



The University of  
**Nottingham**

UNITED KINGDOM · CHINA · MALAYSIA

Raju, Rajeswari (2016) Optimisation of image processing networks for neuronal membrane detection. PhD thesis, University of Nottingham.

**Access from the University of Nottingham repository:**

[http://eprints.nottingham.ac.uk/33948/1/RAJESWARI%20RAJU\\_009225\\_PhD%20Thesis%20in%20Computer%20Science.pdf](http://eprints.nottingham.ac.uk/33948/1/RAJESWARI%20RAJU_009225_PhD%20Thesis%20in%20Computer%20Science.pdf)

**Copyright and reuse:**

The Nottingham ePrints service makes this work by researchers of the University of Nottingham available open access under the following conditions.

This article is made available under the University of Nottingham End User licence and may be reused according to the conditions of the licence. For more details see: [http://eprints.nottingham.ac.uk/end\\_user\\_agreement.pdf](http://eprints.nottingham.ac.uk/end_user_agreement.pdf)

For more information, please contact [eprints@nottingham.ac.uk](mailto:eprints@nottingham.ac.uk)

**OPTIMISATION OF IMAGE  
PROCESSING NETWORKS FOR  
NEURONAL MEMBRANE  
DETECTION**

**RAJESWARI RAJU**

**Thesis submitted to the University of  
Nottingham Malaysia Campus for the degree  
of Doctor of Philosophy**

**DECEMBER 2015**

**OPTIMISATION OF IMAGE  
PROCESSING NETWORKS FOR  
NEURONAL MEMBRANE DETECTION**

**by**

**RAJESWARI RAJU**

**THESIS**

**Presented to the School of Computer Science  
Faculty of Science**

**For the Degree of  
Doctor of Philosophy In Computer  
ScienceThe University of Nottingham  
Malaysia Campus**

**December 2015**

**OPTIMISATION OF IMAGE  
PROCESSING NETWORKS FOR  
NEURONAL MEMBRANE DETECTION**

**RAJESWARI RAJU**

**THE UNIVERSITY OF NOTTINGHAM  
MALAYSIA CAMPUS, 2015**

**First Supervisor: Dr. Tomas Maul  
Second Supervisor: Prof. Andrzej Bargiela**

## ABSTRACT

This research dealt with the problem of neuronal membrane detection, in which the core challenge is distinguishing membranes from organelles. A simple and efficient optimisation framework is proposed based on several basic processing steps, including local contrast enhancement, denoising, thresholding, hole-filling, watershed segmentation, and morphological operations. The two main algorithms proposed Image Processing Chain Optimisation (IPCO) and Multiple IPCO (MIPCO) combine elements of Genetic Algorithms, Differential Evolution, and Rank-based uniform crossover. 91.67% is the highest recorded individual IPCO score with a speed of 280 s, and 92.11% is the highest recorded ensembles IPCO score whereas 91.80% is the highest recorded individual MIPCO score with a speed of 540 s for typically less than 500 optimisation generations and 92.63% is the highest recorded ensembles MIPCO score. Further, IPCO chains and MIPCO networks do not require specialised hardware and they are easy to use and deploy. This is the first application of this approach in the context of the *Drosophila* first instar larva ventral nerve cord. Both algorithms use existing image processing functions, but optimise the way in which they are configured and combined. The approach differs from related work in terms of the set of functions used, the parameterisations allowed, the optimisation methods adopted, the combination framework, and the testing and analyses conducted. Both IPCO and MIPCO are efficient and interpretable, and facilitate the generation of new insights. Systematic analyses of the statistics of optimised chains were conducted using 30 microscopy slices with corresponding ground truth. This process revealed several interesting and unconventional insights pertaining to preprocessing, classification, post-processing, and speed, and the appearance of functions in unorthodox positions in image processing chains, suggesting new sets of pipelines for image processing. One such insight revealed that, at least in the context of our membrane detection data, it is typically better to enhance, and even classify, data before denoising them.

(307 words)

## **LIST OF PUBLISHED PAPERS**

1. The paper ‘Local contrast hole-filling algorithm for neural slices membrane detection’ was presented at the 2014 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE 2014), which was fully sponsored by the IEEE Malaysia Section and IEEE Industrial Electronics and Industrial Applications of Malaysia Joint Chapter. The conference is currently indexed in IEEE Explorer, and its print ISBN No. is 978-1-4799-4352-4. DOI 10.1109/ISCAIE.2014.7010227

2. The paper ‘New image processing heuristic suggested by optimization experiments – Enhance it before you Lose it’ was presented at the 2<sup>nd</sup> International Conference on Intelligent Systems and Image Processing 2014 (ICISIP2014), Kitakyushu, Japan. ICISIP2014 was organised by the Institute of Industrial Applications Engineers (IIAE) and sponsored by The Institute of Electrical and Electronic Engineers Fukuoka Section. The conference DOI identifier 10.12792/icisip2014

3. The paper ‘New image processing pipelines for membrane detection’ has been published in Journal of the Institute of Industrial Application Engineers (JIAE), Vol 3, No 1, pp 15-23, January 2015, Online edition: ISSN 2188-1758; 8811 Print edition: ISSN 2188-1758, 2015. DOI identifier 10.12792/JIAE.3.15

## **LIST OF PAPERS READY FOR PUBLICATION**

4. The paper ‘Image Processing Chain Optimization for Membrane Detection in Neural Slices’ is ready for publication\*.

5. The paper ‘Multiple Image Processing Chain Analysis for Membrane Detection – Optimization Outcome’ is ready for publication\*.

\*Submitted and under review at the time of writing.

## **ACKNOWLEDGEMENTS**

I would like to express my sincere appreciation and thanks to my advisor Assistant Professor Dr. Tomas Henrique Maul, who has been a tremendous supervisor and mentor for me. I would like to thank him for imparting encouragement during hardships and for allowing me to grow as a research scientist. His guidance and advice on my research have been priceless. I would also like to thank my co-supervisor, Professor Andrzej Bargiela for his advice and encouragement. I am grateful to you both for your brilliant and insightful comments and suggestions.

I would also like to express my thanks to the University of Nottingham Malaysia Campus and its members, who were there for me throughout my research. Sincere thanks also goes out to the International Symposium of Biomedical Imaging 2012, especially Albert Cardona and the team for allowing public access to 30 Transmission Electron Microscopy (TEM) images and corresponding ground truth.

I also give special thanks to my family. Words cannot express how grateful I am to my mother, Madam Walli Krishnan, father, Mr. Raju Suppen, mother in law, Madam Muthulechimy, father in law, Mr. Sangaran Nair and children, Omkarra Nair, Trishul Nair and Swasztika Nair for all the sacrifices they made on my behalf. Your prayers for me helped to sustain me thus far. I would also like to thank all of my friends who supported me in writing, and encouraged me to strive towards my goal. Finally, I would like to express my appreciation for my beloved husband Sritharan Sangaran who spent sleepless nights with me and always provided me with support in the moments when there was no one to answer my queries.

**THANK YOU.**

# TABLE OF CONTENTS

|   |      |
|---|------|
| <b>ABSTRACT</b> .....                               | iv   |
| <b>LIST OF PUBLISHED PAPERS</b> .....               | v    |
| <b>LIST OF PAPERS READY FOR PUBLICATION</b> .....   | v    |
| <b>ACKNOWLEDGEMENT</b> .....                        | vi   |
| <b>LIST OF FIGURES</b> .....                        | xiii |
| <b>LIST OF TABLES</b> .....                         | xix  |
| <b>GLOSSARY</b> .....                               | xxii |
| <br>  |      |
| <b>CHAPTER 1 – INTRODUCTION</b> .....               | 1    |
| 1.1 Problem Statement and Main Contributions.....   | 1    |
| 1.2 Image Segmentation.....                         | 3    |
| 1.3 Segmentation in Medical Imaging.....            | 3    |
| 1.4 Problem Formulation .....                       | 5    |
| 1.5 Research Aim and Goal .....                     | 8    |
| 1.6 Research Objectives.....                        | 8    |
| 1.7 Proposed Solution .....                         | 9    |
| 1.8 Research Scope .....                            | 11   |
| 1.9 The Proposed Algorithms .....                   | 11   |
| 1.10 Creation of the Image Processing Network ..... | 12   |
| 1.11 Advantages of the Proposed Algorithms .....    | 14   |
| 1.12 Limitations of the Proposed Algorithms.....    | 17   |
| 1.13 Main Contributions of This Research .....      | 17   |
| 1.14 Thesis Structure .....                         | 19   |
| <br>  |      |
| <b>CHAPTER 2 – LITERATURE VIEW</b> .....            | 21   |
| 2.1 Digital Image Processing .....                  | 21   |
| 2.2 Computer Vision.....                            | 22   |
| 2.3 Segmentation in General.....                    | 23   |
| 2.3.1 Segmentation Techniques.....                  | 24   |
| a) Early stage.....                                 | 24   |



|                                       |  |    |
|---------------------------------------|--|----|
| b)                                    | Middle stage.....  | 24 |
| c)                                    | Continuation stage.....  | 25 |
| 2.3.2                                 | Summary of Monochrome Segmentation Techniques.....                       | 26 |
| 2.3.3                                 | Why segmentation is difficult.....                                       | 27 |
| 2.4                                   | Medical Imaging.....   | 28 |
| 2.4.1                                 | Medical Image Segmentation hurdles.....                                  | 28 |
| 2.4.2                                 | 2015 Competitions in Biomedical Image Analysis.....                      | 30 |
| 2.4.3                                 | Popularity of Biomedical Challenges.....                                 | 33 |
| 2.5                                   | Segmentation in Medical Image Processing.....                            | 34 |
| 2.5.1                                 | History of Medical Image Segmentation.....                               | 34 |
| 2.5.2                                 | Advantages and Limitations of Medical Image Segmentation Algorithms..... | 35 |
| 2.5.3                                 | Comparison MRI and CT.....   | 36 |
| 2.6                                   | Neuron and Cell Segmentation.....  | 37 |
| 2.7                                   | Challenges in Neuron Segmentation.....                                   | 37 |
| 2.8                                   | Gaps filled by the Proposed Algorithms (IPCO and MIPCO).....             | 38 |
| 2.8.1                                 | Comparison with ISBI Competitors.....                                    | 38 |
| 2.8.2                                 | Gaps with other similar area of interest researchers.....                | 40 |
| 2.9                                   | Optimisation of Image Processing Algorithms.....                         | 40 |
| 2.9.1                                 | Genetic Algorithm (GA) and Global Stochastic Optimisation.....           | 40 |
| 2.9.2                                 | Differential Evolution (DE).....   | 41 |
| 2.9.3                                 | Rank-Based Uniform Crossover.....  | 42 |
| 2.10                                  | Conclusion.....  | 42 |
| CHAPTER 3 – RESEARCH METHODOLOGY..... |  | 43 |
| 3.1                                   | Background into the Data Slices used in this Research.....               | 43 |
| 3.1.1                                 | Evaluation Metrics.....  | 44 |
| 3.1.2                                 | The Dataset.....   | 44 |
| 3.1.3                                 | Electron Microscopy.....   | 45 |
| 3.1.4                                 | Image Acquisition.....   | 46 |
| a)                                    | Preparation of the slice - Histology.....                                | 46 |
| b)                                    | The TEM <i>Drosophila</i> Slices.....                                    | 46 |

|       |  |    |
|-------|--|----|
| c)    | The Training data.....                                     | 46 |
| d)    | The Testing Data.....                                      | 47 |
| 3.2   | Performance Measures used for this research.....           | 48 |
| 3.3   | The Platform: MATLAB and the Image Processing Toolbox..... | 50 |
| 3.4   | Creation of the Algorithm.....                             | 51 |
| 3.5   | IPCO And MIPCO Internal Framework for Optimisation. ....   | 52 |
| 3.5.1 | Experimental Design of the Approach.....                   | 52 |
| 3.5.2 | Creation of the algorithm.....                             | 55 |
| 3.5.3 | Flow of the procedure.....                                 | 57 |
| 3.5.4 | Image Processing Function Used .....                       | 61 |
| a)    | Denoising.....   | 61 |
| b)    | Contrast Enhancement.....                                  | 62 |
| c)    | Thresholding.....  | 62 |
| d)    | Hole-Filling .....   | 63 |
| e)    | Watershed .....  | 64 |
| f)    | Morphological Operator.....                                | 64 |
| g)    | Simple combination functions.....                          | 64 |
| 3.5.5 | Filling the gap: Comparison.....                           | 66 |
| 3.6   | Conclusion.....  | 68 |

## **CHAPTER 4 – RESULTS – LOCAL CONTRAST HOLE-FILLING .....69**

|       |  |    |
|-------|--|----|
| 4.1   | Rationale of introducing LCHF .....                          | 69 |
| 4.2   | Initial Startup .....  | 70 |
| 4.3   | Experiments using existing simple segmentation methods ..... | 71 |
| 4.3.1 | Edge Detection.....  | 71 |
| 4.3.2 | Simple thresholding with enhanced membrane features .....    | 72 |
| a)    | Thresholding (TH) and Anisotropic Smoothing (AS).....        | 73 |
| b)    | Thresholding with Gradient Magnitude.....                    | 74 |
| 4.3.3 | Diffusion.....   | 75 |
| 4.3.4 | Graph cuts.....  | 75 |
| 4.4   | First stage of the experiments .....                         | 77 |
| 4.4.1 | Explanation of Functions Selected and Tuned for LCHF .....   | 77 |

|   |     |
|---|-----|
| a) Image Denoising .....  | 77  |
| b) Contrast Enhancement.....  | 79  |
| i) The CLAHE Parameter of Choice – Num Tiles .....                          | 83  |
| ii) Experiments Using Various CLAHE parameters .....                        | 84  |
| c) Thresholding Functions .....   | 85  |
| d) Hole-Filling Functions .....   | 87  |
| i) Performance Measures for Basic Hole-Filling<br>technique results.....    | 88  |
| e) Morphological Operators.....   | 88  |
| f) Dilation.....  | 88  |
| g) Eroding .....  | 89  |
| 4.5 Local Contrast Hole-Filling Algorithm (LCHF).....                       | 90  |
| 4.5.1 LCHF Outcome .....  | 91  |
| 4.5.2 Summary of functions and associated parameters.....                   | 92  |
| 4.5.3 Comparison of LCHF algorithm with simple<br>segmentation methods..... | 92  |
| a) Comparison with Edge Detection method .....                              | 92  |
| b) Comparison of Simple Thresholding with enhanced<br>features.....         | 93  |
| c) Comparison with Weickert’s PDE and Hessian<br>based Diffusion.....       | 94  |
| d) Comparison with and without Denoising function. ....                     | 94  |
| e) Comparison with output of Hole-Filling.....                              | 95  |
| 4.6 Experiments with other datasets .....                                   | 97  |
| 4.7 Strength of the Proposed LCHF Algorithm .....                           | 98  |
| 4.8 Weakness of LCHF .....  | 99  |
| 4.9 Conclusion .....  | 101 |

## **CHAPTER 5 – IMAGE PROCESSING CHAIN OPTIMISATION..... 101**

|   |     |
|---|-----|
| 5.1 IPCO’s training and testing.....    | 103 |
| 5.2 The Novelty .....                   | 104 |
| 5.3 IPCO Processing functions .....     | 104 |
| 5.3.1 Watershed Functions.....          | 106 |
| 5.3.2 Simple Combination Functions..... | 107 |

|           |   |     |
|-----------|---|-----|
| 5.4       | IPCO Computational Flow .....                           | 109 |
| 5.4.1     | Outputs obtained using IPCO .....                       | 111 |
| 5.4.2     | Results obtained using IPCO ( F1 scores > 90%).....     | 112 |
| 5.4.2 (a) | Results obtained using IPCO for Best Chain .....        | 113 |
| 5.4.2 (b) | Optimisation Dynamics .....                             | 114 |
| 5.5       | Observations, New Findings, and Suggestions.....        | 116 |
| 5.6       | Appearance of the Used Function in the Chain .....      | 120 |
| 5.7       | Comparison with state-of-the-art approaches .....       | 121 |
| 5.8       | Comparison of IPCO score with Other Method scores ..... | 123 |
| 5.9       | Ensembles of IPCO chains.....                           | 124 |
| 5.10      | Conclusion .....  | 126 |

**CHAPTER 6 – MULTIPLE IMAGE PROCESSING CHAIN OPTIMISATION..... 128**

|       |  |     |
|-------|--|-----|
| 6.1   | Processing by MIPCO.....                               | 129 |
| 6.2   | Training MIPCO .....                                   | 131 |
| 6.3   | Findings.....  | 132 |
| 6.3.1 | Best Shortest MIPCO functions .....                    | 132 |
| 6.3.2 | Interesting Observation: Morphological Operators ..... | 134 |
| 6.3.3 | MIPCO Function Repetition.....                         | 139 |
| 6.3.4 | Mandatory functions that always appear in chains.....  | 142 |
| 6.4   | Comparison between IPCO and MIPCO .....                | 143 |
| 6.5   | Differences and Similarities (IPCO and MIPCO) .....    | 147 |
| 6.5.1 | For IPCO chain:.....                                   | 148 |
| 6.5.2 | For MIPCO network:.....                                | 148 |
| 6.6   | MIPCO Ensembles Output vs. Other Methods.....          | 151 |
| 6.7   | Conclusion .....                                       | 152 |

