COL780 : Computer Vision

Assignment 1

Automated Evaluation of Micro-sutures

Submitted By : Manish Kumar

Entry No. : 2023MCS2497

Introduction

Micro-Suturing involves the intricate task of performing suturing (stitching) under a microscope. In this assignment we've tried to automate the evaluation of effectiveness of the micro-suturing performed on the basis of some key parameters such as :

1. Number of sutures
2. Inter-suture distances
3. Angle made by suture

Tasks :

Given a set of images, our objective is to process them systematically to extract valuable information. Subsequently, utilize this processed data to judge the effectiveness of micro-suturing.

Task I : Number of micro-sutures

Image Processing Pipeline:

a) Reading and opening image

All images from the specified directory (img_dir) were read and stored in a list. The function **`open_files()`** filtered out non-string filenames to ensure data integrity.

b) Gaussian blurring for noise reduction

A Gaussian blur was applied to each image to reduce noise and smoothen the details. This helped in preparing the image for subsequent processing steps.

Parameters involved :

kernel_size: Size of the Gaussian kernel for blurring (3x3 in our case).

sigma: Standard deviation of the Gaussian distribution (2.0 in our case).

c) Image sharpening to enhance sutures

A sharpened version of the image was obtained by subtracting a blurred version from the original image. I.e, **alpha * (org_img) - beta * (blurred_img)**

This process helped enhance edges and fine details, making sutures more distinguishable.

Parameters involved :

alpha: Sharpening factor.

beta: Blurring factor.

d) Color to grayscale conversion

The RGB images were then converted to grayscale (single channel), since sutures' color information isn't crucial for analysis. Working in grayscale simplifies computations and reduces the dimensionality of the data.

e) Thresholding and inversion for sutures isolation.

A threshold(**value = 120**) was applied to the grayscale image to create a **binary image**, emphasizing sutures against the background.

The image was inverted to highlight suture regions.

f) Morphological operations (`opening`) for further refinement

Morphological opening operations were performed on the inverted binary image.

This involved erosion followed by dilation, which helps in removing small noise and refining the suture regions

g) Suture Counting Algorithm

We've counted the number of sutures in the binary image using **contour analysis**. Also, we're identifying centroids and angles for further analysis.

Steps:

Find **contours** in the binary image.

