# AUTOMATION OF QUALITY CONTROL PROCESS IN MOUSE BRAIN ARCHITECTURE PROJECT

*A Project Report*

*submitted by*

**SNEHA REDDY AENUGU**

*in partial fulfilment of the requirements*
*for the award of the degree of*

**BACHELOR OF TECHNOLOGY & MASTER OF TECHNOLOGY**

**DEPARTMENT OF ELECTRICAL ENGINEERING**
**INDIAN INSTITUTE OF TECHNOLOGY MADRAS.**

**MAY 2016**

# THESIS CERTIFICATE

This is to certify that the thesis titled **AUTOMATION OF QUALITY CONTROL PROCESS IN MOUSE BRAIN ARCHITECTURE PROJECT SUBMITTED TO IIT-M**, submitted by **Sneha Reddy Aenugu**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor of Technology** and **Master of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

**Prof.Partha Mitra**
Research Guide
Professor
Cold Spring Harbor Laboratory
New York, 11724.

**Prof.Kaushik Mitra**
Co-Guide
Assistant Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: 04-05-2016

# ACKNOWLEDGEMENTS

I would like to thank Prof. Partha Mitra for giving this opportunity to work on the Mouse Brain Architecture Project, which is an incredible learning experience. Next I would like to thank Center for Computational Brain Research (CCBR), IIT Madras, for funding my stay in CSHL (Cold Spring Harbor Laboratory), New York, which helped me to interact personally with the Mouse Brain Architecture project team. I would then like to thank Dr. Daniel Ferrante for acting as a bridge between me and CSHL, helping me with valuable feedback.

I personally acknowledge the guidance and feedback provided by Prof. Kaushik Mitra during the whole process. The weekly meetings have provided me a overview of the work being done by my peers and introduced me to a plethora of interesting problem statements. The lab space provided to me was invaluable as it served as my home. In this regard, I would like to extend my gratitude towards my lab mates who have been particularly accommodating. Specifically I'd like to thank Aditi Singh, who is working under the same project as me, for her company in times of need. I'd also like to thank Jayashree with whom I'd been sharing the work compute, for being so accommodating. I would like to thank Prof. Balaram Ravindran for his feedback at the beginning of the project and his words of wisdom.

I would finally like to thank IIT Madras for providing the wonderful opportunities throughout the course of my study and the faculty of Electrical Engineering for inspiring me.

# ABSTRACT

KEYWORDS: Mouse Brain Architecture; Watershed Segmentation; Concavity analysis.

Mouse Brain Architecture Project aims to generate Brain-wide maps of neural connectivity in amouse. To achieve this, the brain of the mouse is injected with viral tracers to stain the neural pathways, which is then sectioned and scanned to generate a data set of stained images.

Biological quality control step ensures that a given brain image is scientifically significant. Tracer volume, stained cell body count and intensity of the tracer are some of the factors considered to quantify the quality of an image. Automation of this process is very essential considering the tedious nature of manual technique and the size of the data set involved. Here focus is placed on the estimation of injection volume and the count of stained cell bodies for images of fluorescent and histochemical imaging techniques.

To estimate the stained cell body count, Image Segmentation techniques like Watershed Segmentation and Concavity point based Segmentation are employed. Concavity based method will be employed finally because of its optimal performance in detecting the cells. For the volume estimation, Pixel Classification techniques are employed to segment out the injection site. Decision trees are found to be an optimal method midway between crude thresholding and computationally intensive Support Vector Machines.

# TABLE OF CONTENTS

# LIST OF FIGURES

# ABBREVIATIONS

| | |
|---|---|
| **AAV** | Adeno-associated Virus |
| **CTB** | Cholera Toxin B |
| **SVM** | Support Vector Machines |

# CHAPTER 1

# INTRODUCTION

## 1.1 Mouse Brain Architecture Project

The Mouse Brain Architecture Project is an ongoing effort to assemble and integrate information about connectivity in the mouse brain. The project aims to deliver the brain-wide maps of inter-regional neuronal connectivity. These maps, will thus specify the input and output regions of major brain region at a mesoscopic level of analysis. The importance of such a systematic effort to map neural connectivity brain-wide, starting with the mouse, is elucidated in (Boland *et al.*, 2009).

Viral tracers are used to determine the inputs and outputs of each brain region. The inputs and the outputs of a brain region are identified by injecting it with viral tracers. The injected virus traces the neural pathways connecting one brain region to the other. Once the tracer spread sufficiently, the extent can be measured either by fluorescence (for dyes) or immunohistochemistry (for biological tracers). To determine the outputs of the brain region (i.e the regions it projects to), it is injected with a small amount of anterograde tracer, which is taken up by the neurons locally and transported down the axons to the target regions. To determine the inputs to the same brain region, a retrograde tracer, which is taken up by the axonal terminals and transported to neuronal cell bodies, is injected into the same location.

Viral tracers like Adeno-associated virus (AAV) and Cholera Toxin B (CTB) are used in the project. AAV tracers express in two types, green fluorescent protein and red fluorescent protein and is used for anterograde tracing (NL and B, 1998).

### 1.1.1 Process Overview

Mice are screened to match specific age and weight requirements. Each selected animal then receives injection of a neuronal tracer (classical or viral) into a predetermined brain region. After the incubation period (tracer specific, but typically 3-21 days), the animal is perfused, and the brain is extracted and frozen in a customized mold. The extracted brain is now sectioned, suitably stained, and each section digitally imaged.

The 2-D images generated from various slices of the brain are then subsequently registered to a common reference atlas. Each injection is placed in a different mouse and the resulting 3D image volumes co-registered in order to obtain the connectivity atlas. The systematic grid of injections spanning the brain is used to generate a brain wide connectivity map.

The images which are generated as mentioned before are stored in different formats such as lossless JPEG2000, lossy JPEG2000, TIFF. The resolution of lossless JPEG2000 images is 18000 x 24000 px.

## 1.2 Quality Control Process

Biological quality control step ensures whether the given data set of images is scientifically significant. This part of the process requires a holistic view of the entire set of brain images to make a decision whether the given dataset is acceptable for further processing or not. The most common parameters for such a judgement are the following

1. **Tissue quality:** In some cases, there are damages in tissues either in the process of extracting the brain from the skull or in the freezing process. Such damages render the data set unacceptable.
2. **Stain quality:** This parameter indicates the quality of the stain i,e, its clarity in the images so as to gather some useful information out of the images.
3. **Tracer:** This quality control ensures that the tracer flows properly in the brain without any leakages.
4. **Injection Precision:** There are cases where the tracer was not injected in the correct portion of the brain. In those cases the data gathered would be insignificant.

Currently in the Mouse Brain Architecture Project, the Quality Control is being done manually. The aim of this project is to automate two of those features, Stain quality and Injection precision. For the estimation of stain quality, we need to know the count of the stained cell bodies and the volume of the tracer. For the injection precision, we need to know the coordinates of the injection. These two problems will be tackled in this thesis.

## 1.3 Chapter layout

Chapter 2 describes the problem statement and states what needs to be achieved in the project. The problem of cell body detection and volume estimation through Image Segmentation is described.

Chapters 3 and 4 deal with the problem of cell body detection through Image Segmentation. Chapter 4 describes Watershed segmentation and its variant Marker-based Watershed Segmentation for solving the problem. Watershed Segmentation results in over-segmentation whereas its variant fails to detect few cells as foreground objects. To overcome these shortcomings the second technique, Concave point analysis is proposed. Chapter 5 describes the Concavity-based Segmentation to achieve the same goal.

Chapter 5 deals with the problem of tracer volume estimation and detection of injection site. Pixel classification methods like SVM, Decision trees are employed to segment the tracer portion in each of the images and thus volume is estimated.

Chapter 6 concludes the report with directions for future work.

# CHAPTER 2

# PROBLEM DESCRIPTION

In the previous chapter, a few features which indicate the quality of tracer are specified. Two of them are chosen and are automated in this product. One of them is the volume of the injection site and the other is the stained cell body count.

## 2.1 AAV tracer Images

AAV (Adeno-associated Virus) is a viral tracer which is obtained in two principle forms, green fluorescent protein and red fluorescent protein. Example of such a tracer image is shown in (2.1). The task is to determine the volume of the injection site in a brain which is imaged by 256 such slices.

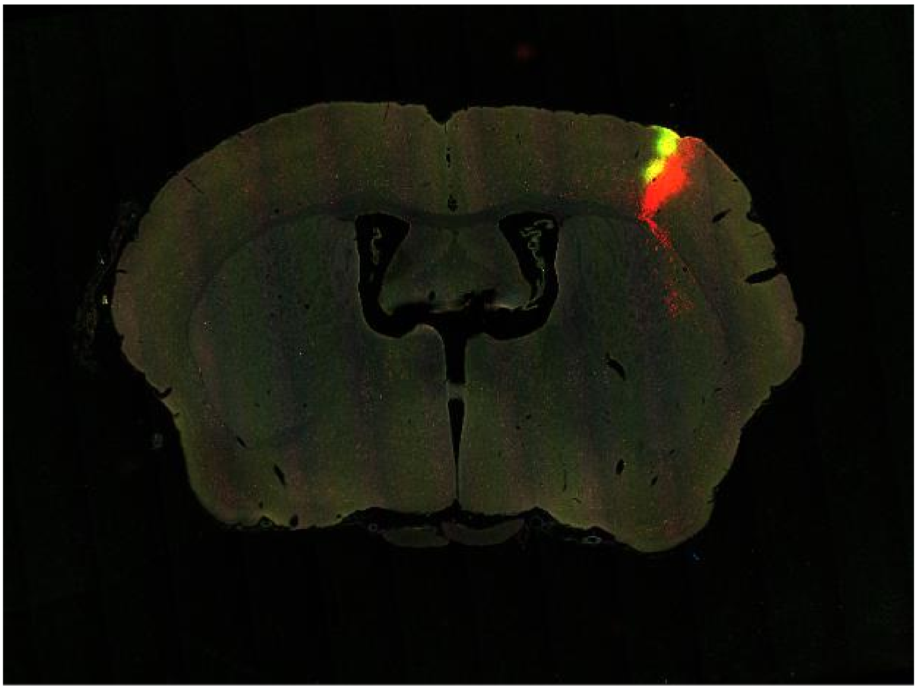


Figure 2.1: AAV tracer image

The second task is to determine the stained cell body count in such images. As the tracer is saturated in the image, it needs to be filtered appropriately and segmented to determine the cell body count. The unfiltered image is shown in (2.2) and the filtered image with cell bodies brought into focus are shown in (2.3).

