COL780: Computer Vision

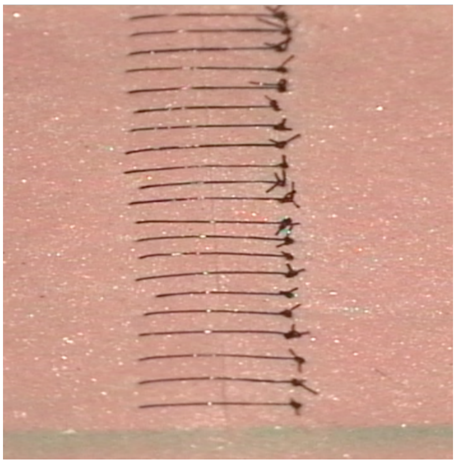
Assignment-1: Automated Evaluation of Micro-sutures

Deadline: $5^{th} February, 2024$

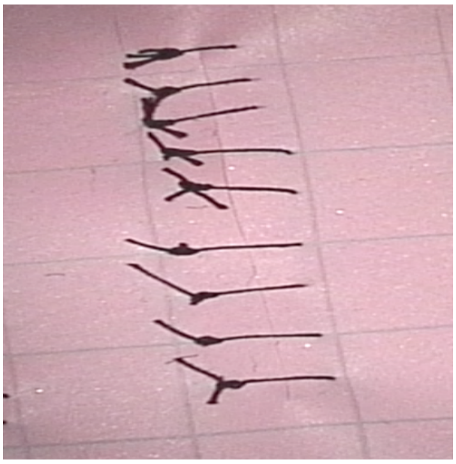
1 Introduction

Micro-Suturing is a task of performing suturing (stitching) under a microscope. It is an indispensable neurosurgical skill. Automated evaluation of micro-suturing would lead to efficient, fast and unbiased training of a trainee neurosurgeon. We propose to evaluate the effectiveness (final image) of micro-suturing using various parameters. In this assignment, you will calculate three parameters of the evaluation using the classical computer vision algorithms. Namely,

- **(a)** Number of sutures
- **(b)** Inter-suture distance
- **(c)** Angulation of the suture



(a) Image-1



(b) Image-2

Figure 1: Sample images of microsutures from the dataset. Image-1 shows a good micro-suturing outcome, where as Image-2 is a relatively bad outcome.

2 Implementation Tasks

Various algorithms that have been discussed in the class can be implemented in order to quantitatively perform the below mentioned tasks for a given image. Some examples include Canny edge detection, Harris corner detection, Contour hunting, etc. The tasks to be implemented are as follows:

2.1 Task-1: Number of micro-sutures

Having a count on the number of micro-sutures in an image can prove to be an important metric in assessing the quality of the surgery. For example, in Image-2, there are 9 micro-sutures in total. Counting the number of sutures in a micro-suturing image involves applying computer vision algorithms to detect and recognize individual sutures. One way to estimate number of sutures is as follows:

- **Preprocessing:** Enhance the image quality by performing operations such as resizing, contrast adjustment, and noise reduction to improve the effectiveness of subsequent algorithms. An important pre-processing step for our application would be to perform **thresholding** in order to effectively separate the background from the sutures.
- **Edge Detection:** Apply edge detection algorithms to highlight the boundaries of sutures. **Canny edge detection** or other suitable edge detection techniques can be used to identify the edges of individual sutures.
- **Filtering and Refinement:** Apply filters or criteria to distinguish sutures from other object details in the image. This may involve filtering based on size, shape, or other characteristics specific to the sutures in the micro-suturing image.
- **Suture Counting:** Once the relevant edges representing sutures are identified, we can have a good estimate on the total number of sutures in the image. You can estimate suture boundaries by considering the **connected components** of the edge pixels. You can use a connected components algorithm or a region-growing approach to group neighboring edge pixels into clusters, each representing a potential suture. Alternatively, you may use **Harris corner detection** to count the number of corners and subsequently, number of sutures can be found out by dividing the number of corners by a fixed constant.

Note: You may need to fine-tune parameters such as the Canny edge detection thresholds and other settings based on the characteristics of the micro-suturing images. Additionally, consider using color information or other features if grayscale edge detection is not sufficient for your specific images. You are free to experiment with any other approaches as well. There may be other approaches to estimate the sutures, for example, using contour detection. Mention the approach that you have used along with some experimentation details in the detailed report that needs to be submitted.

