Automated Cell Segmentation Of Nissl Stained Mouse Brain Images

A Project Report

submitted by

ADITI SINGH

in partial fulfilment of the requirements

for the award of the degree of

BACHELOR AND MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

May 2016

THESIS CERTIFICATE

This is to certify that this thesis, submitted by **Aditi Singh**, to the Indian Institute of Technology, Madras, for the award of **Dual Degree (B.Tech. +M.Tech.)**, is a bonafide record of the research work done by them 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
Principal Investigator
Mitralab
Cold Spring Harbor Laboratory
New York, 11724

Prof. Kaushik Mitra
Research Guide
Assistant Professor
Dept. of Electrical Engineering
IIT Madras, 600036

Place: Chennai

Date:9 May 2016

ACKNOWLEDGEMENTS

I would like to thank the Indian Institute of Technology, Madras for giving me the opportunity to work on a project and my guide Prof. Partha Mitra, for allowing me to work with him and his invaluable help throughout the project, right from the identification of our project topic, to helping mould approaches and identifying new directions to work in when a line of thought had to be abandoned.

I would also like to thank my advisor, Prof. Kaushik Mitra, for all the guidance and encouragement he provided me.

I am deeply indebted to Dr. Toufiq Parag to help with my research at various stages. His inputs on the framing the segmentation algorithms have helped in shaping a lot of my research work.

I would also like to thank the members of Mitralab at CSHL, especially Dr. Daniel Ferrante , for the many helpful suggestions I have been provided. I thank Alexander Tolpygo, for providing me insights of the biological pipeline of the project.

ABSTRACT

KEYWORDS: Nissl; Segmentation; Ilastik; Annotation .

Modern neuroanatomical research relies on whole brain imaging using light microscopic techniques. Applications of current interest include mesoscale mapping[18] of connectivity at a whole-brain scale and other structural/functional whole-brain studies (in mouse, marmoset, and other species). An important step of a neuroanatomical study is "mapping", i.e. identifying the brain compartment in which a labelled cell or process is located. Classically, this mapping was done by visual examination of one or more histochemical stains, of which the Nissl stain is the most widely utilized variety. In the modern era of neuroanatomical research, it is desirable to perform this step using techniques from machine vision, given the large data volumes ($\sim 10^{12}$ pixels/brain for mouse).

As a first step, we have developed algorithms to segment Nissl-stained sections into the component objects, which can be further grouped to obtain information about the brain region involved. Several past studies ([16, 13]) addressed similar problems by fitting preconceived shape models to a set of image pixels. Given the large variation in the shapes and pixel intensities of the objects of interest, conceptually it is more reasonable to learn their appearances and boundaries instead. We adopt such an approach proposed in [27] for electron microscopy data segmentation. Each pixel of a Nissl image is classified into 3 classes: cell interior, cell boundary and background. On the real valued outputs of a pixel detector, we apply a region growing algorithm (watershed) that typically oversegments the cells into fragments or superpixels. We have used interactive interfaces for training the pixel classifier (ilastik) as well as the superpixel boundary classifier (implementation of technique described in [28]) with limited groundtruth data to generate our baseline segmentation results. On a small set of test images with sparsely located cells, we achieved an F-score of 81% of precision-recall values (at 95% recall) in determining the cell centers computed from the segmentation output.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 Motivation

The Mouse Brain Architecture Project[30] aims to generate brain-wide maps of neural connectivity in the mouse, at a mesoscopic level, an intermediate between single-neuron connectivity, at sub-micrometer scale and brain-wide connectivity, at centimeter scale, which will specify the inputs and outputs of major brain regions.

This will provide far more detail than, for instance, MRI-based methods, and yet considerably less detail than is achievable via electron microscopy (EM). The latter approach, while useful for mapping synaptic connections between individual neurons, is feasible on a whole-brain basis only for very small brains (e.g. that of the fruitfly) or very small portions of the mouse brain.

Information about connectivity[18] will lead to major advances in understanding of what makes one unique and will set the stage for future studies of abnormal brain circuits in many neurological and psychiatric disorders.

The mouse brain is represented in about 500 images, each image showing an optical section through a 20 micron-thick slice of brain tissue, with alternate sections being tracer and nissl stained. The tracer injected slices, show the neuron connectivity by tracing the axonal projections, while the nissl stained slices, provide the anatomical knowledge. The information from both nissls and tracers is used to find what region of the brain is connected to what, contributing to formation of connectivity matrix.

Different regions in brain can be distinguished from each other on the basis of cell count, density and the type of cells discovered. This arises the need for segmentation of Nissl stained images.

1.2 Problem Statement

The project has been divided majorly in two stages, namely experimental and data analysis pipeline.
The experimental pipeline involves getting quality images of brain slices, and the data analysis pipeline involves the extraction of relevant information from the same. Data Analysis Pipeline includes 2D Image Analysis (involving Autofocus, Dynamic Ranging, Measuring Tracer Strength, Image Alignment problems), 3D Registration, Detection of Labelled Cells, Reconstruction of Neuronal Projections.
The goal is to develop a software for segmentation of the nissl images. The application developed should have the following features:

- Multiple image handing
- An automated process that segments the cells of interest

1.3 System Overview

The overall system consists of input image, whose groundtruth is already available through annotation.
The segmentation is carried out on it, after required pre-preprocessing steps.
The centers of the segmented regions are compared against those provided through groundtruth.
The performance is analysed through precision and recall values.

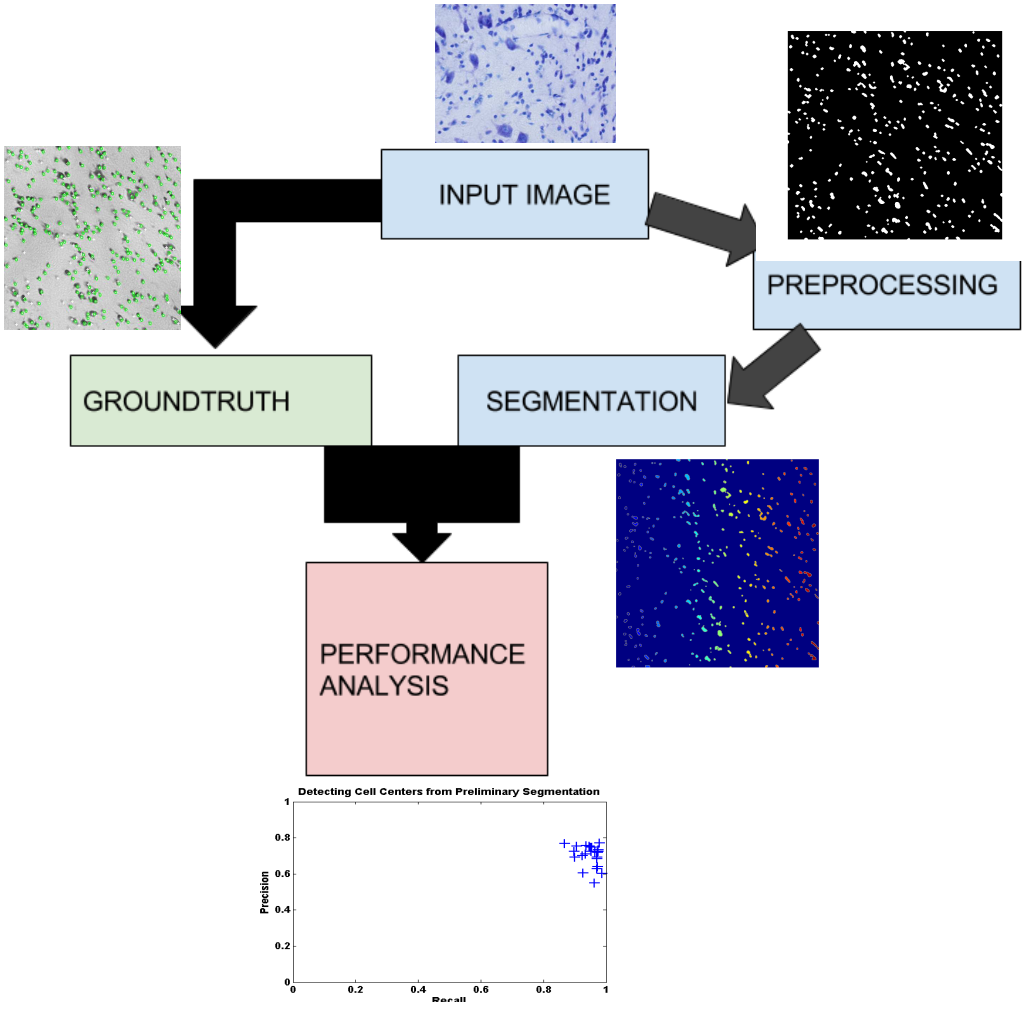

Figure 1.1: System Overview for a sample image segmentation

CHAPTER 2

IMAGE ACQUISITION

2.1 Introduction

Six alternating 20 microns brain slices have been produced by placing on two slides. Each slide that completes a series of slides are named by the following convention PMDXX&XX - with the higher number value representing to the top brain in the column. A consecutive slide number is given to each alternating set; PMDXX&XX - F/N 1,2,3 thus two sets of slides are produced, with alternating section order. Here F stands for fluorescent, N stands for Nissl.