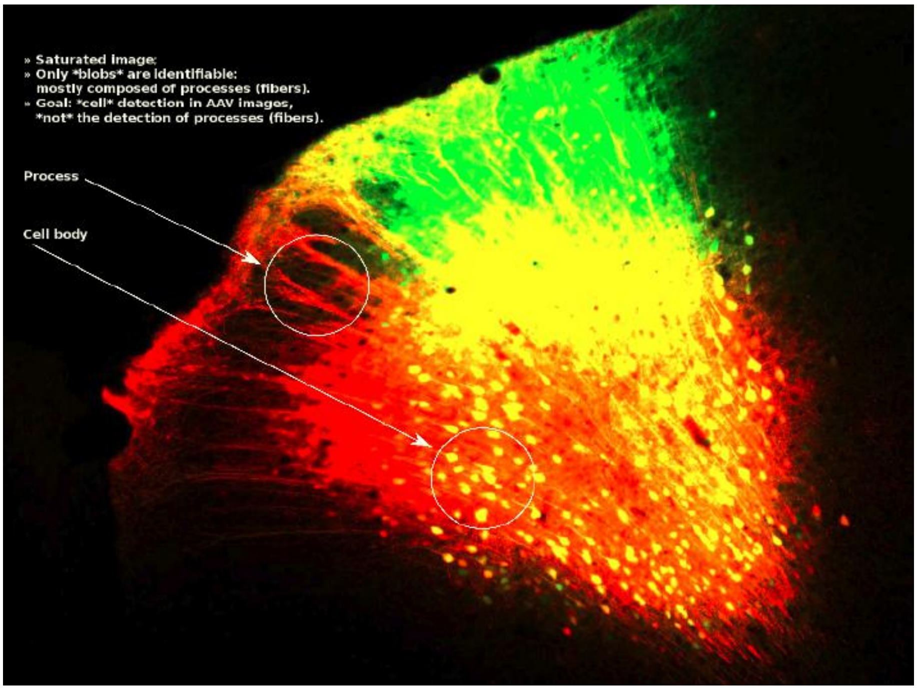

» Saturated image;
» Only *blobs* are identifiable:
mostly composed of processes (fibers).
» Goal: *cell* detection in AAV images,
*not* the detection of processes (fibers).
Process
Cell body

Figure 2.2: Saturated AAV tracer

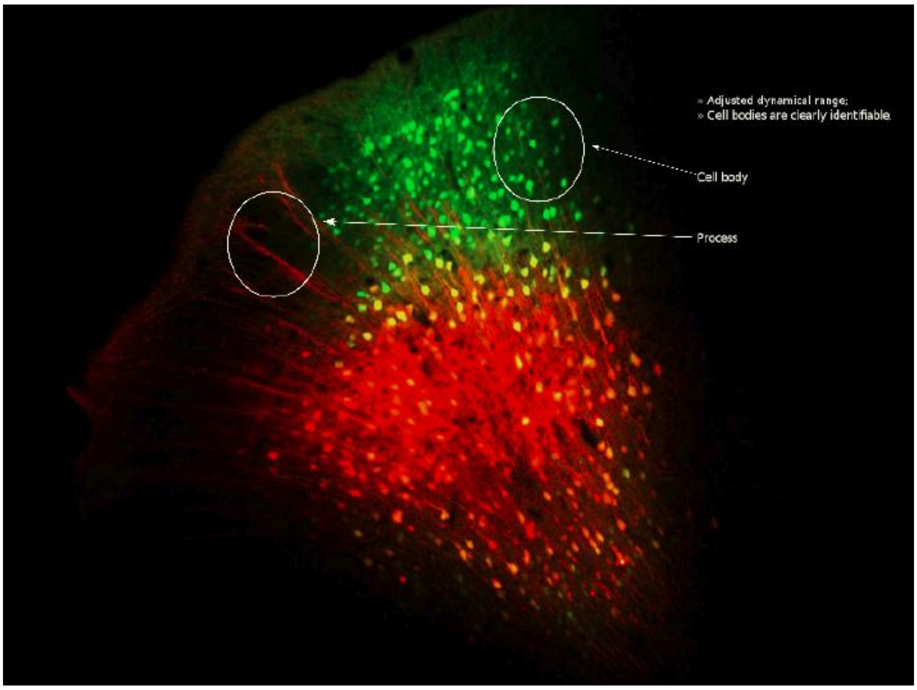

» Adjusted dynamical range;
» Cell bodies are clearly identifiable.
Cell body
Process

Figure 2.3: Filtered AAV tracer image with cell bodies visible

The main step in determining the cell body count is to isolate the cells by segmenting them from the process fibres as shown in (2.3). Two methods are employed for this task which are described in chapters 4 and 5.

## 2.2 CTB tracer Images

Cholera Toxin subunit B (CTB) is used for retrograde tracing. Example of such a tracer image is shown in (4.10). The task is to find the injection area of all the slices in the same brain and collate them to find the volume of the injection site.

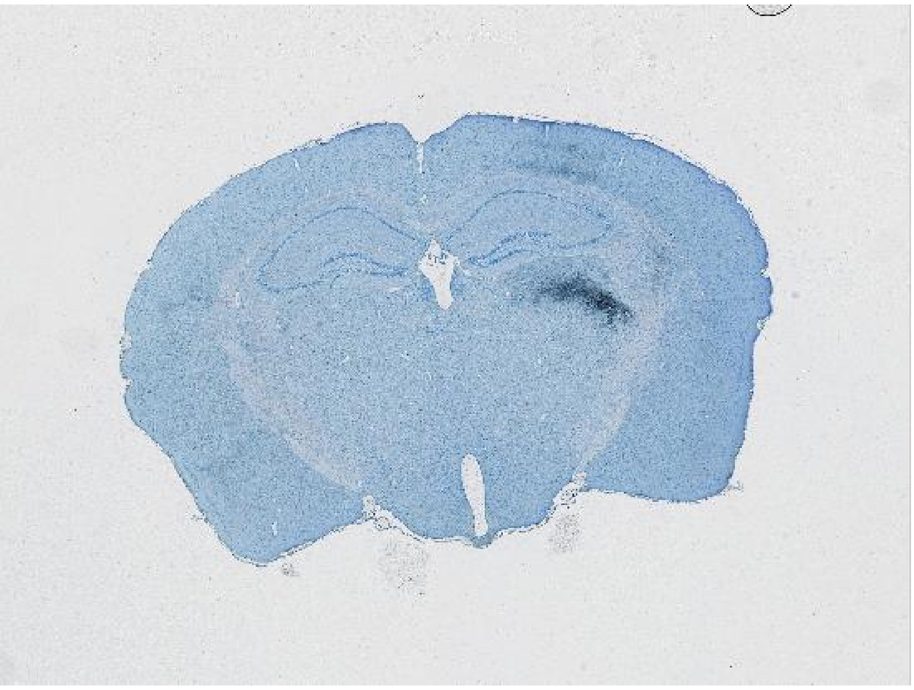


Figure 2.4: CTB tracer image

Stained cell body count is extremely difficult to determine in the CTB images as the tracer is weak. So this analysis is skipped for this image.

# CHAPTER 3

# WATERSHED SEGMENTATION

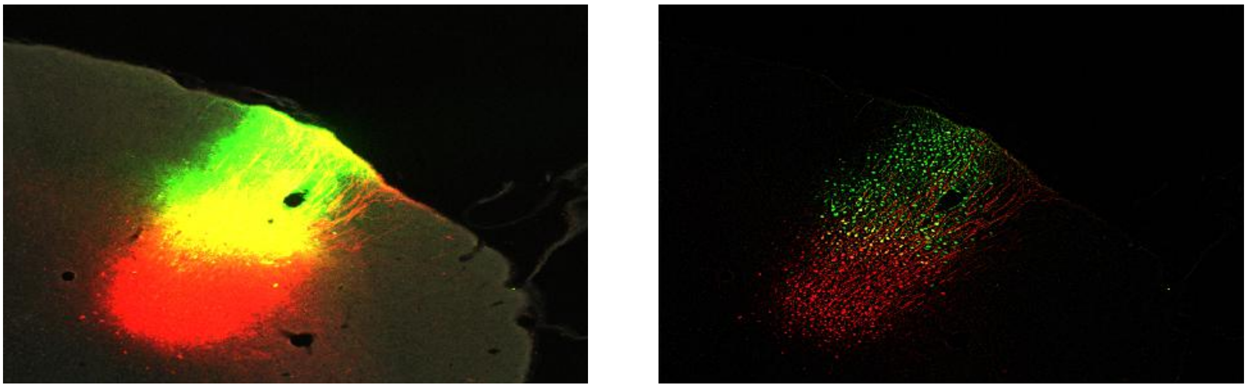
## 3.1 Introduction

The first task is to obtain the stained cell body count. To achieve this target, cells which are touching process fibres need to be isolated by segmentation. Watershed Segmentation is widely used for the problem of touching object segmentation. The principle of watershed algorithm to segment gray level image was introduced by L. Vincent and P. Soille (Vincent and Soille, 1991). The mechanism of watershed assumes that the image is a topographic relief in which the gray level intensity corresponds to the altitude, higher the intensity, higher the altitude. The watershed is a set of *dams* built to separate different *catchment basins*, surfaces such that the water flows down from any point to the same minimum. In watershed, water starts filling the catchment basins from their minima. The place where water from two different minima merge, we build a dam to prevent the merging from happening. These set of dams correspond to the watersheds of the image which are the lines that segment the touching objects.

Watershed segmentation often leads to over-segmentation. Several variants of watershed are proposed to prevent this and some of them include merging of different catchment basins post watershed while some include constrained filling of the water from specific points, instead of all minima. The latter variant is called as Marker-based Watershed Segmentation. This method is employed in this thesis.

