20115621-buidinhhanhdu-lap7

October 14, 2023

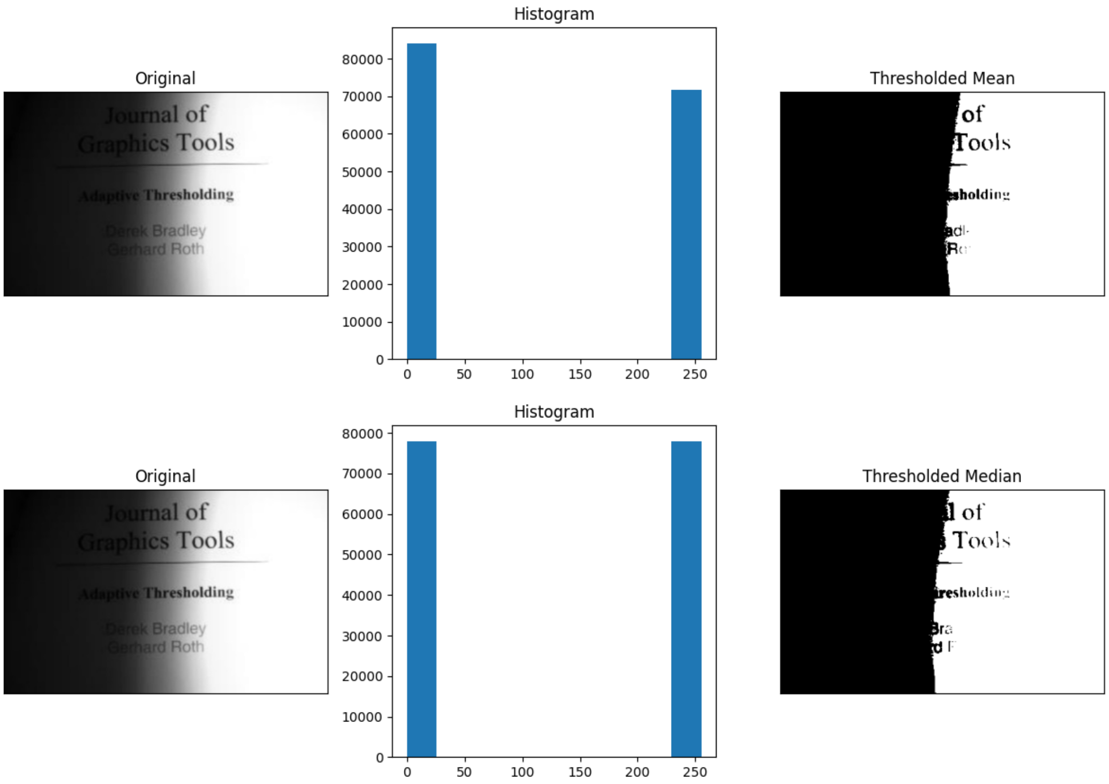
[127]:
```
#Libaries used
import cv2
import matplotlib.pyplot as plt
import numpy as np
```

[128]:
```
#Load an image
image = cv2.imread('im1.jpg', cv2.IMREAD_GRAYSCALE)
```

Lab Task 1: Write a program that threshold the provided image using global mean and median.

[129]:
```
def threshold_global_mean(image):
    mean_value = np.mean(image) # Calculate the mean of the image
    _, thresholded_image = cv2.threshold(image, mean_value, 255, cv2.
 ↪THRESH_BINARY) # Calculate the threshold
    return thresholded_image

def threshold_global_median(image):
    median_value = np.median(image) # Calculate the median of the image
    _, thresholded_image = cv2.threshold(image, median_value, 255, cv2.
 ↪THRESH_BINARY) # Calculate the threshold
    return thresholded_image

# Threshold using global mean
thresholded_mean = threshold_global_mean(image)

# Threshold using global median
thresholded_median = threshold_global_median(image)
```

[130]:
```
#Show results
plt.figure(figsize= (15, 10))

plt.subplot(231),plt.imshow(image, cmap= 'gray'),plt.title('Original')
plt.xticks([]), plt.yticks([])


plt.subplot(232),plt.hist(thresholded_mean.ravel()), plt.title('Histogram')
plt.subplot(233),plt.imshow(thresholded_mean, cmap= 'gray'), plt.
 ↪title('Thresholded Mean')
plt.xticks([]), plt.yticks([])

plt.subplot(234),plt.imshow(image, cmap= 'gray'),plt.title('Original')
plt.xticks([]), plt.yticks([])


plt.subplot(235),plt.hist(thresholded_median.ravel()), plt.title('Histogram')


plt.subplot(236),plt.imshow(thresholded_median, cmap= 'gray'), plt.
 ↪title('Thresholded Median')
plt.xticks([]), plt.yticks([])

plt.show()
```



