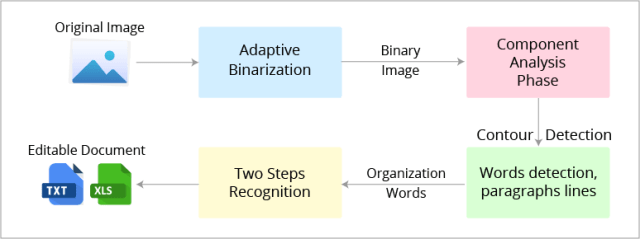
**Tesseract OCR**

Tesseract is an open-source text recognition engine that is available under the Apache 2.0 license and its development has been sponsored by Google since 2006. In the year 2006, Tesseract was considered as one of the most accurate open-source OCR engines. You can use it directly or can use the API to extract the printed text from images. The best part is that it supports an extensive variety of languages. It is through wrappers that Tesseract can be made compatible with different programming languages and frameworks. In this blog, I’ll be using the Python wrapper named pytesseract. It is used to recognize text from a large document, or it can also be used to recognize text from an image of a single text line. Below is the visual representation of the Tesseract OCR architecture as represented in the Voting-Based OCR System research paper.



Talking about the Tesseract 4.00, it has a configured text line recognizer in its new neural network subsystem. These days people typically use a Convolutional Neural Network (CNN) to recognize an image that contains a single character. Text that has arbitrary length and a sequence of characters is solved using Recurrent Neural Network (RNNs) and Long short-term memory (LSTM) where LSTM is a popular form of RNN. The Tesseract input image in LSM is processed in boxes (rectangle) line by line that inserts into the LSTM model and gives the output.

By default, Tesseract considers the input image as a page of text in segments. You can configure Tesseract’s different segmentations if you are interested in capturing a small region of text from the image. You can do it by assigning **–psm** mode to it. Tesseract fully automates the page segmentation but it does not perform orientation and script detection. The different **configuration parameters for Tesseract** are mentioned below:

**Page Segmentation Mode (–psm):** By configuring this, you can assist Tesseract in how it should split an image in the form of texts. The command-line help has 11 modes. You can choose the one that works best for your requirement from the table given below:

|  |  |
| --- | --- |
| mode | Working description |
| 0psm | Orientation and script detection (OSD) only |
| 1 | Automatic page segmentation with OSD |
| 2 | Automatic page segmentation, but no OSD, or OCR |
| 3 | Fully automatic page segmentation, but no OSD (Default) |
| 4 | Presume a single column of text of variable sizes |
| 5 | Assume a single uniform block that has a vertically aligned text |
| 6 | Assume a single uniform block of text |
| 7 | Treat the image as a single text line |
| 8 | Treat the image as a single word |
| 9 | To treat the image as a single word in a circle |
| 10 | Treat the image as a single character |
| 11 | Sparse text. Find as much text as possible not in a particular order |
| 12 | Sparse text with OSD |
| 13 | Raw line. Treat the image as a single text line, bypass hack by Tesseract-specific. |

**Engine Mode (–oem):** Tesseract has several engine modes with different performance and speed. Tesseract 4 has introduced an additional LSTM neutral net mode that works the best. Follow the table given below for different OCR engine modes:

|  |  |
| --- | --- |
| OCR engine mode | Working description |
| 0 | Legacy engine only |
| 1 | Neural net LSTM only |
| 2 | Legacy + LSTM mode only |
| 3 | By Default, based on what is currently available |

**Limitations of Tesseract**

* The OCR’s accuracy is not as apt as compared to some currently available commercial solutions.
* It is not capable of recognizing handwritten text.
* If a document contains languages that are not supported by Tesseract then results will be poor.
* It requires a clear image as input. A poor quality scan may produce poor results in OCR.
* It doesn’t give accurate results of the images affected by artifacts including partial occlusion, distorted perspective, and complex background.
* It is not good at analysing the normal reading order of documents. For example, you might fail to recognize that a document contains two columns, and might try to join the text across those columns.
* It does not expose the font family’s text information.

In the end, it can be concluded that Tesseract is perfect for scanning clean documents and you can easily convert the image’s text from OCR to word, pdf to word, or to any other required format. It has pretty high accuracy and font variability. This is very useful in case of institutions where a lot of documentation is involved such as government offices, hospitals, educational institutes, etc. In the current release 4.0, Tesseract supports OCR based deep learning that is significantly more accurate.