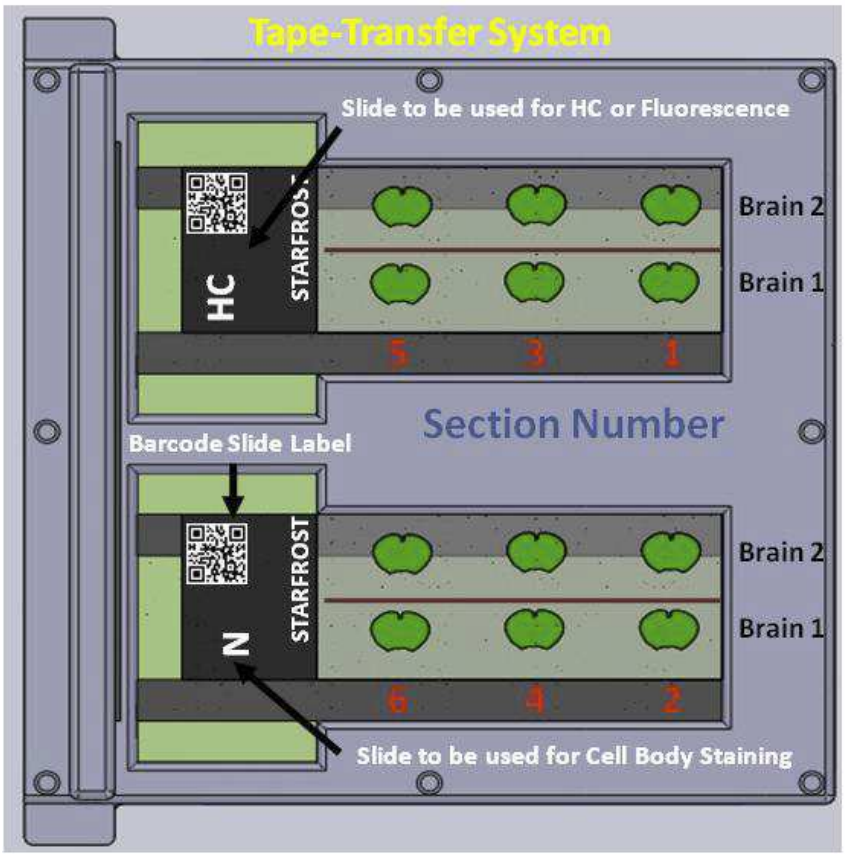

Figure 2.1: Brain slices placed on slides

2.2 Process Description

2.2.1 Slide Loading

Nanozoomer HT 2.0 Virtual Microscopy system (Hamamatsu /Olympus), a digital slide scanner is used to scan the slides in order to produce high quality digital images. Scanning can be done in both bright field (8 bits per color channel) or fluorescence mode (12 bits per color channel). Brightfield scanning is used for Nissl, IHC, and HC stained slides. Fluorescence scanning at 12-bit depth per pixel, per color channel, is used for all other samples (Rabies and AAV injections). A Lumen Dynamics X-Cite exacte light source is used to produce the excitation fluorescence.The slides are loaded into the Nanozoomer for scanning.

Figure 2.2: Nanozoomer HT 2.0

The slides are scanned so that one brain slice is focussed at once. For the proper focussing of the brain slices Regions of Interest (ROIs)/ cropping boxes are marked around each section and 9 - 12 focus points are generated per section. For the fluo-

rescence mode focus points are generated automatically whereas for the bright field scanning it has to be manually done.

2.2.2 Conversion

The .NGR format slide images are cropped into images of individual brain slices. These cropped sections are then converted into JPEG2000 format. Both lossless JP2 and lossy JP2 conversion is done. In cases where the loss in the quality of the images is tolerated lossy JP2 images can be used as they consume less storage space compared to lossless JP2 images. Along with the JPEG2000 format, PNG, TIF and metadata files are generated and are stored in a folder corresponding to that brain slice.

2.2.3 Quality Control

The JPEG2000s generated in the previous step are subjected to quality control. The possible errors in the process include wrong order of the cropped sections, missing sections, errors in cropping, focus issues, duplicates generation and damages in conversion. These errors are identified in the quality control and the corresponding damaged sections are rescanned, refocused and converted manually. If the cropped sections are damaged, the required sections are cropped again.

2.2.4 Analysis

Once the quality check is enforced, the images should be subjected to the following analysis.

- Dynamic range: The range of the pixels is obtained and the contrast is adjusted accordingly for the improvement of the quality of the images.
- Tracer strength: The strength of the tracers is determined.
- Pre-registration: Alignment of images is done to get them ready for registration.

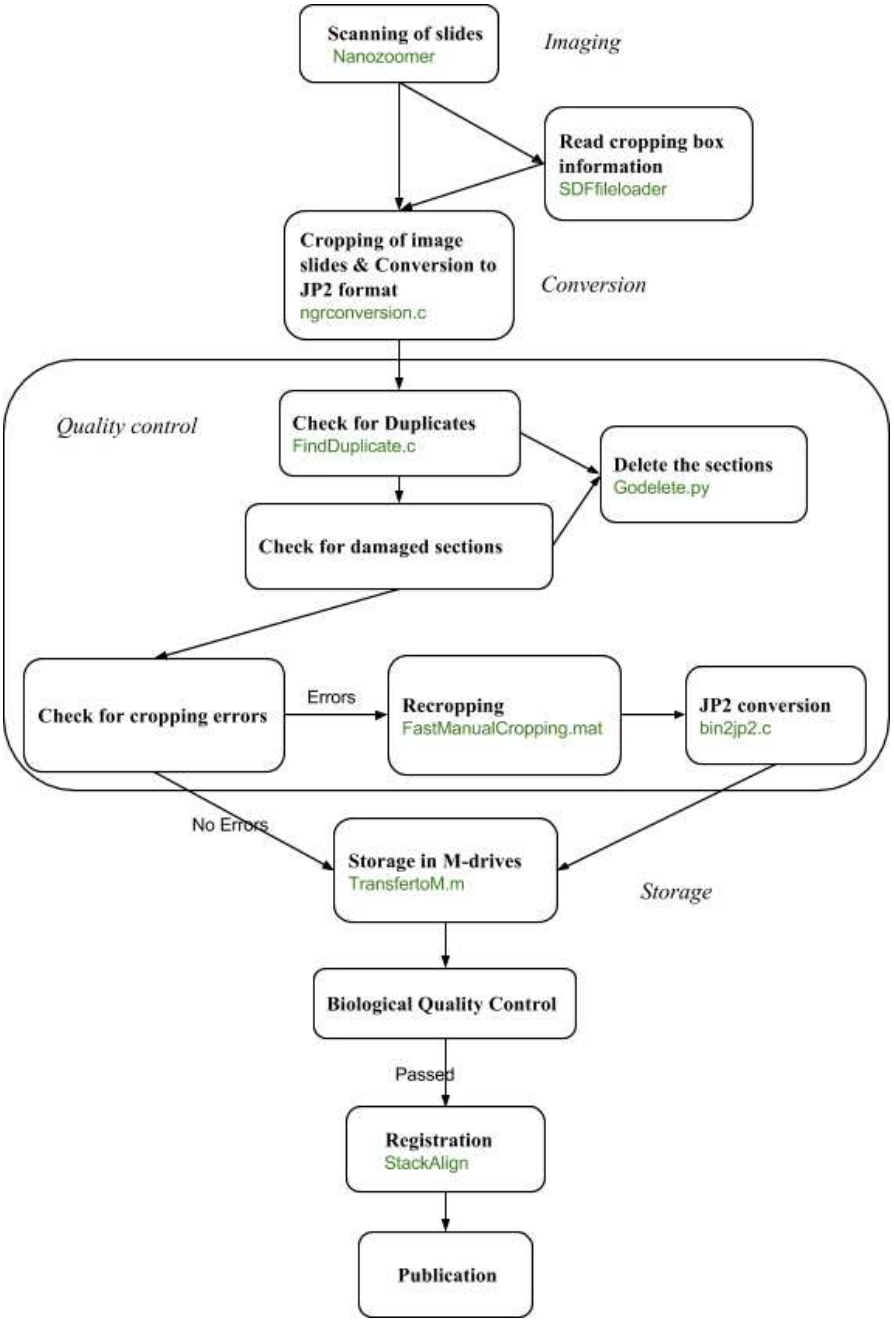

Figure 2.3: Image Acquistion Process Description

CHAPTER 3

BACKGROUND KNOWLEDGE

3.1 Neuron theory

The nerve tissue is composed of individual cells which are fully functional units. Although the morphology of various types of neurons differs in some respects, they all contain four distinct regions with differing functions: the cell body, the dendrites, the axon, and the axon terminals.
The structure of a standard neuron can be seen below.

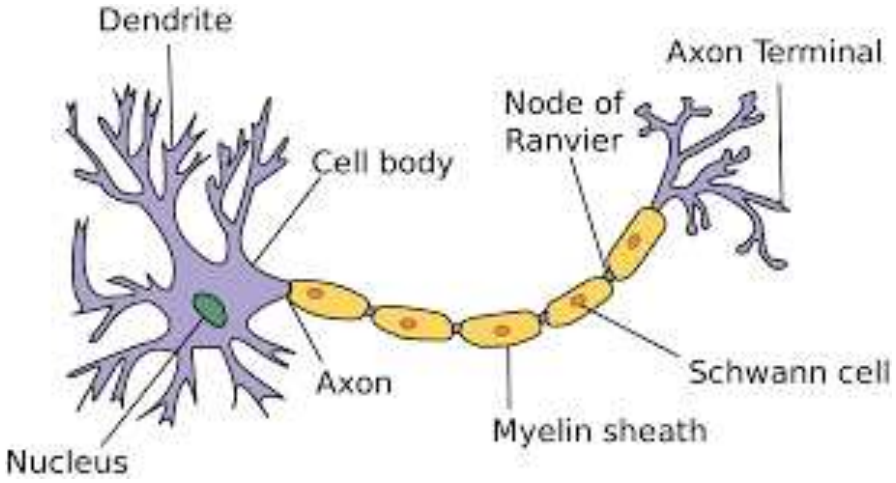

Figure 3.1: Neuron Structure

3.2 Tracers

The anterograde and retrograde tracers are used for visualization of the biological process of axonal transport as described here[35].
Anterograde tracing is a research method which is used to trace axonal projections from their source (the cell body or soma) to their point of termination (the synapse). While retrograde tracing traces neural connections from their termination to their source (i.e. synapse to cell body).

Tracer	Type	Color
biotinylated dextran amine (BDA)	classical anterograde	Blue
adeno-associated virus	viral anterograde	Red
cholera toxin subunit B (CTB)	classical retrorograde	Black
modified rabies virus RV-4GFP (RV)	viral retrorograde	Green

Table 3.1: Tracers used in the process

3.3 Nissl Staining

Staining is basically the process of making a compound of interest (DNA, proteins) more easily identifiable by enhancing its visibility through certain chemical means. The online medical dictionary defines staining as 'the use of a dye, reagent, or other material for producing coloration in tissues or microorganisms for microscopic examination'.
Nissl cell body is a dense granular mass often found in nerve cells. These granules are of rough endoplasmic reticulum (RER) with rosettes of free ribosomes, and are the site of protein synthesis.
Nissl staining stains the Nissl bodies in cells, technically called the endoplasmic reticulum. The staining procedure, discussed here[20] consists of sequentially dipping the mounted brain slices in about a dozen different solutions for specified amounts of time.

