Sauvola Method

Youssef Lamzaouak
Ecole des Mines de Saint-Etienne

1 Introduction

Optical Character recognition has been a subject of a variety of reserrach paers during the last ten years. This intense research is due to development of AI and then the necessity to get more informations from documents like passport documents, invoices, bank statements, computerized receipts, business cards and so on. But the problem was that most of the developed algorithms rely on statistical methods, not considering the special nature of document images. However, recent developments on document types, for example documents with mixed text and graphics, call for more specialized binarization techniques. In the current literature ,different algorithms were proposed to deal with different degradation problems ranging from changes in lumination (illumination), scanning errors and resolution, poor quality of the source document to complexity in the document structure and more .we are going to cite here major algorithms problems and what problem they deal with . First,O'Gorman proposes a global approach calculated from a measure of local connectivity information.Second, Yang et al came out with a thresholding algorithm that uses a statistical measurement, called 'largest static state difference'. Rosenfeld and Smith presented a global thresholding algorithm to deal with noise problem using an iterative probabilistic model when separating background and object pixels.And then Perez and Gonzalezdesigned an algorithm to manage situations where imperfect illumination occurs in an image.

2 The authors approach:

The authors approach is innovative as it deals with complex and diverse documents content.First, a fast classification is done to distinguish picture parts from text parts.Then two algorithms are proposed to deal with each category;the Soft decision Method(SDM) which includes noise and signal tracking capabilities and text binarization method(TBM) which is used to separate text component from bad backround(dark regions ,to luminant regions and transparent texts etc. . .).We are not going to get in details of SDM ,our work here will focus on just TBM .But before dealing with TBM ,we should talk about Niblack as the authors method were inspired by him.So the Niblack algorithm is an algorithm that determines a threshold value to each pixel-wise by sliding a rectangular window over the gray level image . The size of the rectangle window may differ. The threshold is calculated based on the local mean m and the standard deviation S of all the pixels in the window and is given by the following derivation: Tniblack=m+k*S where T represent the threshold value, m is the average value of the pixels pi, and k is fixed depends upon the noise still live on the background it may be.the problem is that this method does not work well for cases in which the background contains light texture as the grey values of these unwanted details easily exceed threshold values.the authors modification,is that the threshold is computed with the dynamic range of standard deviation, R. Furthermore,the local mean is utilized to multiply terms R and a fixed positive fixed value k. This has the effect* *of amplifying the contributionof standard deviation in an adaptive manner as shows the formula : T=m* [1+k*((S/R)-1)]

[13]:
```
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
import skimage
import os
from skimage.filters import (threshold_otsu, threshold_niblack)
```

[14]:
```
os.getcwd()
os.chdir('C:\\Users\\hp\\Desktop\\imagestoolbox\\')

Image=skimage.io.imread('CapT.PNG')
```

[15]:
```
def Sauvola_Method(Image,Wind_size,R=128,k=0.5):
    imsize=Image.shape
    if(len(imsize)==3):
        Image=skimage.color.rgb2gray(Image)
        imsize=Image.shape
    Mask=np.zeros(imsize)
    nrows=imsize[0]
    ncols=imsize[1]

    #detrminig the mean and the sd for each pixel
    half_wind=Wind_size//2
    for i in range(half_wind,nrows):
        for j in range(half_wind,ncols):
            Window=Image[(i-half_wind):(i+half_wind),(j-half_wind):(j+half_wind)]
            mean=np.mean(Window)
            std=np.std(Window)
            Mask[i][j]=mean*(1 + k * ((std / R) - 1))
    return Mask
```

[16]:
```
matplotlib.rcParams['font.size'] = 9


image = skimage.color.rgb2gray(Image)
binary_global = image > threshold_otsu(image)


window_size = 25
thresh_niblack = threshold_niblack(image, window_size=window_size, k=0.5)
binary_niblack = image > thresh_niblack
Mask=Sauvola_Method(image,25)
```

```
I=image>Mask
```

[17]:

