Build your own OCR(Optical Character Recognition) for free

Balaaji Parthasarathy Feb 20, 2018 · 6 min read

Optical Character Recognition, or OCR is a technology that enables you to convert different types of documents, such as scanned paper documents, PDF files or images captured by a digital camera into editable and searchable data. It converts these documents into machine coded text.

OCR has been gaining recent popularity and being able to identify what is present in the image opens up a new horizon of opportunities.

For the past few years OCR frameworks has evolved a lot but not to an extent where they could be 100% for any image size or any image quality.

In order to get it more closer to 100%, requires a lot of tuning and training. There is a lot of pre-processing work involved before the most accurate information could be retrieved.

There are many softwares/APIs available out there which could be do a pretty good job of processing an image and based on what they could do and how well they do the prices vary.

Let us go over some of these in detail.

OCR has been gaining recent popularity and being able to identify what is present in the image opens up a new horizon of opportunities.

For the past few years OCR frameworks has evolved a lot but not to an extent where they could be 100% for any image size or any image quality.

In order to get it more closer to 100%, requires a lot of tuning and training. There is a lot of pre-processing work involved before the most accurate information could be retrieved.

There are many softwares/APIs available out there which could be do a pretty good job of processing an image and based on what they could do and how well they do the prices vary.

Let us go over some of these in detail.

Some popular OCR APIs

Google vision api(https://cloud.google.com/vision/**)** is one of the most popular API's available and it gets you the most accurate information. Vision API is more of an image processing framework than just an optical character recognition framework. If the intention is to just identify what characters are present in the image, this framework has a lot more to it. This framework is really expensive unless your base set of images are a few.

Below is the pricing information.

https://cloud.google.com/vision/pricing

Amazon Rekognition(https://aws.amazon.com/rekognition/) is again an image processing framework just like the Google's Vision API.This framework uses deep learning technology to identify objects, image and faces. This is little less expensive than Vision API.

Below is the pricing information.

https://aws.amazon.com/rekognition/pricing/

OCR Space(https://ocr.space/) is a more of a budget friendly option compared to the first 2 options. This SDK does a neat job of getting the needed information but not to the level of Rekognition and Vision APIs. If your requirement is less than 25K request a month you can even get away for free.

Below is the pricing information.

https://ocr.space/ocrapi

Open Source Frameworks:

There are a couple of open source frameworks that can be used to build an OCR framework in house. They are effective too as long as you know how to train it for your requirements. Listed below are a couple of such frameworks.

Python pyocr

PyOCR(https://github.com/jflesch/pyocr) is an optical character recognition (OCR) tool wrapper for python. That is, it helps using OCR tools from a Python program.It has been tested only on GNU/Linux systems. It should also work on similar systems (*BSD, etc). It may or may not work on Windows, MacOSX, etc.

PyOCR can be used as a wrapper for google's Tesseract-OCR or Cuneiform. It can read all image types supported by Pillow, including jpeg, png, gif, bmp, tiff, and others. It also support bounding box data.

Tesseract-OCR

Tesseract is an optical character recognition engine for various operating systems.It is free software released under the Apache License, Version 2.0, and was originally developed at Hewlett-Packard Laboratories Bristol and at Hewlett-Packard Co, Greeley Colorado between 1985 and 1994, with some more changes made in 1996 to port to Windows, and some C++izing in 1998. In 2005 Tesseract was open sourced by HP. It was later developed and sponsored by Google since 2006. Tesseract is considered as one of the most accurate open-source OCR engines currently available.