## 3.2 Pre-Processing

The AAV tracer images are too heavily saturated to get the cell body information. So initial filtering is done to bring the cell bodies into focus. A method of filtering prescribed in Arce *et al.* (2013) is followed. The image is initially passed through a Gaussian filter and the filtered image is subtracted from the original raw image to get the desired image. This final image has cell bodies clearly identifiable as seen in (4.3b)

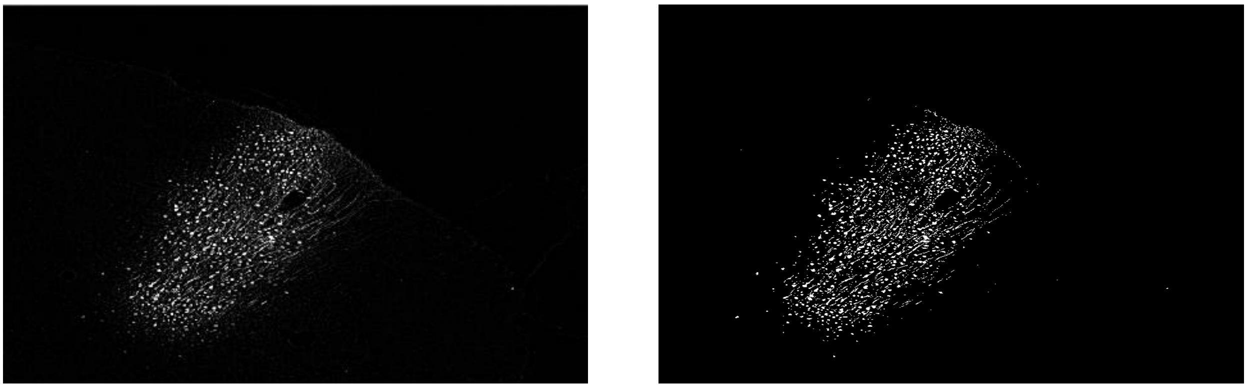


(a) Original image

(b) Filtered image

Figure 3.1: Filtering of Image

The filtered image is then converted to gray scale as seen in (4.4a). Otsu thresholding is applied on this gray scale image to give a binary image as seen in (4.4b). The segmentation algorithm is applied on the binary image.



(a) Grayscale image

(b) Binary image

Figure 3.2: Gray to Binary Conversion

## 3.3 Watershed Segmentation

Plain watershed segmentation results in over-segmentation as shown in (3.3).

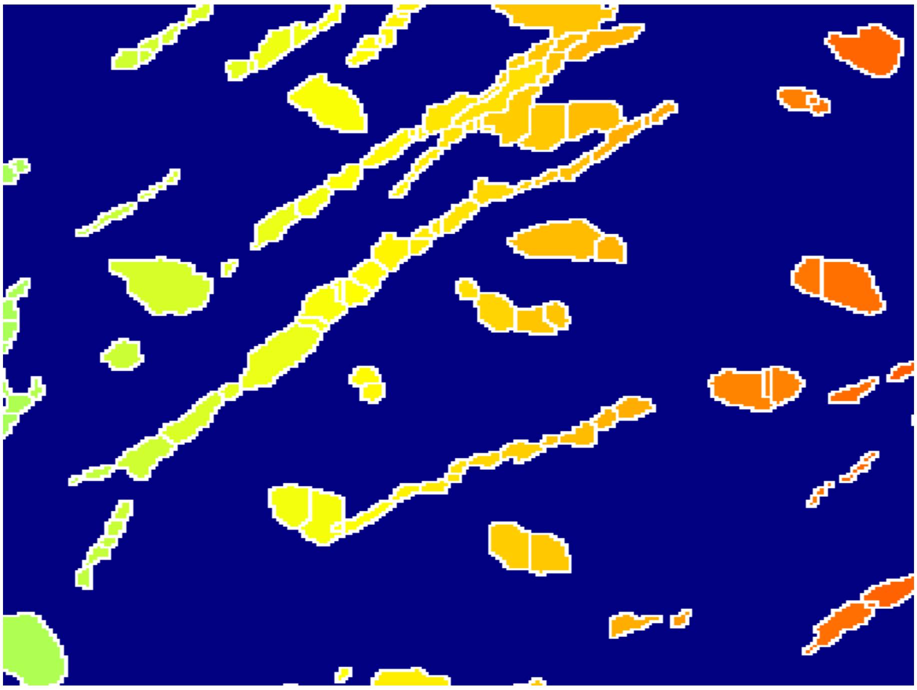


Figure 3.3: Watershed Segmentation

## 3.4 Marker-based Watershed Segmentation

To prevent over segmentation a Marker-based Watershed SegmentationZhang and Jiang (2011) is implemented. Segmentation using the watershed transform works better if we can mark foreground and background objects. Markers for foreground and background are obtained from the morphological operations like erosion and dilation. The Marker image obtained by eroding the binary image is shown in (3.4).

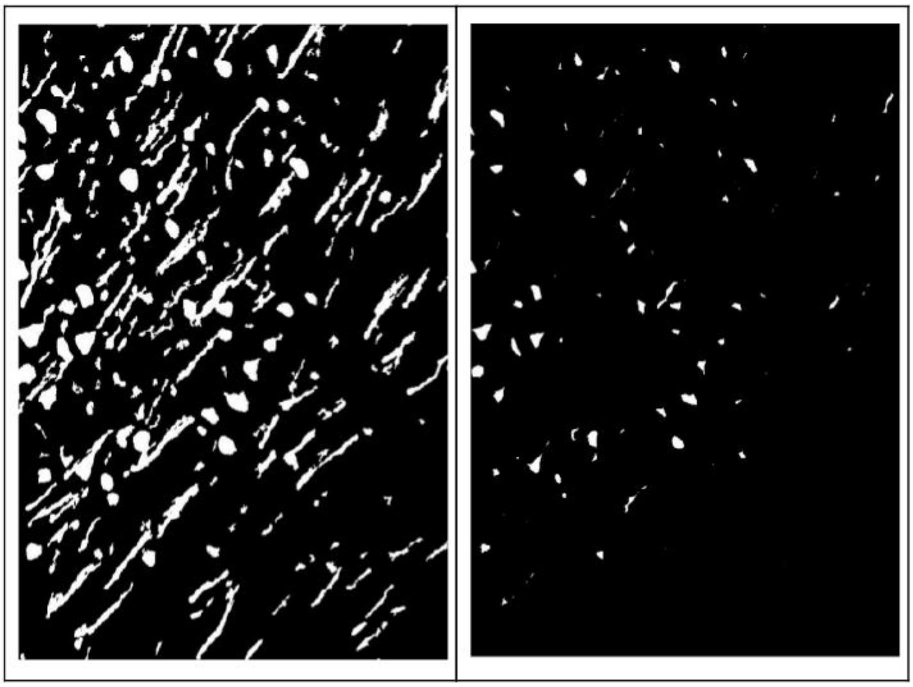


Figure 3.4: Marker Image

The segmentation boundaries for marker based watershed segmentation can be seen in (3.6)

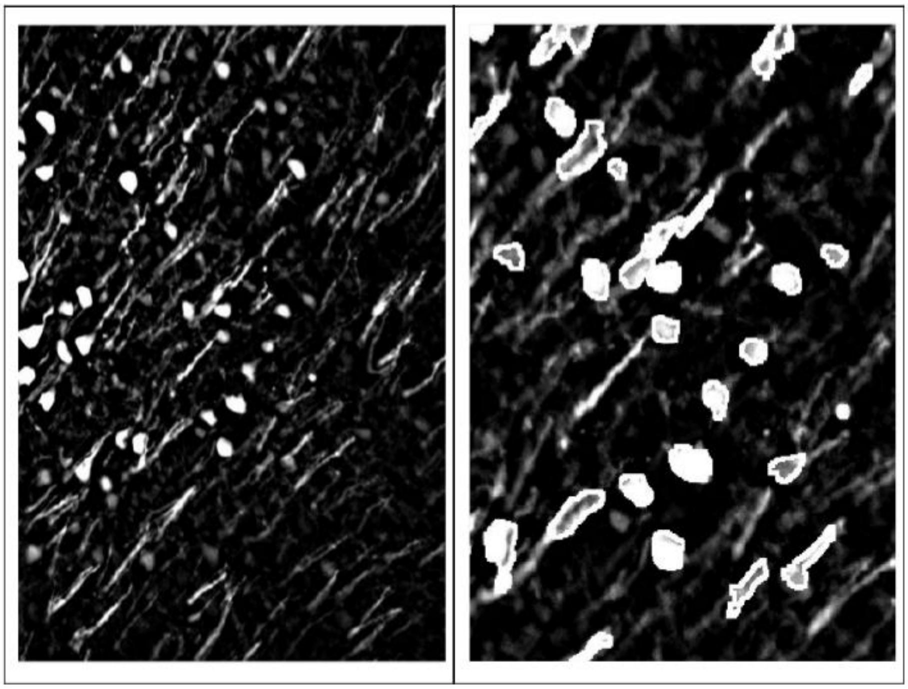


Figure 3.5: Segmentation Boundaries

The Marker-based watershed segmentation performed on a portion of the image is shown in (**??**).

