**Name**

Zai-ul-Abideen

**Roll no.**

BCSF17E018

**Class**

BSCS (7th Self)

**Project**

Image To Text Conversion

**Tittle**

Week 4 (Plane Submission)

**Sir**

Fahad Maqbool

## Performance Tuning

The most important factor in the speed of an OCR job is in fact the quality of the input image. The less background noise that is present and the higher the dpi, with a perfect target dpi at about 200 dpi, will cause the fastest and most accurate OCR results.

This is not, however, necessary, as Iron OCR shines at correcting imperfect documents (though this is time-consuming and will cause your OCR jobs to use more CPU cycles).

If possible, choosing input image formats with less digital noise such as TIFF or PNG can also yield faster results than lossy image formats such as JPEG.

#### Image Filters

The following Image filters can really improve performance:

* ****OcrInput.Rotate( double degrees)**** - Rotates images by a number of degrees clockwise. For anti-clockwise, use negative numbers.
* ****OcrInput.Binarize()**** - This image filter turns every pixel black or white with no middle ground. May Improve OCR performance cases of very low contrast of text to background.
* ****OcrInput.ToGrayScale()**** - This image filter turns every pixel into a shade of grayscale. Unlikely to improve OCR accuracy but may improve speed
* ****OcrInput.Contrast()**** - Increases contrast automatically. This filter often improves OCR speed and accuracy in low contrast scans.
* ****OcrInput.DeNoise()**** - Removes digital noise. This filter should only be used where noise is expected.
* ****OcrInput.Invert()**** - Inverts every color. E.g. White becomes black : black becomes white.
* ****OcrInput.Dilate()**** - Advanced Morphology. Dilation adds pixels to the boundaries of objects in an image. Opposite of Erode
* ****OcrInput.Erode()**** - Advanced Morphology. Erosion removes pixels on object boundariesOpposite of Dilate
* ****OcrInput.Deskew()**** - Rotates an image so it is the right way up and orthogonal. This is very useful for OCR because Tesseract tolerance for skewed scans can be as low as 5 degrees.
* ****OcrInput.DeepCleanBackgroundNoise()**** - Heavy background noise removal. Only use this filter in case extreme document background noise is known, because this filter will also risk reducing OCR accuracy of clean documents, and is very CPU expensive.

**Some Code Like**

**Imports IronOcr**

**Private Ocr = New IronTesseract()**

**' Configure for speed**

**'INSTANT VB TODO TASK: The following line could not be converted:**

**Ocr.Configuration.BlackListCharacters = "~`$#^\*\_}{][|\\@"**

**Ocr.Configuration.PageSegmentationMode = TesseractPageSegmentationMode.Auto**

**Ocr.Configuration.TesseractVersion = TesseractVersion.Tesseract5**

**Ocr.Configuration.EngineMode = TesseractEngineMode.LstmOnly**

**Ocr.Language = OcrLanguage.EnglishFast**

**Using Input = New OcrInput("img\Potter.tiff")**

**Dim Result = Ocr.Read(Input)**

**Console.WriteLine(Result.Text)**

**End Using**