**Name**

M.Qasim Ijaz

Asmat

Zai-ul-Abideen

**Roll no.**

BCSF17E002

BCSF17E007

BCSF17E018

**Class**

BSCS (7th Self)

**Project**

Image To Text Conversion

**Tittle**

Week 3 (Plane Submission)

**MODULE Cropping Of Image**

Reading Cropped Regions of Images

As you can see from the following code sample, Iron's fork of Tesseract OCR is adept at reading specific areas of images.

We may use a System.Drawing.Rectangle to specify, in pixels, the exact area of an image to read.

This can be incredibly useful when we are dealing with a standardized form which is filled out, where only a certain area has text which changes from case to case.

**Example: Scanning an Area of a Page**

We can use a System.Drawing.Rectangle to specify a region in which we will read a document. The unit of measurement is always pixels.

We will see that this provides speed improvements as well as avoiding reading unnecessary text. In this example we will read a student's name from a central area of a standardized document.





**Code Example**

using IronOcr;

var Ocr = new IronTesseract();

using (var Input = new OcrInput())

{

// a 41% improvement on speed

var ContentArea = new System.Drawing.Rectangle() { X = 215, Y = 1250, Height = 280, Width = 1335 };

Input.AddImage("img/ComSci.png", ContentArea);

var Result = Ocr.Read(Input);

Console.WriteLine(Result.Text);

}

This yields a 41% speed increase - and allows us to be specific. This is incredibly useful for .Net OCR scenarios where documents are similar and consistent such as Invoices, Receipts, Checks, Forms, Expense Claims etc.

ContentAreas (OCR cropping) is also supported when reading PDFs.