EE5175: Image Signal Processing - Lab 10 report

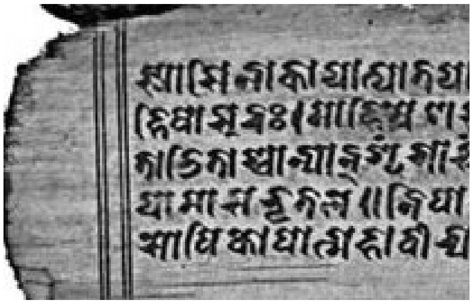
Ruban Vishnu Pandian V (EE19B138)

April 8, 2023

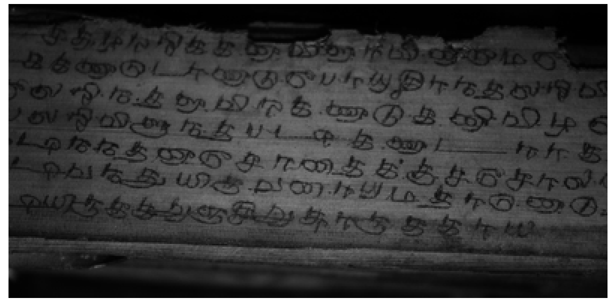
Contents

1 Otsu's thresholding

Aim: To binarize two given grayscale images by finding the optimal threshold using Otsu's thresholding algorithm. The input images `palmleaf1.png` and `palmleaf2.png` are shown below:



(a) Input image 1



(b) Input image 2

2 Otsu's thresholding algorithm:

Let the input image be $f(x, y)$. Let i denote the 8-bit gray level ($i = 0, 1, ..., 255$) and $F(i)$ denote the number of times intensity i appears in the given image, i.e., intensity histogram of the given image. Let t denote the binarization threshold. Let the new image be $g(x, y)$. Then:

$$g(x,y) = \begin{cases} 0 & f(x,y) \leq t \\ 255 & f(x,y) > 0 \end{cases}$$

Otsu's algorithm is a method to find the optimal threshold t. The optimization metric in Otsu's algorithm is the between class variance. With the notations defined above, the following quantities are defined:

$$N = \sum_{i=0}^{255} F(i) \quad (Total\ image\ size)$$

$$N_1 = \sum_{i=0}^{t} F(i) \quad (First\ class\ size)$$

$$\mu_1 = \sum_{i=0}^{t} i\frac{F(i)}{N_1} \quad (First\ class\ mean)$$

$$\sigma_1^2 = \sum_{i=0}^{t} (i - \mu_1)^2 \frac{F(i)}{N_1} \quad (First\ class\ variance)$$

$$N_2 = \sum_{i=t+1}^{255} F(i) \quad (Second\ class\ size)$$

$$\mu_2 = \sum_{i=t+1}^{255} i\frac{F(i)}{N_2} \quad (Second\ class\ mean)$$

$$\sigma_2^2 = \sum_{i=t+1}^{255} (i - \mu_2)^2 \frac{F(i)}{N_2} \quad (Second\ class\ variance)$$

$$\mu_T = \sum_{i=0}^{255} i \frac{F(i)}{N} \quad (Total\ mean)$$

$$\sigma_T^2 = \sum_{i=0}^{255} (i - \mu_T)^2 \frac{F(i)}{N} \quad (Total\ variance)$$

Based on the quantities defined above, the final two important metrics are defined as follows:

$$\sigma_w^2 = \sigma_1^2 \frac{N_1}{N} + \sigma_2^2 \frac{N_2}{N} \quad (Within\ class\ variance)$$

$$\sigma_b^2 = (\mu_1 - \mu_T)^2 \frac{N_1}{N} + (\mu_2 - \mu_T)^2 \frac{N_2}{N} \quad (Between\ class\ variance)$$

It turns out that the following equation is always true:

$$\sigma_b^2 + \sigma_w^2 = \sigma_T^2$$

We want to minimize the within class variance and maximize the between class variance as much as possible for optimal thresholding. Fortunately, because of the above equation, minimizing one quantity will automatically maximize the other and vice-versa (since σ_T^2 is independent of the threshold). Hence, the Otsu algorithm becomes a simple minimization (or maximization) algorithm. In this report, maximization of the between class variance is discussed.

From the definition of between class variance, we can easily show that:

$$\sigma_b^2 = \frac{\mu_1^2 N_1 + \mu_2^2 N_2}{N} - \mu_T^2$$

Let us define the following functions of t:

$$N_1 = \sum_{i=0}^{t} F(i) = p(t)$$

$$\mu_1 N_1 = \sum_{i=0}^{t} iF(i) = q(t)$$

Then we have:

$$\sigma_b^2 = \frac{q(t)^2}{p(t)N} + \frac{(\mu_T N - q(t))^2}{(N - p(t))N} - \mu_T^2$$

From the definitions of $p(t)$ and $q(t)$, we can also show that:

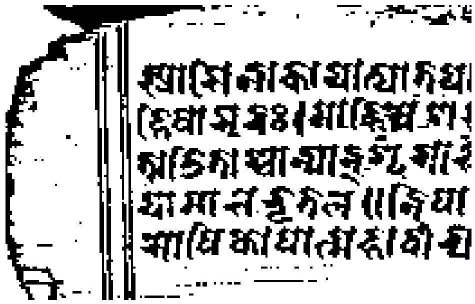
$$p(t+1) = p(t) + F(t+1)$$

$$q(t+1) = q(t) + (t+1)F(t+1)$$

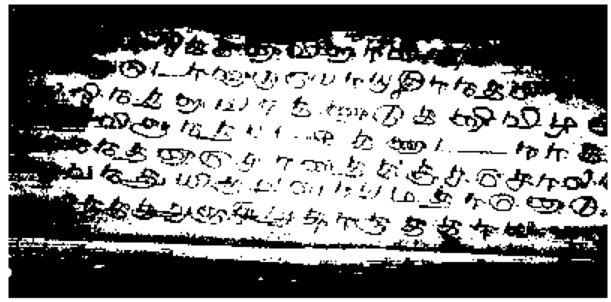
Hence, implementing this maximization method is also easy since we don't need to recompute the class means again. We can use the previously computed values of $p(t)$, $q(t)$ and update them recursively.

3 Results on the given images:

The Otsu algorithm described in the previous section was applied on both the input images and the results are shown below:



(a) Output image 1



(b) Output image 2

The results are convincing. Most of the alphabetical letters can be distinguished clearly from the background. The output of first image looks better than that of the second.

4 Observations and Conclusions

1. The output of first image is better than that of second since it has uniform illumination. The illumination present in the second image is non-uniform (It can be clearly seen that the central portion is brighter than the edges of the second image).

2. To obtain a better binarization for second image, the Otsu's algorithm can be applied locally instead of finding a global threshold. The image can be divided into small windows and Otsu's algorithm can be applied to find a threshold for each window. In this way, the impact of non-uniform illumination can be minimized.