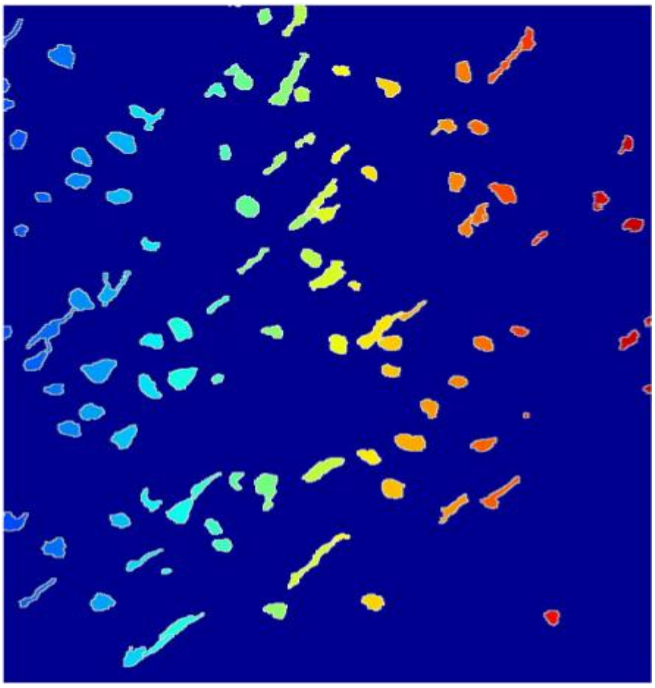


Figure 3.6: Marker-based Watershed Segmentation

After segmentation, the next step is to determine which portions are cells by shape feature. The darker portions in (**??**), are identified as process fibres and the lighter portions are identified as cells.

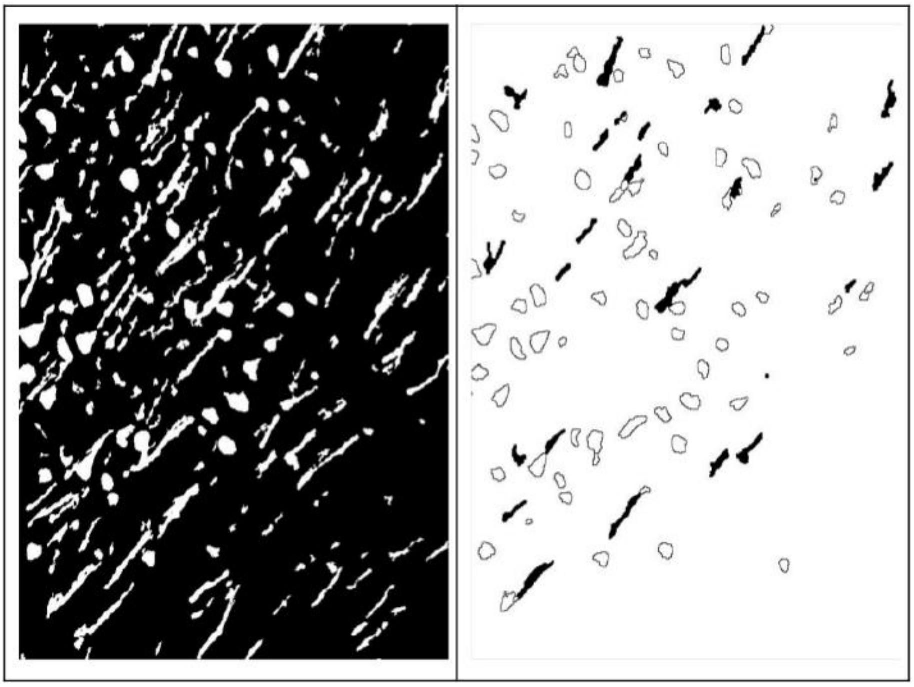


Figure 3.7: Segregated cells and process fibres using shape features

## 3.5 Results

Segmentation performed using the Marker image will output the segmented image shown in (3.8).

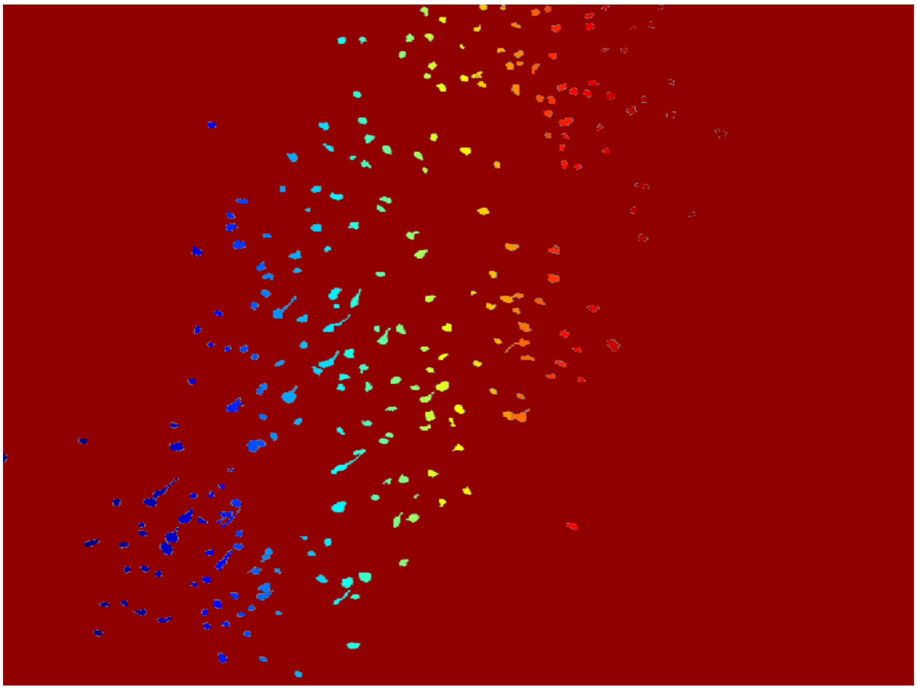


Figure 3.8: Marker-based Watershed Segmentation

Because of insufficient markers, cells are less populated in the segmented image compared to the original image. Almost 40% of cells are not reported as cells. A new method to identify the appropriate markers is to be implemented. In this thesis, this method is forsaken and concavity analysis is implemented to segment the cell bodies.

# CHAPTER 4

# SEGMENTATION BY CONCAVITY POINT ANALYSIS

## 4.1 Introduction

In this chapter, a new algorithm based on concave point analysis is described. The algorithm proceeds by generation of concave points and then determines the split lines for segmentation, with the concave points as the vertices for the segmentation of the object.

## 4.2 Literature Review

Segmentation using Concavity analysis has two stages. The first stage is detection of Concavity points and the second stage is choosing appropriate line segments to split the clumped objects. Literature survey is done to chose a suitable methods to detect the concave points in the contour of an object and to split the clumped cells.

### 4.2.1 Finding the Concavity Points

The icut algorithm presented in (He *et al.*, 2015), describes a method of determining the concaveness of each point on the contour of the clumped object and then a threshold is applied to these concaveness values. In this paper a mask is applied at every contour point of a binary object. The number of pixels common to the mask and the object at each contour point is its concaveness value. This technique has been attempted but is unsuccessful as there are many redundant local concavities that came into focus upon thresholding. A larger mask which would usually solve the problem could not be employed as the fibres to be segmented are thin, elongated structures.

A method to trace a contour and identify all the points that make a concave angle with the adjacent points is described in (Rafferty *et al.*, 2012). This is not particularly effective for the current problem as the concave points which are at a greater depth from the convex hull but do not satisfy the angle requirement are discarded and a few redundant points arose due to local concavities.

The paper, (Plissiti *et al.*, 2013), calculates the concave points by first determining the convex hull of the binary object. Then, for each line segment of the convex hull that does not belong to the global boundary, the points of the boundary that have the largest distance from the convex hull are identified. A variant of this method is employed in the current thesis.

Concave points are determined using a curvature metric assigned to each point in the contour of the object in (Mouelhi *et al.*, 2011). The curvature metric is expressed using Gaussian derivatives. The initial concave points are determined by keeping the local minima of the curvature.

A more robust method for the detection of concave points is described in (LaTorre *et al.*, 2013). In this paper three measures are used jointly to describe the concavity of a point. The three measures are as follows: the area of the concave region which is between the yellow and the magenta lines as shown in (5.1), the length of the green line, the ratio of the length of the magenta line to the corresponding yellow line.

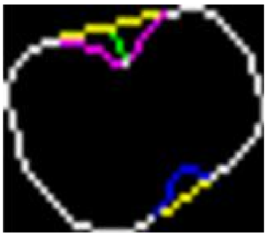


Figure 4.1: Concavity measure

In (Wang and Song, 2007), the medial axis of a binary object is found and the distance of every contour point from the medial axis is determined. By thresholding this distance, the suitable concave points can be found. This method cannot be employed as there are thin, elongated structures whose distances from the medial axes are small but do not contain the concave points.

A method described in (Farhan and Yli-Harja, 2013) is found to be suitable for the purpose of this problem. In this method, concavity depth at each point on the contour is found by a method described later in section —- and then a threshold is applied to this value to find the concave points.

A variant of the method described in (Wang *et al.*, 2011) is used in the thesis. In this paper clumps are split into two equal parts by determining the bottleneck of the figure. The example of bottleneck is shown in (**??**). For any couple of points given in the contour of the object, a metric is defined, which on minimizing gives the split line for the figure. This method is effective but cannot be directly employed as it splits the figure into just two parts. An iterative version of this algorithm was employed.

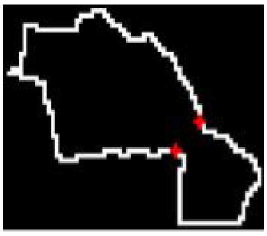


Figure 4.2: Bottleneck points

### 4.2.2 Determination of split lines for segmentation

The split lines in (Fan *et al.*, 2013) are determined by using concave regions instead of concave points. For every binary object there is, the object is subtracted from its convex hull to determine the concave regions. The split lines connect two concave regions by joining the points from the two regions which are at a minimum distance from each other. While this is an excellent method to split circular-like objects, arbitrary shapes result in redundant concave regions and splitting according to concave regions leads to over-segmentation.

A set of features namely Saliency, Concavity-concavity alignment, and Concavity-line alignment are taken for all the concavity points in the list of candidate split lines in (**?**). A Split line is chosen so that it as short as possible in addition to having concave enough regions at both of its ends. This is ensured by Saliency by considering Concavity depth as well as the distance between the two concavity points.

In (Wang and Song, 2007), the best split line among the candidate lines that join the concave points is found by employing certain constraints like distance between the vertices of a splitting line, the area ratio of two parts resulting from the splitting, etc. The splitting process is proceeded iteratively.

Watershed segmentation is used in conjunction with concave points to determine the split lines for segmentation in (Mouelhi *et al.*, 2011). The split lines are first determined by watershed segmentation which inevitably results in over-segmentation. Among these split lines those which have concave points as their vertices are chosen.