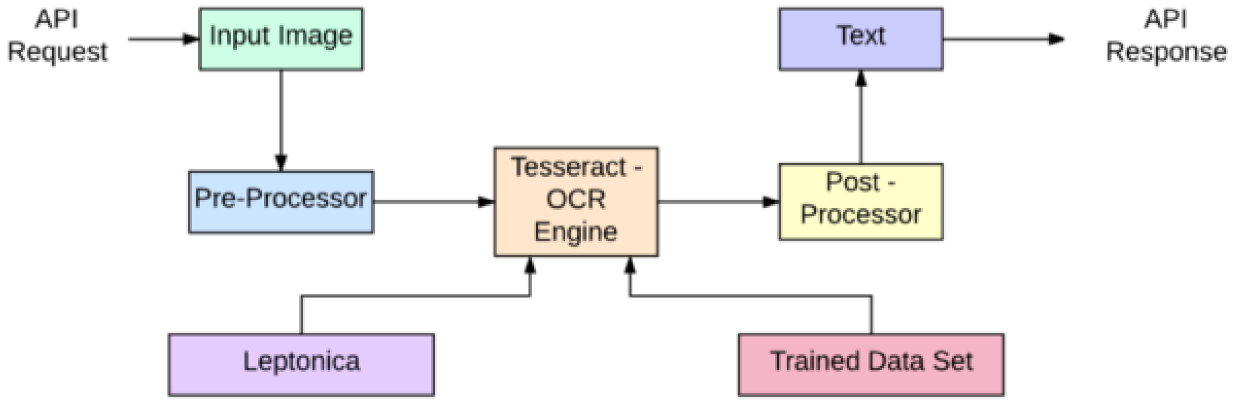
There were not many open source options for being able to build on your own. In this document, we will be do a deep dive into the Tesseract framework and how to have it setup and how good or bad would the outcomes be.

Most OCR frameworks out there is probably built on top of Tesseract and it is the most popular among the bunch which has pretty good outcomes.

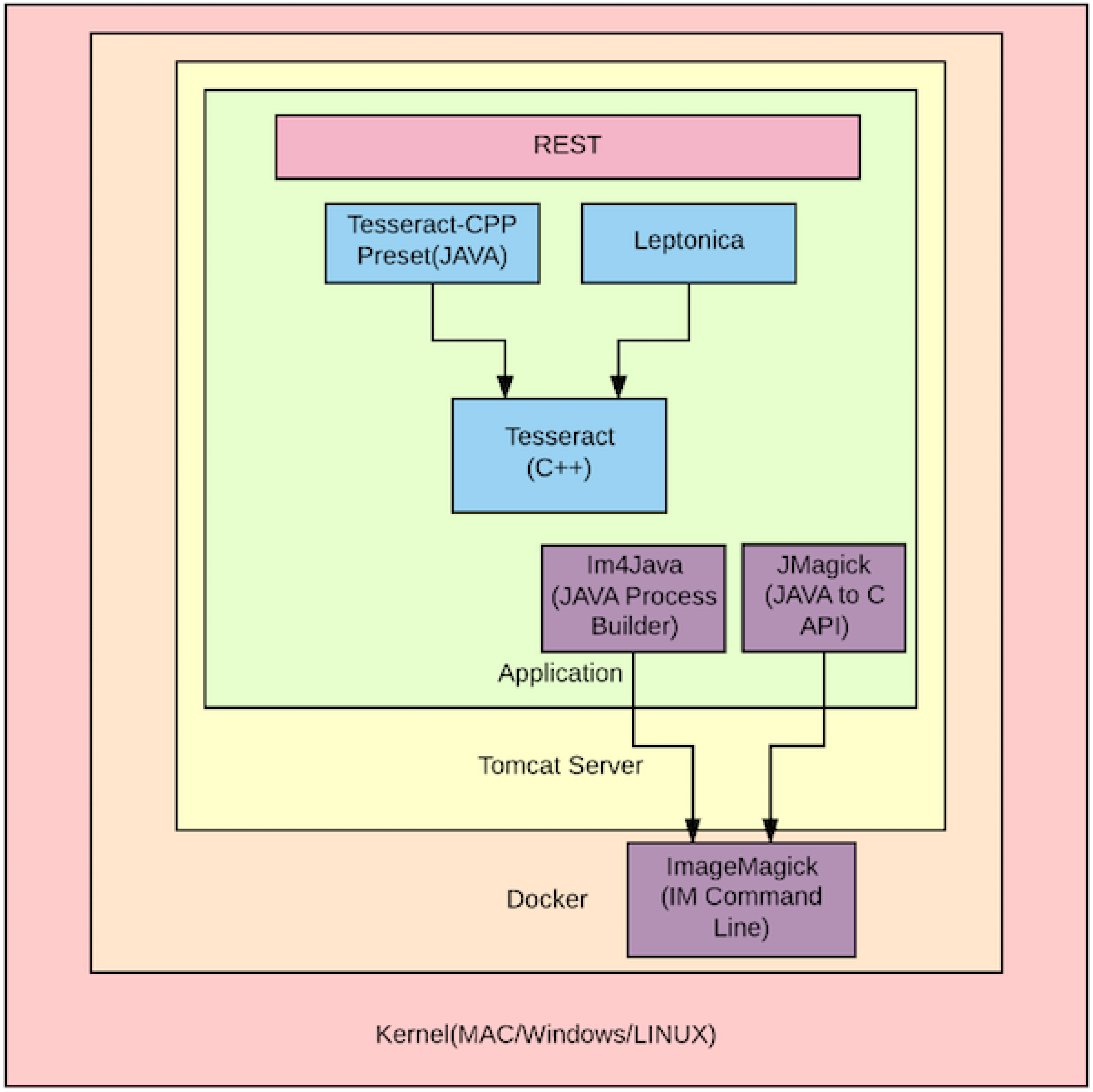
Tesseract supports a whole slew of languages like no other framework. It supports English, Spanish, Thai all the way upto Tamil, Uzbec and Yiddish. It will be hard to find something that is not supported.