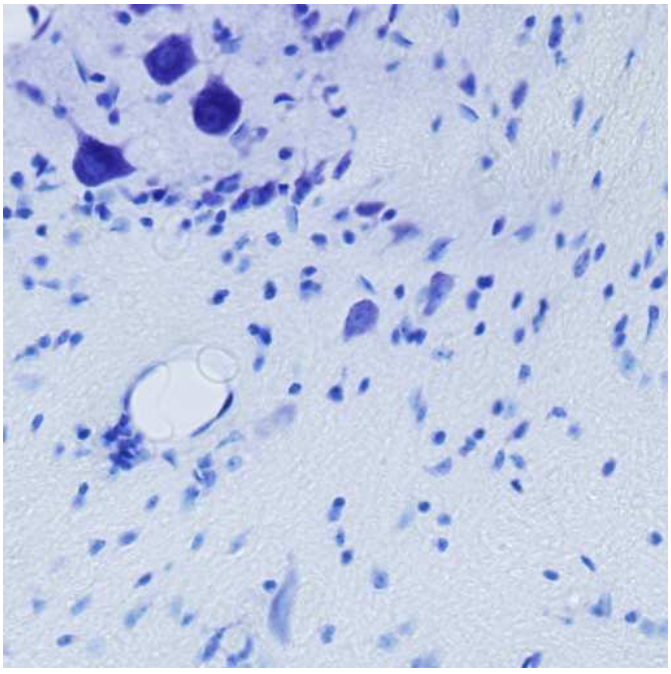

Figure 3.2: Nissl Stained Image

3.4 Dataset

Alternate slices are nissl stained and tracer stained, such that the they can be registered together. Nissl images provide information about neuroanatomy, whereas tracer images help with the neuronal connectivity[18].

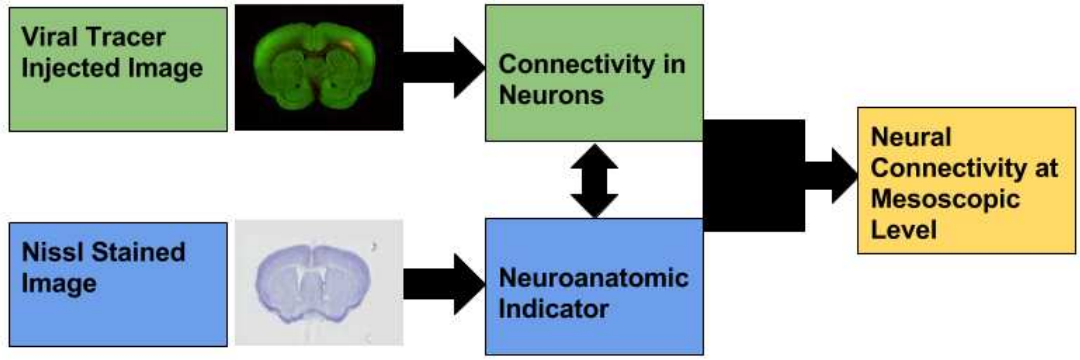

Figure 3.3: Process flow

240 to 260 slices are generated from each mouse brain, with size 18000 x 24000 pixels. Each slice has been cropped into smaller region of interests of size 512 x 512 pixels, for easier implementation.

The main issue with Nissl images are that most of the cells are clumped together and there are sometimes with some background halo.

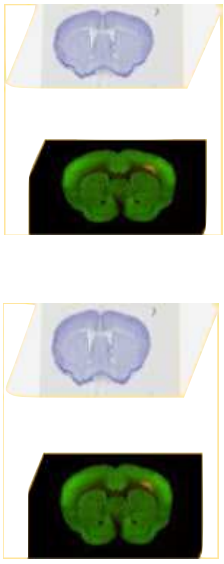

Figure 3.4: Alternate Nissl and Fluroscent sections

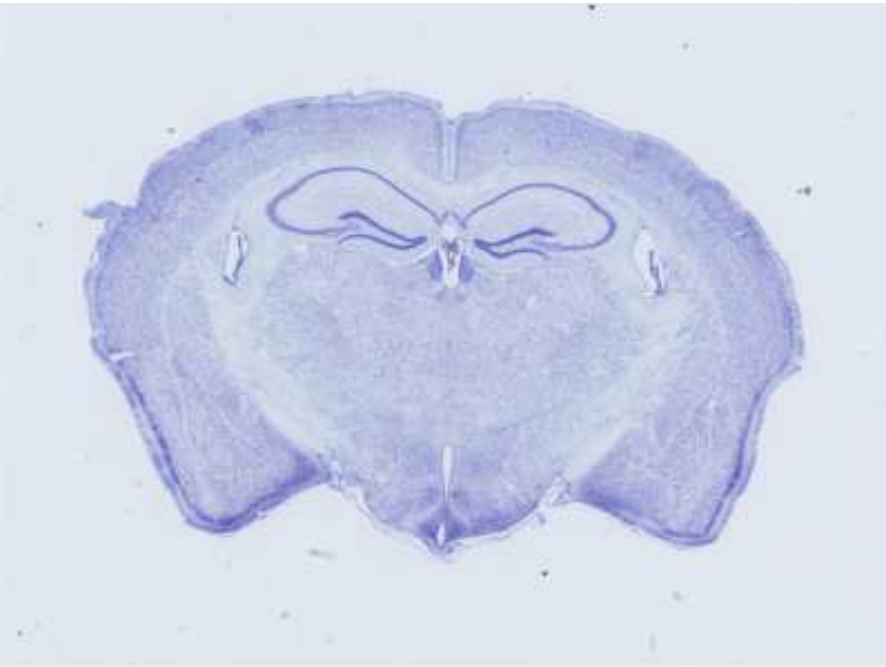

Figure 3.5: Image Slice full(18000 x 24000 px)

3.5 Project description

The Mouse Brain Architecture project[30] has been divided majorly in two stages, namely experimental and data analysis pipeline.
The experimental pipeline involves getting quality images of brain slices, and the data analysis pipeline involves the extraction of relevant information from the same. While, data Analysis Pipeline includes 2D Image Analysis (involving Autofocus, Dynamic Ranging, Measuring Tracer Strength, Image Alignment problems), 3D Registration, Detection of Labelled Cells, Reconstruction of Neuronal Projections.

3.6 Softwares/Tools/Libraries

There are numerous softwares, tools and libraries available for digital image processing in the medical domain. These perform vast functions like segmentation, 3D model building, registration, image correction, tissue classification, object recognition and tracking, visualization for different image modalities.
Vigra[37], itk-SNAP[40], Advanced Normalization Tools (ANTS)[4], Ilastik[33], ImageJ[31] can be used support most of the image processing functions for various image modalities. There are toolkits available which especially deal with optical imaging such as

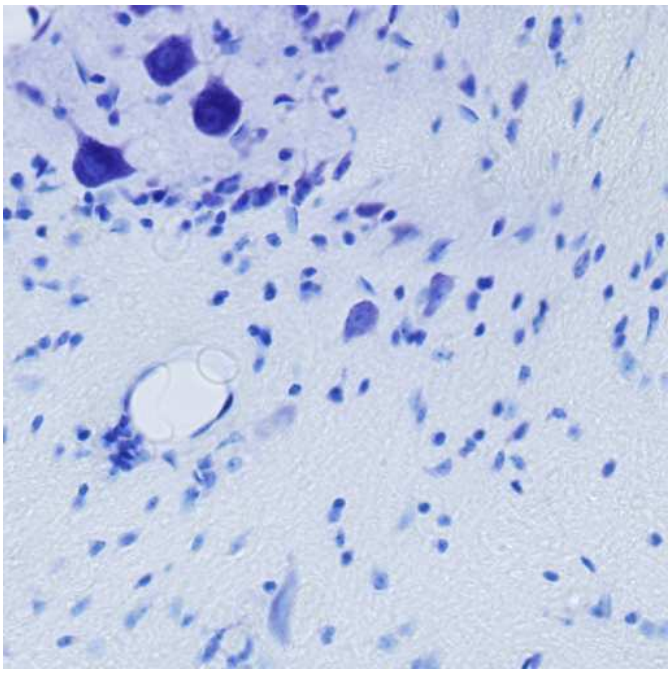

Figure 3.6: ROI extracted from slice(512 x 512 px)

Farsight[8], CMTK (Computational Morphometry Toolkit)[1] is widely used for MR images. For EM images; Gala[25], CytoSeg can be used.

A study was conducted on these, comparing the programming languages, image modalities supported by each, quality of online help available, along with the ease of use.

Ilastik[33] has been used as a training software in this thesis. It is a simple, user-friendly tool for interactive image classification, segmentation and analysis. It is built as a modular software framework, which currently has workflows for automated (supervised) pixel- and object-level classification, automated and semi-automated object tracking, semi-automated segmentation and object counting without detection. It also allows easy batch processing.

CHAPTER 4

LITERATURE SURVEY

The main goal of segmentation is to partition the image into regions sharing similar features. Few common segmentation methods, compared here[10] can be classified as:

- Region Based
- Edge Based
- Intensity Based

4.1 Region Based Segmentation

In this type of segmentation, discussed here[12], seeds are chosen, based on user criterion, and based on region membership criterion, growing[19] or splitting or merging[36, 7] happens. This depends highly on seed[3] picked, and can fail if seed lies on an edge. We have discussed one such method, based on concept of watersheds[23] in the thesis. This category also includes clustering based approaches[38], where pixels are clustered in a feature space using any discriminating feature associated and then connecting regions are found. A graph based segmentation[11] method has been implemented for our purpose.
In such methods, post processing is often necessary.

4.2 Edge Based Segmentation

These methods[5] rely on the discontinuities in the image data to locate boundaries, and hence segment. It includes gradient operator based edge detection[9], laplacian of gaussian zero crossing edge detection[15], short response hilbert transform[29] . Edge detection based methods have been used as a intermediate steps in both second and third approaches, for the same.
These type of methods, often lead to false edge detections, missed edge detections,

and difficulty in selecting an appropriate threshold for separating significant and non-significant edge information.

4.3 Intensity Based Segmentation

The image histogram is analysed, and the peaks and valleys are used to get information of number of clusters in the image, as described here[21, 24, 32]. The result obtained by this method is not perfect, for complex uses. This also is a vital step in pre-processing.

CHAPTER 5

ANNOTATION

Image annotation is required for both training and testing purposes. This has been achieved by manually labeling the cell and non-cell regions, and by crowdsourcing the images for annotation.

5.1 Data Preparation

Since the data consisted of small sized ROI images(512 x 512 px), there were several images with no cell regions. This posed a need to automate classification of images into tissue and non-tissue images. This was achieved on the basis of entropy and background pixel calculation after preprocessing, with an error rate or 0.36%.