The icut algorithm in (He *et al.*, 2015) uses the spatial information as well as concave contours to fill up a weight matrix of a graph. A normalized cut is then performed to determine the split lines for segmenting the clumped objects.

Ellipse fitting algorithm is employed in (Plissiti *et al.*, 2013) instead of generating split lines for better segmentation, please find our proposed offer below:

on of overlapping cells. Ellipse fitting is not necessary for the purpose of this thesis as the main aim of segmentation is to get the count of the cells by not necessarily retaining the shape.

In (Farhan and Yli-Harja, 2013), the split lines are identified by using a new technique called Rectangular Window-based Concavity Point Pair Search. The algorithm an algorithm which finds the minimum/maximum intensity splitting path using the directional vector associated with the concavity point which guides the algorithm in the right direction by means of a window and prohibit it from straying.

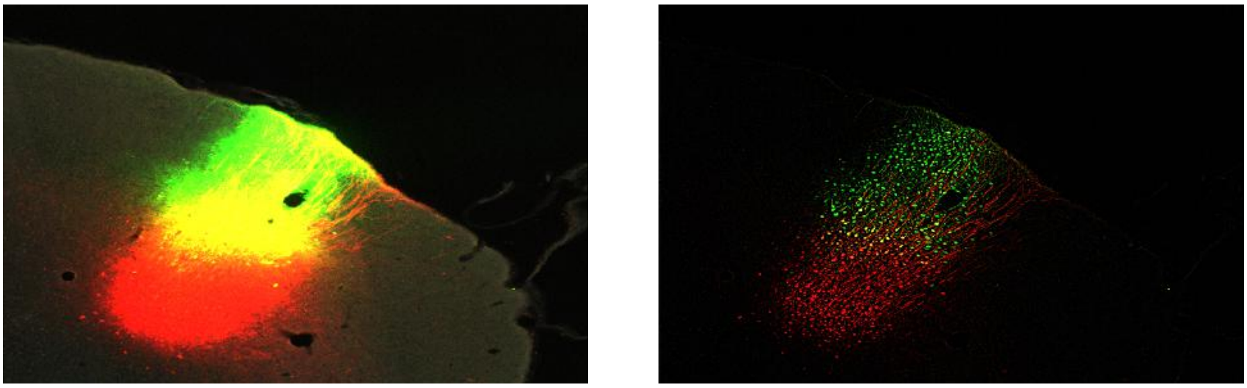
Concave vertex graph is employed in (Yang *et al.*, 2008*a*) to find the best split line. A graph is formed with vertices as the concave points of the object's contour and edges detected in the region of touching cells. The segmentation is then treated as an optimal grouping of pixels, which can be solved by recursively searching optimal shortest path in the concave vertex graph using Dijkstra algorithm.

## 4.3 Segmentation by Concavity analysis

In this method, split lines for the segmentation are formed by the detection of concave points in the contour of the objects in the image. The steps followed in the process are described below.

### 4.3.1 Preprocessing

The AAV tracer images are too heavily saturated to get the cell body information. Initial filtering is done to bring the cell bodies into focus. A method of filtering prescribed in Arce *et al.* (2013) is followed. The image is initially passed through a Gaussian filter and the filtered image is subtracted from the original raw image to get the desired image. This final image has cell bodies clearly identifiable as seen in (4.3b)

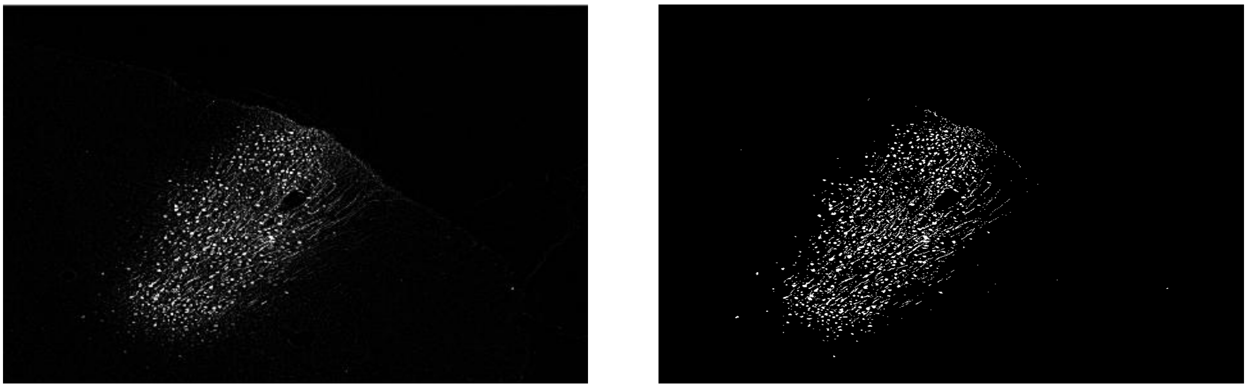


(a) Original image

(b) Filtered image

Figure 4.3: Filtering of Image

The filtered image is then converted to gray scale as seen in (4.4a). Otsu thresholding is applied on this gray scale image to give a binary image as seen in (4.4b). The segmentation algorithm is applied on the binary image.



(a) Grayscale image

(b) Binary image

Figure 4.4: Gray to Binary Conversion

### 4.3.2 Classification of objects based on shape

The objects in the binary image are classified based on shape into three categories.

1. Objects which are isolated cells
2. Objects which are cells but have process fibres attached to them
3. Objects which are process fibres

To realize this classification, different metrics are used.

### 4.3.3 Metric to identify cells

Isolated cells in the image can be identified by a measure of their compactnessYang *et al.* (2008*b*).

$$Compactness = \frac{(Perimeter)^2}{Area} \tag{4.1}$$

Circle, which is the most compact object has the compactness value of $4\pi$. The objects are identified as isolated cells if their compactness value lies within 20% of that of circle.

### 4.3.4 Metric to identify process fibres

Process fibres are elongated objects. Elongation of an object can be measured by taking the aspect ratio of the rotated minimal bounding boxYang *et al.* (2008*b*) of the object. The vertices of the rotated minimal bounding box can be seen in the (4.5).

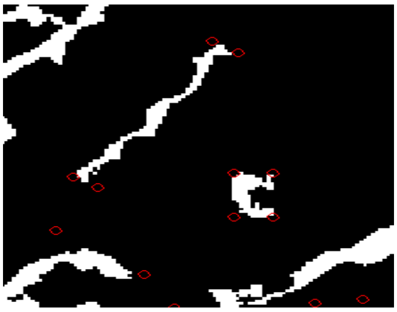


Figure 4.5: Vertices of a Rotated Minimum Bounding Box of an object

The Aspect Ratio, i.e the ratio of the width of the Bounding box to its height, is a reliable metric. But a few objects have very low Aspect Ratio but they are not be identified as process fibres. To deal with such cases, a new metric has been proposed to measure the elongation, which takes into account both SolidityYang *et al.* (2008*b*) and Aspect Ratio of the object.

$$Solidity = \frac{Area\ of\ the\ object}{Area\ enclosed\ by\ its\ Convex\ Hull} \tag{4.2}$$

$$Elongation = \frac{Solidity}{Aspect\ Ratio\ of\ the\ Rotated\ Minimal\ Bounding\ Box} \tag{4.3}$$

An object is identified as a process fibre if its elongation value is greater than 3. A sample process fibre with its elongation value is shown in (4.6).

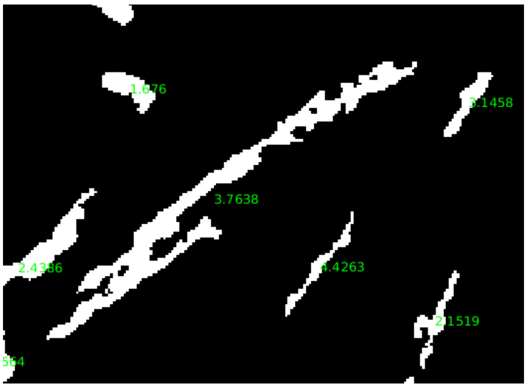
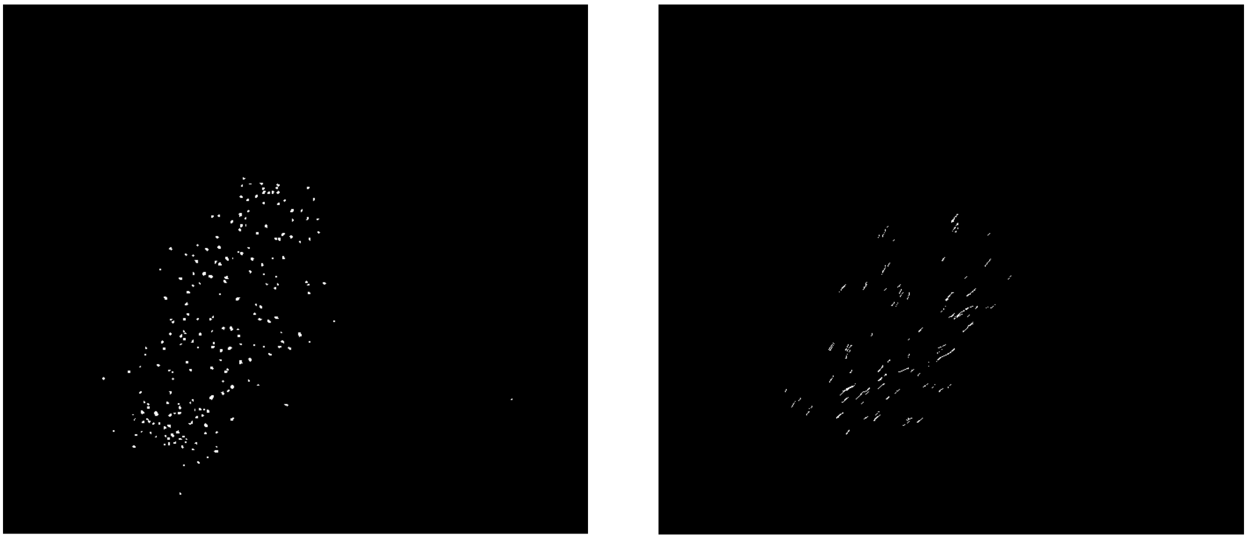


Figure 4.6: $\frac{Solidity}{AspectRatio}$ of a sample process fibre

The binary image is thus split into three categories as shown in (4.7a), (4.7b) & (4.8). The algorithm proceeds iteratively segmenting out the image which has both cells and fibres ($I_{work}$). Once new objects are created by segmentation, they are added to either categories of cells ($I_{cell}$) and fibres ($I_{process}$).