**CHAPTER 7 – DISCUSSION..... 154**

|     |                              |     |
|-----|------------------------------|-----|
| 7.1 | Research Outcome .....       | 155 |
| 7.2 | Implication of Findings..... | 155 |

|   |   |         |
|---|---|---------|
| 7.3                                     | Comparison with existing knowledge (literature).....            | 157     |
| 7.4                                     | The Three Proposed Algorithms for Membrane Detection .....      | 159     |
| 7.4.1                                   | LCHF (1 <sup>st</sup> ) Algorithm .....                         | 159     |
| 7.4.2                                   | IPCO chain (Automated Algorithm) .....                          | 161     |
| 7.4.3                                   | MIPCO network (Automated Algorithm, Parallel Network)....       | 164     |
| 7.5                                     | Suggestion of new findings.....                                 | 165     |
| 7.6                                     | Similarities and differences of IPCO and MIPCO.....             | 168     |
| 7.7                                     | Algorithm Performance and Improvement .....                     | 168     |
| 7.8                                     | Empirical Analysis: Reliability of the Proposed Algorithm ..... | 169     |
| 7.9                                     | Suggestions for Future work .....                               | 171     |
| <br><b>CHAPTER 8 – CONCLUSION</b> ..... |   | <br>175 |
| 8.1                                     | Interesting Findings and Suggestion .....                       | 176     |
| 8.2                                     | Contribution of the Proposed Algorithms.....                    | 179     |
| <br><b>REFERENCES</b> .....             |   | <br>182 |
| <br><b>APPENDIX 1</b> .....             |   | <br>193 |
| <b>APPENDIX 2</b> .....                 |   | 196     |
| <b>APPENDIX 3</b> .....                 |   | 199     |

## LIST OF FIGURES

|   |    |
|---|----|
| Figure 1.1: (Left to right) 1. Microscopy image; 2. Ground truth; 3. IPCO processing result.....  | 10 |
| Figure 1.2: (Left to right) 1. Ground truth; 2. MIPCO processing result.....  | 10 |
| Figure 1.3: (a) Example of output result using IPCO chain. (b) Example of output using MIPCO network.....   | 15 |
| Figure 1.4: Image Processing Chain outputs using IPCO.....  | 17 |
| Figure 3.1: Data slices and their corresponding ground truths.....  | 47 |
| Figure 3.2: Examples of ssTEM images for test data.....   | 47 |
| Figure 3.3: Flowchart showing the overall computational flow in a specific chain, with fine-tuning in selection of favoured functions and its parameterisation.....   | 57 |
| Figure 3.4: Flowchart showing the overall computational flow in a specific chain consisting of three functions. In: input image. Ot: output image. FunAa: single-input function such as denoising. FunBb: multiple-input function such as image blending.....   | 58 |
| Figure 3.5: Ensemble.....   | 60 |
| Figure 3.6: Flowchart showing the overall computational flow in a specific network consisting of three functions, and three layers of chains (for illustration purposes). Im: input image. Ot: output image. FunAa: single-input function such as denoising. FunBb: multiple-input function such as image blending..... | 61 |
| Figure 4.1: Segmentation result obtained using the LCHF algorithm, IPCO, and MIPCO network compared with the original image and corresponding ground-truth image.....   | 69 |
| Figure 4.2: Simple comparison of different edge detection methods for <i>Drosophila</i> dataset (greyscale image).....  | 72 |
| Figure 4.3: Output using thresholding alone and thresholding with anisotropic smoothing (TH with AS) for three diferent dataset.....  | 73 |

|  |    |
|--|----|
| Figure 4.4: Output obtained using thresholding with gradient magnitude for three different datasets.....                   | 75 |
| Figure 4.5: Comparison of diffusion-based approaches.....  | 76 |
| Figure 4.6: (a) Original noisy image, (b) result of median denoising, (c) result of denoising with Wiener filter .....     | 79 |
| Figure 4.7: Original image before contrast enhancement and image after contrast enhancement (Histogram Equalisation) ..... | 80 |
| Figure 4.8: Output obtained using default and prescribed high and low values for contrast limit.....                       | 81 |
| Figure 4.9: Original image before application of CLAHE and after application.....  | 81 |
| Figure 4.10: Histogram of original image vs. histogram of the processed image using 64 bins.....                           | 81 |
| Figure 4.11: Values for NumTiles and their corresponding output.....   | 85 |
| Figure 4.12: Original Image vs. Thresholded Image ( $v = 16$ , $v = 25$ ) with colourmap.....                              | 86 |
| Figure 4.13: Effect of using various threshold values on <i>Drosophila</i> dataset.....                                    | 86 |
| Figure 4.14: Hole Filling for Greyscale and Binary Image.....  | 87 |
| Figure 4.15: Effects of Hole-filling.....  | 88 |
| Figure 4.16: Effects of Dilation.....  | 89 |
| Figure 4.17: Effects of Eroding.....   | 90 |
| Figure 4.18: Summary of LCHF Functions.....  | 92 |

|   |     |
|---|-----|
| Figure 4.19: Comparison using LCHF with the other edge detection method.....  | 92  |
| Figure 4.20: Comparison of diffusion-based approaches to LCHF.....  | 94  |
| Figure 4.21: Output result obtained using LCHF (with squiggly lines).....   | 100 |
| Figure 4.22: Microscope image (left) with the corresponding LCHF result (middle), and corresponding Ground truth (right).....   | 100 |
| Figure 5.1: Left: ssTEM section from <i>Drosophila</i> first instar larva; Middle: corresponding ground-truth maps for cell membrane (maroon); Right: segmentation result using IPCO .....  | 102 |
| Figure 5.2: List of all the functions used in IPCO (represent in Diagram).....  | 106 |
| Figure 5.3: Flowchart showing the combine Average processing for illustration.  | 107 |
| Figure 5.4: Flowchart showing the combine Addition processing for illustration.   | 107 |
| Figure 5.5: Flowchart showing the combine Multiply processing for illustration.   | 108 |
| Figure 5.6: Flowchart showing the combine MinMaxTwo processing for illustration.  | 108 |
| Figure 5.7: Flowchart showing the overall computational flow in a specific IPCO chain consisting of three functions. I: input image. O: output image. $F_A$ : single-input function such as denoising. $F_B$ : multiple-input function such as image blending ( <i>e.g.</i> , Combiner MinMax)..... | 110 |
| Figure 5.8: Sample output images obtained using an IPCO chain.  | 112 |
| Figure 5.9: Sequence of functions and processed images of a chain with an F1 score of 91.67%.....   | 114 |



|   |     |
|---|-----|
| Figure 5.10: Best Cost versus iteration count for five chains out of the 10 best IPCO chains (as per Table 5.2).....  | 115 |
| Figure 5.11: Average F1 value versus iteration count for the 10 best IPCO chains (as per Table 5.2).....  | 115 |
| Figure 5.12: Iteration Graph for 1000 generations.....  | 116 |
| Figure 5.13: (a) Binary image, (b) Watershed Image, (c) Result of application of Morphological Eroding.....   | 118 |
| Figure 5.14: (a) Original Image, (b) Image added with ‘Salt and Pepper’Noise, (c) Image added with Gaussian noise.....  | 118 |
| Figure 5.15: Shows output after adding Salt and Pepper extra noise.....   | 119 |
| Figure 5.16: Shows output after adding Gaussian extra noise.....  | 119 |
| Figure 5.17: Figure 5.15: Shows output after adding Poisson extra noise.....  | 120 |
| Figure 5.18: Ensemble flowchart.....  | 124 |
| Figure 5.19: Shows the output image obtained using IPCO method on the ISBI Test Image (Slice 2).....  | 125 |
| Figure 5.20: Ensemble output for the training Image.....  | 125 |
| Figure 6.1: Left: ssTEM section from <i>Drosophila</i> first instar larva; Middle: corresponding ground-truth maps for cell membrane (black); Right: segmentation result obtained using Multiple Image Processing Chain Optimisation Network (MIPCO)..... | 128 |
| Figure 6.2 shows a flowchart of a MIPCO network consisting of three chains and four layers. In the experiments, MIPCO networks were allowed to use a maximum of eight functions per chain, and five parallel chains.....                                  | 130 |

|   |     |
|---|-----|
| Figure 6.3: (a) MIPCO example output with three layers, flows and the selected functions, (b) Final output of the three chosen chains.....  | 136 |
| Figure 6.4(a): Another example of MIPCO chosen functions, function flows for single, and combination function to perform its final output6.4(b).....  | 137 |
| Figure 6.5: Example of final output images of network using MIPCO that have morphological operators in various positions (specified in the figure), in the front, middle, and end portions of the chains with F1 scores > 91% vs. the Ground-Truth image (rightmost)..... | 139 |
| Figure 6.6: (a), (b), (c) Examples of chains with function selection and repetition for visual inspection as per Table 6.4.....   | 141 |
| Figure 6.7(a): IPCO output function for single best chain.....  | 143 |
| Figure 6.7(b) shows that the best MIPCO network consists of eight functions and five chains with F1 score of 91.80%.....  | 144 |
| Figure 6.8 shows the output reflected in Table 6.5 for the best MIPCO network.....  | 145 |
| Figure 6.9: Another visual representation of MIPCO output versus Ground truth for score of 88.96%.....  | 151 |
| Figure 6.10: Visual representation of example of MIPCO best score output versus Ground truth for score with 8% difference to reach 100%.....  | 151 |
| Figure 6.11: MIPCO Ensembles vs. Other Methods.....   | 152 |
| Figure 7.1: LCHF Output.....  | 160 |
| Figure 7.2: IPCO Output.....  | 161 |
| Figure 7.3: LCHF output using ISBI Test Slice.....  | 162 |

|   |     |
|---|-----|
| Figure 7.4: MIPCO Output.....   | 164 |
| Figure 7.5: Sample output obtained using the IPCO processing chain..... | 172 |

## LIST OF TABLES

|   |    |
|---|----|
| Table 2.1: Monochrome Image Techniques.....   | 27 |
| Table 2.2: List of 2015 Competitions (a few examples).....  | 30 |
| Table 2.3: MRI vs. CT scan.....   | 36 |
| Table 3.1: Main categories of processing functions available to IPCO in the implementation reported in this research (there is no order restriction and it can appear in any order).....        | 59 |
| Table 3.2: Main categories of processing functions available to MIPCO networks in the implementation reported in this research (there is no order restriction; it can appear in any order)..... | 60 |
| Table 3.3: Summary of Proposed Algorithms and Functions Used.....   | 65 |
| Table 3.4: Gaps Filled by IPCO chain and MIPCO network.....   | 66 |
| Table 4.1: Measurement values for denoising filter for three experimental functions (denoising, thresholding, and hole-filling).....  | 78 |
| Table 4.2: Randomly picked slices from the 30 stacks dataset.....   | 82 |
| Table 4.3: Average performance values after using different contrast enhancement techniques.....  | 83 |
| Table 4.4: Showing the Cost function result using various NumTiles parameter values.....  | 84 |
| Table 4.5: Explanation and F1 results for various CLAHE parameters.....   | 84 |
| Table 4.6: Measurement values for basic Hole-Filling.....   | 88 |

|   |     |
|---|-----|
| Table 4.7: Images and corresponding output obtained using the LCHF algorithm.....   | 93  |
| Table 4.8: LCHF with and without the denoising function.....  | 95  |
| Table 4.9: Results of experiments conducted using thresholding with hole-filling only and using LCHF.....   | 96  |
| Table 4.10: Experimental results obtained using other datasets (published in a conference paper presented at IEEE Symposium)  | 97  |
| Table 4.11: Measured values of final LCHF method as per the above visual output.....  | 100 |
| Table 5.1: List of all the functions used in IPCO.....  | 105 |
| Table 5.2: Main classes of IPCO functions with their corresponding image processing phases and general purposes.....  | 109 |
| Table 5.3: IPCO chains with F1 score greater than 90%, averaged over all training images.....   | 113 |
| Table 5.4: Frequency of Function appearance in 50 trial experiments.....  | 120 |
| Table 5.5: Participating groups for first challenge on 2D segmentation of neuronal processes in electron microscopy images in the International Symposium on Biomedical Imaging (ISBI) 2012 challenge workshop (Leaders Group)..... | 122 |
| Table 5.6: Testing performance score of IPCO vs. Post-Processing for <i>Drosophila</i> dataset.....   | 123 |
| Table 5.7: IPCO chains used in the best ensembles.....  | 124 |
| Table 6.1: Best shortest network (with three chains) for MIPCO with score >91%.....   | 133 |

|   |     |
|---|-----|
| Table 6.2: Appearance position of earliest morphological operators(row shows average scores and positions).....   | 138 |
| Table 6.3: Function repetitions for 50 trials using MIPCO network.....  | 140 |
| Table 6.4: Examples of chains with function selection and repetition.....   | 140 |
| Table 6.5: MIPCO best chain and function occurrences (rearrange for better view).....                             | 145 |
| Table 6.6: Performance score for each slice of the dataset for the score of 91.80%.....                           | 146 |
| Table 6.7: Denoising and Thresholding Function Appearance Frequency (reproduced from Chapter 5, Table 5.4).....   | 149 |
| Table 6.8: Combination Function Appearance Frequency (reproduced from Chapter 5, Table 5.4).....                  | 149 |
| Table 6.9: Thresholding and Combination Functions Appearance Frequency for MIPCO (reproduced from Table 6.3)..... | 150 |
| Table 7.1: Contribution of the research with general targetresearch activity.....                                 | 156 |
| Table 7.2: Sample cumulative pixels showing false positive and false negative..                                   | 175 |
| Table 8.1: Research output satisfying the aims, goals, and objectives.....  | 179 |

## GLOSSARY

### Glossary of Acronyms

| Acronyms | Meaning  |
|----------|--|
| CT       | Computer Tomography  |
| DE       | Differential Evolution   |
| GA       | Genetic Algorithm  |
| GSO      | Global Stochastic Optimisation   |
| IPCO     | Image Processing Chain Optimisation  |
| ISBI     | Internal Symposium of Biomedical Imaging   |
| LCHF     | Local Contrast Hole-Filling  |
| MATLAB   | Matrix Laboratory. It is a multi paradigm numerical computing environment and 4 <sup>th</sup> generation programming language. Matlab being used as platform for this research |
| MIPCO    | Multiple Image Processing Chain Optimisation   |
| MRI      | Magnetic Resonance Imaging   |
| RBUC     | Rank Based Uniform Crossover   |
| TEM      | Transmission Electron Microscopy   |

### Glossary of Terms

| Acronyms                 | Meaning   |
|--------------------------|---|
| Algorithm                | As for this research, there are 3 main algorithms: LCHF, IPCO and MIPCO.  |
| Blending                 | To combine 2 images to get a new output image.  |
| <i>Drosophila Larvae</i> | A fruit fly. This research used the TEM images of the <i>Drosophila Larvae</i> .  |
| Combine Function         | Newly developed function for image blending using different techniques, such as combining 2 images using averaging of the pixel values, adding and multiplying the pixel values to create a new output image and finding the minimum or maximum value from the pixel values of the image and using the new minimum or maximum value to create the new output image. |

Continue...

continued...


|                        |   |
|------------------------|---|
| Contrast Enhancement   | Image processing technique to enhance contrast in images.   |
| Denoising              | Image processing technique to remove noise from images.   |
| Functions              | Image processing functions, such as Contrast Enhancement, Denoising, Thresholding, Morphological Operators, Hole-Filling, Watershed, Combination Function (Addition, Multiply, Average, Finding Minimum and Maximum Value).   |
| Hole Filling           | Image processing technique to fill holes in images.   |
| Morphological Operator | Image processing technique to dilate and erode images.  |
| Slices                 | Data slices also known as sections in Biology.  |
| Stages                 | Levels in algorithm development.<br>First Stage – Initial Stage in development of the algorithm, introducing the first created algorithm.<br>Second Stage – Second phase in development of the algorithm, in introducing the second created algorithm.<br>Thirds Stage – Third phase in development of the algorithm, in introducing the third created algorithm. |
| Thresholding           | Image processing technique frequently used for separating foreground and background regions in images.  |
| Watershed              | Image processing technique for segmenting images.   |



# CHAPTER 1

## INTRODUCTION

*The most beautiful thing we can experience is the mysterious. It is the source of all true art and science. He to whom this emotion is a stranger, who can no longer pause to wonder and stand rapt in awe, is as good as dead: his eyes are closed.*



*(Albert Einstein)*

Life mysteries and the curiosity that constantly surround researchers with wonders are what drive them to conduct great research and pursue difficult-to-grasp answers for each and every question that arises. Of the five given senses (hearing, seeing, feeling, smelling, and tasting), seeing is perhaps the noblest, because it allow us to examine the mysteries of the universe. It is a masterpiece of nature's work. Humans are largely responsive to visual cues, and cognitive level images are unconsciously persuasive. The adage 'A picture speaks a thousand words' was coined almost 100 years ago, and since then the consensus has been that a complex idea can be conveyed with just a single still image. In today's modern age, this adage still has significance for computing with images.