2.2 Task-2: Inter-Suture Spacing

In micro-suturing, suture spacing refers to the distance between individual sutures or stitches made during a surgical procedure involving small or delicate tissues. The **mean and variance of inter-suture distance** (relative to the height of image) are important metrics to measure in order to assess the suturing done by the neurosurgical trainee.

After we have estimated sutures in Task-1, one way to estimate the inter-suture spacing by **calculating centroids** of the sutures as follows: For each connected component, calculate the centroid, and then compute the distances between consecutive centroids to measure the inter-suture spacing.

Note: You should visualize the calculated distances and centroids on the original image. This approach is based on the assumption that connected components obtained from the edges from Canny detection reasonably represent the suture boundaries. In case you have followed any other approach, mention appropriate experimentation details in the report.

2.3 Task-3: Angulation of the suture

Ideally, we would like have sutures parallel to the x-axis. However, in reality, different sutures may be at different angles to the x-axis. Also, more the alignment amongst the sutures, better the surgery. Hence, we should also consider the **mean and variance of the angle formed** by various sutures and the x-axis in order to assess the performance of the trainee.

One way to find the angulation of the sutures and assess their alignment with the x-axis is as follows: Once we have found the centroid for each suture in Task-2, we can calculate the angle formed by the **line connecting the centroid to a reference point** (e.g., the leftmost point in the image) and the x-axis.

2.4 Task-4: Comparison of two micro-suturing outcomes

Once we are able to extract the above features from the image, we can quantitatively compare two micro-suturing outcomes on each feature, and conclude which image is better amongst the two with respect to that feature. This will help us guide if our implementation is good enough or not.

For this task, you will be given multiple pair of images, and for each pair, you need to output which image is "better" with respect to both features (inter-suture distance and angulation of sutures) individually. **Better image is one which has lower variance with respect to that feature.**

3 Dataset and Evaluation

- **Dataset:** We shall be using a subset of a large micro-suturing dataset (more information regarding the dataset can be found here). You can find the dataset to experiment on inside the `data` directory, wherein you have been provided with ten sample images to test your implementation. We have a separate held-out test dataset (which would be similar to the sample images given). We shall test the performance of your algorithms on that dataset.

- Entire code must be submitted in the form of python files, along with a `main.py` which will be executed in two parts during evaluation/demo as follows:

  ```
  python3 main.py <part_id> <img_dir> <output_csv>
  ```

  There are two tasks:

  - For the first part, you are required to iterate through all the images in the given directory and generate a `csv` file specifying the various parameters that you have calculated. It should have the following columns : `image_name, number of sutures, mean inter suture spacing, variance of inter suture spacing, mean suture angle wrt x-axis, variance of suture angle wrt x-axis`. `main.py` will be executed as follows:

    ```
    python3 main.py 1 <img_dir> <output_csv>
    ```

- For the second part, you will be given pairs of images and you need to output which image is a better suture outcome with respect to inter-suture distance and angulation of sutures. `main.py` will be executed as follows:

  ```
  python3 main.py 2 <input_csv> <output_csv>
  ```

  The `input_csv` file has the following two columns: `img1_path, img2_path`, where as `output_csv` should have the following four columns: `img1_path, img2_path, output_distance, output_angle`, where `output_distance` and `output_angle` take the values either 1 or 2, depending upon which image is better with respect to that feature.

  A sample `output.csv` file will been provided later for this part which has ground truth labels in the `output` column for some of all possible pairs of images in `data` directory. Note that these have been manually labeled by experts.

4 Note

- You can use the inbuilt opencv functions to read and write the images and for visualization purposes only. You **CANNOT** use any other inbuilt image processing functions unless explicitly exempted in the description of the assignment.
- Entire code must be submitted in the form of python files, along with a `main.py` which will be executed during evaluation/demo as explained earlier.
- Grading rubrik (out of 100 marks):

Task	Marks
Task-1	40
Task-2	25
Task-3	15
Task-4	10
Report	10

Table 1: Evaluation Criteria

- You are required to submit the code, data, and a detailed report for this assignment and zip the contents with file name `ENTRY_NUMBER.zip` (For example: `2019CS50768.zip`). The submission can be done using Moodle. Any update in the submission instructions will be shared later.
- This is an individual assignment. Mention the names of all the students you have discussed with in the report. Any sort of plagiarism will lead to zero in the assignment straightaway and stricter punishment depending on the case.