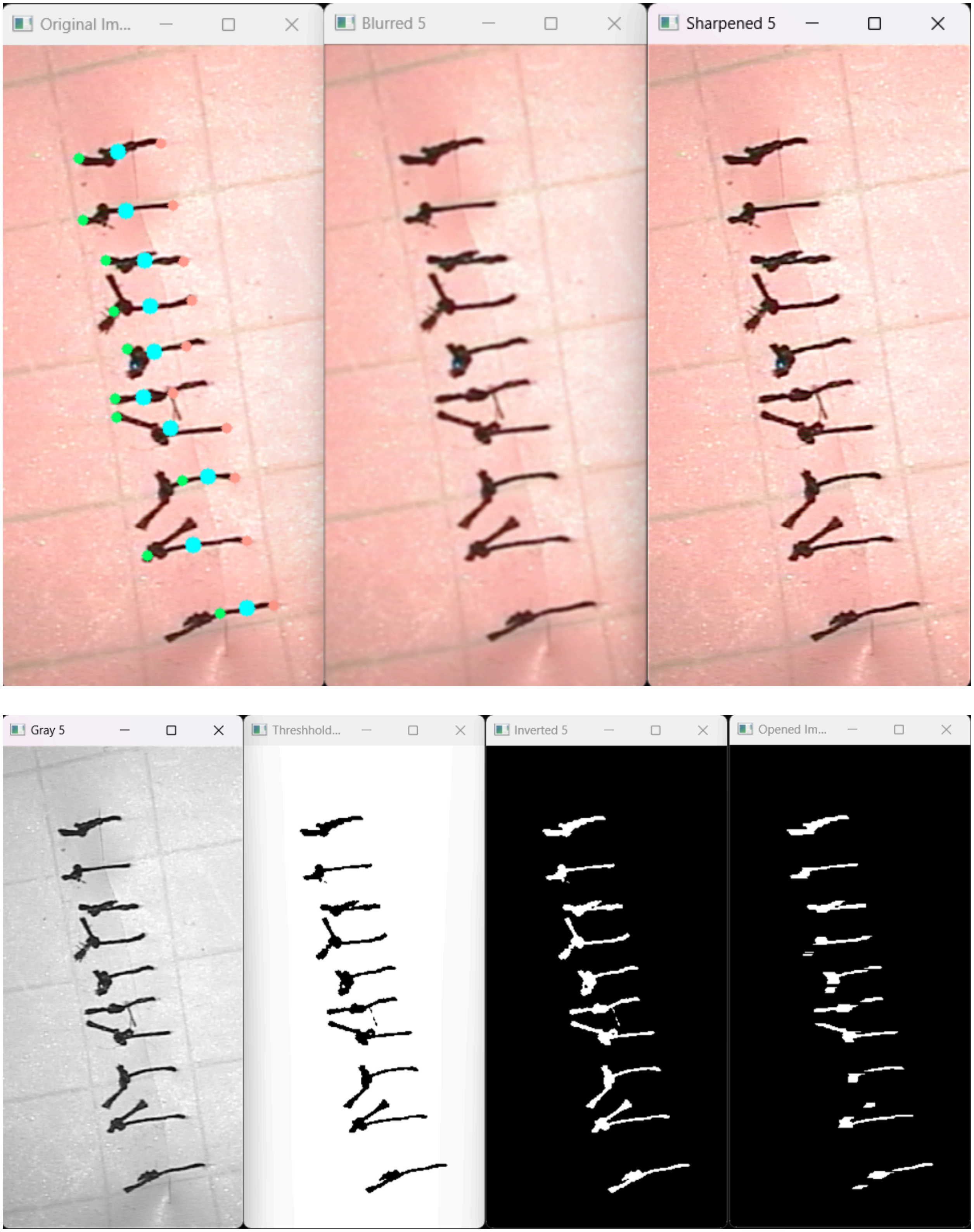
Calculate **contour lengths** and determine an average and median contour length.

Count sutures based on a threshold (e.g., **average contour length - 20**).

Mark **leftmost, rightmost,** and **centroid** points on the original image.

Calculate **angles** between the leftmost point and the centroid for each suture.

Sample Image & its Processing details :



Output :

For the above image, we've obtained the following results :

```
C:\Users\manis\OneDrive\Desktop\CV\Assignment1\practice>python3 final_def.py
[SUCCESS]FOund 10 image(s)...Ready to process!

[LOG]Applied Gaussion Blur to all images!
[LOG]Sharpened all images!
[LOG]Converted all images to Single Channel!
[LOG]Converted all images to Binary images!

[LOG]Inverted all images!
[LOG]Applied Errosion & Dilation to -ve images!

[SUCCESS]Following are the Sutures count :
Image 5: 10
```

<u>Task II</u> : Inter-Suture Spacing

After identifying the sutures in the images, we'll save their leftmost, rightmost points &

Centroid points by extracting contours. The centroid is considered as the representative point for each suture.

We're then using the Euclidean distance formula to calc. Distance between 2 consecutive contours/sutures using the numpy fn. **`np.linalg.norm()`**

Lastly, the result distances, measured in pixels, are then used to calculate mean and the variance. These statistical measures offer insight into the distribution & arrangement of sutures under microscopic images.

Task III : Angulation of the Suture

The angle calculation involves determining the orientation of each suture relative to its centroid. For this, we've taken the **leftmost** point of all sutures as the reference point.

The **angle** of the suture is then computed using the arctangent function **`np.arctan2()`**, resulting in values stored in **radians.**

i.e.,

```
angle = np.arctan2(leftmost[y] - centroid[y], leftmost[x] - centroid[x])
```

where, the leftmost point (x,y) & centroid(x,y) are specific to a particular suture.

Summary

In this image processing pipeline for the evaluation of micro-suturing effectiveness, we have implemented a series of key steps to extract meaningful parameters from a set of images. The obtained parameters, including **`suture count`**, **`inter-suture distances`**, `and` **`suture angles`**, serve as quantitative measures for evaluating micro-suturing effectiveness.

Lastly, to facilitate record-keeping and further analysis, the obtained results are meticulously stored in the 'output.csv' file.