### 1.1 Problem statement and Main Contributions

This research dealt with the problem of neuronal membrane detection, in which the core challenge is distinguishing membranes from organelles. Although many segmentation algorithms are available, new algorithms are still needed because no standard algorithm that satisfies or suits all existing conditions for all datasets currently exists. Standard segmentation algorithms tend to over-or under-segment microscopic images of neuronal membranes, mainly because of the similarity between membrane and non-membrane (*e.g.*, organelles) material. Moreover, sample based training is generally difficult and time-consuming and needs specialised high end hardware with high cost.

Furthermore, many neural network approaches are practically black boxes, which means that we tend to have limited knowledge of their internal workings. This raises the issue of interpretability, because it is difficult to determine how these networks solved specific problems. To add more to this problem, many existing algorithms do not have the capability to retrain or continuously learn. Moreover, in the context of image processing pipelines, some unconventional insights can only be revealed by non-restriction of function ordering. So, it is clear that many research gaps exist. This research is working to address these gaps. There is still room for improved algorithms in terms of accuracy, speed, generality, and utilisation of low end hardware for cost saving. This research proposes an algorithm which is high in speed in detecting neuronal membranes with usage of low end hardware. The research aims at the utilisation low end hardware with minimal cost, whilst obtaining accuracy levels comparable with the state of art. As for the capability of the proposed approach, it has been tested in an open challenge in which medical imaging researchers showcased their best methods and participated in direct head-to-head comparisons, with standardised datasets that capture the complexity of a real-world problem in a controlled experimental design and metrics to evaluate the results. The research uses existing image processing functions, but optimises the way in which they are configured and combined. The approach differs from related works in terms of the set of functions used; the parameterisations allowed, the optimisation methods adopted; the combination framework and the testing and analyses conducted. It uses a larger set of functions and the combination framework is less rigid in structure, and provides reordering flexibility with no ordering constraints. This process revealed several interesting and unconventional insights pertaining to denoising and morphological operators, which found new sets of chains and suggested new sets of pipelines for image processing. These algorithms also incorporated single and multiple input functions such as ‘image blending’, and used special purpose ‘combiner functions’ specifically designed to encourage chains to form different representations and transformations. The highest recorded F1 score was 92.63% for ensembles and 91.80% for individual scores.

## **1.2 Image Segmentation**

Image segmentation is a process in which an image is partitioned in a semantically meaningful way (same surface, object, material, *etc.*). It is a common task, but its execution details vary widely. Image segmentation's goal is to move from an array of pixel to a collection of regions by understanding the component of the image, and to extract objects and boundaries of interest to give more than one class of regions (L. Shapiro & Stockman, 2001, L. G. Shapiro & Linda, 2002).

Driven by the increased capacity of imaging devices, tools that are highly adapted to the application have become a necessity to achieve good performance. Current technologies enable researchers to enhance their research abilities, make suggestions, and contribute more benefits to the community (Kaynig, Fischer, & Buhmann, 2008). Over time, image processing research has advanced from basic low-level operations to high-level image interpretation analysis and understanding, and has resulted in easier processing of images.

Segmentation is often used as a preprocessing technique in many image analysis procedures (L. Shapiro & Stockman, 2001, L. G. Shapiro & Linda, 2002). Segmentation is present in many image driven processes, *e.g.*, text, object, iris or face detection and recognition, fingerprint recognition, detection of deviations in industrial pipelines, tracking of moving people/cars/airplanes, image editing, image compression, traffic, meteorological, military, medical areas, and satellite image processing.

## **1.3 Segmentation in Medical Imaging**

Segmentation occurs naturally in the human visual system, thus it can help to segment objects. Humans can detect edges, shapes, lines, and patterns using visual information, and subsequently make decisions. However, in general, manually processing all images is not feasible for humans. It is definitely not

feasible when there are many images, because much time, money, and energy are required. Moreover, humans can get bored carrying out this process. Consequently, humans have created tools to assist them. Tools are needed to assist humans in browsing through large images and to extract meaningful information, especially in medical imaging. Segmentation tools can help medical staff to browse through large images created using today's modern technology, and segmentation can extract meaningful information and output models of organs, and other structures for further analysis, in order to detect abnormalities such as tumours and quantify changes in follow-up studies or for simulation. Modern medical imaging modalities generate increasingly larger images which simply cannot be examined manually by a human as such a task is exhausting. This fuels a need for development of more efficient image segmentation methods because to date there is no general method for solving all segmentation problems.

Although many segmentation algorithms are available, new algorithms are still needed because no standard algorithm that satisfies or suits all existing conditions for all datasets currently exists. This situation exists because the segmentation problem is inherently ill-posed. According to Hadamard (1923), a problem is referred to as being ill-posed when no solutions exist, or when the existing solutions are not unique or do not vary continuously with the input data. Segmentation is regarded as ill-posed because of the large number of possible partitions that can result for a single input. All the existing algorithms are suited for a specific purpose, with corresponding advantages. In other words, improved algorithms are still needed.

As stated above, there is still room for improved algorithms. Some of the areas in which improvements are needed are as follows:

- Accuracy
- Speed
- Generality
- Robustness to noise

- Cost of hardware(low end and no specialised hardware)

In this research, the above areas were analysed and efforts made to improve them, such as higher accuracy, faster speed, and lower cost in hardware (personal computer).

#### **1.4 Problem Formulation**

The presentation of an image can be changed and simplified through image segmentation, in which the image is divided into different parts comprising multiple sets of pixels. This process is conducted with the aim of presenting the data in a more meaningful manner that facilitates much easier analysis and extraction of high-level information. The extracted meaningful information can be used for further analysis. Following the development of an algorithm that can extract needed information, the next step is to judge that algorithm's performance.

However, the question remains of how a segmentation algorithm should be judged; perhaps through visual comparison of two images? Although visual comparison can help researchers to get a better picture of the performance of the algorithm, this method is still not acceptable because subjective evaluation is inconsistent. For example, human view and decision may differ, and it is very difficult to measure the differences and similarities. Moreover, visual comparison is difficult to replicate. Thus, the best way to measure the performance of the algorithm is to use the performance score of the segmentation algorithm on a standard segmentation benchmark or by comparing it with an available gold standard (if such is available). A higher score guarantees a higher performance for an algorithm.

More specifically, the problem of membrane detection (or segmentation) is characterised by several issues. These issues include the following:

1. Standard segmentation algorithms tend to over-or under-segment microscopic images of neuronal membranes, mainly because

of the similarity between membrane and non-membrane (*e.g.*, organelles) material.

2. Sample-based training approaches are generally difficult and time-consuming, partly because a sufficiently large number of labelled training samples need to be provided in order to get a desirable outcome. Many algorithms depend on the existence of ground-truth samples for training. These ground-truth samples need to be prepared by one or more experts, which is an expensive and time-consuming process.

3. In order to carry out the task, specialised hardware is often required for initialisation and calibration procedures, prior knowledge of the medical domain under consideration, advanced programming skills, *etc.*

4. Many approaches (*e.g.*, Deep Neural Networks), are practically black boxes, which means that they can only be viewed in terms of their inputs and outputs, without any knowledge of their internal working. This raises the issue of interpretability, because it is difficult to determine how these networks solved specific problems.

5. Many algorithms are not flexible and cannot be applied to many different types of datasets.

6. Many existing algorithm also do not have a retrainable capability, and do not have the capacity to form different representations and transformations. Some unconventional insights can only be revealed by non-restriction in function ordering. (This aspect is demonstrated in the outcome of this research).

The issues outlined above have contributed directly and are a major reason for this research and the approach consequently proposed. Further, to add to the capability of the proposed approach, it has been tested in an open challenge in which medical imaging researchers showcased their best methods and participated in direct head-to-head comparisons, with standardised datasets that

capture the complexity of a real-world problem in a controlled experimental design and metrics to evaluate the results. The challenge involved segmentation of neuronal structures using 30 slices of the *Drosophila Larvae* Transmission Electron Microscopy (TEM) dataset. The challenge, called Segmentation of neuronal structures in Electron Microscopy stacks (IEEE International Symposium on Biomedical Imaging, ISBI 2012), was carried out in a premier forum for presentation of technological advances in theoretical and applied biomedical imaging and image computing. The provider allowed public access to the 30 TEM images and their corresponding ground truth.

As part of the research process and for comparison of the proposed method with current state-of-the-art approaches, a submission was sent to the ISBI challenge workshop, as a 32-bit TIFF 3D image. The aim of the challenge was to compare and rank the different competing methods based on their pixel and object classification accuracy. The algorithm was tested in an open challenge in which medical imaging researchers showcased their best methods and participated in direct head-to-head comparisons, with standardised datasets that capture the complexity of a real-world problem, and using a controlled experimental design and metrics to evaluate the results. The approach proposed in this research, Image Processing Chain Optimisation (IPCO), obtained an F1 score of 90% on the unseen test datasets, with the highest score being 94%.

This research was conducted and a solution proposed despite existing solutions for the dataset because (for example) even though the solution of the winning method that scored 94% in the challenge was marginally better in quantitative terms (4% more), it required almost a week of training time on specialised hardware. Consequently, it is much more difficult to apply in real-world scenarios than the proposed method. This issue of speed and specialised hardware requirements can be minimised by adopting a simpler approach such as that exemplified by the algorithms proposed in this research. The proposed algorithms are fast to fine-tune and/or optimise, and can be trained and manipulated even after they have already been optimised. This definitely enhances the capability, efficiency, and transparency of the suggested algorithms. The simplicity, efficiency, interpretability, and usability of the

algorithm, makes it easier for researchers or non-computer scientists with limited experience of computer vision and machine learning to adopt it.

## **1.5 Research Aim and Goal**

The focus of this research is on the problem of neuronal membrane detection, in which the core challenge is distinguishing membranes from organelles. The aim/goal is to propose an algorithm with the following characteristics that can detect membranes and eliminate organelles:

- High accuracy
- High speed
- Low hardware cost. The research is aiming at utilising low end hardware.
- Interpretability
- Usability
- Easy to adopt by new researchers in the area of Image Segmentation and Classification.

## **1.6 Research Objectives**

The objectives set for this research were as follows:

- To adopt a hybrid algorithm that combines high-level knowledge with low-level information.
- To develop a membrane detection algorithm with accuracy close to the state-of-the-art, but with additional features such as: efficient training, interpretability, usability, and easy adoption by new researchers.
- To develop a membrane detection algorithm with improved speed close to that of the state-of-the-art.
- To develop a simple and efficient approach based on several basic processing steps, including local contrast enhancement,



thresholding, denoising, hole-filling, watershed segmentation, and morphological operators.

- To obtain insights into new types of useful image processing pipelines.

This research was conducted in three main stages: (i) Local Contrast and Hole-Filling (LCHF), (ii) Image Processing Chain Optimisation (IPCO) chain, and (iii) Multiple Image Processing Optimisation (MIPCO) network.

The aim of the first stage was to select the most effective tuning for a predefined processing pipeline. Because the component methods are critically dependent on some parameters, this stage served also to determine the ranges of the effective values of parameters in the processing pipeline for the detection of cell membranes which were simultaneously capable of ignoring organelles.

Next is the automated stage, in which the sequences (or chains) of image processing functions are optimised using a global stochastic optimisation approach, with the overall process called IPCO.

To further boost performance, ensembles were created from several high-scoring IPCO chains. This idea were used to develop another enhanced parallel algorithm, called the MIPCO network. MIPCO is the result of efforts to further boost the performance of IPCO.

## **1.7 Proposed Solution**

With the above list of issues that exist in membrane detection, this research was conducted with the aim of addressing the listed issues and proposing an algorithm that is efficient, simple, and accurate in dealing with the membrane detection problem. In this research, the ability to discriminate between organelles and membranes is at the core of the problem to be solved. The figures below show the outputs obtained using the two algorithms proposed in this research.

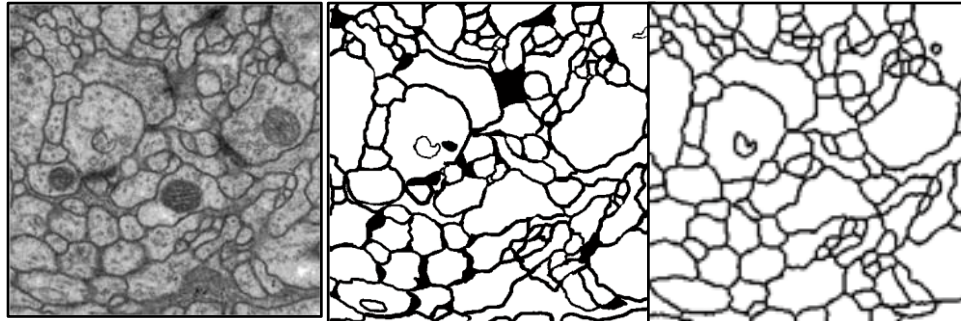


Figure 1.1: (Left to right) 1. Microscopy image; 2. Ground truth; 3. IPCO processing result

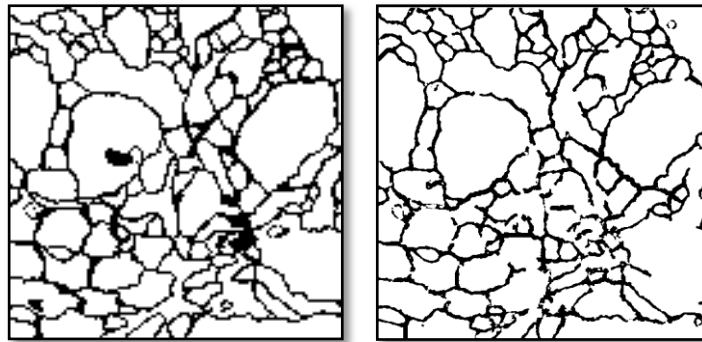


Figure 1.2: (Left to right) 1. Ground truth; 2. MIPCO processing result

The above figures show that the proposed approach is highly desirable and competitive. Both algorithms attained competitive accuracy levels, with F1 scores higher than 90%. To place this score in perspective, the highest score at present is 92.63% on the F1 measure of test accuracy score. Moreover, the approach does not involve an excessively long tuning stage. The approach requires only 10 seconds to process a data slice. The approach also does not require specialised hardware, and it is simple and easy to use. The research was conducted using a standard average personal computer with a 2.40 GHz Intel Core processor, 4 GB RAM, a 32 bit OS, and the MATLAB image processing toolbox by MathWorks (MatLab, 2012). The approach results in chains consisting of short sequences of basic processing steps which are efficient and easy to interpret. Although it is a simple design feature, it is critical for choosing optimal pipelines for specific datasets. The approach uses various sets of functions and the combination framework is less rigid in structure and provides reordering flexibility—the approach has no ordering constraints, *e.g.*,

‘classification’ may be done before ‘preprocessing’. This order flexibility, although simple, provided the research with new insights into image processing pipelines, with classification often being performed before denoising, at least in the domain of membrane detection. This finding could not have been obtained by forcing function order using the standard image processing workflow.

## **1.8 Research Scope**

The experiments conducted in this research were carried using the *Drosophila* first instar larva ventral nerve cord (VNC). The dataset was obtained from the ISBI site and consisted of 30 slices of Transmission Electron Microscopic images, imaged at a resolution of  $4 \times 4 \times 50$  nm/pixel and covering a  $2 \times 2 \times 1.5$  micron cube of neural tissue with its corresponding ground-truth slices. For this research, subsections of some of the initial slices were used for training. The research training and testing were solely conducted using this dataset. In some experiments outcome, the algorithm was tested with other neuronal images in order to obtain comparison results for the algorithm.

The dataset indicated above was chosen for the following reasons:

- It is an extensive dataset with a significant number of benchmark results for comparison.
- The provider granted public access to 30 TEM slices of training images, 30 TEM slices of testing images, and 30 ground-truth images corresponding to the training images.

## **1.9 The Proposed Algorithms**

The research is divided into three Main stages/algorithms (there are also some minor stages involved for data collection, and variable fine-tuning, which are further explained in the Methodology and Result chapter.

**Each of the stages below corresponds to a different category in the algorithm:**

1. **Algorithm 1:** Manual Tuning of Image Processing Chains. In this category, a new algorithm called the LCHF algorithm using non-Learning approach was proposed. This approach achieved an F1 score of 71% for identification of the membrane in comparison with the benchmark (ground truth) images.

2. **Algorithm 2:** Automated Fine-Tuning of IPCO. In this case, the process was conducted automatically to detect membranes and eliminate organelles. A hybrid global stochastic optimisation method, which included elements of genetic algorithms, differential evolution, and rank-based uniform crossover, was adopted. To further boost performance, ensembles (combinations of several different classifiers) of IPCO chains were used to improve the generalisation capabilities of the classification approach.