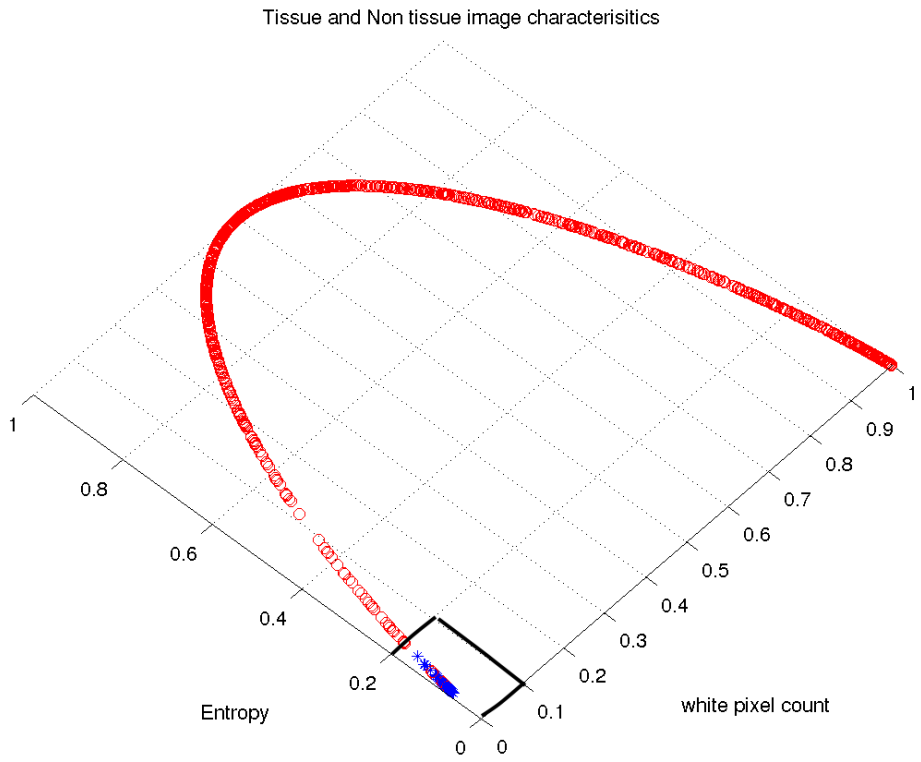

Figure 5.1: Separation surface for tissue and non tissue images

5.2 Manual Annotation

Interactive training software, ilastik is used to train the pixel classifier, for cellular, non-cellular and boundary regions as three labels .The training workflow, has been used for predicting regions in a new image. A separate workflow has been trained to be used as groundtruth for performance of the algorithms.

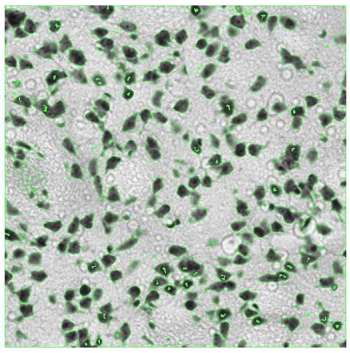

Figure 5.2: Manually labelling

5.3 Crowdsourcing

Amazon Mturk and Crowdflower were used for image annotation to get an exhaustively labelled dataset. Study about the dependence of task performance in crowdsourcing on various parameters was considered, as described in [17] and image size, task size, reward for task was set as per this. However the results achieved from these exhibited high inaccuracy, due to lack of prior knowledge.

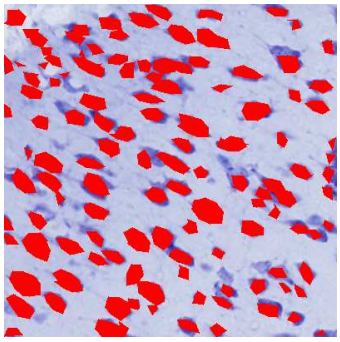

Figure 5.3: Labelling from crowdflower

CHAPTER 6

GRAPH-BASED SEGMENTATION

This algorithm[11] represents the problem of image segmentation as a graph based problem, graph $G = (V, E)$, where each pixel in the image becomes a node $v_i \in V$, and the edges E connect certain pairs of neighbouring pixels. With every edge a weight is associated based on some property of the pixels that it connects, such as their image intensities.

There are different ways to measure the quality of a segmentation but in general we want the elements in a component to be similar, and elements in different components to be dissimilar. This means that edges between two vertices in the same component should have relatively low weights, and edges between vertices in different components should have higher weights.

6.1 Pairwise Region Comparison

A minimum spanning tree(MST) is a spanning tree of a connected, undirected graph. It connects all the vertices together with the minimal total weighting for its edges.

The internal difference of a component $C \subseteq V$ to be the largest weight in the minimum spanning tree of the component, $MST(C, E)$.That is,

$$Int(C) = max_{e \in MST(C,E)} w(e)$$

The difference between two components C_1,$C_2 \subseteq V$ is the minimum weight edge connecting the two components. That is,

$$Dif(C_1, C_2) = min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w(v_i, v_j)$$

If there is no edge connecting C_1 and C_2 we let $Dif(C_1, C_2) = \infty$.

The pairwise comparison predicate, is defined as:

$$D(C_1, C_2) = \begin{cases} true & \text{if } Dif(C_1, C_2) > min(Int(C_1) + \tau(C_1), Int(C_2) + \tau(C_2)) \\ false & \text{otherwise} \end{cases}$$

where,τ is a threshold function and $\tau(C) = k/|C|$, with k being a constant parameter.

6.2 Algorithm

Weights on each edge measure the dissimilarity between pixels(intensities).

For two points, p and q on the image, $edgeweight(p1, p2) = \frac{((r_p - r_q)^2 + (g_p - g_q)^2 + (b_p - b_q)^2)^{1/2}}{((x_p - x_q)^2 + (y_p - y_q)^2)^{1/2}}$

- Sort edgeweights, E into (o_1, ..., o_m), by non-decreasing edge weight.
- Initial segmentation, S^0: Each vertex is in its own component.
- Internal difference of a component C to be the largest weight in the minimum spanning tree of the component.
- Construct S^q from S^{q-1} : Let v_i and v_j denote the vertices connected by the q^{th} edge in the ordering, i.e., o_q = (v_i, v_j). If v_i and v_j are in disjoint components of S^{q-1} and w(o_q) is small compared to the internal difference of both those components, then merge the two components otherwise do nothing.
- Return $S = S^m$ after m iterations.

6.3 Results

Results of the graph based segmentation are below, with different region has been labelled with a unique color.

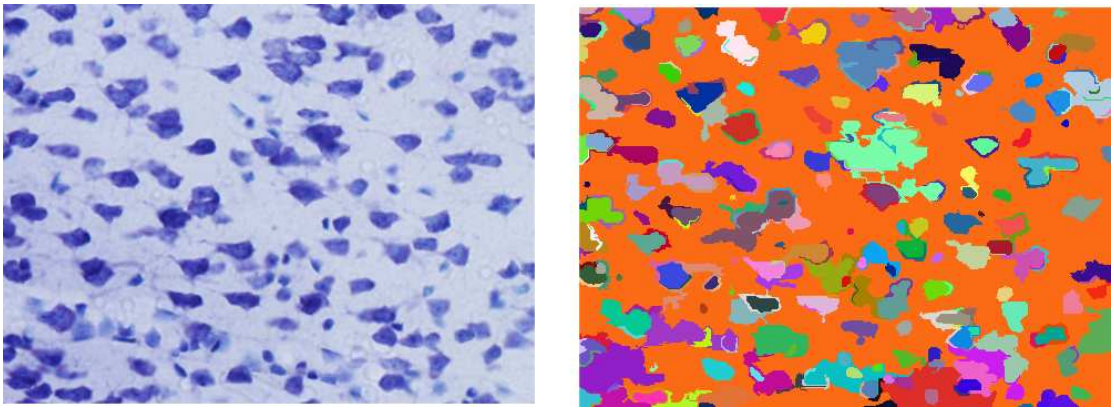

Figure 6.1: Graph Based Segmentation Results(1)

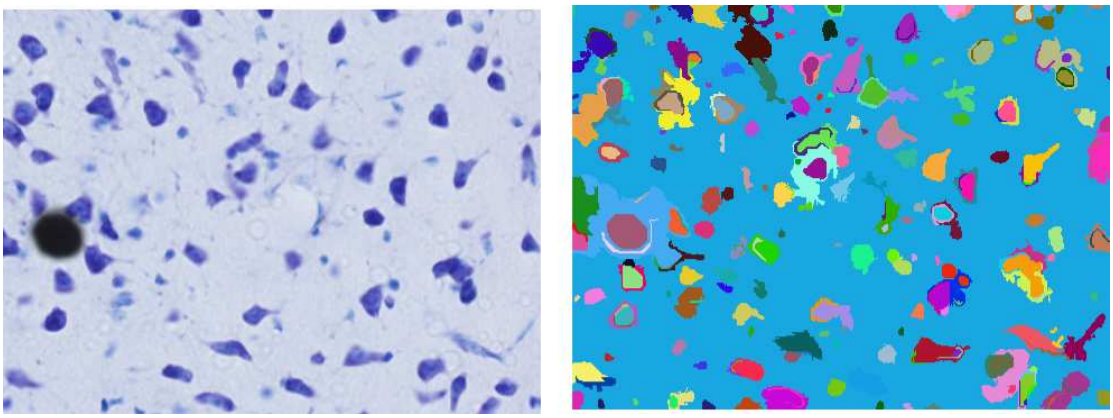

Figure 6.2: Graph Based Segmentation Results(2)

6.4 Performance Analysis

The performance has been measured using precision and recall values, comparing to the groundtruth values.