(a) Image with isolated cells, $I_{cell}$

(b) Image with isolated Process fibres, $I_{process}$

Figure 4.7: Categorizing of Objects in the Image

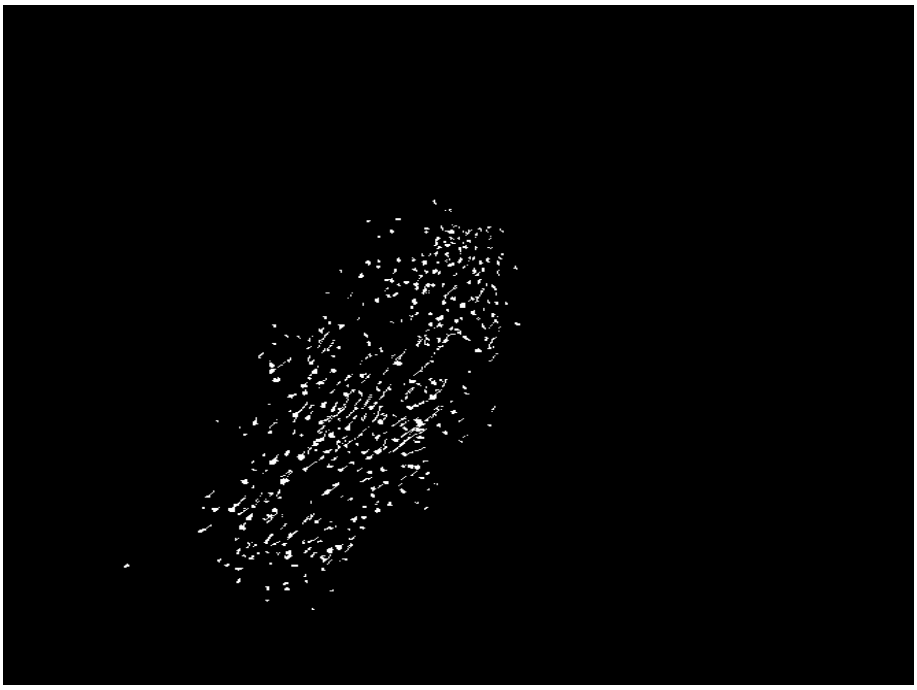


Figure 4.8: Image which contain clumps of cells and process fibres, $I_{work}$

### 4.3.5 Detection of Concave points

For the segmentation of clumps of cells or cells with process fibres attached to them, concave point analysis is used. The concave points are determined based on the definition of concavity. The idea is to take two contour points which are joined by a line. Taking too far off or close by points may cause error in the detection of concave points. In this algorithm, the contour points are chosen such that they are 15 pixels away from each other. If the entire line defined by these two contour points lies solely within the object, the mid point of the contour defined by the two points is the point of convexity. On the other hand if the entire line passes through the background, the mid point of the contour defined by the two points is the point of concavity. In the latter case, the concaveness of the mid point of the contour is determined by taking its distance from the mid point of the line defined by the two points.

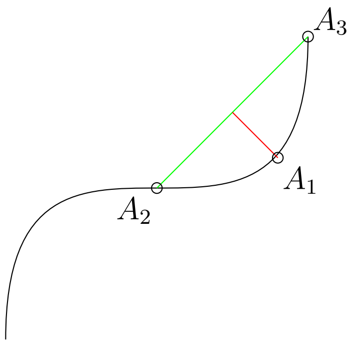


Figure 4.9: Detection of Point of Maximum Concavity

Consider a pixel on the contour of the object, $A_1$ in (4.9). Two points, $A_2$, $A_3$ are chosen 7 pixels away from it. The line $\overline{A_2A_3}$, the green line, lies entirely in the background so the point $A_1$ is the point of concavity. The distance of the point $A_1$ from the line $\overline{A_2A_3}$, the red line, is a measure of concavity at the point $A_1$.

By using this technique, point of maximum concavity (i.e, the greatest depth) is found for every object. Points of maximum concavity (shown as red circles) for few objects is shown in (4.11a), (4.12a).

### 4.3.6 Detection of a Split line containing the Point of Maximum Concavity

Once the point of maximum concavity has been detected, the next step is to detect another point which will help split the object into two parts. The other point is chosen such that the pair of points is a bottleneck in the object. The example of a bottleneck is shown in the figure (4.10)In order to achieve, a method described in Wang *et al.* (2011) is implemented.

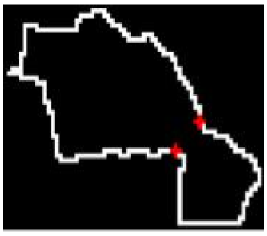


Figure 4.10: Bottleneck points

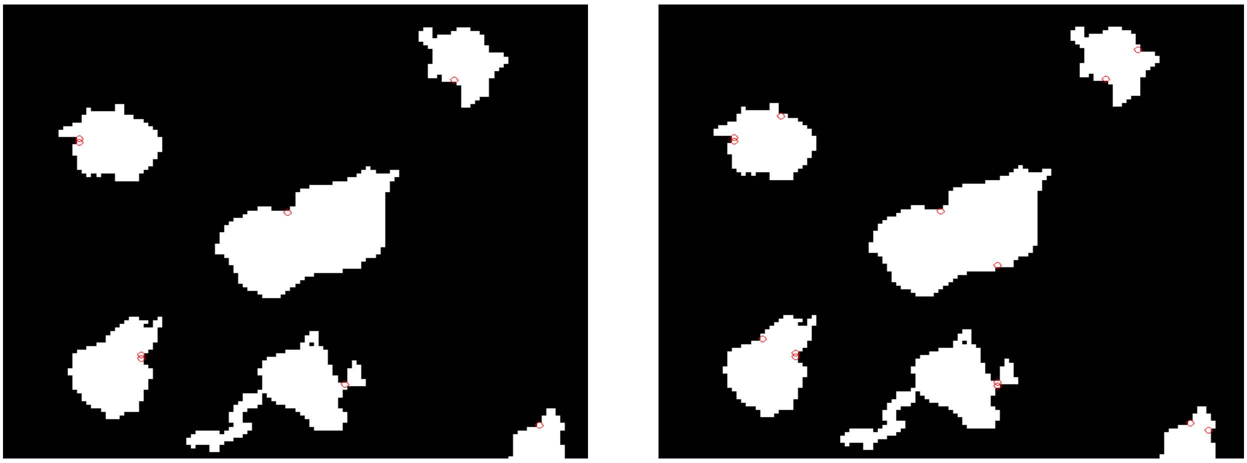
Let $A$, $B$ be two different points on the contour of an object. We can define a cost function between points $A$ and $B$ as

$$E_s(A,B) = \frac{dist(A,B)}{min\{length(A,B),\ length(B,A)\}} \tag{4.4}$$

where $dist(A, B)$ represents the Euclidean distance between the points A and B, $length(A, B)$ denotes the clockwise length from point $A$ to $B$ on the boundary of the object. $minlength(A, B),\ length(B, A)$ represents the smaller value between the two lengths.
For any point of maximum concavity $A$, a point $B^*$ is found which minimizes the cost $E_s$.

$$(A, B^*) = \underset{B}{argmin}\ E_s(A, B) \tag{4.5}$$

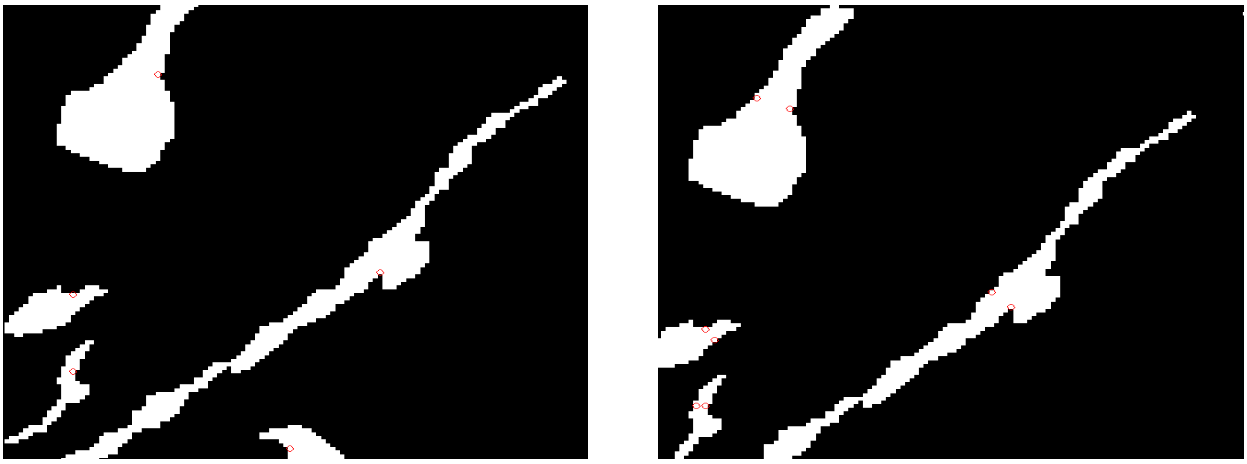
The points of maximum concavity for a few objects are shown in (4.11a), (4.12a) and the split points determined by equation (5) are shown in (4.11b), (4.12b).