Overview

OCR Process Flow



Machine Level Architecture


REST
Tesseract-CPP Preset(JAVA)
Leptonica
Tesseract (C++)
Im4Java (JAVA Process Builder)
JMagick (JAVA to C API)
Application
Tomcat Server
ImageMagick (IM Command Line)
Docker
Kernel(MAC/Windows/LINUX)

How to setup and get it to work?

For this exercise I use a Dockerized Java Spring — boot application with a Gradle build.

Gradle dependency needs to be added for Tesseract, Leptonica, JMagick and Im4Java. Lets discuss in a few on what are these

- **Gradle dependency**

```
dependencies {

compile group: 'org.bytedeco.javacpp-presets', name: 'tesseract', version: '3.03-rc1-1.0'

compile group: 'org.bytedeco.javacpp-presets', name: 'tesseract', version: '3.03-rc1-1.0', classifier: opencvBinaryClassifer

compile group: 'org.bytedeco.javacpp-presets', name: 'leptonica', version: '1.72-1.0', classifier: opencvBinaryClassifer

compile group: 'jmagick', name: 'jmagick', version: '6.6.9'

compile group: 'org.im4java', name: 'im4java', version: '1.4.0'

}
```

- Tesseract -CPP Preset — It is the Java wrapper for Tesseract which is built on a CPP framework.
- Leptonica — Its a dependency for Tesseract, through which we get support to several image formats. It also gets position and page layout Information.
- JMagick — JMagick is the java interface for ImageMagick C-API.
- Im4Java — It is the Java wrapper for ImageMagick. This fires command line ImageMagic commands using Java Process builder.

We would also have to make sure we have ImageMagick setup in our machine. This could be easily done using brew.

- brew install imagemagick
- brew info imagemagick — We can run this command to make sure the installation was successful.

With this you are done with the setup and now you can start coding.