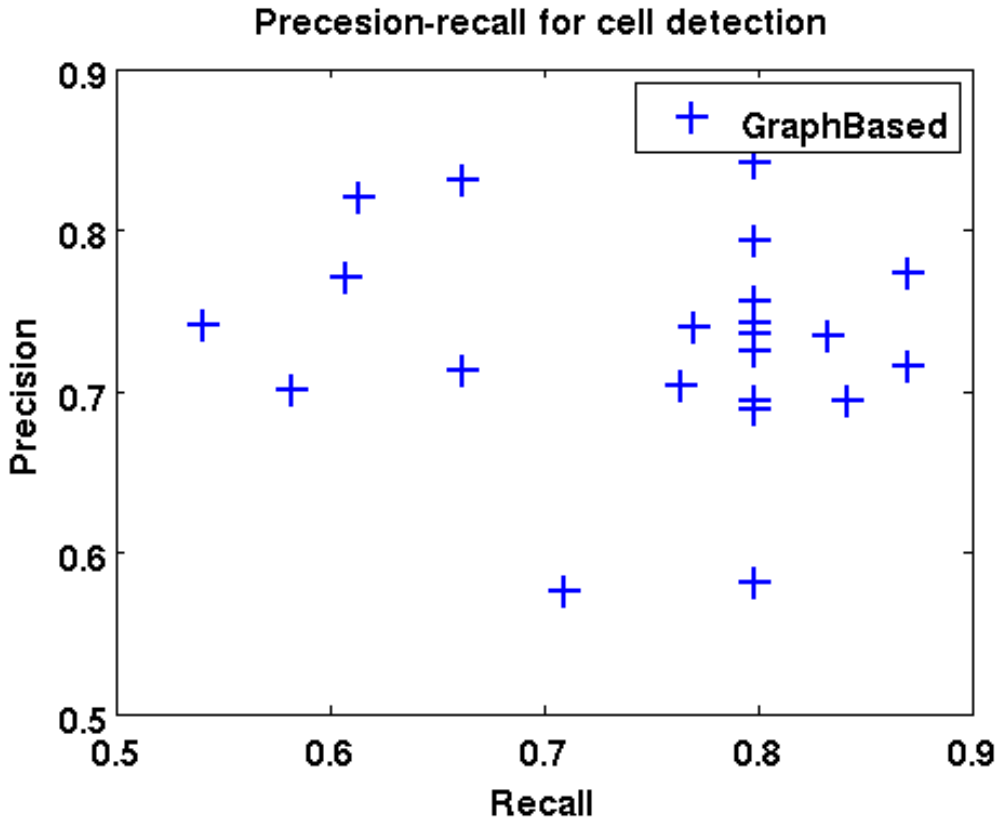

Figure 6.3: Graph Based Segmentation Results

Measurement	Value
Precision	0.6723
Recall	0.7506
F-score	0.7093

Table 6.1: Performance analysis for graph based segmentation

CHAPTER 7

WATERSHED SEGMENTATION

7.1 Introduction

The term watershed refers to a ridge that divides areas drained by different river systems. A catchment basin is the geographical area draining into a river or reservoir.
A watershed is introduced as the set of points where a drop of water, falling there, may flow down towards several catchment basins of the relief.

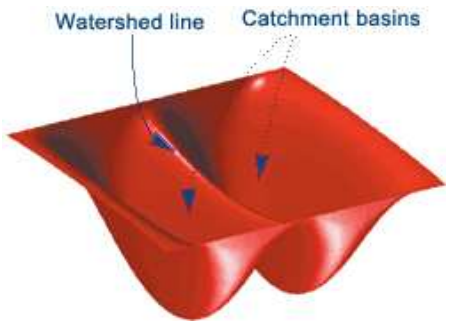

Figure 7.1: Watershed lines and basins

Every grayscale image can be considered as a topographic surface. The gray level of a pixel becomes the elevation of a point. The basins and valleys of the relief correspond to the dark areas, whereas the mountains and crest lines correspond to the light areas[23].

7.2 Algorithm

The following steps are involved in the watershed based segmentation:

1. **Training predictions** The workflow trained on a separate set of images, is used to predict the labels for test image.

Figure 7.2: Watersheds on a grayscaled image

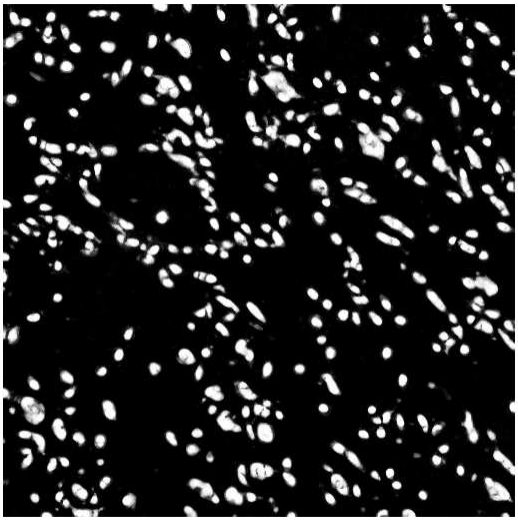

Figure 7.3: Cell region labels prediction

2. **Otsu Thresholding**
   The method[2] assumes that the image contains two classes of pixels following bi-modal histogram (foreground pixels and background pixels), it then calculates the optimum threshold separating the two classes so that their intra-class variance is minimal, and their inter-class variance is maximal.

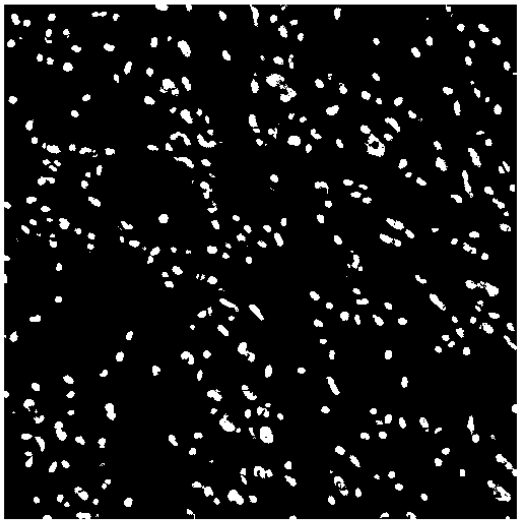

Figure 7.4: Otsu Thresholding result

3. **Closing**
   This morphological operation involves dilation followed by erosion. It is used to fill background regions of the image.

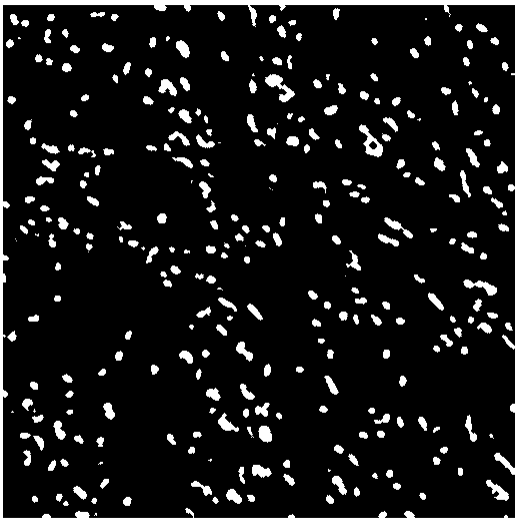

Figure 7.5: Closing result

4. **Opening**
   This morphological operation involves erosion followed by dilation. It is used to remove noise from the image.

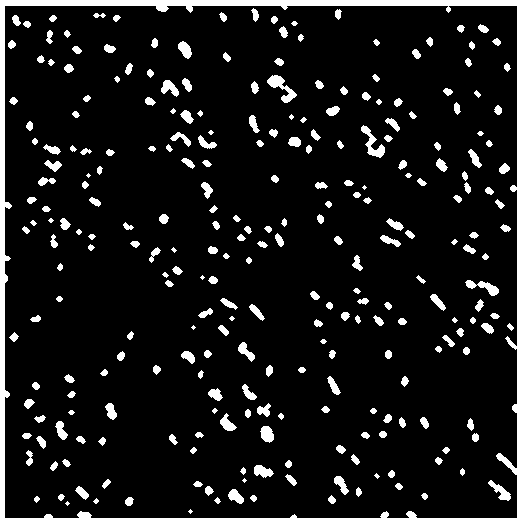

Figure 7.6: Opening result

5. **Distance Transform**
   This morphological operator[26, 22] gives a graylevel image that looks similar to the input image, except that the graylevel intensities of points inside foreground regions are changed to show the distance to the closest boundary from each point. This provides us with a topographic surface.

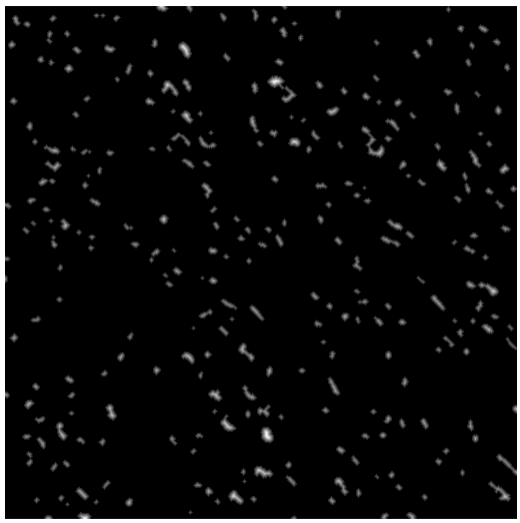

Figure 7.7: Distance transform result

6. **Watershed**
   Watershed transform is applied to the result of distance transform to separate touching objects.

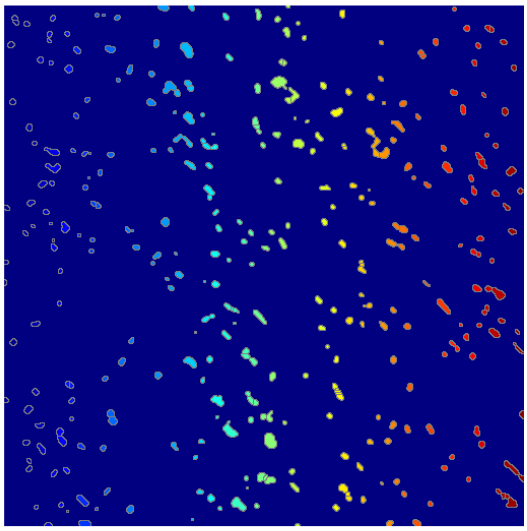

Figure 7.8: Watershed result

7.3 Results

Each segmented area has been labelled with a unique color id.