(a) Point of maximum concavity

(b) A split point for the point of maximum concavity

Figure 4.11: Generation of a split point

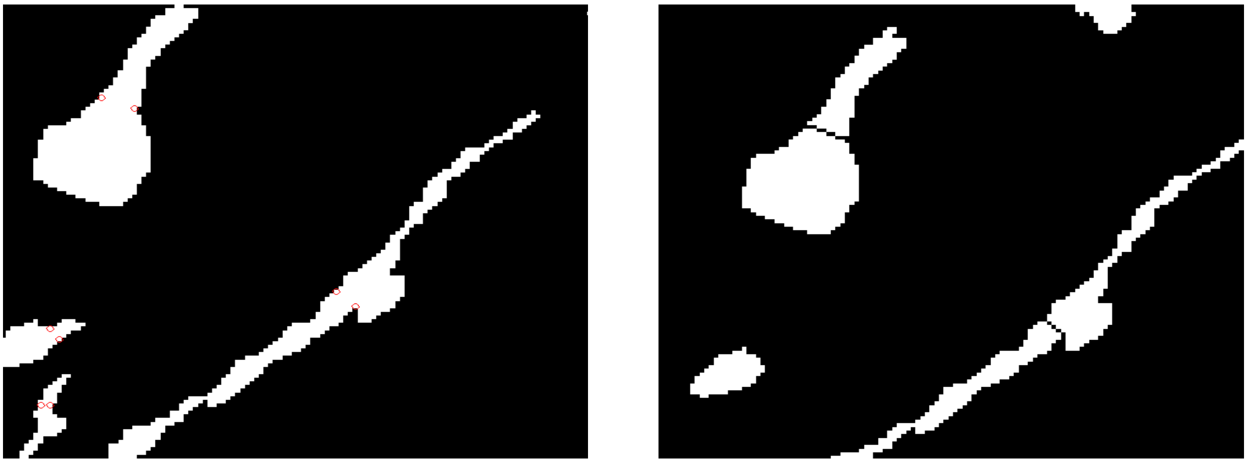


(a) Point of maximum concavity

(b) A split point for the point of maximum concavity

Figure 4.12: Generation of a split point

Now that the two split points for an object have been identified, the split line is determined passing through these two points using Bresenham Line Algorithm as shown in (4.13b).



(a) Split points for objects

(b) Split lines through Points of Maximum Concavity

Figure 4.13: Splitting of cells using split points

The algorithm is described in (1). The algorithm proceeds segmenting the cells iteratively and the newly segmented cells are classified according to shape into cells or process fibres. When there are no more cells left to segment, the program ends with the segmented cells as the output.

**Algorithm 1:** Segmentation by Concavity analysis

```
$I_{Filt} = Filter(I_{original})$ ;
$BW = Binary(I_{Filt})$ ;
$[I_{Cell}, I_{Work}, I_{Process}] = shapeclassify(BW)$ ;
; // Classify the binary image into three categories based on the shape of the objects
while $num\_of\_objects(I_{Work}) > 0$ do
    ; // Looping till all the objects in the $I_{Work}$ are segmented and classified into $I_{Cell}$ and $I_{Process}$
    $Cpts = maxconcavepoints(I_{Work})$ ;    // Calculates points of maximum concavity
    $Spts = splitpoints(Cpts)$ ; // Calculates split points for each of the concave points
    $I_{Work-Seg} = bresenham(Cpts, Spts)$ ;    // Segments the objects based on split and concave points
    $[I_{Cell\_New}, I_{Work\_New}, I_{Process\_New}] = shapeclassify(I_{Work\_Seg})$ ;    // Classify the objects in the newly segmented image into cells and process fibres
    $I_{Cell} = I_{Cell} \mid I_{Cell\_New}$ ;    // Adding newly identified cells into $I_{Cell}$
    $I_{Process} = I_{Process} \mid I_{Process\_New}$;  // Adding newly identified Process fibres into $I_{Process}$
    $I_{Work} = I_{Work} - I_{Cell\_New} - I_{Process\_New}$ ;    // Updated $I_{Work}$ image after the segmentation
$I_{Cell\_Segmented} = I_{Cell}$;
```

### 4.3.7 Results

The result of the segmentation can be seen in (4.14). The original image and the image with the discarded process fibres are shown in (5.4a) and (5.4b).

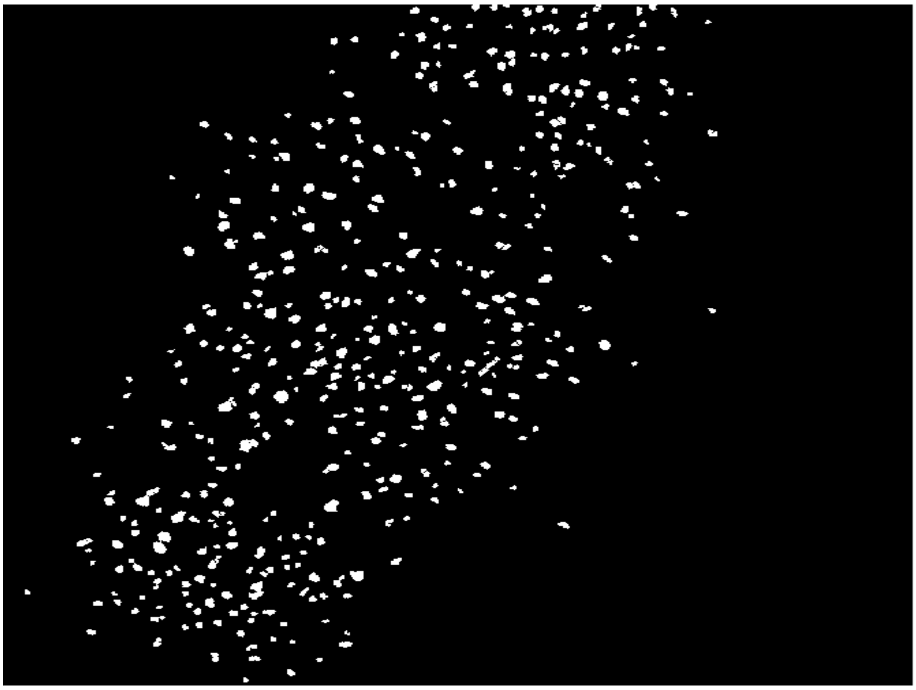
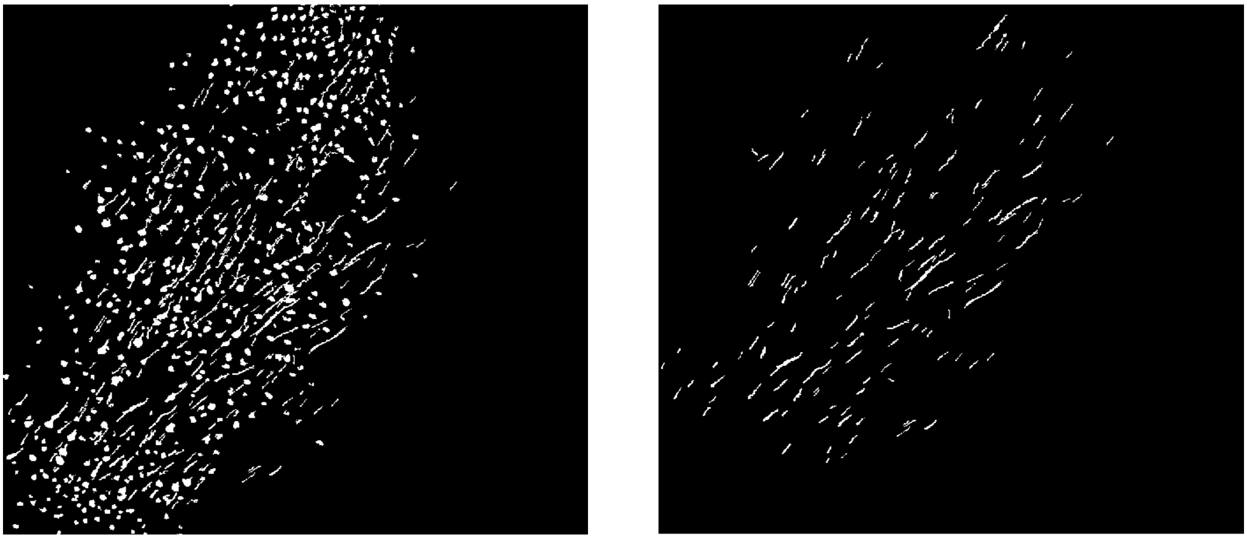


Figure 4.14: Image with cells segmented, $I_{cell}$



(a) Image to be Segmented

(b) Image with isolated Process fibres, $I_{process}$

Figure 4.15: Imaqe with Process fibres

### Precision and Recall

One tracer image is annotated and is compared to the segmented image. The values are given in the table below. As seen in the table, Precision is 95% and Recall is 87%.

| **Measure** | **Value** |
|---|---|
| True Postives | 480 |
| False Positives | 25 |
| False Negatives | 67 |
| Precision | 0.95 |
| Recall | 0.87 |

# CHAPTER 5

# VOLUME ESTIMATION

## 5.1 Introduction

Each mouse brain is sliced into 256 sections and are imaged using fluorescent or histochemical techniques depending upon the tracer employed. In order to determine the volume of the injection site, the area of tracer in each of the 26 sections is to be determined. To achieve that, the tracer portion in the image needs to be segmented.

AAV tracers uses fluorescent imaging technique and the tracer is manifested in three colors. AAV tracer is present in two colors red and green, but the places where the two colors combine, yellow color is observed. In CTB images, the tracer can be observed to be black in color. The next section describes the segmentation methods used to achieve this.

## 5.2 Segmentation

Color image segmentation can be done using thresholding, but thresholding doesn't take into account the status of the neighbouring pixels. Since we are interested in the blob detection rather than scattered tracer detection, thresholding requires a fair bit of post-processing. So pixel classification methods are employed to segment the tracer part.

