Document-image binarization using modified Sauvola's algorithm

Soumyadeep Dey Pratik Jawanpuria
Emails: {desoumya,prjawanp}@microsoft.com

Abstract

We propose a novel binarization technique that improves upon the Sauvola's algorithm [2] by preserving more foreground information in the binarized document-images. The computational complexity of the proposed algorithm is same as the Sauvola's algorithm, thereby making it suitable even for mobile-computing based applications.

1 Introduction

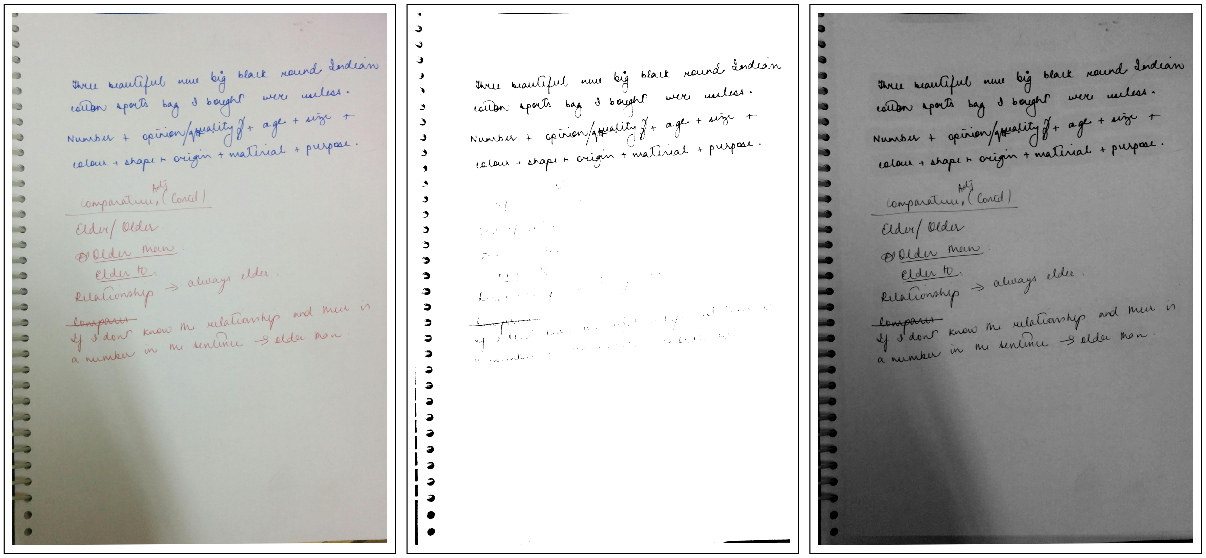
Binarization is a technique to segment foreground from the background pixels. A simple technique for binarization is thresholding of gray-level or color document scanned images. Different segmentation methods related to binarization of document images have been reported in the literature, such as, [1], [2], [3], etc. In this work, we develop a method to segment foreground from the background pixels using modified Sauvola's algorithms [2].

In [2], a threshold is computed for each pixel using the Eq 1, where, for an input image I, $R = \frac{\max(I) - \min(I)}{2}$.

$$T_W(p) = m_W^p \times [1 + k \times (\frac{s_W^p}{R} - 1)] \tag{1}$$

The threshold T is computed for each pixel (p) based on a window W of size $n \times n$ surrounding it, where m_W^p, s_W^p respectively represent mean and standard deviation of W around pixel p, and k lies between $0 \leq k \leq 1$.

Empirically, it has been observed that binarization using the the thresholds obtained from Eq 1 misses foreground pixels in many scenarios. An example of such failure is shown in Figure 1. In the following, we propose an improvement on the binarization technique proposed in [2].



(a) (b) (c)

Figure 1: (a) input image; (b) binary image using [2]. Please observe the loss of foreground information in this binarized image; (c) confidence image using the proposed Eq 2. It should be noted that the confidence image (c) is an intermediate result of our algorithm. For final binarized output, please see Figure 2.

2 Proposed algorithm

We define a confidence value, C, for each pixel p as Eq 2, where $\max(I)$ represent maximum value of any pixel of an input image I.

$$C_W(p) = \begin{cases} \frac{I(p)-T_W(p)}{\max(I)-T_W(p)} & \text{if } I(p) > T_W(p) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

A new image I_c corresponding to an input image I is generated using the confidence value given by the Eq 2. An example of such an image is shown in Figure 1 (c). Later, Eq 1 is used to compute thresholds for the image I_c. These thresholds are further used to generate the new binary image. A few examples of binarization using this new method is shown in Figure 2.

Implementation details: The current implementation of the algorithm is done using the support of python libraries *numpy, skimage,* and *opencv*. Overall algorithm to obtain the binary image is given in Algorithm 1.

