**Methodology Followed in API:**

At present we have two API’s 1) http:/[/<host>:<port>/](http://172.17.1.127:3000/) (<http://172.17.1.215:4565/>)

2) http://<host>:<port>[/benchmark](http://172.17.1.127:3000/benchmark) (http://172.17.1.215:4565/benchmark)

* to upload an individual file and get recognised text, confidence values, mean Confidence and acceptance or rejection of the image and
* to compare the results of different images.

The present API’s are build using Tesseract java application ( Tess4j). To improve the accuracy of tesseract we modified the training data set.

For Pre-Processing we build a separate packages for Thresholding (Bradley Local Thresholding and Ostu) and few image enhancement ,filtering techniques(using JMagick library) , but not implemented in the present API.

**Source Code:**

https://github.com/worldofprasanna/ocrengine

Below are the other local thresholding methods which can be used to improve the binary images.

**Bradley Local Thresholding:**

Performs local thresholding of a two - dimensional image (GrayScale Image).

Every image's pixel is set to black if its brightness is T percent lower than the average brightness of surrounding pixels in the **window** of the specified size, otherwise it is set to white.

Window: Ex- Say [3,3] is the window size. T: 10%

Now for calculating the local threshold for each pixel (p) of the image, algorithm selects a integral image i.e, of size 3X3 where p is at the centre. Average brightness is calculated for this integral image and if brightness value of p is less than 10% of the average value then the pixel (p) is set to black else it is set to white.

**Then how the threshold value is calculated for the boundary pixels?**

Padding Technique: <http://www.mathworks.in/help/images/ref/padarray.html>

**Niblack Local Thresholding:**

Niblack’s algorithm calculates a pixel-wise threshold (T) based on the local mean **m** and the standard deviation **s** of all the pixels in the window. If the value of pixel (p) is less than the T then its value is set to black else set to white.

Equation:

T = m + k\*s ( k = constant and is fixed to -0.2)

This Algorithm fails when there is large variation in the illumination values of the Gray Image and the value of k suits for limited documents.

**Sauvola's Local Thresholding:**

Sauvola's algorithm is a modification of Niblack's algorithm, this algorithm calculated local thresholding value (T) as:

T = m\*(1 – k\*(1 – s/R) )

where k is set to 0.5 and R to 128. This method outperforms Niblack’s algorithm in images where the text pixels have near 0 gray-value and the background pixels have near 255 gray-values. However, in images where the gray values of text and non-text pixels are close to each other, the results degrade significantly.

**Wolf’s Local Thresholding:**

To address the issues in Sauvola’s algorithm, Wolf proposed another algorithm which is an extension of Niblack's algorithm to normalize the contrast and the mean gray value of the image.

Equation:

T = (1-k)\*m + k\*M + k\*s/R \*(m-M)

where k is fixed to 0.5, M is the minimum gray value of the image and R is set to the maximum gray-value standard deviation obtained over all the local neighborhoods windows.

Limitation of this algorithm is that if there is a sharp change in background gray values across the image. This is due to the fact that the values of M and R are calculated from the whole image. So even a small noisy patch could significantly influence M and R values, thus eventually calculating misleading binarization thresholds.

How to improve the accuracy of tesseract:

1. Add more training examples in the training data for tesseract
2. Meeting the requirements needed for tesseract
3. Other appropriate pre-processing steps.