5.2 IMAGE PROCESSING DEVELOPMENT.

The importance of the image processing phase is to create the best environment possible when capturing the stylistic features of the handwriting text. Ideally, the resultant data available prior to the image pre-processing phase will be a binary image containing representations of only pen marks of the subject made to represent text.

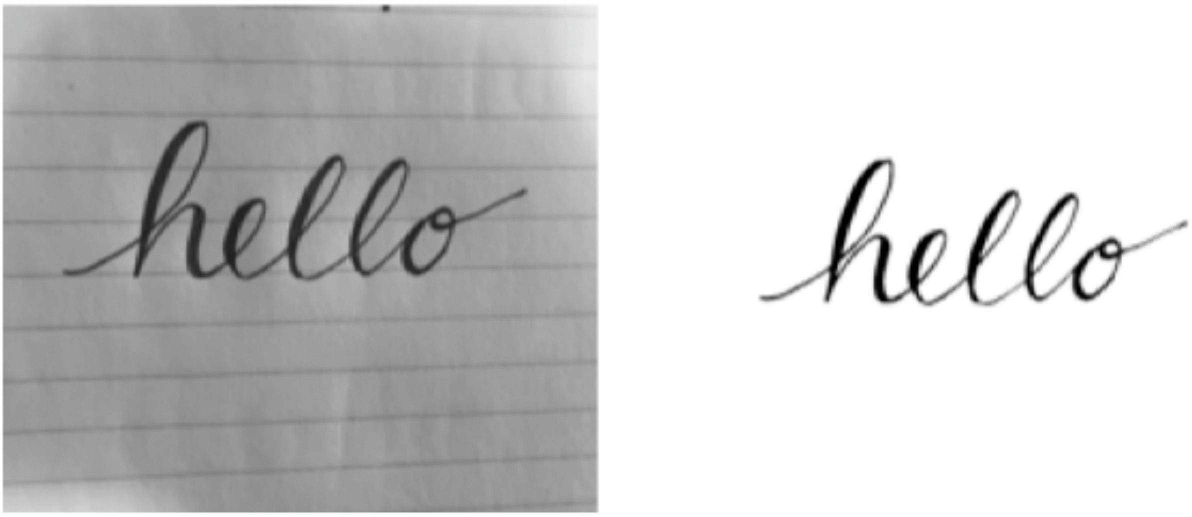

Figure 20: Original image and desired output from preprocessing

5.2.1 IMAGE PROCESSING CONSIDERATIONS.

Isolation of the text region by removing: - Any non-text regions of the image such as the table behind a photographed notepad of text.

Removal of all noise including:
1. Notepad lines that may appear on the image. (line detection)
2. Pen marks showing though thin paper. (smoothing filter, light blurring then thresholding)
3. Shadows on folded paper. (smoothing filter, blurring)
4. Impulse noise/Spotty artifacts due to poor image quality. (smoothing filter, light blurring)

Contrast enhancement.
- Increase of the dynamic range using contrast stretching (Gonzalez and Woods, 2002, p. 85)

Finally to create the binary representation.
- thresholding.

Dealing with low resolution images. Thresholding of low resolution image will can result in (jagged) lines of the pen marks. In addition where a user can crop an image maybe need that the image is scaled up.
- Scaling up of cropped images.

5.2.2 POINT OPERATIONS OPERATIONS.

Point operations simply perform a provided action on a specified pixel.

(Gonzalez and Woods, 2002, p. 77) provides the follow expression.

$$g(x, y) = T[f(x, y)]$$

Where $f(x, y)$ is the input and $g(x, y)$ is the output , with $T[\ldots]$ being the function defined over neighborhood (x, y)

(Gonzalez and Woods, 2002, p. 77)

5.2.2.1 GLOBAL THRESHOLDING/HARD THRESHOLDING.

Global thresholding is the matter of applying the same threshold value to every pixel in the representation.

“The grey level value remains constant over the selected neighborhood”

(Castleman, 1995, p. 452)

(Gonzalez and Woods, 2002, p. 86) provides the follow expression to represent a threshold operation.

$$g(x, y) = \begin{cases} 1 & \textit{if} \quad f(x, y) \quad > \quad T \\ 0 & \textit{if} \quad f(x, y) \quad \leq \quad T \end{cases}$$

From this the following pseudo code can be developed.

```
// Pseudo Code

T = threshold

for each row y in representation
    for each column x in row
    if pixel(x, y) > T then
        pixel(x, y) greylevel = 1
    else
        pixel(x, y) greylevel = 1
```

Figure 21: Threshold function pseudo code

```
- (NSBitmapImageRep *) thresholdWithValue:(int)value
{

    NSBitmapImageRep *output = [self grayScaleRepresentationOfImage:self.image;
    unsigned char *data = [output bitmapData];

    for ( int y = 0; y < self.height; y++ )
    {
        for ( int x = 0; x < self.width; x++ )
        {
            int index = x + (y * self.width);
            if ( data[index] < value) {
                data[index] = 0;
            } else {
                data[index] = 255;
            }
        }
    }

    return output;
}
```

Figure 22: Threshold function implementation