3. **Algorithm 3:** Automated Fine-Tuning of MIPCO. This approach involved the application of a hybrid global stochastic optimisation to image processing networks, in which the network is processed in parallel. MIPCO is fully automated and is a more powerful approach. The optimisation algorithm has several basic image processing functions available to it, which it configures in different sequences and with different parameter settings, in response to the cost function, defined as the F1 score relative to a subset of the training images. MIPCO consists of multiple networks, in which the networks are optimised together and interact with each other to produce the best output with the highest score.

## **1.10 Creation of the Image Processing Network**

1. Experiments were first conducted with basic preliminary functions in the experimental phase. In the initial stage, various algorithms were written and tested with a main testing function. Each

algorithmic variant was coded in a separate function, and the optimal parameter required for each algorithm was hard-coded within the main testing function. This optimal parameter was found through different fine-tuning experiments carried out as a ‘starter’ for this research. In this case, algorithm parameters were not passed as arguments but were specified within the main testing function itself.

2. In the later stage, many avenues were considered for more innovative contribution. One natural path that follows from this work is formalisation of the processing chain into a parameterisable solution that can then be optimised using different optimisation algorithms. A simple function was created to run Image Processing Optimisation. The goal that was set for this function was to optimise the processing chain in order to find the optimal processing chain.

3. Many experiments were conducted using the created function as a basis and many useful questions were asked to reach the set goal of this function. Among the questions were the following:

- a. What is the optimal processing chain? Can the chain achieve a performance of more than 90%?
- b. What is the best and fastest chain possible?
- c. What is the optimal chain for a specific number of functions in a chain?
- d. What is the best type of segmentation algorithm that can be used in this Image Processing Chain?

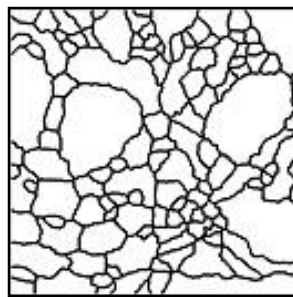
4. The algorithm achieved the set goal and a performance greater than 90%. The algorithm is not only capable of highlighting the membrane boundaries, but also manages to remove the internal structures (the organelles) successfully.

5. This second stage algorithm was called the IPCO algorithm. The IPCO algorithm can receive inputs from earlier functions; this in some sense can be seen as a network.

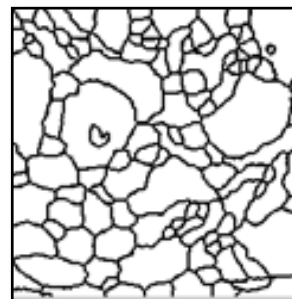
6. To further enhance the approach for accuracy, ensembles from several high-scoring chains were created. Subsequently, the idea to create multiple networks was conceived. Thus, the next improvement stage, called the Multiple Image Processing Chain Optimisation (MIPCO) network, was entered. MIPCO is essentially a direct application of global stochastic optimisation to multiple image processing networks. These networks execute in parallel and can exchange intermediate information. MIPCO has various functions which it configures in different sequences and with different parameter settings. It computes layer by layer and there is no dependency of functions in the same layer. Functions in a layer can receive input from any other function in previous layers. Thus, a layer must complete all computation before the next layer can initiate its own computation; MIPCO is fully automated.

7. Both approaches are efficient and interpretable, and facilitate the generation of new insights. Many interesting insights were obtained and reported in executing the algorithm. A new set of pipelines for image processing was also suggested.

### 1.11 Advantages of the Proposed Algorithms



F1 score: 90.37%



F1 score: 91.63%

(a)

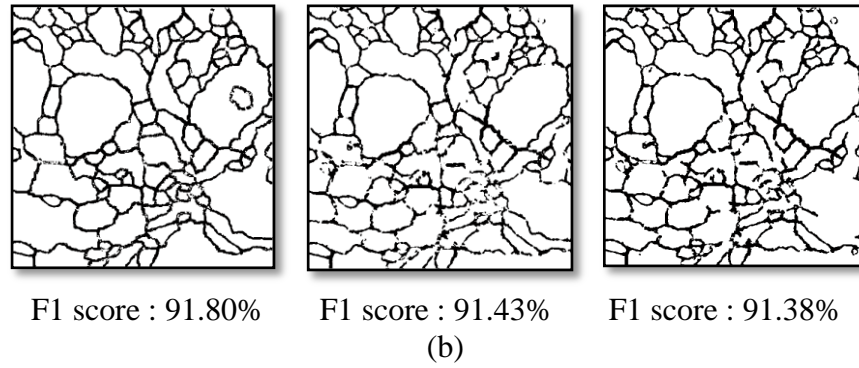


Figure 1.3: (a) Example of output result using IPCO chain. (b) Example of output using MIPCO network

The above figures demonstrate the efficacy of both algorithms (IPCO and MIPCO) in detecting membranes and eliminating unwanted intracellular cells. The ability of the algorithms to discriminate between membranes and organelles is shown. This strongly emphasises the advantages listed below:

- 1) The algorithms (IPCO and MIPCO) not only highlight membrane boundaries, but also remove internal structures (eliminate organelles) successfully.
- 2) The implemented IPCO and MIPCO chains efficiently detected membranes in the ISBI 2012 challenge dataset. IPCO combines the simplicity and efficiency of simple sequences of image processing functions and involves automated fine-tuning of an algorithm relative to a dataset. Further, MIPCO networks are optimised together and interact with each other to produce the best output with the highest score.
- 3) The constraint of a sufficiently large number of labelled training samples can be overcome by IPCO and MIPCO because both the IPCO and MIPCO algorithms can work with relatively small samples. In the training conducted in this research, IPCO and MIPCO used only about 2% of the training data, but performed well in distinguishing membranes and organelles, thus satisfying the original goal.
- 4) IPCO and MIPCO have a relatively fast convergence speed.

5) IPCO and MIPCO have a consistent optimisation process which leads to a variety of useful and easily interpretable solutions.

6) The algorithms do not require specialised hardware. Based on current hardware constraints, training classifiers with a large number of free parameters can require weeks of computation, even when high performance machines with high data transfer rates are used. This involves significant monetary (hardware) and energy costs (time spend). The proposed approaches are more environmentally friendly. Moreover, long hours of training and specialised hardware are usually not feasible for small researchers. The research is aiming at utilising low end hardware.

7) IPCO and MIPCO's simplicity, efficiency, interpretability, and usability make them easier to use and deploy. Their simplicity facilitates easier deployment by researchers with limited knowledge of image segmentation. For example, the algorithms involve simple programming steps with basic functions that can typically be found in MATLAB standard image processing libraries. The toolbox is useful for processing, visualisation, and analysis of images, whilst MATLAB is convenient for rapid prototyping (MatLab, 2012).

8) Using the algorithms, reasonable results are obtainable with relatively little effort. The best F1 score to date is 92.63% and the algorithms do reasonably well distinguishing membranes and organelles, thus satisfying the original goal.

9) Another advantage of IPCO and MIPCO is that they require relatively small sample sizes.



## 1.12 Limitations of the Proposed Algorithms

1) Among the issues that need to be addressed in future work is further improvement of accuracy.

2) A clear example of this is shown in Figure 1.4 (Image Processing Chain with IPCO). In the bottom rightmost sub-figure (Ground Truth (GT) overlapped Processing Output (PO)), the colour representations are as follows:

- Black = True Negative
- Yellow = True Positive
- Green = False Negative
- Red = False Positive

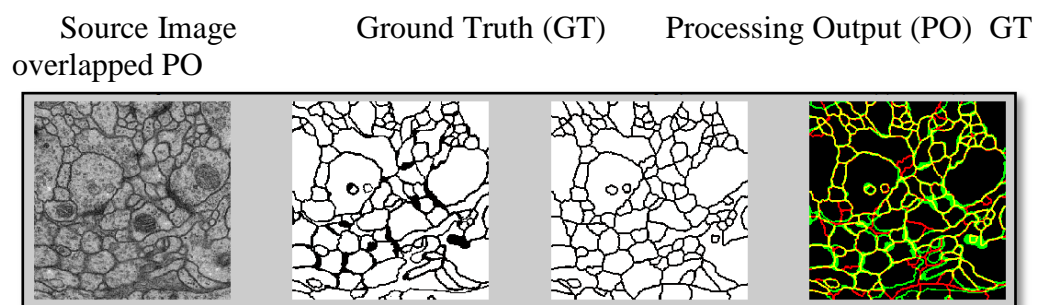


Figure 1.4: Image Processing Chain outputs using IPCO

## 1.13 Main Contributions of This Research

### Discoveries and proposals ...

1) This research does not propose any new individual image processing functions; it uses existing functions and optimises the way in which they are configured and combined. The approach optimally selects, configures, and combines existing functions.

2) Work by other researchers in this area typically differs from this approach in one or more ways—specifically, the set of functions used,

the parameterisations allowed, the optimisation methods adopted, the combination framework, and the testing and analyses conducted.

3) In the research framework, a special-purpose ‘combiner’ function is specifically designed to encourage chains to form different representations and transformations. The combiner function was adopted to integrate with other functions, when the chain was designed in such a manner that the function can receive input from earlier functions and this capability of the processing chain enables it to be regarded as a processing network. In analysing the output of the processing network, from the combiner function viewpoint, a useful process is performed and not just copying of the previous input image. Moreover, the existence of the ‘combiner’ function also results in a better performance score. The existence of the function can be considered a contribution to the processing network.

4) The approach adopts a hybrid global stochastic optimisation method, which includes elements of genetic algorithm, differential evolution, and rank-based uniform crossover. The optimisation algorithm is easy to further manipulation online as a result of its simplicity and transparency. Moreover, the interpretability of the image processing network is higher than that of neural networks because neural networks are practically black boxes, which means that we can only view them in terms of their inputs and outputs, without any knowledge of their internal working. It is difficult to ascertain how a neural network solves specific issues or problems.

5) This is the first time this approach has been applied in the context of the *Drosophila* first instar larva VNC, imaged at a resolution of  $4 \times 4 \times 50$  nm/pixel and covering a  $2 \times 2 \times 1.5$  micron cube of neural tissue.

6) In this research, systematic analyses of the statistics of optimised chains were conducted, and several interesting and unconventional

insights pertaining to preprocessing, classification, post-processing, and speed were obtained. In other words, the types of analyses conducted were novel, and revealed, for example, interesting insights pertaining to denoising and morphological operators and their appearance in unorthodox positions in image processing pipelines. Moreover, the image processing networks can be extremely varied or robust; for example, many different configurations can perform very well.

7) Based on the outcome (results) of this research, several papers related to the findings have been published.

#### **1.14 Thesis Structure**

The remainder of this thesis is organised as follows:

- Chapter 2 discusses related work conducted by other researchers in the interest area of this research. Various studies relevant to the area are discussed and their proposed methods compared. Current state-of-the-art results relevant in the research area (both published and commercial) are also highlighted.
- Chapter 3 outlines the tools and technologies used in the experiments conducted in this research and to create the proposed algorithms. The chapter also includes explanations of the hardware, software, and techniques used, with background details into the dataset used, other related information and about the performance measures chosen and used in the research.
- Chapter 4 explains in detail the work carried out in the initial stage of this research to develop the algorithms, such as fine-tuning the parameters, and creating the first stage algorithm, called the LCHF algorithm. The chapter comprises many subsections describing the experimental stages and findings.

- Chapter 5 describes the work carried out to develop the second stage algorithm, called the IPCO algorithm. This chapter also comprises many subsections explaining the algorithm, experimental stage findings, and results and analysis of the IPCO algorithm.
- Chapter 6 explains in detail the development of the third stage algorithm, called the MIPCO network. The chapter also comprises many subsections explaining the algorithm, experimental stage findings, and results and analysis of the MIPCO algorithm.
- Chapter 7, the Discussion chapter, explains the research and its achievement in relation to the aims and objectives outlined in the Introduction chapter. The novel contribution of the research to image processing pipelines and a guide for new research are also highlighted. The limitations of the research and suggested future work are also discussed.
- Finally, Chapter 8, the Conclusion chapter, briefly explains the conclusions drawn from this research.
- References and an appendices section are also included.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 Digital Image Processing

Antonie van Leeuwenhoek's, a Dutch tradesman and scientist, became the first man to make and use a real microscope in his research in the late 17<sup>th</sup> century. Using his microscope, Antoni discovered many biological discoveries, and contributed to the study of microbiology. He is known as the first person in history to observe single-celled organisms (animalcules, now known as microorganisms). He was instrumental in the development of microscopes and is called the father of microbiology. His work was studied and further enhanced by the English scientist Robert Hooke in the year 1665(Hooke & Gunther, 1938). Fast forward centuries to the current era in which the current state-of-the-art comprising advanced technological methods and equipment allows researchers to easily acquire large images in fewer hours (Dobell, 1932). According to Vonesch *et al.* (2006), one of the tools that contributed to research on images, especially medical images is the appearance of light microscopy. As early as the 1920s, newspaper images were being transmitted across the Atlantic using the Bartlane cable picture transmission system. This initial system supported only five grey levels and required a significant amount of time to transmit an image. In 1964, NASA's Jet Propulsion Laboratory used computer algorithms for its images of the moon(DeJong & Green, 1997). Presently, in the new digital era, the typical images produced by scanners and other modalities can support more than 65,000 shades of grey.

Images and videos are used in our everyday life to create and showcase our visual experiences (Milanova, 2014). Many applications engage with images and video, especially in computer vision, and help to duplicate the effect of human vision via technology and devices. It is arguable and being hotly debated that human vision is poor at judging the colour and brightness of the details in images, as it is comparative rather than quantitative. Overington (1985, 1988) disagrees with these claims but, unfortunately, there is no presence of counter-evidence for his objection. This information is available in

Russ (2011). Thus, tools, for example a segmentation tool, are needed to automate the process and enable us to not simply depend on human vision or to carry out the task manually. In this research, the goal is to automate the membrane detection process in order to eliminate or reduce human resource and time costs. Using human capability to detect details in images can be unreliable and gives results that vary from person to person. Thus, a segmentation tool which can automate the process and cost less in terms of time, energy, and money is desired.

## **2.2 Computer Vision**

Computer vision is developing in parallel with mathematical techniques. Recovering the 3D shape and appearance of objects is possible with computer vision. With computer vision the objective is to recover some unknowns given insufficient information in this rich, complex world (Szeliski, 2010).

The primary goal in the computer vision field is to exceed human vision using computer software and hardware. The computer vision field can be divided into subcategories such as low-level vision, in which images are processed for feature extraction. In low-level computer vision, very minor knowledge of the content of the images and video is used. Next is middle-level vision, which deals with object recognition, segmentation, motion analysis, and 3D reconstruction. This level receives inputs from the low-level vision category. Next comes high-level vision, which deals with the interpretation of inputs or information obtained from middle-level vision. High-level computer vision uses major knowledge, well set goals, and structured plans to achieve the goal. High-level vision imitates human cognition. High-level vision will also direct the task that should be performed by the middle and low-level vision. In the next section, the segmentation process, which can be regarded as a bridge between low and high-level vision is discussed.

In the work conducted in this research, a simple algorithm which bridges the high-level knowledge and low-level information is proposed. The optimisation

heuristics used can be considered high-level knowledge, whereas the manner in which they are used and their details can be considered lower level knowledge.

### **2.3 Segmentation in General**

The purpose of segmentation is to partition an image by defining the boundaries in non-overlapping regions. Many image segmentation algorithms have been developed. Some of these algorithms segment the image based on the object it represents, which is referred to as object-based segmentation, whereas others segment automatically, which is referred to as automatic segmentation. In automated image segmentation, the image pixels of interest are segmented into needed segments or regions (Tasdizen & Seyedhosseini, 2014). According to Orkonselenge (2004), automated image processing carries out the process based on similarity criteria across an image using an algorithm or by applying independent operators. This opinion is supported by Neubert *et al.* (2006), J. Chen *et al.* (2008), and Taye (2011). Darwish, Leukert, & Reinhardt (2003) state that local homogeneity criteria (colour and shape) (Blaschke, (2010)), play a key role in merging the decision of the automated process. Object based segmentation focuses on a group of pixel that constitute a desired object or features in the input image. Its focus is on spectral properties, shape, orientation, and adjacency to other features (Malladi, Sethian, & Vemuri, 1993).

Segmentation or labelling is often considered to be more of an art than a science, and is also often regarded as the cornerstone of image processing and analysis. It simplifies the understanding of the image from thousands of pixels to a few regions (Estellers *et al.*, 2012), Sonka, Hlavac, & Boyle (1998), and Alvarez, Jernigan, & Nahmias (1999) also stated that segmentation is one of the most important techniques for image processing, and is essential in vast areas of computer vision, (Kass, Witkin, & Terzopoulos, 1988), (Zosso *et al.*, 2011). As a result of the importance of image segmentation, researchers in this area of interest have been proposing a number of algorithms. Further, this field has become an interdisciplinary field because application of image segmentation in computer vision can be utilised in many applications, such as

remote sensing, electronics, medical, machine learning, and industrial applications (Singh & Singh, 2010).