Algorithm 1 ModSauvolaBin($I, \tilde{n}, \tilde{k}, \hat{n}, \hat{k}$)

Require: An input image I

Require: An optional parameter $\tilde{n}$. By default $\tilde{n} = 0.05 \times \min(h, w)$, where h, w are height and width of image I

Require: An optional parameter $\tilde{k}$. By default $\tilde{k} = 0.35$

Require: An optional parameter $\hat{n}$. By default $\hat{n} = 0.1 \times \min(h, w)$, where h, w are height and width of image I

Require: An optional parameter $\hat{k}$. By default $\hat{k} = 0.35$

Ensure: A binary image (I_{bin})

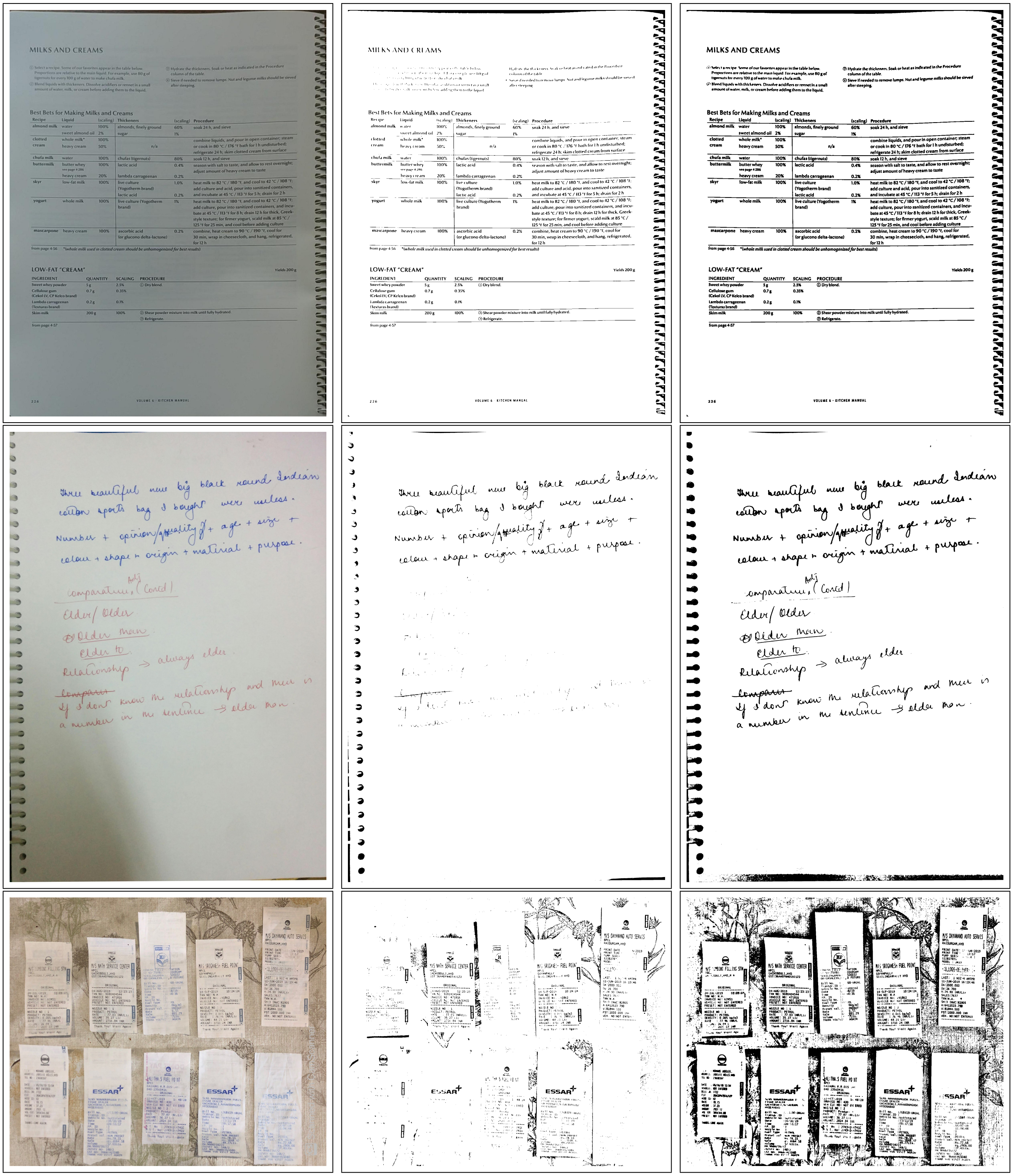
1: $I_g \leftarrow$ gray scale version of I
2: $\widetilde{W} \leftarrow \tilde{n} \times \tilde{n}$
3: $T_{\widetilde{W}} \leftarrow$ Using Eq 1 with $W = \widetilde{W}$ and $k = \tilde{k}$, $\forall p \in I_g$
4: $C_{\widetilde{W}} \leftarrow$ Using Eq 2 with $T_W = T_{\widetilde{W}}$, $\forall p \in I_g$
5: $I_{conf} \leftarrow C_{\widetilde{W}}$
6: $\widehat{W} \leftarrow \hat{n} \times \hat{n}$
7: $T_{\widehat{W}} \leftarrow$ Using Eq 1 with $W = \widehat{W}$ and $k = \hat{k}$, $\forall p \in I_{conf}$
8: $I_{bin} = 0$ if $I_{conf} \leq T_{\widehat{W}}$
9: $I_{bin} = 255$ if $I_{conf} > T_{\widehat{W}}$
10: **return** I_{bin}

References

[1] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernatics*, 9(1):62–66, 1979.

[2] J. Sauvola and M. Pietikäinen. Adaptive document image binarization. *Pattern Recognition*, 33:225–236, 2000.

[3] B. Gatos, I. Pratikakis, and S. J. Perantonis. Adaptive degraded document image binarization. *Pattern Recognition.*, 39(3):317–327, 2006.



(a) (b) (c)

Figure 2: (a) input image; (b) binary image using [2]; (c) binary image using proposed algorithm