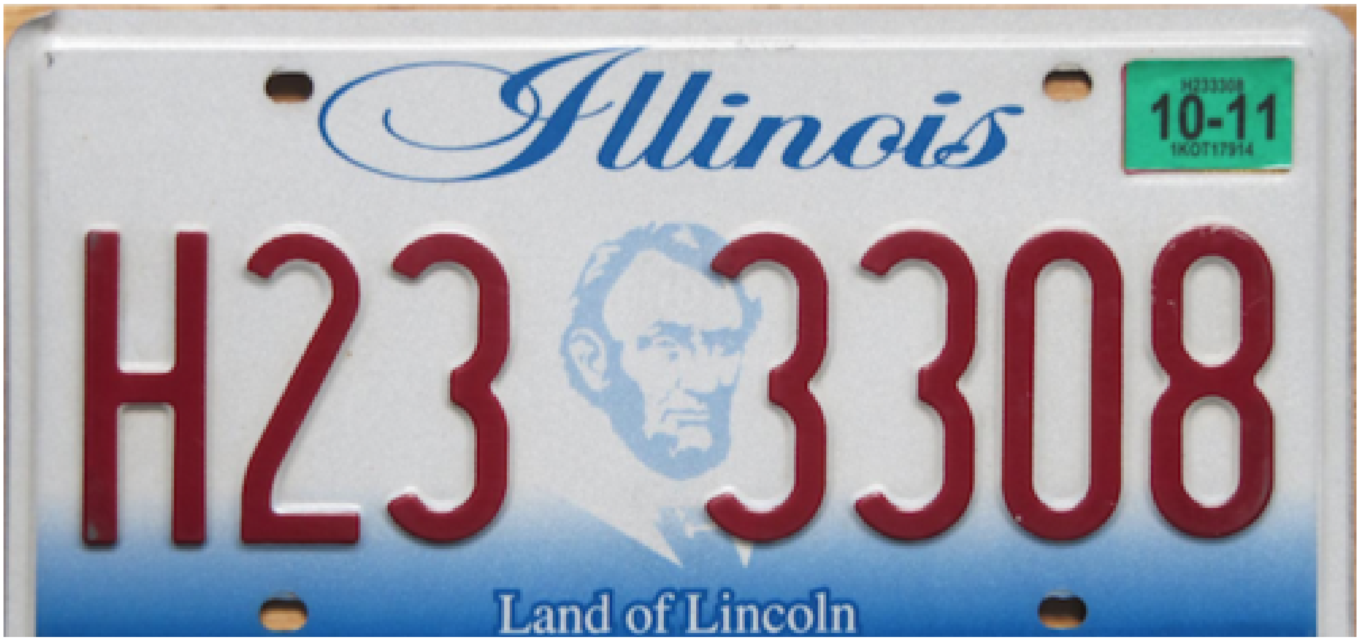
How to improve the efficiency of the output using tesseract?

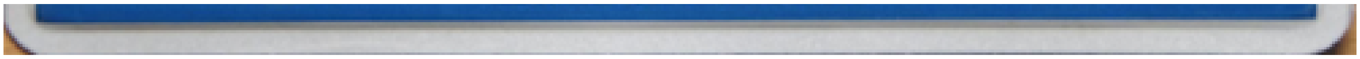
- In order for Tesseract to work its best you would have to make sure the image is as clear as possible.
- Which could might mean we would have to perform image modifications such as resizing, colorspace, contrast, morphology, filter(Gaussian, Triangle, Spline, etc), edge detection.
- For this reason we will make use of JMagick which has a slew of functions which makes use of ImageMagick under the skin to perform image modifications.
- Here are some useful links for performing Image Modifications