### **2.3.1 Segmentation Techniques**

A general algorithm that works for all images does not exist because there is no general image understanding system. For example, a 2D image can represent an infinite number of possibilities (Fu, 1981). To build such a system requires vast storage and knowledge (Kass, Witkin, & Terzopoulos, 1988), (Xu & Prince, 1998). The growth of segmentation techniques is outlined below:

#### **a) Early stage**

**This stage can be categorised into three classes:**

- Clustering or characteristic feature thresholding (Rosenfeld, 1977 and 1984; Fu & Mui, 1981)
- Edge detection
- Region extraction

#### **b) Middle stage**

**This stage can be divided into three approaches:**

- Classical approach (based on histogram thresholding, edge detection, relaxation, semantic and syntactic) (Pal & Pal, 1993).
- Fuzzy mathematical approach (based on edge detection, thresholding, relaxation). According to Pal & Pal (1993), more than 30 different researchers support this approach (Mohamed, Ahmed, & Farag, 1999)
- Attempts made to use neural networks (Hopfield and Kohonen).



### c) Continuation stage

A continuation from the past years, the current method in medical imaging can be divided into eight main groups (Pham, Xu, & Prince, 2000; V. Martin & Thonnat, 2008; Zhang, Fritts, & Goldman, 2008; Dzyubachyk, Niessen, & Meijering, 2008):

- Thresholding: Binary partitioning of the image intensity (Cheng, Lin, & Mao, 1996) with filtering (Pitas & Venetsanopoulos (1990), Astola & Kuosmanen, (1997)).
- Region growing approaches: Extraction of the region based on predefined criteria (Pohle & Toennies, 2001).
- Classifiers: Pattern recognition techniques.
- Clustering approaches: Performance as with the classifier method, in which the training is unsupervised (*i.e.*, there are no output labels, only input data) (Ng *et al.*, 2006).
- Markov random field models: Statistical models used in the segmentation method.
- Artificial neural networks: Simulate biological learning.
- Deformable models: Use mathematical foundations to represent object shape and approximation theory (mechanism for data measurement and need manual interaction) (McInerney & Terzopoulos, 1995, 1996).
- Atlas guided models: The anatomy atlas is used as a reference frame in segmentation.

The types of images being used for computing can be divided into monochrome images and colour images. Because this research used monochrome images, the Table 2.1 shows summarised monochrome segmentation techniques information.

**2.3.2 Summary of Monochrome Segmentation Techniques** (Sridevi & Mala, 2012)

Table 2.1: Monochrome Image Techniques

| <b>Technique</b>              | <b>Description</b>                        | <b>Strength</b>                                      | <b>Limitation</b>  |
|-------------------------------|---|--|--|
| Histogram Thresholding        | Number of peaks correspond to a region    | Do not need prior knowledge of image                 | Do not perform well on objects with no obvious peak  |
| Edge Detection                | Detection of discontinuity                | Perform well for images with good contrast           | Do not perform well for ill-defined edges. Less immune to noise than clustering and thresholding   |
| Feature Clustering            | Each region forms a separate cluster      | Easy implementation                                  | Image dependent and feature selection unclear to obtain satisfactory results   |
| Watershed                     | Use concept of topological interpretation | Stable result and continuous detection of boundaries | Sensitivity to noise and over-segmentation   |
| Partial Differential Equation | Based on differential equations           | Fast, good for time critical applications            | Solution of a Partial Differential Equation (PDE) depends very strongly on the boundary conditions, and do not easily yield to general solutions |

Continue...

...continued

|                |                                      |  |   |
|----------------|--------------------------------------|--|---|
| Region based   | Group pixel to homogenous region     | More noise immune than edge detection methods  | Quite expensive in terms of computational time and memory. Inherent dependence on seed selection for region |
| Fuzzy          | Use ambiguity rather than randomness | Can be used for approximate inference          | Lack of universal methods for fuzzy system design   |
| Neural Network | For classification or clustering     | Utilise the parallel nature of neural networks | Longer training time needed. Need to avoid over training  |

### 2.3.3 Why segmentation is difficult

As stated above, no single algorithm is adequate for all types of segmentation. Further, segmentation plays a key role and happens to have a central position in many problems (Fu & Lu, 1977). Thus, the discussion as to why segmentation is difficult is ongoing. Image processing researchers need to be aware of this fact before engaging in the segmentation process.

Image segmentation is generally a difficult task, and the output of algorithms is affected for the following reasons (Sharma & Aggarwal, 2010):

- Missing edges
- Lack of texture contrast between the background and the region of interest.
- Partial volume effect; that is, a single image voxel may contain several types of tissues owing to the finite spatial resolution of the imaging device (Uryasev & Pardalos, 2013).

- Noisy images

## **2.4 Medical Imaging**

Medical Imaging is a process that uses technologies for visual representations to view the human body (internal structures) to diagnose, monitor, analyse, and treat diseases and disorders or abnormalities. As a discipline, it is a part of biological imaging and is known as biomedical imaging. It incorporates many imaging technologies, including the following (Haidekker, 2013):

- X-ray radiography
- Magnetic Resonance Imaging (MRI)
- Medical Ultrasonography (Ultrasound)
- Endoscopy
- Elastography—Mapping of the elastic properties of soft tissue
- Tactile Imaging—Translation of the sense of touch into a digital image
- Thermography—Primarily used for breast imaging for cancer detection
- Medical Photography
- Positron Emission Tomography (PET) (Wong, 2002)

### **2.4.1 Medical Image Segmentation hurdles**

As with per image segmentation, medical image segmentation also faces hurdles such as the following (Vovk, Pernuš, & Likar, 2007):

- Intensity inhomogeneity arises from the imperfections of the image acquisition process and reduces the segmentation accuracy.
- Presence of artefacts
- Signs of clinical interest are subtle (Mathew, Khan, & Niranjana, 2011)
- Closeness in grey level of different soft tissues

- Often textured in complex ways (Mathew, Khan, & Niranjana, 2011)
- Relatively poorly sampled, with many pixels containing more than one tissue type (same as with the partial volume effect above).
- Objects or structures of interest have complex shapes (Mathew, Khan, & Niranjana, 2011)

Up to 2010, five billion medical imaging studies had been conducted worldwide (Roobottom, Mitchell, & Morgan-Hughes, 2010).

Currently (2015), special sessions, PhD forums, tutorials, and workshops are being organised in this area to boost and encourage researchers to work harder and contribute to image processing research. As can be seen by current publications in this area researchers are still engaging in image processing research. Publications from late 2014 to the beginning of 2015 in the area of image processing include the following (to name a few):

- Guo, Zheng, & Huang (2015) with research on image watermarking.
- Stühmer & Cremers (2015) with a proposed method of fast projection for connectivity constraints in image segmentation.
- Nayak *et al.* (2015) with research in graphical models for image tracking and recognition, and Koppal & Narasimhan (2015) on photography with illumination mask.
- Dar and Bruckstein (2015) with motion compensated coding.
- Punnappurath *et al.* (2015) with face recognition research.
- Bhuyan & Borah (2014) with fundamental concepts for medical images.

The above are but a few examples of researchers who published their work in the area of image processing and segmentation. More of biomedical imaging competitions that took place over the past 10 years and some that will occur in the future are listed below. They illustrate the various advancements happening globally in the area of image processing over the years. Image processing, especially biomedical image processing, is experiencing rapid technological

development and has moved from basic research to clinical application, with funding in the billions of dollars.

#### 2.4.2 2015 Competitions in Biomedical Image Analysis

A few of the various image analysis competitions are listed in Table 2.2 below.

Table 2.2: List of 2015 Competitions (a few examples).

| <b>Competition</b>                                  | <b>Brief Description</b>  |
|---|---|
| Leaf Segmentation and Counting Challenge            | Demonstrates the difficulty of segmenting all the leaves in an image of plants, using images of tobacco plants and arabidopsis plants—associated with Computer Vision Problems in Plant Phenotyping (CVPPP, 2015).  |
| Endoscopic Vision Challenge                         | Provides a formal framework for evaluating the current state-of-the-art, gathering researchers in the field and providing high quality data with protocols for validating endoscopic vision algorithms—associated with the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI, 2015). |
| Gland Segmentation Challenge in Histological Images | Validates the performance of existing or newly invented algorithms on the same standard dataset, with Haematoxylin and Eosin (H&E) stained slides—associated with MICCAI2015 (GLAS, 2015).  |

Continue...

...continued

|   |   |
|---|---|
| Medical Imaging Methods                                   | For ischemic stroke lesion segmentation, provides a on multi-spectral MRI images (ISLES, 2015).   |
| Medical Classification                                    | Deals with image retrieval in CLEF to work on compound figures of the biomedical literature and to separate them if possible and/or attach to the sub-parts labels about the content—associated with PubMed Central (CLEF, 2015). |
| CSI 2015—The Spine Workshop & Challenge                   | Covers both theoretical and very practical aspects of computerized spinal imaging—Computational Methods and Clinical Applications for Spine Imaging (CSI, 2015).  |
| Diabetic Retinopathy Detection                            | Identify signs of diabetic retinopathy in eye images—associated with California Healthcare Foundation (DR2015).   |
| Anatomy3 Challenge  | Segmentation of abdominal organs and localisation of anatomical landmarks—associated with ISBI 2015 (VISCERAL, 2015, VISCERAL Lesion, 2015).  |
| Automatic Polyp Detection Challenge in Colonoscopy Videos | Evaluates new and existing polyp detection algorithms on a large dataset, collected and annotated at Mayo Clinic in Arizona and Hospital Clinic Barcelona (POLYP, 2015).  |

Continue...

...continued

|  |   |
|--|---|
| <p>Neonatal and Adult Brain Segmentation</p> <p>White matter Modelling Challenge</p> | <p>Provides insight into the main differences and similarities, and evaluates automatic algorithms for segmenting grey matter, white matter and cerebro-spinal fluid (NEO, 2015).</p> <p>Aims to identify the mathematical model for diffusion MRI that best describes the signal from in-vivo human brain white matter, (BRAIN, 2015).</p> |
| <p>Lung Nodule Classification Challenge</p>  | <p>Deals with quantitative image analysis methods for the diagnostic classification of malignant and benign lung nodules, (LUNG, 2015).</p>   |
| <p>Cell Tracking Challenges</p>  | <p>Expands the previous years benchmark, and fosters the development of automated tools for extremely challenging datasets (CELL, 2015).</p>  |
| <p>Retinal Cyst-Segmentation Challenge</p>   | <p>Evaluates new and existing SD-OCT retinal cyst-segmentation algorithms on a uniform dataset, Ophthalmic Image Analysis (OPTIMA, 2015).</p>   |
| <p>The Longitudinal Multiple Sclerosis Lesion Segmentation Challenge</p>             | <p>Competition in which teams apply their automatic lesion segmentation algorithms to MR neuroimaging data acquired at multiple time points from patients (Longitudinal, 2015).</p>   |
| <p>Dental Image Analysis, Bitewing Radiography Caries Detection Challenge</p>        | <p>Investigates automated methods for detection of caries in 120 bitewing X-rays (Bitewing, 2015).</p>  |

Continue...



...continued

|  |   |
|--|---|
| Diagnosis in Cephalometric X-ray Image                     | Automated detection and analysis for prediction of the locations of 19 landmarks and classification of anatomical types based on eight standard measurement methods (Chal, 2015). |
| Overlapping Cervical Cytology Image Segmentation Challenge | Extracts the boundaries of individual cytoplasm and nucleus from overlapping cervical cytology images (CYTO, 2015).   |

### 2.4.3 Popularity of Biomedical Challenges

Affordable technology solutions for clinical medical problems are favoured in nowadays, and this can be done through scientific research. The availability of good funding can contribute to good research. Today, many organisations, both educational and non-educational, are showing interest in undertaking research to benefit nations and to gain popularity. Over the past 10 years, the biomedical imaging has gained significant popularity and attention (Suzuki, 2014). Many challenges and competitions have taken place during this period. More information on the past competition and challenges is given in the Appendix section, which list information from the past 10 years; example, for some biomedical imaging competitions. The examples listed in the Appendix section are just a few of the thousands of real life competitions occurring around the world in the area of medical imaging to promote and to provide a better platform for assisting medical personnel. The following are some of the tools involved:

- a) Functional Imaging
- b) Spectroscopic Imaging
- c) Optical Imaging
- d) Image Fusion
- e) Image-guided intervention

Biomedical imaging is gaining acceptance and has moved from research at the cellular level to whole organ level research. To date, research in the area of image processing and analysis continues because it is useful and many unsolved (or partially solved) mysterious problems still exist. Segmentation is one such unsolved (or partially solved) problem, which happens to have a central position in many other problems, as applications and components depend on it. This is one of the reasons why this area of research will never diminish in years to come.

## **2.5 Segmentation in Medical Image Processing**

The aim of segmentation in medical image processing is to extract clinically relevant information from medical images. This area of image processing focuses on computational analysis of the images, not their acquisition (Suzuki, 2014)

### **2.5.1 History of Medical Image Segmentation**

Medical image segmentation can be divided into three generations (Withey & Koles, 2007; Dzyubachyk, Niessen, & Meijering, 2008). Each level involves additional and advanced algorithmic complexity added to the next level. For example, the first level deals with image analysis, the following level deals with optimisation methods and models, and the next level with the advance of technology incorporating knowledge into the process. It then progresses towards a fully automated process.

The initial level uses low-level techniques, where little information is needed; for example, thresholding, edge tracking, and region segmentation. The next level includes statistical information, such as pattern recognition, neural networks, and clustering. The need for knowledge appears to provide accurate results which spur incorporation of higher level knowledge such as expert-defined rules and shape models.

## **2.5.2 Advantages and Limitations of Medical Image Segmentation Algorithms**

The image segmentation process is crucial in medical image processing. Further, variations in intensity, contrast, and shape of cells in high resolution electron microscopy images result in the segmentation task being even more challenging as inaccurate segmentation results will affect other processing stages. To date, there is no single universal algorithm for segmentation of anatomical structures (Smistad *et al.*, 2015) in medical image segmentation. Each of the currently available algorithms has strengths and limitations. However, with the development of advance technology (X-ray, CAT, MRI, Ultrasound, Scanning Electron Microscope (SEM), TEM, Nuclear Medicine, *etc.*). 2D and 3D images can more easily be captured and information inside the body revealed for easy and accurate diagnosis and treatment planning (Huang & Tsechpenakis, 2009). Medical image segmentation reveals and facilitates visualisation of the interest portion of the images which contain a lot of information (Smistad *et al.*, 2015). As medical imaging data continue to grow, many computationally efficient methods are needed (Scholl *et al.*, 2011), and fast segmentation algorithms are becoming important and favoured. Table 2.3 compares the advantages and limitations of the most common medical imaging methods (MRI and CT scans).

### 2.5.3 Comparison MRI and CT (Mogoseanu *et al.*, 2003)

Table 2.3: MRI vs. CT scan.

| Method  | Advantages   | Limitations   |
|---------|--|---|
| MRI     | Excellent for soft tissue imaging at high resolution, and is capable of using multi-channel images with variable contrast.   | Has to take care of bias field noise (Intensity inhomogeneities in the RF field), longer time than CT scan, more difficult to obtain uniform image quality.                                 |
| CT scan | <p>Better bone detail, better in cases of trauma and emergent situations.</p> <p>Less expensive than MRI, easy to interpret by radiologists and physicians.</p> <p>Wide availability.</p> <p>Short scan time.</p> <p>Higher sensitivity than MRI for sub-arachnoids haemorrhage and intracranial classification.</p> | <p>Expensive compared to X-ray.</p> <p>In general, less sensitive than MRI (except for certain areas).</p> <p>Radiation exposure.</p> <p>Inferior soft tissue contrast compared to MRI.</p> |

Some general explanation was given above for general medical imaging. As this research is based on neuronal membrane segmentation, the next section discusses neurons and cell segmentation.

## **2.6 Neuron and Cell Segmentation**

The broad area of research interest, such as digital image processing, computer vision, segmentation in general, and medical image segmentation have been discussed above. We will now look at the flow of information for neuron and cell description as the research is about membrane cell detection in medical images.

Cell theory was developed in the 19th century (Meijering, 2012). More than a century and a half afterward, the first computer aided cell analysis was conducted in the mid-1950s. It appeared to automate the cell classification which applied thresholding for one-dimensional scans (Tolles, 1955). This was followed by automated processing of 2D images (Prewitt & Mendelsohn, 1966). Multiple computers for parallel task analysis of images appeared in the mid-1970s (Preston, 1976). Further advancements in microscopes for tracing and engaging with morphological analysis also occurred (Meijering, 2010). The research in this area is developing at great speed, with the current existence of advanced technology, and further with greater research funding and more researchers, various beneficial outputs can be presented.

## **2.7 Challenges in Neuron Segmentation**

Neuron segmentation is considered difficult for many reasons. A few of those reasons are listed below:

- Membrane contrast and thickness
- Large physical separation between shape, position, and sections, and changeable between adjacent sections.
- Presence of intracellular structures
- Ill-posed problem exist if the following conditions are not satisfied: Differences in lighting, variations or inconsistencies in inter-layer distances (Donoser, Urschler, Hirzer, & Bischof, 2009).
- Slight changes in image gradient affect the neighbouring regions.

