directly or on the command line. The documentation for EasyOCR is not very extensive and does not contain many examples. However, the EasyOCR documentation does contain descriptions of all of the methods and parameters that are used within the package. EasyOCR supports over seventy different languages. The installation simply uses python's package manager. EasyOCR also supports the option to utilize either your GPU or your CPU in order to generate results. At the time of writing, this software has recent updates, and active community support.

PaddleOCR is based on python. The documentation is very extensive, but it does not seem to be organized very well. The installation can simply be managed by python's package manager, but they also offer support for docker to make it easier to manage dependencies. PaddleOCR has a unique online experience that allows you to understand how the technology works without having to install anything yourself. Unfortunately, the online experience is not in English. It seems like the main way of utilizing the software is directly via python. It looks like they have other methods of utilizing the software, but since the documentation is not well organized it is not very easy to find. PaddleOCR has the option to utilize either your GPU or CPU for processing data. At the time of writing, this software has recent updates, and active community support.

The installation for Tesseract was the simplest and most straight to the point. The apt-get package manager installed all of the necessary dependencies. EasyOCR and PaddleOCR both were easy to install, assuming python was properly installed on the machine. Interestingly enough, both EasyOCR and PaddleOCR both errored out due to missing dependencies and required a google search in order to figure out what dependency was missing. The actual use for Tesseract was the most simple and only required two parameters, input and output. EasyOCR was a little more complicated but was still able to be accessed via the cli and was easily aided with the cli help command. PaddleOCR required actual code to written in python, which I needed help from the documentation to figure out.

After testing each OCR toolkits with the three different images I got a better idea on the functionality and accuracy of each one. The first image I choose was a snapshot of the first page of the United States constitution. I imagined that this would be next to impossible for the OCR to detect, because of the resolution of the image and the type of font that was used. The next image I picked was a bill from AT&T, this example is interesting because although the font is easily readable the format is strangely structured. The third image is an image of a basic advertisement with very little text. Tesseract, EasyOCR, and Paddle OCR all spit out random non relevant characters for the image of the United States constitution. Unexpectingly, Tesseract errored out and was not able to process the bill from AT&T. PaddleOCR and EasyOCR were both able to process the AT&T bill but were not 100 percent accurate. All three of the toolkits were able to process the simple advertisement of text. EasyOCR was the most accurate for the simple advertisement of text, Paddle OCR was the next most accurate, and Tesseract came in last.

In the future, If I were to build a project that required OCR, I am most compelled to use EasyOCR, because of its direct python support; accuracy; and ease of install. I am a little surprised with the output of Tesseract, given that it is the most popular of the three. It is important to note, that although Tesseract performed the weakest of the three, it did not require you to specify the input language, while the other two did. If I were to do this experiment again, I would use a greater variety of documents, multiple languages, and also dig in deeper to advanced parameters that may create a greater accuracy inside of the output.

Sources

https://github.com

https://github.com/PaddlePaddle/PaddleOCR

https://github.com/tesseract-ocr/tesseract

https://github.com/JaidedAI/EasyOCR

Images

https://wordtohtml.net/images/word-to-html-sharing-image-2020.png

https://noveltydocumentusa.com/wp-content/uploads/2019/03/Att.jpg

https://upload.wikimedia.org/wikipedia/commons/6/6c/Constitution_of_the_United_States%2C_page_1.jpg