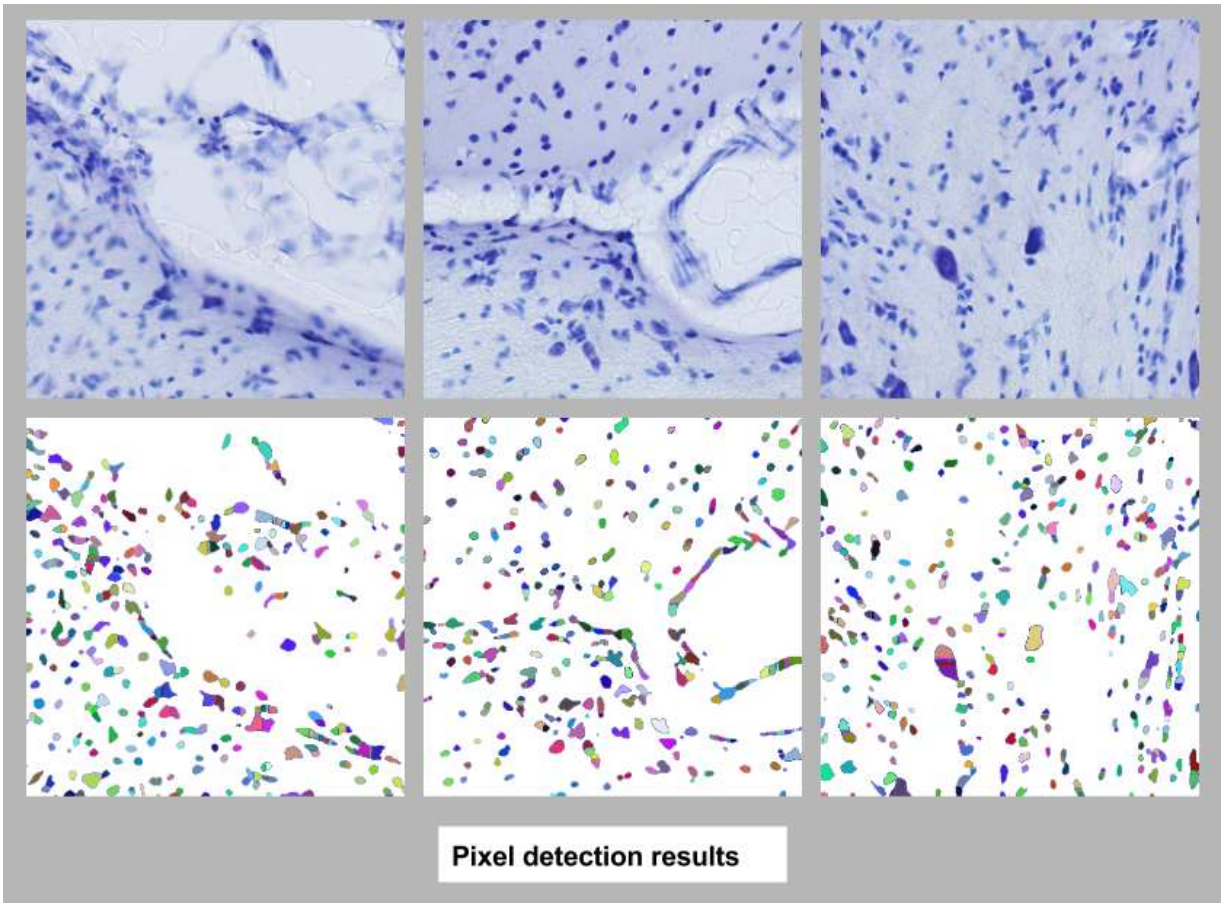

Figure 7.9: Results

7.4 Performance analysis

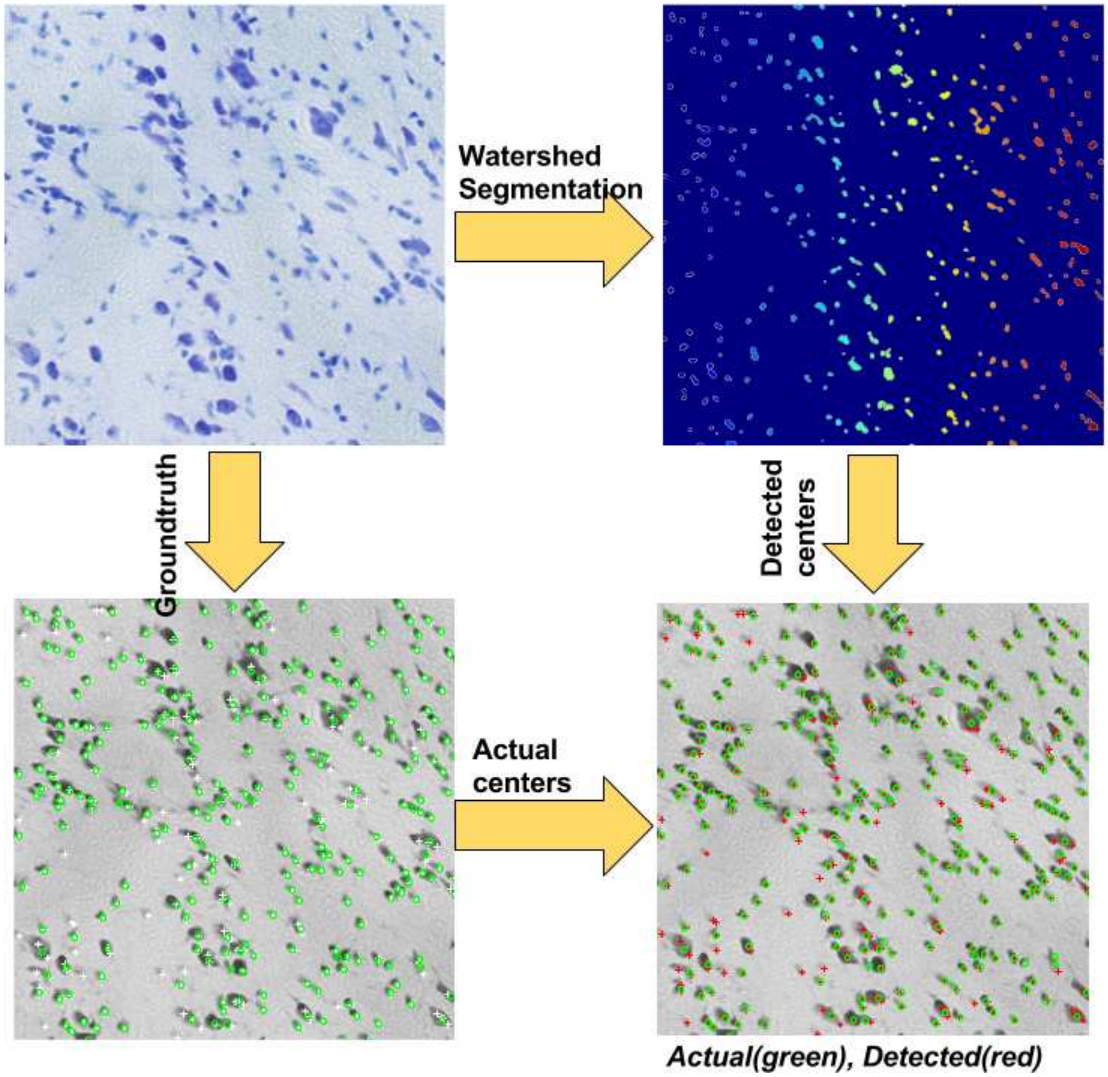

Figure 7.10: Performance evaluation process flow

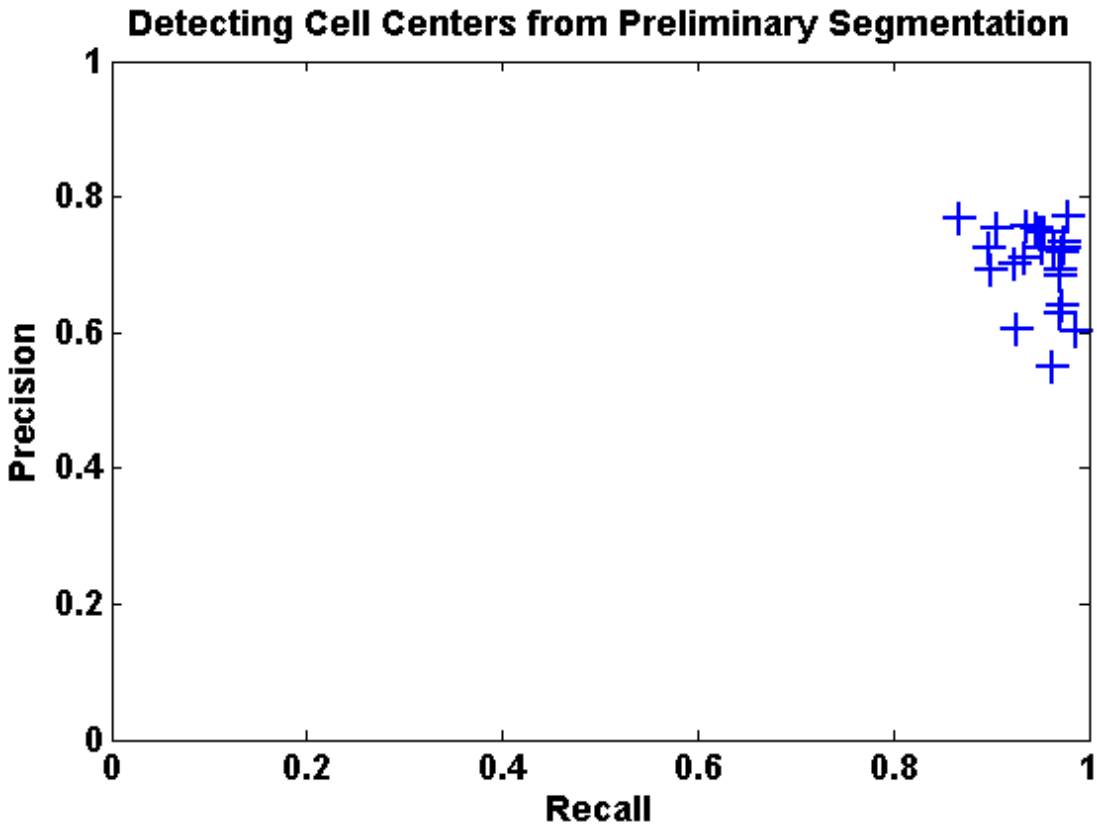

Figure 7.11: Precision recall curve

Measurement	Value
Precision	0.7066
Recall	0.9557
F-score	0.8125

Table 7.1: Performance analysis for watershed segmentation

CHAPTER 8

CONCAVE POINT BASED SPLITTING

8.1 Introduction

Splitting touching cells is important for medical image processing and analysis system. The watershed based result showed undersegmented regions, at places where touching cells have been encountered. A concave point extraction has been used to split touching cells as in [34, 39], which are not handled by watershed. There are other methods which perform graph cut aimed at segmenting touching cells[14], but these have not been implemented considering the image size.

8.2 Algorithm

This algorithm uses convex hull and the nearby points to detect concave points, as discussed here [34].

Figure 8.1: Input image

8.2.1 Pre-processing

- The image is smoothed to remove any noise, using gaussian filter.
- This is followed by image enhancement to fix any contrast based issues.
- The image is binarized using Otsu's technique.
- The regions with very small area have been removed as noise.

Figure 8.2: Image post preprocessing

8.2.2 Concave Point Detection

The binary image is subtracted from its convex hull, giving us image with concave regions. This is done for each connected component discovered in the image.

Figure 8.3: Binary image

Figure 8.4: Convex Hull

Figure 8.5: Difference

The image with the concave regions is binarized and its centroid O is computed.
For every concave region, two regions in the image here, the point closest to the centroid O are acquired, J and $J^{'}$ in the image below, which are at distances $d1$ and $d1^{'}$ respectively from O.
Two neighbor points of the point $J(x, y)$ on the contour are firstly founded, that is, the

previous point $J1$ and the following point $J2$. The horizontal coordinate of $J1$ is x-δ, and the horizontal coordinate of $J2$ is x+δ, where $\delta = 7px$.