- Local ambiguity, difficult to find object boundaries, and context needs to increase to segment the images.
- Small objects (thin lines) are difficult to trace.
- Different structures are hard to categorise by intensity differences.
- Presence of noise and microstructures (Ciresan, Giusti, Gambardella, & Schmidhuber, 2012).

A problem is classified as well-posed if it satisfies the conditions below (Tohka, 2002, 2014):

- A solution exists
- The solution is unique
- The solution depends continuously on the data

## **2.8 Gaps filled by the Proposed Algorithms (IPCO and MIPCO)**

### **2.8.1 Comparison with ISBI Competitors**

The research is concerned with the problem of neuronal membrane detection in which the core challenge is distinguishing membranes from organelles. Deep Neural Network (DNN), an early precursor to Artificial Neural Network, exploded into popularity around 2006 following a significant breakthrough achieved by Hinton, Osindero, & Teh (2006). However, DNN had many problems: it assumes that segmentation has already been done; when discrimination is difficult, it does not learn to sequentially attend to the most informative parts of objects; it is weak in handling perceptual invariances, *etc.* The ISBI 2012 winner, Ciresan, Giusti, Gambardella, & Schmidhuber (2012), adopted this method, and as published by them, DNN is slow to train, the approach needs long hours (or several days) for training. Even after the network is trained, it still took about 1/2 hour on four Graphics Processing Unit (GPUs) to conduct testing of the whole stack of the dataset. Laptev, Vezhnevets, Dwivedi, & Buhmann (2012) (the runner up of the ISBI challenge) in commenting on Dan's approach, said that the solution is slightly

better in quantitative terms, but it requires almost a week (seven days) of training time with the use of specialised hardware, and it is therefore much more difficult to apply in real-world scenarios. Laptev, Vezhnevets, Dwivedi, & Buhmann (2012) also used high-end hardware. The need for long hours of training and specialised hardware can be seen to counterbalance the advantage of both methods.

Kamentsky (2012) use freely available open source software called CellProfiler (Carpenter *et al.*, 2006; Lamprecht, Sabatini, & Carpenter, 2007) in their research with *Drosophila* images. However, a need of user judgement for smoothing and values, can cause uncertainty in the resulting data (Collette, 2015).

According to Burget, Uher, & Masek (2012), a participant in ISBI 2012, the segment-level segmentation they used succeeded in the removal of small objects, but it fails to remove some bigger objects because the objects are connected to the membrane. They also stated that their method could not connect the broken line and other promising enhancements needed to reconnect the broken (membrane) lines. Further, they suggested that using an extended set for better feature extraction would give better results for pixel error criteria.

Other researchers using the *Drosophila* dataset, Seyedhosseini *et al.* (2011, 2012) from University of Utah, used the Contextual Hierarchical Model (CHM) for scene labelling. The method only uses patch information and not shape models, but the model needs to learn hundreds of parameters (Seyedhosseini, M., & Tasdizen, 2015). According to the researchers, CHM can be prone to error due to absence of any global constraints. They suggest that some other post-processing should accompany CHM to enforce consistency and global constraints. Moreover, according to them, the CHM needs 30 hours of training time on the CPU.

Other researchers such as Iftikhar & Godil (2012) and Tan & Sun (2012) used Support Vector Machines as a classifier. According to Burges (1998), the limitation of Support Vector Machine lies in its speed, size for training and

testing data, slow test phase, choice of appropriate kernel, selection of kernel function parameters, high algorithmic complexity and, for large-scale tasks, extensive memory requirements.

### **2.8.2 Gaps with other similar area of interest researchers**

Rahnamayan & Mohamad (2010) proposed a variant of image processing chain optimisation for tissue segmentation in medical images, but the method does not have reordering flexibilities for functions with rigid structuring.

Nagao & Masunaga (1996), proposed a method for image transformation from an original image to target image with a series of filters using Genetic Algorithms. However, the sequence needs to determine adequate transformation. Aoki & Nagao (1999) use sequential image transformation, which has speed limitations.

## **2.9 Optimisation of Image Processing Algorithms**

### **2.9.1 Genetic Algorithm (GA) and Global Stochastic Optimisation**

GA was first introduced in the 1970s by Holland at University of Michigan, United States (Holland, 1975). GA is a method to solve both constrained optimisation problems, which optimise an objective function with respect to some variables in the presence of constraints on those variables and unconstrained optimisation problems. It works well in mixed (continuous and discrete) combinatorial problems. It belongs to a class of stochastic search methods, but operates on a population of solutions. GA solves problems based on a natural selection process, and repeatedly modifies a population of individual solutions (Low *et al.*, 2010). It can work on various problems and the parameter can be tweaked. It is modelled after the biological process, through computer simulation.

GA can be divided into two categories: deterministic and stochastic. Although there are two categories, deterministic GAs are not favoured as they are



unconventional, poorly researched, and have not yet shown much potential. Moreover, they are considered slow when it comes to even problems with more than a few parameters. Theoretically, stochastic GAs are more favoured and are good at widely exploring the potential solution space (Pardalos,2001), (Pardalos & E, 2002). However, these algorithms are slow at finding the local maximum, but their performance improves on finding a good area of the solution space. Lonnie *et al.* (2007) stated that global optimisation algorithms are a class of algorithms that seek to avoid getting trapped in local minima because of the diversion (fragmentation) in the population.

Several researchers use GA in multi-background problems. Chun (2014) used GA to reduce the computational time of most metaheuristics in solving combinatorial optimisation problems, Bandlaney (2006) used GA for control flow testing. Oh, Harman, & Yoo (2011) used GA for transition coverage of state flow models. Haga & Suehiro (2012)used GA to generate automatic test cases. Aishwarya & Anto (2014)proposed a clinical decision support system based on GA and Extreme Learning Machine (ELM) for medical diagnosis.

## **2.9.2 Differential Evolution (DE)**

DE is favoured because of two main advantages: (1) limited use of control parameters, and (2) fast convergence. DE uses operators which are related to those of GA; *i.e.* crossover, selection, and mutation. According to Saha et. al (2013) and Nurhan and Bahadir (2004), when considering global optimisation methods for filter design, GA is a good choice. Filters designed by GA have the potential to obtain near global optimality(S. Chen, 2000). However, in terms of convergence speed, it has disadvantages which can be partly addressed by DE, which is a simple and yet powerful evolutionary algorithm first introduced by Storn &Price (1995). Early in the literature, according to Karaboga & Cetinkaya (2004), the DE algorithm was not as common as GA (Nurhan & Bahadir, 2004), but it has picked up tremendously over the years partly because of its effectiveness and partly because of its relative simplicity. DE has been convincingly successful in solving single-objective optimisation

problems (Robič & Filipič, 2005), and several researchers are currently trying to match this success in the domain of multi-objective optimisation problems (Arunachalam, 2014).

### **2.9.3 Rank-Based Uniform Crossover**

Uniform crossover was first proposed by Ackley (1987). The operator has been successfully used in several different applications (*e.g.*, Duarte-Mermoud, Beltrán, & Salah (2013)) and has been studied theoretically at length (*e.g.*, Chicano, Whitley, & Alba (2014)). The operator involves creating a new solution, by scanning parental parameters (or alleles) one-by-one, and copying each parameter (or allele) from the best parent with probability  $P$ . Although in many studies,  $P = 0.5$  meaning that both parents are equally likely to contribute a parameter (this is referred to as equiprobable uniform crossover by Semenkin & Semenkina (2012)), in this study, the  $P$  is biased towards the stronger solution, and therefore  $P = 0.75$ . This bias towards the stronger parent is reflected in the rank-based half of the term rank-based uniform crossover (RBUC).

## **2.10 Conclusion**

This Chapter described, in general, digital image processing, computer vision, segmentation, and medical image processing. The major focus was on gaps existing in comparison with algorithms that use the same dataset and participate in the grand segmentation challenge. For every gap identified will explain in the next chapter how the IPCO and MIPCO networks work to fill it. Some comparison was also carried out with other researchers with work considered to be very much related to the interest area of this research. This proves that this research area and scope are also of interest to other researchers and it is recent in a timely manner (2012-2014). Some explanations of the stochastic global optimisation approach and adopted method were also given before concluding the chapter.

Further details and step-by-step elaboration of techniques are provided in the Methodology chapter.

## CHAPTER 3

### RESEARCH METHODOLOGY

Several gaps were identified in the Literature Review chapter. Those gaps are addressed in this chapter and brief information is given on which gaps are filled by the Image Processing Chain Optimisation (IPCO) and Multiple Image Processing Chain Optimisation (MIPCO) networks.

At the beginning of the chapter, the dataset slices and the open challenge competition in which they are used are discussed. Then, detailed explanation about the dataset, the image acquisition, the type of dataset, and other related information is given. The subsequent sections describe the software and hardware used. This is followed by the performance measures of the technique used, the reason for choosing the method and various comparisons. The final two sections describe the tools, processing functions, and techniques used to carry out this research, and how the proposed method fills the identified gaps.

#### **3.1 Background into the Data Slices used in this Research**

The dataset for the experiments was obtained from the IEEE International Symposium on Biomedical Imaging challenge. The provider allowed public access to 30 Transmission Electron Microscopy (TEM) images and their corresponding ground truth. The challenge involved segmentation of neuronal structures using the provided *Drosophila* dataset: The challenge was called Segmentation of neuronal structures in Electron Microscopy stacks (ISBI, 2012), and this symposium was the premier forum for the presentation of technological advances in theoretical and applied biomedical imaging and image computing.

As part of the research progression, and to compare the proposed method with current state-of-the-art approaches, the submission was sent to the ISBI challenge organiser as a 32 bit TIFF 3D image, with values between 0 (100% membrane certainty) and 1 (100% non-membrane certainty). The aim of the

challenge was to compare and rank the different competing methods based on their pixel and object classification accuracies. The algorithm was tested in an open challenge in which medical imaging researchers showcased their best methods and participated in direct head-to-head comparisons using standardised datasets that capture the complexity of a real-world problem. Further, a controlled experimental design and metrics were used to evaluate the results. The proposed approach (IPCO) obtained a F1 score of 90% on the unseen test datasets, in which the highest score was 94% (see the IPCO result chapter for the list of participants and placings).

### 3.1.1 Evaluation Metrics used in the competition

(The below metrics were used by IEEE International Symposium on Biomedical Imaging Challenge, 2012 for their competition evaluation)

- **Warping Error:** A segmentation metric that penalises topological disagreements (*i.e.*, object splits and mergers). However, this measure places relatively high computational demands. Instead of focusing on the pixel disagreement it focuses on segments, accounts for the number of neuron splits and mergers to obtain the desired output from gold standard, and measures the topological error (Jain *et al.*, 2010).
- **Rand Error:** Defined as  $1 - F_{\text{rand}}$ , where  $F_{\text{rand}}$  represents the  $F_1$  score of the Rand index (Rand, 1971; Unnikrishnan, Pantofaru, & Hebert, 2007). It measures the accuracy with which pixels are associated with their respective neurons. (This score is considered in the competition; the lower the score, the better, the placing).
- **Pixel Error:** Defined as  $1 - F_{\text{pixel}}$ , where  $F_{\text{pixel}}$ , represent the  $F_1$  score of pixel similarity. It expresses the square of the number of disagreements between image and ground truth.

### 3.1.2 The Dataset

The dataset used is a set of 30 sections of a serial section Transmission Electron Microscopy (ssTEM) dataset of the *Drosophila melanogaster* first-instar larva ventral nerve cord (VNC). It is a species of flies in the family

*Drosophilidae* and in the taxonomic order *Diptera*. The fly is commonly known as vinegar fly or fruit fly (Pierce, 2015). Starting with Woodworth's proposal about the use of this species as a model organism (Pierce, 2015), according to Reiter *et al.* (2001), *Drosophila melanogaster* continues to be widely used for biological research in studies mainly because about 75% of known human disease genes have a recognisable match in the genome of fruit flies (Atli, 2013), and 50% of fly protein (Atli, 2013) sequences have mammalian homologs (Reiter *et al.*, 2001).

### **3.1.3 Electron Microscopy**

#### **a) Background**

Traditionally, cell biology has relied on phosphorescence and fluorescence optical microscopy in order to analyse cells and tissues instead of using reflection and absorption electron microscopy (EM), which allows biologists to analyse sub-cellular structures such as mitochondria and nuclei.

#### **b) Transmission Electron Microscopy (TEM)**

In this research, TEM images were used. TEM was invented by Max Knoll and Ernst Ruska in 1931. TEM requires the sample to be prepared in a TEM grid and placed in the middle of a specialised chamber of the microscope. The image is produced by the microscope via fluorescent screens. TEM can be used to reveal the fine structural details of different materials, and is currently one of the most useful technologies available for visualising neuronal structures (Vu & Manjunath, 2008). D. Martin, Fowlkes, Tal, & Malik (2001) stated that a reliable automated segmentation of neuronal structures in TEM stacks is infeasible with the current image processing techniques. A solution to this problem is essential for any automated pipeline reconstruction or for mapping of neural connection in 3D images.

### 3.1.4 Image Acquisition

#### a) Preparation of the slice—Histology

Russ (2011) explained in detail about the preparation of the freshly dissected instar fly brains.

#### b) The TEM *Drosophila* Slices

Cardona *et al.* (2010), the *Drosophila larva* dataset provider, used a software package (TrakEM2) and Leginon software package (Automated Molecular Imaging group at the Scripps Institute, San Diego, CA) to automate the TEM images. They (Cardona and team) created the dataset to test their approach towards a comprehensive anatomical reconstruction of neuronal microcircuitry and delivers microcircuitry comparisons between vertebrate and insect brains (Cardona *et al.*, 2010).

#### c) The training data

The dataset used in this research is a stack of 30 images from a serial section Transmission Electron Microscopy (ssTEM) dataset of the *Drosophila* first instar larva VNC. Albert Cardona and his team provided other researchers in this interest area with public access to 30 slices of TEM images and their corresponding ground-truth images for training (Cardona *et al.*, 2010). The microcube has dimensions  $2 \times 2 \times 1.5$  microns approximately, with a resolution of  $4 \times 4 \times 50$  nm/pixel and each 2D section is  $512 \times 512$  pixels. The corresponding binary labels were annotated by an expert neuroanatomist, who marked membrane pixels with zero and the rest of pixels with one (in-out fashion). According to the provider, the images are representative of actual images in the real world, containing some noise and small image alignment errors, but none of these problems led to any difficulties in the manual labelling of each element in the image stack by the expert human neuroanatomist. As shown in below Figure 3.1, the white is for the pixels of segmented objects and black for the rest of the pixels (which correspond mostly to membranes) (Cardona *et al.*, 2010).

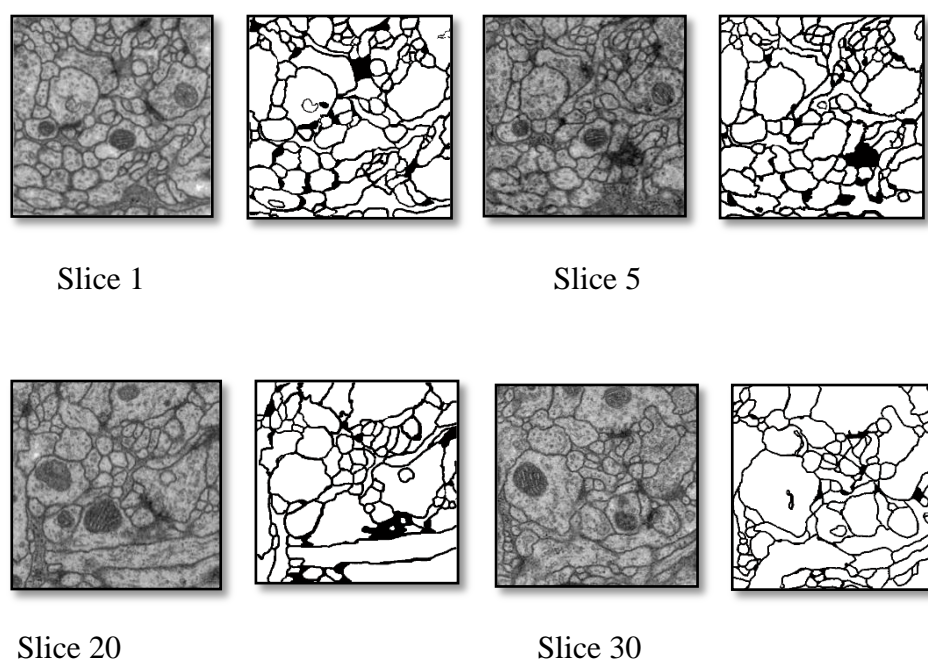


Figure 3.1: Data slices\* and their corresponding ground truths\*.