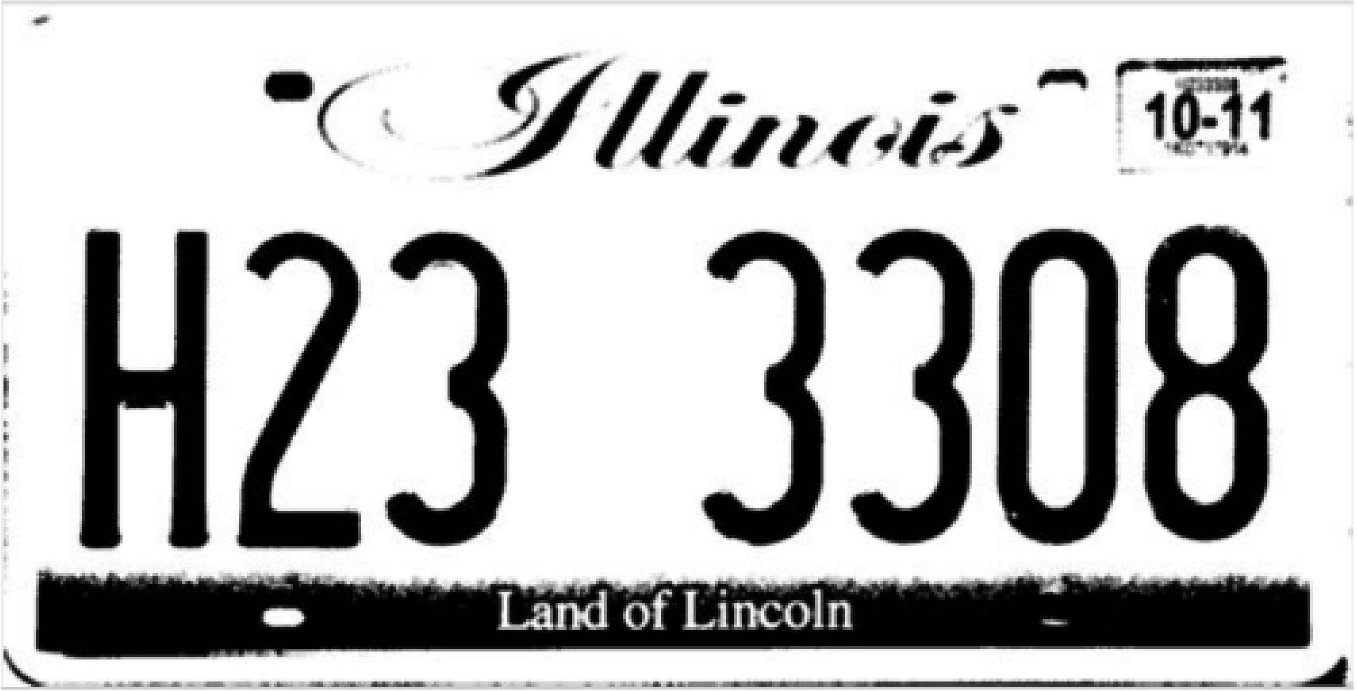
http://www.fmwconcepts.com/imagemagick/downsize/index.php

http://www.imagemagick.org/script/index.php

- Below are sample images of what it was and how it needs to be for Tesseract to understand and perform OCR.







Font Recognition?

You can come across circumstances where Tesseract does not recognize respond with all the text seen on the image this could be due to the fact that Tesseract was not programmed to understand the font in the image. For this reason it becomes mandatory to identify the font, install and generate trained data files for the needed fonts.

Below are the steps to be able to achieve the same

- Install Tesseract on the machine

```
brew install — with-training-tools tesseract
```

- Download and Install JTessBox Editor

https://sourceforge.net/projects/vietocr/?source=typ_redirect

- Identify the font in the image and install it on the system
- Open the JTessBox Editor and choose the needed font and type in a sentence with all the needed characters.

- Clicking on generate, would create .box and .tif files.
- Now update the font name in the below code and run the python script using the below command
- python tesseract-trainer.py

Python Tesseract Script Expand source

- Once the python script run successfully it will generate a slew of file and will add the same to the tesseract installable. Although you would need to copy them and add it to the tessdata folder in your project.

Useful links:

http://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=1416&context=etd_projects

https://en.wikipedia.org/wiki/Tesseract

http://im4java.sourceforge.net/

https://www.smashingmagazine.com/2015/06/efficient-image-resizing-with-imagemagick/

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

http://www.programcreek.com/java-api-examples/index.php?api=org.im4java.core.ConvertCmd

http://im4java.sourceforge.net/docs/dev-guide.html

https://medium.com/@sathishvj/training-tesseract-ocr-for-a-new-font-and-input-set-on-mac-7622478cd3a1#.ju5p3mv47