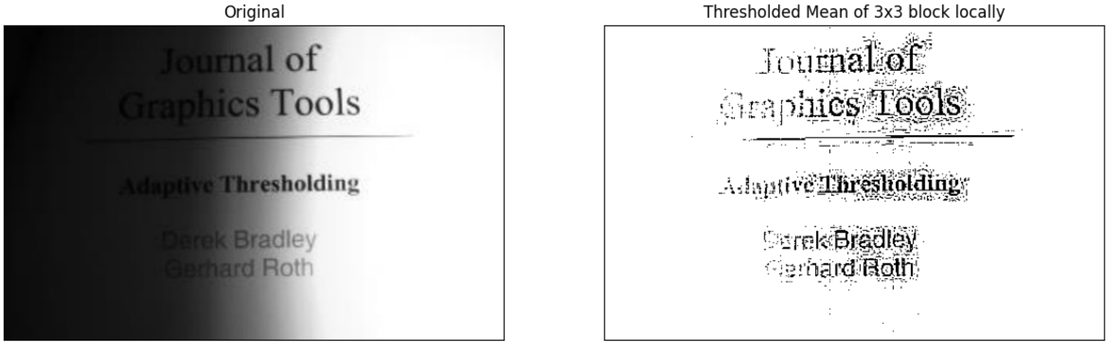
Lab Task 2: Now threshold the image by taking threshold value mean of 3x3 block locall

[131]:
```
kernel_size = 3 # Set the size of the kernel

#Threshold using local mean of 3x3 block
```

```
thresholded_local_mean = cv2.adaptiveThreshold(image, 255, cv2.
  ↪ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY, kernel_size, 2)
```

[132]:

```
#show results
plt.figure(figsize= (15, 5))

plt.subplot(121),plt.imshow(image, cmap= 'gray'),plt.title('Original')
plt.xticks([]), plt.yticks([])


plt.subplot(122),plt.imshow(thresholded_local_mean, cmap= 'gray'), plt.
  ↪title('Thresholded Mean of 3x3 block locally')
plt.xticks([]), plt.yticks([])

plt.show()
```



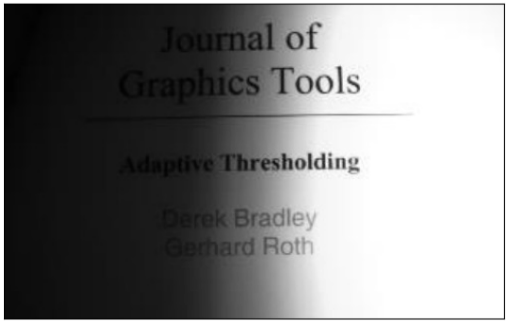
Lab Task 3: Implement global adaptive thresholding algorithm to threshold given image

[133]:

```
#Threshold global
thresholded_global = cv2.adaptiveThreshold(image, 255, cv2.
  ↪ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 11, 2)
```

[134]:

```
#Show results
plt.figure(figsize= (15, 5))

plt.subplot(121),plt.imshow(image, cmap= 'gray'),plt.title('Original')
plt.xticks([]), plt.yticks([])


plt.subplot(122),plt.imshow(thresholded_global, cmap= 'gray'), plt.
  ↪title('Thresholded Global')
plt.xticks([]), plt.yticks([])

plt.show()
```

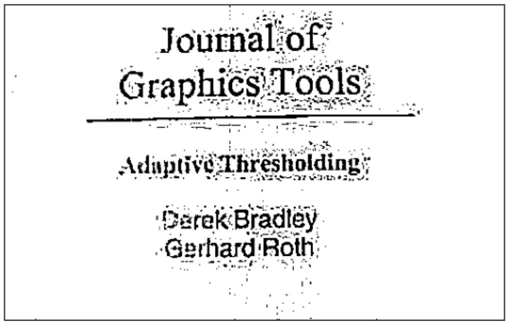


Original

Journal of
Graphics Tools

Adaptive Thresholding

Derek Bradley
Gerhard Roth



Thresholded Global

Journal of
Graphics Tools

Adaptive Thresholding

Derek Bradley
Gerhard Roth