#### d) The testing data

The test data were another volume from the same *Drosophila first instar larvae* VNC used as the training dataset. The ground truth of the test data was not publicly available because the contesting segmentation methods were to be ranked by their performance on a test dataset and the contest was still open for participation.

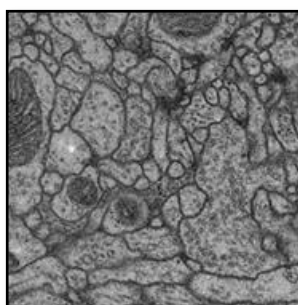


Figure 3.2: Examples of ssTEM images\* for test data.

\*The figure is a reproduction, and is to use for the purpose of generating or testing non-commercial image segmentation software (Cardona *et al.*, 2010).

### 3.2 Performance Measures used for this research

The proper choice of a metric is favoured and plays a more important role in supervised learning than in conventional hand-designed approaches (Jain *et al.*, 2010). According to Jain *et al.*, if the boundary detection algorithm is designed by hand then the performance metrics can be created later in the process, but this is not possible for supervised learning. The ideal metric suggested for machine-human disagreement should firstly tolerate minor differences in boundary location and penalise the topological disagreements (Dollar, Tu, & Belongie, 2006).

The performance of the proposed three approaches (Local Contrast Hole-Filling (LCHF), IPCO, and MIPCO) was measured in terms of precision (*i.e.*,  $tp/(tp + fp)$ ), recall (*i.e.*,  $tp/(tp + fn)$ ), and the F1 score (*i.e.*,  $2 \times (\text{precision} \times \text{recall})/(\text{precision} + \text{recall})$ ), where  $tp$  is the number of true positives,  $fp$  is the number of false positives, and  $tn$  is the number of true negatives. For each slice, a confusion matrix was computed followed by corresponding precision, recall, and F1 scores. The final performance values were averaged from the output results for each slice of the 30 slices.

The F1 score measures consider both Precision and Recall measures, and take the harmonic mean of the two measures instead of a simple arithmetic mean. For example, if Precision is 0 and Recall is 1; then, by using arithmetic mean there is 50% correct and returning 0.5 despite being the worst possible output, whereas using the harmonic mean would return F1 measures of zero. In other words, precision and recall both have true positives in the numerator and different denominators. To average them, it really only makes sense to average their reciprocals; thus, the best way is by using harmonic mean. Consequently, a high F1 score requires both high precision and recall.

As stated above, the performance of the algorithm was measured in terms of Precision, Recall, and F1 score:



$$Precision = tp / (tp + fp) \quad (1)$$

where  $tp$  is true positives (*i.e.*, the number of pixels correctly labelled as belonging to the positive class) and  $fp$  is false positives (*i.e.*, number of pixels incorrectly labelled as belonging to the membrane class).

$$Recall = tp / (tp + fn) \quad (2)$$

where  $tp$  is true positives and  $fn$  is false negatives (*i.e.*, number of pixels which were not labelled as belonging to the positive class, but should have been).

Pixels that are falsely identified as a boundary in the output, but are classed as the cell interior pixels in the ground-truth image are referred to as false positives. Conversely, pixels that are identified as interior in the output, but are classed as a boundary in the ground-truth image are referred to as false negatives.

$$F1 = 2((Precision \times Recall) / (Precision + Recall)) \quad (3)$$

where F1 is a measure of a test's accuracy. The F1 score can be interpreted as a weighted average of the precision and recall, with the F1 score reaching its best value at one and worst score at zero.

For each slice, a confusion matrix was computed followed by corresponding precision (1), recall (2), and F1 scores (3). The final performance values were averaged from the results corresponding to each one of the 30 slices.

In this research, F1 measures were used instead of Rand index (as per the ISBI challenge), because the Rand penalises even slightly misplaced borders. The frequency of pixels belonging to which objects is considered in Rand error calculation and it gives equal weight to false positives and false negatives. In this research also, the Warping error measurement was not adopted because it completely disregards non-topological error information. Ciresan, Giusti, Gambardella, & Schmidhuber (2012), the winner of the ISBI 2012 challenge

stated that even for their experiment, Rand and Warping error are not a choice and are just minimised as a side-effect, but never explicitly accounted for during the training process. According to them, the pixel classifier method is used with the aim of minimising pixel error. The pixel error metric is simple and does not lead to qualitative differences in the output image.

### **3.3 The Platform: MATLAB and the Image Processing Toolbox**

The research algorithm was created based on the sequence of basic image processing functions adapted from MATLAB. MATLAB is a mathematical computing software, and the image processing toolbox is one of the most useful and popular toolboxes. It is very useful for researchers and students in the area of image processing. This toolbox is useful for the processing, visualisation, and analysis of images, while MATLAB is convenient for rapid prototyping, has proved necessary in research laboratories, similar to the way Microsoft Office is used in office settings. MathWorks is the provider of MATLAB(MatLab, 2012).

#### **Hardware used in experiments and for creation of the algorithm**

Computer Processor: Intel Core i3 CPU

2.40 GHz

Installed memory (RAM): 4.00 GB

System type: 32 bit Operating System

The algorithm was also tested on a lower specification personal computer with 1.60 GHz processor and 1.48 GB of RAM, and was shown to run efficiently without crashing.

### 3.4 Creation of the Algorithm

The research effort was not to create new individual image processing functions, but to optimally select, configure, and combine existing functions.

In carrying out the research, from the initial to the final stage of development, many techniques were introduced, tested, and analysed. Finally, the approach used adopted hybrid global stochastic optimisation, which combines elements of GA, Differential Evolution (DE), and rank-based uniform crossover (RBUC) (the probabilistic mingling and RBUC are the same). The research used the adopted method to implement the IPCO and MIPCO frameworks.

The proposed algorithms use a larger set of functions and the combination framework is less rigid in structure, and provides reordering flexibility with no ordering constraints, compared to Rahnamayan & Mohamad(2010), who use image processing chain optimisation for tissue segmentation in medical images.

The algorithm proposed is similar in capability to tree structural image transformation, where it is possible to have single and also multiple input functions such as image blending. In contrast to the work of Aoki & Nagao, (1999) and Nakano *et al.* (2010), the approach differs in terms of optimisation method, parameterisations allowed, set of filters, type of functions, adoption of combiner functions, choice of dataset, and types of analyses conducted. In this research framework, the research included a new category of special-purpose ‘combiner’ functions specifically designed to encourage chains to form different representations and transformations. This research was conducted using systematic analyses of the statistics of optimised chains, and revealed several interesting and unconventional insights pertaining to preprocessing, classification, post-processing, and speed. In other words, the types of analyses that were conducted are novel, and have, for example, revealed interesting insights pertaining to denoising and its appearance in unorthodox positions in image processing pipelines (several papers were published to showcase these results).

### **3.5 IPCO And MIPCO Internal Framework for Optimisation.**

In the implementation of Global Stochastic Optimisation (GSO) for this research, the GSO used three main heuristics (*i.e.*, Genetic Algorithm (GA), Differential Evolution (DE), and Rank Based Uniform Crossover (RBUC)); mutation and crossover are heuristics within GAs. Further details can be found in the Appendix section.

#### **3.5.1 Experimental Design of the Approach**

Following the development of both algorithms (IPCO and MIPCO), the following experiments were designed and conducted to evaluate their performance.

##### **a) Experiment 1**

###### **Evaluation of the efficacy of IPCO and MIPCO on datasets.**

Algorithm: IPCO, MIPCO

Objective: To test and measure the effectiveness of IPCO and MIPCO

Experimental procedure:

The experiments were executed 50 times using IPCO, and 50 times using MIPCO. In the results obtained, the occurrence of each functions and chains was analysed. The information was then plotted, viewed graphically, and further analysed.

##### **b) Experiment 2**

###### **Experiments to obtain an optimal value for IPCO.**

Algorithm: IPCO, MIPCO

Experiment with varied chain lengths.

Objectives:

- To study the trends resulting for each experiment.

- To determine the mandatory functions for image segmentation that should be chosen for optimisation.
- To observe the occurrence, and the frequency of repetition.
- To study the shortest and longest possible chains for all scores > 91% or > 92% (if available).

Experimental procedure:

Chain lengths were varied from one to eight.

Experiments were executed >50 times with the IPCO version frozen.

The differences in speed vs. accuracy for the shortest and longest possible chains scoring > 91% were measured.

Prediction:

The shortest chain will consist of Thresholding as the choice of function.

The second shortest chain will consist of Denoising + Thresholding or Contrast Enhancement + Thresholding.

The longest and best chain will consist of hole file + watershed function.

### c) **Experiment 3**

#### **Comparison of IPCO to MIPCO.**

Algorithm: IPCO, MIPCO

Objective: Learn and analyse the sensitivity and inconsistencies in the scores, and type of chains and functions being chosen. The structure can also be modified and rearranged to determine the best combination out of the 30 images.

Experimental procedure:

Questions arising from the experiments:

- i) Which method performs better to achieve the set target? Compute the performance for the variations (grow the algorithm step-by-step). Find the single best algorithm that repeats and gives a constant result.

- ii) Find the shortest functions and shortest chains that score  $> 91\%$  or  $92\%$  for both IPCO and MIPCO. Determine the differences and similarities.
- iii) Identify the mandatory function that always appears in chains with the following characteristics:
  - a. F1 score greater than  $90\%$
  - b. F1 score greater than  $91\%$
  - c. Determine the parameter being used for each chosen function.
  - d. Discuss the sensitivity of the results. What is being directly affected by the sensitivity of the score results? For much higher scores, what information are lost in comparison with the original image and ground truth? Plot a visual graphical image for inspection. What is the suggestion?
- iv) Discuss the inconsistencies. Different images require different specific levels. Consequently, successive sets of five images in a total of 30 images were used:
  - a. First five images (Images 1-5)
  - b. Next five images (Images 6-10)
  - c. Next five images (Images 11-15)
  - d. Next five images (Images 16-20)
  - e. Next five images (Images 21-25)
  - f. Next five images (Images 26-30)

Prediction:

For Question (iii), the mandatory function will be thresholding + denoising for the shortest chain.

The longest chain will consist of Thresholding + Denoising + Morphological Operators + Watershed + Hole-Filling, for both scores  $> 90\%$  and  $> 91\%$ .

For Question (iv), the higher the score, the more the membrane is ignored. The scores will differ for (a-f). However, in choice of functions, the result may be the same.

#### **d) Experiment 4**

**Compare the gaps in IPCO and MIPCO (several variations) and both approaches with the ISBI competitor.**

Algorithm: IPCO, MIPCO

Objective: To compare the limitations of the competitor with the strength of IPCO or MIPCO networks.

### **3.5.2 Creation of the algorithm**

The algorithm was created in five stages:

#### **Stage 1:**

##### **a) Manual Tuning**

Several fine-tuning experiments were carried out in order to obtain a favourable set of functions and parameterisations in terms of accuracy (*i.e.*, F1 score) and speed, vis-à-vis the ssTEM images from the ISBI 2012 challenge, as will be explained in the Result chapter.

##### **b) Best Optimal Parameter for LCHF**

This stage is known as the LCHF stage, to obtain the Best Optimal Parameter for Functions used in the Creation of the First Stage of the Algorithm. Using the favourable set of functions and parameterisations in Stage 1(a), Stage 1(b) outputs the result using the selected best optimal parameter, and creates an algorithm known as the LCHF algorithm.

#### **Stage 2: Automated Stage – IPCO chain**

This stage is known as the IPCO stage. The first stage of the algorithm is improved with the adoption of a hybrid GSO method in its framework, which includes combinations of elements of GA, DE, and RBUC.

### **Stage 3: Preprocessing and post-processing stages**

Several observations pertaining to denoising functions and morphological operators and their appearance in an unorthodox position in image processing chains, and suggestion of a new set of pipelines for image processing are made.

### **Stage 4: Performance Booster by creating ensembles**

From the experimental results, it was discovered that the ensemble of the algorithm gave better results (from several high scoring IPCO chains). This resulted in the new idea of further modifying the algorithm to perform better and return a much higher score.

### **Stage 5: Automated Stage - MIPCO network.**

This improved version of the algorithm is better than ensembles because the chains can optimise together and interact with each other. It processes the information in parallel and combines the results for better performance and accuracy. This is in contrast with ensembles which train separately and combine later.



### 3.5.3 Flow of the procedure

#### Flowchart of Stage 1:

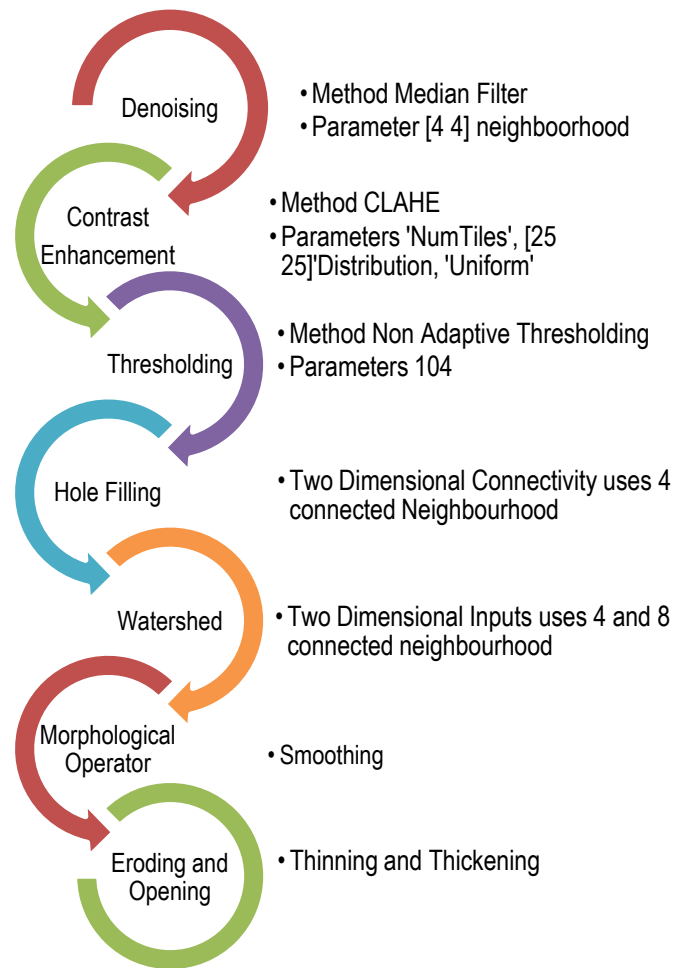


Figure 3.3: Flowchart showing the overall computational flow in a specific chain, with fine-tuning in selection of favoured functions and its parameterisation.

The proposed algorithm, called Local Contrast Hole-Filling based Membrane Detection (LCHF), recognises cell membranes while simultaneously ignoring organelles. At this stage the aim was to select the most effective tuning of a predefined processing pipeline. Because the component methods are critically dependent on some parameters, this stage serves also to determine the ranges of the effective values of parameters in the processing pipeline for the detection of cell membranes which were simultaneously capable of ignoring organelles. LCHF essentially consists of a sequence of preprocessing steps (*i.e.*, denoising and contrast enhancement), classification steps (*i.e.*, thresholding and hole-filling), and post-processing steps (*i.e.*, smoothing with morphological

operators). Each processing step has its own parameters which require some data-dependent fine-tuning.

Thresholding is primarily responsible for membrane detection, whereas hole-filling is primarily responsible for organelle elimination. Finally, the algorithm proceeds to smooth (post – processing) the results via morphological operators such as erosion and dilation. In order to evaluate the algorithm, and based on the processed output and ground-truth data, a confusion matrix and related performance metrics are also computed.

### Flowchart of Stage 2:

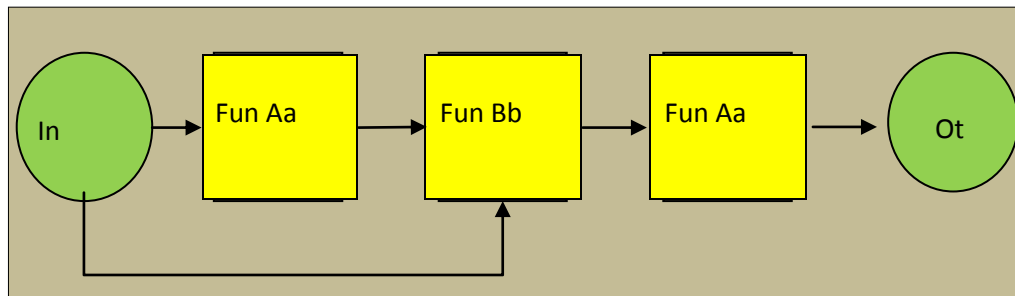


Figure 3.4: Flowchart showing the overall computational flow in a specific chain consisting of three functions. In: input image. Ot: output image. FunAa: single-input function such as denoising. FunBb: multiple-input function such as image blending.

In this stage (Stage 2), the automated algorithm is called the IPCO algorithm, and is in essence application of GSO to image processing chains. IPCO is fully automated and incorporates elements of GA, DE, and RBUC, in an effort to obtain a more robust approach. The optimisation algorithm has several basic image processing functions available to it, which it selects and configures in different sequences and with different parameter settings in response to the cost function, defined as the F1 score relative to a subset of the ISBI 2012 training images. In this part of the research, the goal is to preserve the simplicity and efficiency of LCHF while allowing for a more systematic and powerful approach.