Then the midpoint J_0 of the point $J1$ and the point $J2$ are computed, and the distance $d2$ between the point O and the point $J0$ can be acquired.

If $d1 < d2$, then the point of interest, J in this case is a concave point.

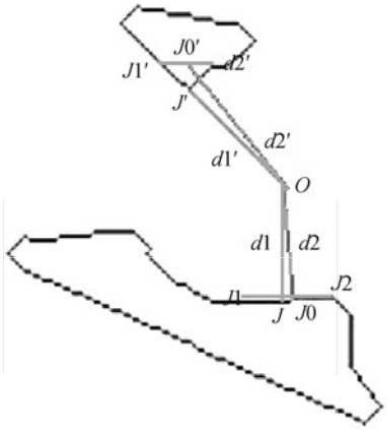

Figure 8.6: Detection of concave points

From the sample image, point J is a concave point, whereas point $J^{'}$ is not a concave point.

8.2.3 Cell Type Decision

Touching cells have been categorized into three types, namely, series, parallel and complex. Let the number of concave points be A and the number of touching cells, be M.

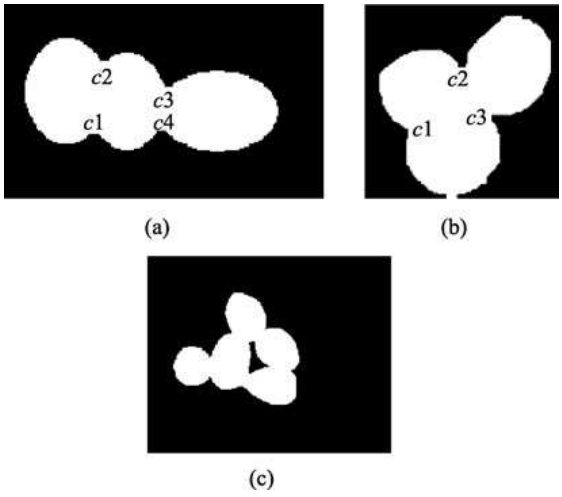

Figure 8.7: (a) Series, (b) Parallel, (c) Complex

If $A = M$, parallel cell arrangement. $2M - 2 = A$, implies a series cell arrangement. Every other, cell arrangement is taken as complex.

The number of touching cells has been found by eroding the initial binarized image repeatedly, and counting every single region which disappears.

8.2.4 Splitting series cells

In case of series touching cells, as seen in fig 7.7(a), the split is made along the axes perpendicular to cell direction.

For this the skeleton and the joints of each skeleton are found. The concave points are

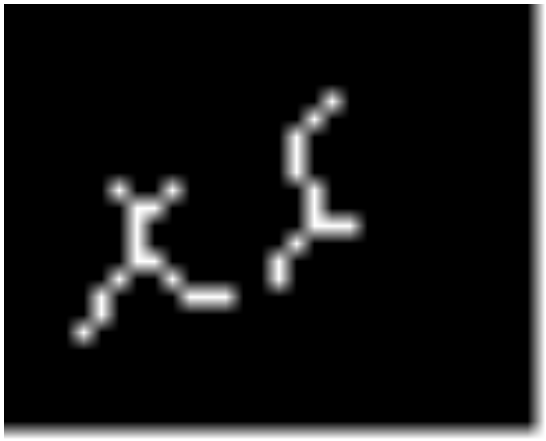

Figure 8.8: Skeleton

Figure 8.9: Skeleton Joints

paired in groups of two as per closest to centroid of a region in joints, and farthest from other regions.

The pairs obtained thus are used to split the cell.

8.2.5 Splitting parallel cells

In case of parallel touching cells, as seen in fig 7.7(b), the centroid of the component is found, and the split is made by joining the centroid to each of the concave points.

This results in M cells.

8.2.6 Splitting complex cells

For complex cells, watershed has been implemented with preprocessing as discussed in the previous chapter.

The splitting has been done by find the two points to be split through and using bresenham's line algorithm[6] for the splitting, in all cases.

Figure 8.10: Input subimage

Figure 8.11: Output after splitting

8.3 Results

This algorithm was tested on the same set of images, as the previous algorithm. The input was the predictions from the training workflow.

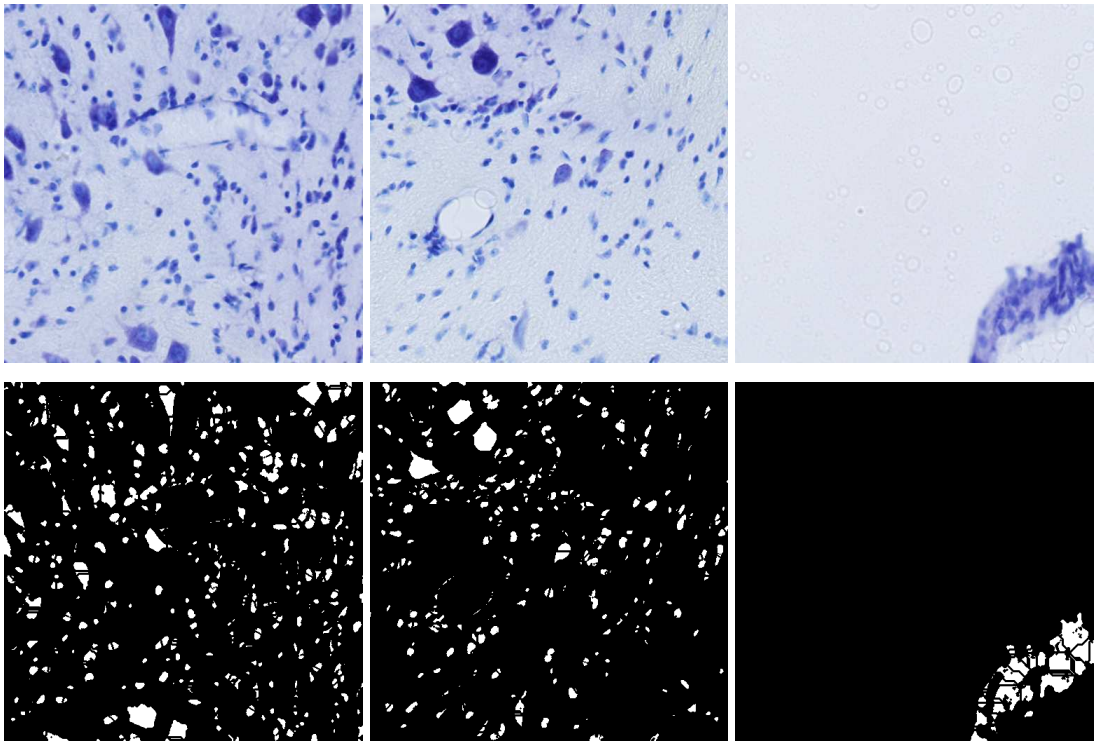

Figure 8.12: Splitting results(bottom row) for images(top row)

8.4 Performance Analysis

The precision recall plot shows two outliers as the algorithm fails in the case of highly dense tissue regions.

Removing those, the results seem to improve the performance from the last algorithm.

Measurement	Value
Precision	0.7122
Recall	0.9695
F-score	0.8211

Table 8.1: Performance analysis for concave point based splitting

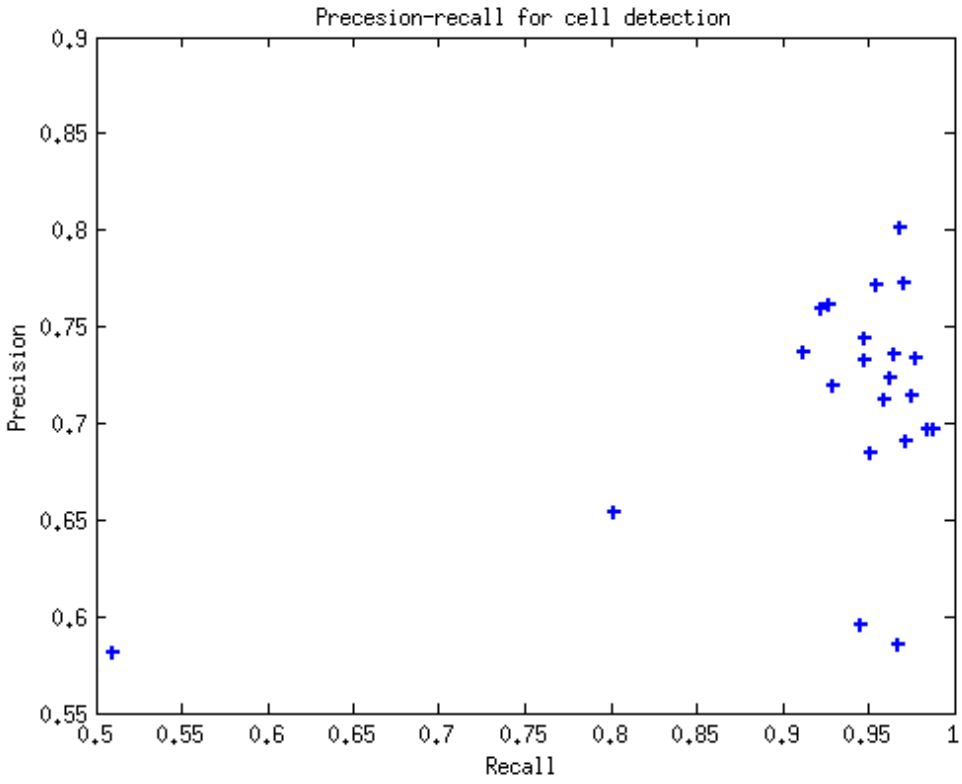

Figure 8.13: Precision recall plot

CHAPTER 9

CONCLUSION AND FUTURE WORK

This thesis discusses three approaches towards Nissl stained cell segmentation, namely Graph Based Segmentation, Watershed Transformation based segmentation and Concave Point Based Splitting.

Both the approaches perform better as compared to the graph based segmentation.

We compare the other two approaches, to find that Concave Point Based Splitting performs better in general, but fails in the presence of dense cells. While watershed algorithm works well in both cases.