### 5.2.1 Support Vector Machines

The primary method employed is Support Vector machines (Wang and Wang, 2011). The steps involved in the SVM are

1. Generation of training data set
2. Feature vector
   - The R,G,B values of a pixel are 3 features
   - The R,G,B values of a median filtered image are 3 features. This takes care of the effect of the neighbourhood in deciding the pixel label

- It is a 6 feature vector

3. SVM
    - For a 200 x 200 image, SVM with 4 classes and 6 features takes 70 s to run
    - If we crop the rough area of the tracer, we have 2000 x 2000 image which is 100 times bigger. Such an image takes almost 2 hours to run. SVM takes a lot of time but accurately classifies pixels
    - Other classifying methods are to be explored

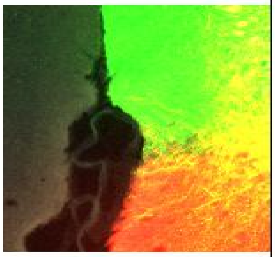
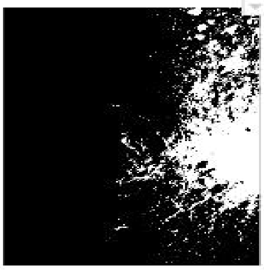
| Cropped image | Red Portion | Green portion | Yellow portion<br>(both red and green) |
|---|---|---|---|
|  | | |  |

Figure 5.1: SVM segmentation of tracers

Because of the high computational complexity of SVM, this approach is rejected.

### 5.2.2 Decision trees

The sample cropped figure which is used for comparing SVM and decision trees (G.Smolinski and G.Milanova, 2008) is shown in (5.2). The results of classifications for both SVM and decision trees are shown in (5.3).

Here SVM is chosen as the standard annotation for measuring the accuracy of the decision tree classification. The accuracy is computed to be 95.4%.

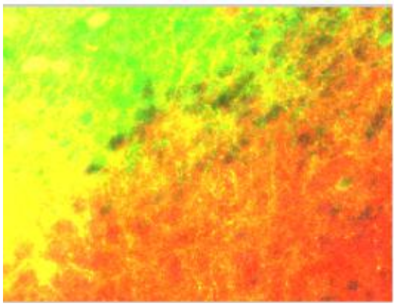


Figure 5.2: Cropped image

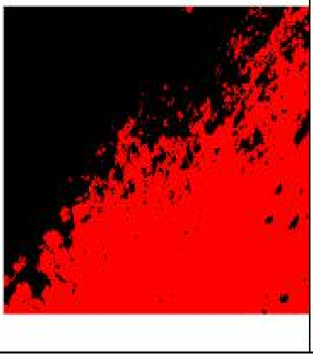
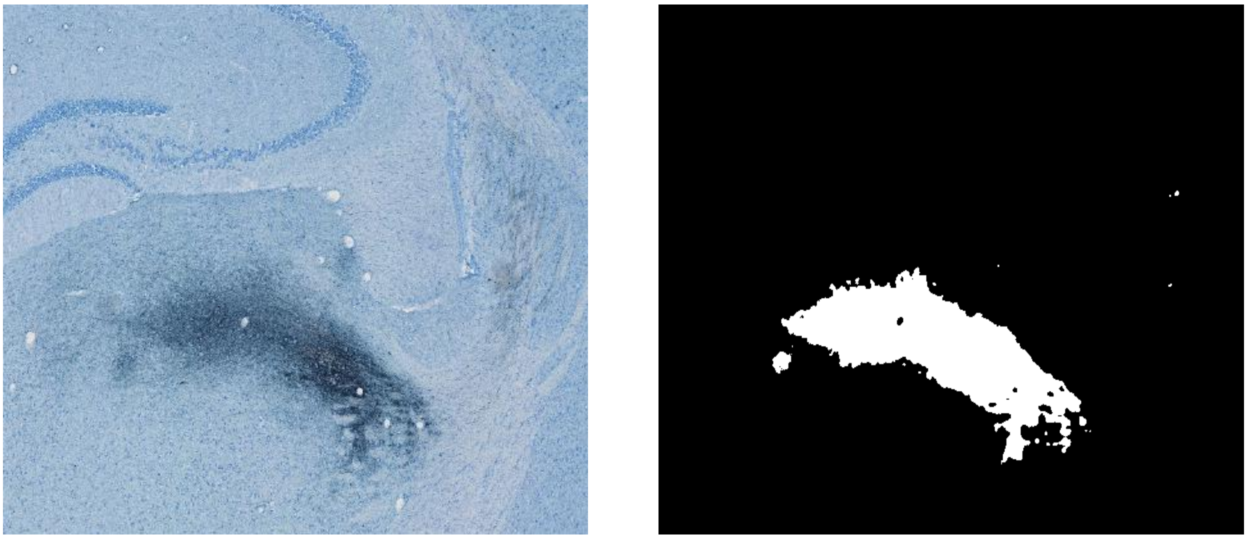
| Classifier | Red portion | Green portion | Yellow portion |
|---|---|---|---|
| Decision trees | | | |
| SVM |  | | |

Figure 5.3: Comparison between SVM and Decision trees

The same analysis applied for CTB tracer gives the following results



(a) CTB Image to be Segmented

(b) Image with tracer segmented

Figure 5.4: CTB tracer segmentation

## 5.3 Volume Estimation

To calculate the volume of the injection site, the areas of the tracer in all the images are stored in an array. The injection coordinates of the injection site in the image are also noted to ensure that the areas of the two consecutive slides in fact do overlap. The volume occupied by the two consecutive slides is approximated by a conical frustum and is calculated by the formula,

$$V = (A_1 + A_2 + \sqrt{A_1 * A_2}) * \frac{h}{3}$$

Where $A_1$ and $A_2$ are areas between the consecutive slices and h is the distance between the two slices which is $40\mu m.$ A pixel is supposed to be $5\mu m$ in size so the area is taken to be $25(\mu m)^2$. The injection coordinates are noted down to evaluate the injection precision.

# CHAPTER 6

# CONCLUSIONS AND FUTURE WORK

The problems of stained cell body count and volume of the tracer have been addressed in the thesis. For the problem of the cell body staining, Concavity point based analysis is found to be optimal. The algorithm proceeding in the iterative fashion using shape features of the objects segmented give superior performance compared to its counterpart Watershed Segmentation. The precision and recall obtained are 95% and 87%.

For the problem of Volume estimation, various methods of Pixel Classification are experimented with. Support Vector Machines give the best results, but they are computationally expensive. Hence the method is rejected. The one that came closest to SVM without the time complexity issues are Decision trees. The accuracy of Decision trees is computed to be 95.4%.

The next step in this project, now that the optimal algorithm were found, is to further optimize the code so that it could be incorporated in the project pipeline. The resolution of each image is 18000x 24000 px and the algorithm needs to work on such a large data set. This is still in process. The proposed method to deal with such a high resolution image is as follows. The lossy JPEG2000 image is to be subjected to thresholding methods to roughly identify the injection site. The relevant portion is then cropped out and is passed to the algorithm to extract stained cell body count and the volume of tracer.

The end goal is to develop a modular code which can be incorporated into the pipeline of the project to facilitate the Quality Control process.

# REFERENCES

1. **Arce**, **Wu**, and **Tseng** (2013). Fast and accurate automated cell boundary determination for fluorescence microscopy. *Scientific Reports*, **3**.

2. **Boland, J. W.**, **C. Wu**, **H. Barbas**, and **H. Bokil** (2009). A proposal for a coordinated effort for the determination of brainwide neuroanatomical connectivity in model organisms at a mesoscopic scale. *PLOS Computational Biology*.

3. **Fan, J.**, **Y. Zhang**, and **S. Li** (2013). A separating algorithm for overlapping cell images. *Journal of Software Engineering and Applications*, **6**.

4. **Farhan, M.** and **O. Yli-Harja** (2013). A novel method for splitting clumps of convex objects incorporating image intensity and using rectangular window-based concavity point-pair search. *Pattern Recognition*, **46**.

5. **G.Smolinski, T.** and **M. G.Milanova**, *Computational Intelligence in Biomedicine and Bioinformatics*. Springer, 2008.

6. **He, Y.**, **H. Gong**, **B. Xiong**, and **X. Xu** (2015). icut: an integrative cut algorithm enables accurate segmentation of touching cells. *Nature*.

7. **LaTorre, A.**, **L. Alonso-Nanclares**, and **S. Muelas** (2013). Segmentation of neuronal nuclei based on clump splitting and a two-step binarization of images. *Expert Systems with Applications*, **40**.

8. **Mouelhi, A.**, **M. Sayadi**, and **F. Fnaiech**, Automatic segmentation of clustered breast cancer cells using watershed and concave vertex graph. *In International Conference on Communications, Computing and Control Applications (CCCA)*. 2011.

9. **NL, C.** and **D. B** (1998). Recombinant adeno-associated virus vector: use for transgene expression and anterograde tract tracing in the cns. *PubMed*.

10. **Plissiti, M. E.**, **E. Louka**, and **C. Nikou** (2013). Splitting of overlapping nuclei guided by robust combinations of concavity points. *Expert Systems with Applications*, **40**.

11. **Rafferty, K.**, **S. Drury**, **G. Hartshorne**, and **S. Czanner** (2012). Use of concave corners in the segmentation of embryological datasets. *Review of Bioinformatics and Biometrics (RBB)*, **1**.

12. **Vincent, L.** and **P. Soille** (1991). Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE transactions in Pattern Analysis and Machine Learning*.

13. **Wang, H.**, **H. Zhang**, and **N. Ray**, Clump splitting via bottleneck detection. *In 18th IEEE International Conference on Image Processing*. 2011.

14. **Wang, W.** and **H. Song**, Cell cluster image segmentation on form analysis. *In Third International Conference on Natural Computation (ICNC 2007)*. 2007.

15. **Wang, X.-Y.** and **Q.-Y. Wang** (2011). Color image segmentation using automatic pixel classification with support vector machine. *Neurocomputing*.

16. **Yang, L.**, **O. Tuzel**, and **P. Meer**, Automatic image analysis of histopathology specimens using concave vertex graph. *In Med Image Comput Comput Assist Interv*. 2008*a*.

17. **Yang, M.**, **K. Kpalma**, and **J. Ronsin** (2008*b*). A survey of shape feature extraction techniques. *Pattern Recognition*, **43**.

18. **Zhang, W.** and **D. Jiang** (2011). The marker-based watershed segmentation algorithm of ore image. *IEEE 3rd International Conference on Communication Software and Networks (ICCSN).*