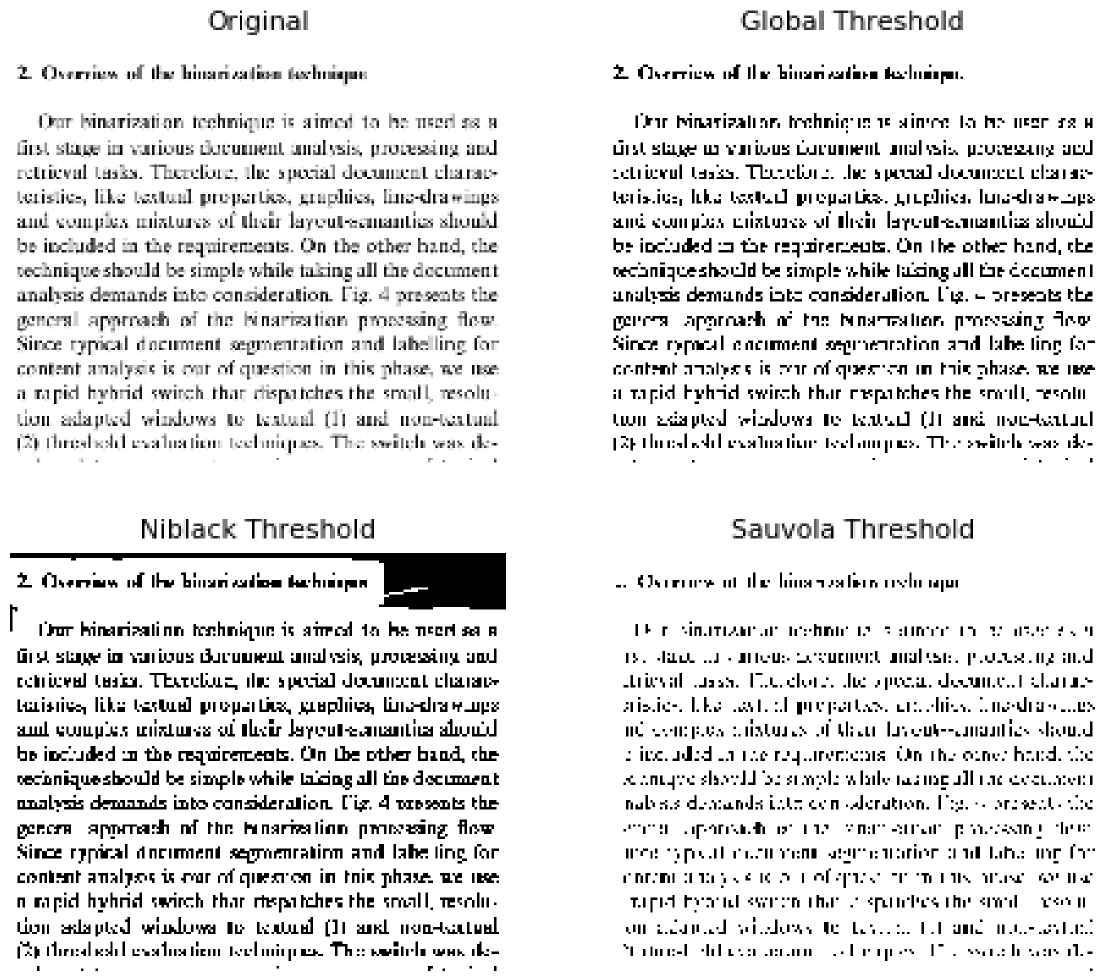
```
plt.figure(figsize=(8, 7))
plt.subplot(2, 2, 1)
plt.imshow(image, cmap=plt.cm.gray)
plt.title('Original')
plt.axis('off')

plt.subplot(2, 2, 2)
plt.title('Global Threshold')
plt.imshow(binary_global, cmap=plt.cm.gray)
plt.axis('off')

plt.subplot(2, 2, 3)
plt.imshow(binary_niblack, cmap=plt.cm.gray)
plt.title('Niblack Threshold')
plt.axis('off')

plt.subplot(2, 2, 4)
plt.imshow(I, cmap=plt.cm.gray)
plt.title('Sauvola Threshold')
plt.axis('off')

plt.show()
```


Original
Global Threshold
Niblack Threshold
Sauvola Threshold

2. Overview of the binarization technique

Our binarization technique is aimed to be used as a first stage in various document analysis, processing and retrieval tasks. Therefore, the special document characteristics, like textual properties, graphics, line-drawings and complex mixtures of their layout-semantics should be included in the requirements. On the other hand, the technique should be simple while taking all the document analysis demands into consideration. Fig. 4 presents the general approach of the binarization processing flow. Since typical document segmentation and labelling for content analysis is out of question in this phase, we use a rapid hybrid switch that dispatches the small, resolution adapted windows to textual (1) and non-textual (2) threshold evaluation techniques. The switch was de-

3 Overall Results and Commments:

To test an allgoritm efficiency lot of criterias can be involved ,in our case we are going to compare this algorithms on the following criterias:noise impact ,illumination ,time needed for computing and object shape preservation. 1-Sauvola algorithm and all others had a tolerable time needed for computing so there is not much added value on this aspect. 2-For object shape preservation, the proposed algorithm behaves robustly compared to other techniques. Since most of the pixels in synthetic images are judged by the soft control method, the threshold between objects and non-object candidates seems very clear. 3-Concerning noise,all algorithms have shown a good behaviour since they kept an overall performance not inferior than 80% for 20% noise penetration,but here Sauvola algorithm outperforme its peers as he maintained this performance value until a 40% penetration rate. 4-For illumination ,the proposed algorithm is still better and more this is the reason why the authors modified the Niblack one,it is because Niblack does not work well for cases in which the background contains light texture as the grey values of these unwanted details easily exceed threshold value (And we could see this clearly in our example since some black stains appeared in Niblack and are not present in Sauvola's one.) Finally,we could say that the proposed algorithm hybrid approach seems to be efficient dealing with different defect types and even the severe one as it performed well against all comparison techniques.