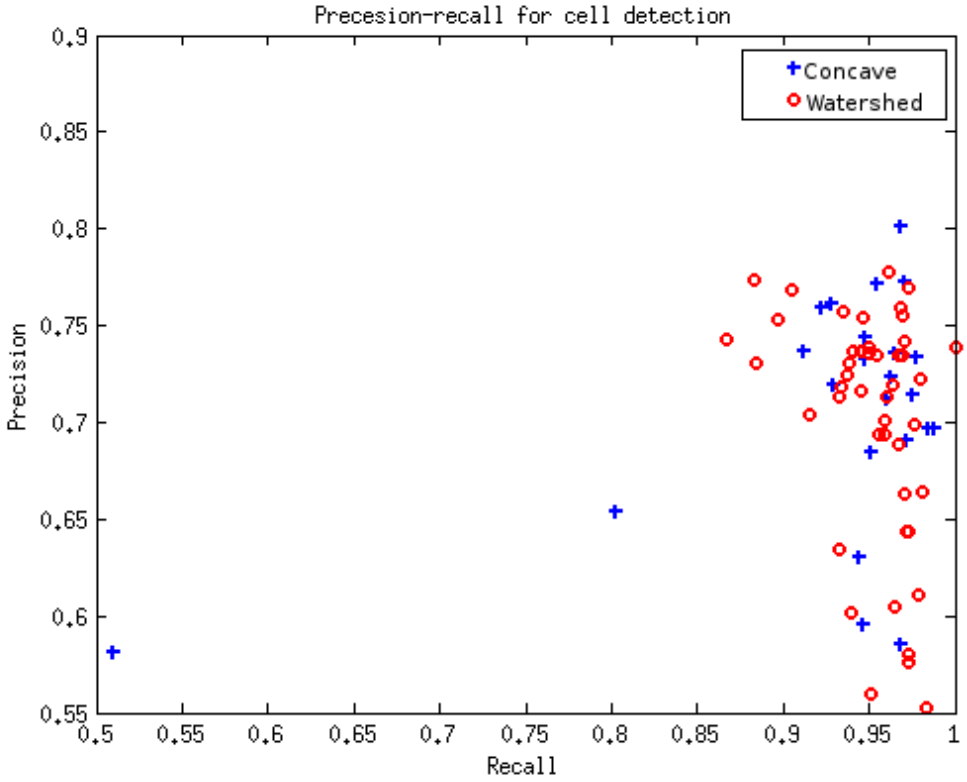

Figure 9.1: Precision recall comparison

Below are the failure cases for second approach.

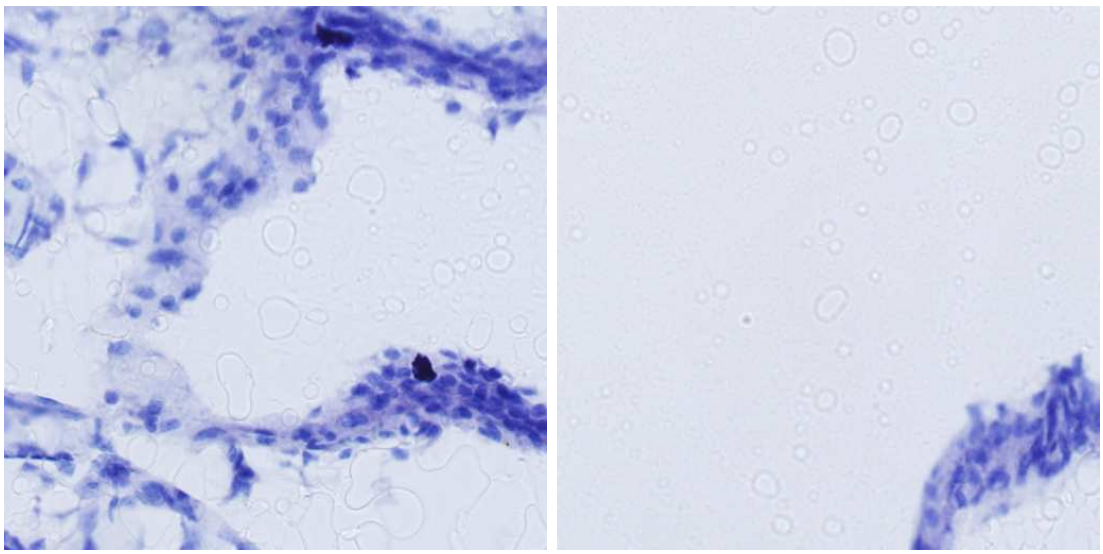

Figure 9.2: Failure images for concave point based splitting

The watershed algorithm can be used in more high density tissue regions, while concave point based splitting can be used otherwise to produce high quality segmentation results.

A region merging algorithm can be used in addition to these since most of these algorithms, lead to over-segmented results.

Cells segmented hence, need to be analysed to study morphological parameters like shape, size, color, density at various areas. These help us differentiating one area in the brain from the other.

Once there is anatomic information through the Nissl stained images, it can be used to derive the connectivity matrix with the help of neuronal connectivity knowlodge.

REFERENCES

[1] The computational morphometry toolkit (cmtk).

[2] A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, Jan 1979.

[3] R. Adams and L. Bischof. Seeded region growing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(6):641–647, 1994.

[4] B. B. Avants, N. J. Tustison, J. Wu, P. A. Cook, and J. C. Gee. An open source multivariate framework for n-tissue segmentation with evaluation on public data. 2011/2012.

[5] M. Brejl and M. Sonka. Object localization and border detection criteria design in edge-based image segmentation: automated learning from examples. *Medical Imaging, IEEE Transactions on*, 19(10):973–985, 2000.

[6] J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems journal*, 4(1):25–30, 1965.

[7] P. C. Chen and T. Pavlidis. Segmentation by texture using a co-occurrence matrix and a split-and-merge algorithm. *Computer graphics and image processing*, 10(2):172–182, 1979.

[8] B. CS, G. Lin, Y. Al-Kofahi, N. A, K. L. Smith, W. Shain, and B. Roysam. Associative image analysis: a method for automated quantification of 3d multi-parameter images of brain tissue. *Journal of neuroscience methods*, 2008.

[9] S. Di Zenzo. A note on the gradient of a multi-image. *Computer vision, graphics, and image processing*, 33(1):116–125, 1986.

[10] F. J. Estrada and A. D. Jepson. Benchmarking image segmentation algorithms. *International Journal of Computer Vision*, 85(2):167–181, 2009.

[11] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181.

[12] S. Gould, T. Gao, and D. Koller. Region-based segmentation and object detection. In *Advances in neural information processing systems*, pages 655–663, 2009.

[13] He. *Scientific Reports 5, no. 12089*, 2015.

[14] Y. He, H. Gong, B. Xiong, X. Xu, A. Li, T. Jiang, Q. Sun, S. Wang, Q. Luo, and S. Chen. icut: an integrative cut algorithm enables accurate segmentation of touching cells. *Scientific Reports*, 2015.

[15] A. Huertas and G. Medioni. Detection of intensity changes with subpixel accuracy using laplacian-gaussian masks. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (5):651–664, 1986.

[16] Inglis. 2008.

[17] H. Irshad, L. Montaser-Kouhsari, G. Waltz, O. Bucur, J. Nowak, F. Dong, N. Knoblauch, and A. H. Beck. Crowdsourcing image annotation for nucleus detection and segementation in computational pathology: Evaluating experts, automated methods and the crowd.

[18] B. JW, W. C, B. H, B. H, B. M, and B. HC. A proposal for a coordinated effort for the determination of brainwide neuroanatomical connectivity in model organisms at a mesoscopic scale. 2009.

[19] S. Kamdi and R. Krishna. Image segmentation and region growing algorithm. *International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume*, 2, 2012.

[20] W. Kuhnel. Color atlas of cytology, histology, and microscopic anatomy. *page 182,ISBN 978-3-13-562404-4*, 2003.

[21] M. E. Leventon, O. Faugeras, W. E. L. Grimson, and W. M. Wells. Level set based segmentation with intensity and curvature priors. In *Mathematical Methods in Biomedical Image Analysis, 2000. Proceedings. IEEE Workshop on*, pages 4–11. IEEE, 2000.

[22] C. Maurer, R. Qi, and V. Raghavan. A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 2*, pages 265–270, 2003.

[23] F. Meyer. Topographic distance and watershed lines. *Signal processing*, 38(1):113–125, 1994.

[24] R. Nagarajan. Intensity-based segmentation of microarray images. *IEEE Transactions on Medical Imaging*, 22(7):882–889, July 2003.

[25] J. Nunez-Iglesias, R. Kennedy, T. Parag, J. Shi, and D. B. Chklovskii. Machine learning of hierarchical clustering to segment 2d and 3d images. *PLoS ONE*, 8(8):1–11, 08 2013.

[26] D. Paglieroni. Transforms distance: Properties and machine vision applications. *Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing, Vol. 54, No. 1*, pages 57–58, 1992.

[27] T. Parag, D. C. Ciresan, and A. Giusti. Efficient classifier training to minimize false merges in electron microscopy segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 657–665, 2015.

[28] T. Parag, S. Plaza, and L. Scheffer. Small sample learning of superpixel classifiers for em segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2014*, pages 389–397. Springer, 2014.

[29] S.-C. Pei, J.-J. Ding, J.-D. Huang, and G.-C. Guo. Short response hilbert transform for edge detection. In *Circuits and Systems, 2008. APCCAS 2008. IEEE Asia Pacific Conference on*, pages 340–343, Nov 2008.

[30] M. PP. The circuit architecture of whole brains at the mesoscopic scale. 2014.

[31] C. A. Schneider, W. S. Rasband, K. W. Eliceiri, et al. Nih image to imagej: 25 years of image analysis. *Nat methods*, 9(7):671–675, 2012.

[32] L. Shafarenko, M. Petrou, and J. Kittler. Histogram-based segmentation in a perceptually uniform color space. *Image Processing, IEEE Transactions on*, 7(9):1354–1358, 1998.

[33] C. Sommer, C. Straehle, U. KÃűthe, and F. A. Hamprecht. Ilastik: Interactive learning and segmentation toolkit. In *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 230–233, March 2011.

[34] H. Song, Q. Zhao, and Y. Liu. Splitting touching cells based on concave-point and improved watershed algorithms. *Frontiers of Computer Science*, 8(1):156–162, 2013.

[35] van Eden, C. G., Lamme, V. A. F., and Uylings. Heterotopic cortical afferents to the medial prefrontal cortex in the rat. a combined retrograde and anterograde tracer study. 1992.

[36] S. Vigus, D. Bul, and C. N. Canagarajah. Video object tracking using region split and merge and a kalman filter tracking algorithm. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 1, pages 650–653. IEEE, 2001.

[37] Wikipedia. Vigra — wikipedia, the free encyclopedia, 2016. [Online; accessed 5-May-2016].

[38] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(11):1101–1113, 1993.

[39] B. X, S. C, and Z. F. Touching cells splitting by using concave points. In *Digital Image Computing: Techniques and Applications*, page 271âĂŞ278, 2008.

[40] P. A. Yushkevich, J. Piven, H. Cody Hazlett, R. Gimpel Smith, S. Ho, J. C. Gee, and G. Gerig. User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *Neuroimage*, 31(3):1116–1128, 2006.