Using IPCO, the algorithm runs automatically to reach the target cost of zero or a maximum of 10000 generations, whichever occurs first. The Results section discusses the best result obtained thus far and how IPCO can lead to a diverse set of useful chains, many of which consist of unorthodox sequences and choices of functions.

Table 3.1: Main categories of processing functions available to IPCO in the implementation reported in this research (there is no order restriction and it can appear in any order).

| <b>Main Processing Functions</b> | <b>Parameter Choice in IPCO</b>                             |
|----------------------------------|---|
| Thresholding                     | Single and Double Thresholding Value                        |
| Contrast Enhancement             | CLAHE (NumTiles, Alpha, ClipLimit)                          |
| Denoising                        | Median Filter and Wiener Filter                             |
| Watershed                        | Two Dimensional Inputs uses 4 and 8 connected neighbourhood |
| Hole Filling                     | Two Dimensional Connectivity uses 4 connected Neighbourhood |
| Combination Function             | MinMax, Average and Multiply                                |
| Morphological Operators          | Eroding and Opening   |

The end result of IPCO processing is image pixels classified as membrane being labelled '1' and pixels classified as non-membrane being labelled '0'. The 0-labelled pixels include various organelles that are eliminated from the image. These binary 0-1 images are compared with the binary images of the ground truth to find pixels that have been identified correctly and incorrectly.

### **Flowchart of Stage 3:**

The flowchart in this stage is same as the flowchart of Stage 2, but with pre- and post-processing and their appearances in unorthodox positions which boost the performance and reveal interesting findings highlighted.

**Flowchart of Stage 4:**

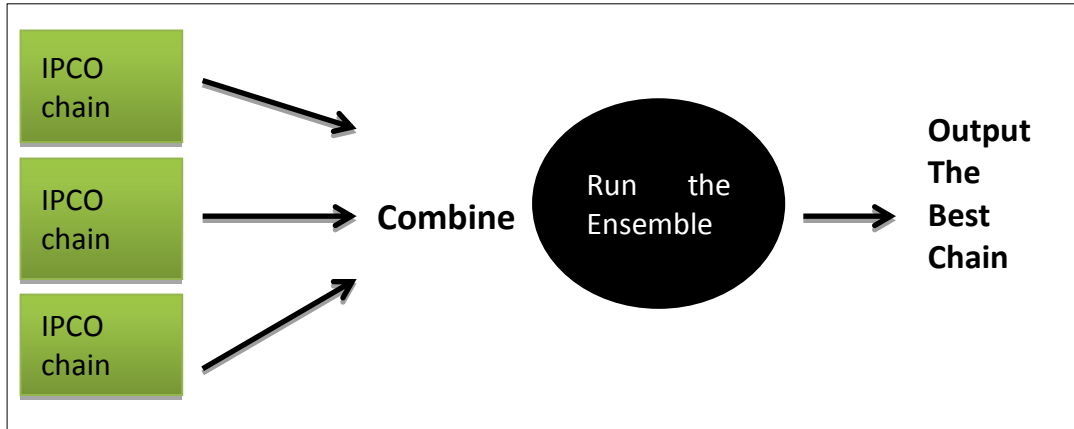


Figure 3.5: Ensemble

**Flowchart of Stage 5:**

Table 3.2: Main categories of processing functions available to MIPCO networks in the implementation reported in this research (there is no order restriction; it can appear in any order)

| <b>Main Processing Functions</b> | <b>Parameter Choice in MIPCO networks</b>                            |
|----------------------------------|--|
| Thresholding                     | Single and Double Thresholding Value                                 |
| Contrast Enhancement             | CLAHE (NumTiles, Alpha, ClipLimit), Histogram Equalization, ImAdjust |
| Denoising                        | Median Filter, Wiener Filter, Imfilter                               |
| Edge Detection                   | Sobel, Prewitt, Roberts, Log, Zerocross, Canny                       |
| Watershed                        | Two Dimensional Inputs uses 4 and 8 connected neighbourhood          |
| Hole Filling                     | Two Dimensional Connectivity uses 4 connected Neighbourhood          |
| Combination Function             | MinMax, Average, Multiply, Subtract, Addition                        |
| Morphological Operators          | Eroding and Opening  |

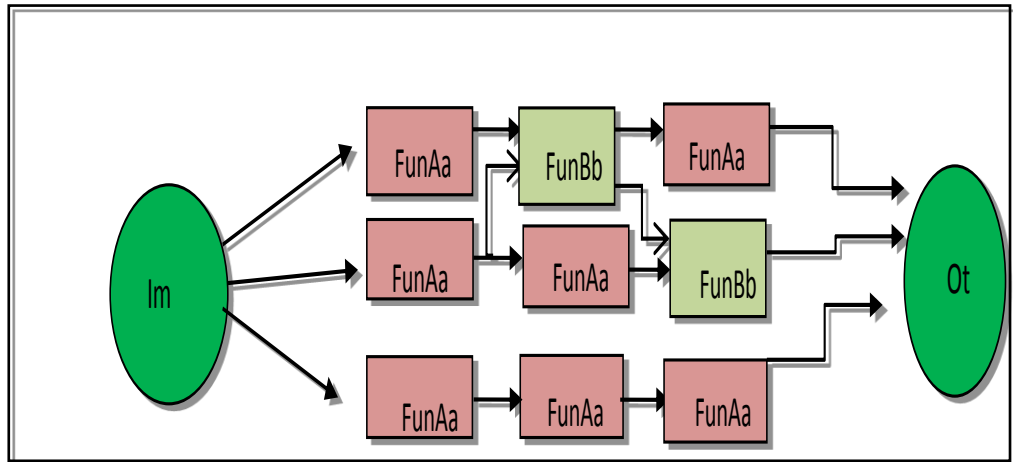


Figure 3.6: Flowchart showing the overall computational flow in a specific network consisting of three functions, and three layers of chains (for illustration purposes). Im: input image. Ot: output image. FunAa: single-input function such as denoising. FunBb: multiple-input function such as image blending.

At the algorithm creation stage, the improved version of the algorithm is called MIPCO networks. The algorithm at this stage consists of multiple chains that operate in parallel, optimise together, and interact with each other to produce the best output with the highest score. As per IPCO, the end results from MIPCO networks classified as membrane are labelled ‘1’ and pixels classified as non-membrane are labelled ‘0’.

### 3.5.4 Image processing functions used

Fine-tuning experiments were conducted to determine the most favourable set of parameters in terms of accuracy (*i.e.*, F1 score) and speed.

#### a) Denoising

In the experiments conducted, various types of denoising algorithms, such as Median, Gaussian, Wiener, Average, and Laplacian, were tried.

## **b) Contrast Enhancement**

With suitable parameter choices, CLAHE significantly improves accuracy, and it exchanges the grey value of the pixels with those of neighbouring pixels to improve local contrast (Jurrus *et al.*, 2009; Venkataraju *et al.*, 2009). Before choosing CLAHE as an essential function in the algorithm, experiments were carried out using Adaptive Histogram Equalisation (AHE)), and several global contrast enhancement methods (*i.e.*, Histogram Equalisation (HE), Adjusting Image Intensity Values (Imadjust), and Contrast Limited Histogram Equalisation (CLHE)). It was discovered that CLAHE can reduce over-amplification of noise using Adaptive Histogram Equalisation. The algorithm design aim was to provide a simple and computationally efficient method for cellular membrane detection.

In Contrast Limited Adaptive Histogram Equalisation, the approach consists of processing small regions of the image (called tiles) using histogram specification, (Gonzalez, Woods, & Eddins, 2010)for each tile individually.

The operation of CLAHE is as follows:

- $I_m$ : Image that needs to be processed for contrast enhancement
- $T_m$ : The output image following contrast enhancement
- $R_w$ : Window that moves to change the pixel value
- $(m, m)$ : Determines the height and width of  $R_w$

First, the image  $I_m$  is padded with  $(m - 1)/2$  pixels on all sides to prevent it meeting the border. The window,  $R_w$ , rearranges each pixel in  $I_m$  to exchange its value with that of neighbouring pixels, according to the defined window size and type, and outputs the result as  $T_m$ .

Experiments illustrating the different performance effects of various contrast enhancement techniques are shown in the Result chapter.

## **c) Thresholding**

Thresholding is a simple form of image segmentation which can convert greyscale images to binary images (L. Shapiro & Stockman, 2001, L. G.

Shapiro & Linda, 2002). It replaces each pixel with white and black pixel accordingly. Researchers such as Hu, Hoffman, & Reinhardt (2001) used grey-level thresholding to develop a technique to recognise lungs automatically. Farag, El-Baz, & Gimelfarb (2004) applied optimal grey-level thresholding and Antonelli, M., Lazzarini, B., & Marcelloni, F. (2005) used an iterative grey-level thresholding to perform segmentation. In this research, thresholding was adopted to perform membrane detection. Further, thresholding is favoured in this research in optimised chains and several experiments show that thresholding performs well in all chains (refer to the Results chapter for further details).

The thresholded (binary) image  $g(x, y)$  is defined as (Gong, J., 1998), (Gonzalez, Woods, & Eddins, 2010):

$$g(x, y) = \begin{cases} a & \text{if } f(x, y) > T \\ b & \text{if } f(x, y) \leq T \end{cases} \quad (1)$$

Pixels labelled  $a$  correspond to objects, whereas pixels labelled  $b$  correspond to the background.

Multiple (dual) thresholding classifies a pixel at  $(x, y)$  as belonging to  $c$  if  $f(x, y) \leq T_1$ , to  $b$  if  $T_1 < f(x, y) \leq T_2$ , and to  $a$  if  $f(x, y) > T_2$ . That is, the segmented image is given by (Gonzalez, Woods, & Eddins, 2010):

$$g(x, y) = \begin{cases} a & \text{if } f(x, y) > T_2 \\ b & \text{if } T_1 < f(x, y) \leq T_2 \\ c & \text{if } f(x, y) \leq T_1 \end{cases} \quad (2)$$

where  $a$ ,  $b$ , and  $c$  are three distinct intensity values, and the user converts them into greyscale values for easy visualisation.

#### **d) Hole-Filling**

Hole-filling was incorporated in this research for indirect classification of organelles. According to Wang & Oliveira (2003), the identification of holes and the reconstruction of missing parts using appropriate parameters are the main issues that need to be solved for each hole-filling process.

MATLAB's built-in hole-filling function is based on morphological reconstruction, and works on binary and greyscale images (MatLab, 2012). The function also allows for manual selection of points of interest, but because at this stage of development of the algorithm the aim is for an automated algorithm, the algorithm does not involve any manual selection of points of interest for hole-filling.

#### **e) Watershed**

Watershed is a popular image processing method, but sometimes it is not favoured owing to its tendency for over-segmentation. Proposals are being made by many researchers to merge most initial over-segmentations to give a good final segmentation. The algorithm used in the Image Processing Toolbox is adapted from Meyer's flooding algorithm (Meyer, 1994).

In the initial stage of the research, integration of watershed into the algorithm was adopted to eliminate a 'jutting line' artefact. In the latter stage of the research, it was observed that, of the output results, the watershed function typically appears later in chains, in which the output image (with only membrane lines left over) does not allow for much over-segmentation to occur, at least for this membrane segmentation problem.

#### **f) Morphological Operator**

Two morphological functions are available to the optimisation process: opening (erosion followed by dilation) and eroding. Note that although these functions are typically categorised as post-processing functions, optimised chains often show them in unorthodox positions (even in early stages), which calls for caution in the categorisation of functions.

#### **g) Simple combination functions**

The following five combination functions were mainly used successfully in the algorithm:



- i.*        Combine-Average: This function computes the average of the output of the previous processing step and the output of any random previous processing step;
  
- ii.*       Combine-Addition: This function adds the output of the previous processing step to the output of any random previous processing step;
  
- iii.*      Combine-Subtraction: This function subtracts the output of any random previous processing step from the output of the previous processing step;
  
- iv.*       Combine-Multiply: This function computes the product of the output of the previous processing step and the output of any random previous processing step and multiplies the result by a scaling factor;
  
- v.*        Combine-MinMaxTwo: This function compares the output of the previous processing step to the output of any random previous processing step, pixel by pixel, and takes either the minimum or the maximum (depending on which function is selected).

Below Table 3.3 gives a summary of the proposed algorithms used in the research and the corresponding functions used.

Table 3.3: Summary of Proposed Algorithms and Functions Used

| <b>Algorithm</b> | <b>List of Function Used</b>   |
|------------------|--|
| LCHF             | Contrast Enhancement<br>Denoising<br>Thresholding<br>HoleFilling<br>Morphological Operator for Smoothing |

Continue...

...continued

|                               |   |
|-------------------------------|---|
| <p>IPCO<br/>and<br/>MIPCO</p> | <p>Contrast Enhancement<br/>Denoising<br/>Thresholding<br/>HoleFilling<br/>Edge Detection<br/>Watershed<br/>Morphological Operator<br/>Combination Function</p> |
|-------------------------------|---|

The algorithm creation, process, framework, functions, and other related information have been explained above; the following highlight the capabilities of the created algorithm. In Chapter 2, gaps were identified; here a brief explanation of how the gaps are filled by IPCO and MIPCO is given.

### 3.5.5 Filling the gap: Comparison

Chapter 2 discussed the gaps in this research area. Below Table 3.4 shows the corresponding gaps/deficiencies filled for various researchers.

Table 3.4: Gaps Filled by IPCO chain and MIPCO networks

| Competitor   | Gap filled   |
|--|--|
| Ciresan, Giusti, Gambardella, & Schmidhuber (2012) | IPCO is fast to fine-tune and optimise. No specialised hardware is required in the IPCO and MIPCO approaches.                    |
| Laptev, Vezhnevets, Dwivedi, & Buhmann, (2012)     | No specialised hardware is required for IPCO and MIPCO approaches, a standard Personal Computer is used for average performance. |
| Kamensky (2012)                                    | IPCO and MIPCO combine multiple approaches to create a competitive algorithm which can be modified and manipulated.              |

Continue.....

...continued

|  |   |
|--|---|
| Burget, Uher, & Masek (2012)                     | IPCO and MIPCO can remove both small and large objects.   |
| Seyedhosseini <i>et al.</i> (2012)               | IPCO and MIPCO are speedy, not time-consuming, and are accompanied by pre- and post-image processing for better and more accurate results.  |
| Iftikhar & Godil (2012)<br>Tan & Sun (2012)      | The limitations of Support Vector Machine (SVM) generally counterbalance its performance. As stated, IPCO and MIPCO are both fast in the training and testing phase, and are very accurate (above 90% F1 score).  |
| Rahnamayan & Mohamad (2010)                      | The proposed approach uses a larger set of functions and the combination framework is less rigid. For instance, IPCO chain and MIPCO network provides reordering flexibility ( <i>i.e.</i> , IPCO and MIPCO has no ordering constraints—classification can be conducted before preprocessing). This order flexibility, although simple, provides new insights into image processing pipelines, with classification often being done before denoising, at least in the domain of membrane detection. |
| Nagao & Masunaga (1996)                          | IPCO and MIPCO also do not place any restrictions on the order of functions.  |
| Aoki & Nagao (1999), Nakano <i>et al.</i> (2010) | The approach differs in terms of optimisation method, set of filters, types of functions, adoption of combiner functions, choice of datasets, and types of analyses and testing conducted.  |

IPCO was also tested in an open challenge in which medical imaging researchers showcased their best methods and participated in direct head-to-head comparisons, with standardised datasets that capture the complexity of a real-world problem, using a controlled experimental design and metrics to evaluate the results. IPCO obtained an F1 score of 90% on the unseen test

dataset, in which the highest score was 94% (see the Results chapter for further details).

### **3.6 Conclusion**

This chapter described the background of the dataset, the performance measure, the platform used for both software and hardware, the internal framework adopted, and the creation of the algorithm with an optimisation approach. The stages involved in the creation of the algorithms and flowcharts for visual representation of the flow of the algorithms were also discussed. Explanation of the experimental design was given to show how the statistics of the experiments were collected. Finally, the gaps identified in Chapter 2 were addressed at the end of this Chapter. For further details on step-by-step technique elaboration and the outcome of the result see the Results chapter.

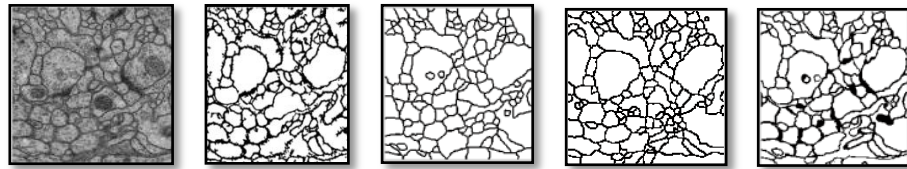
## CHAPTER 4

### RESULTS

#### LOCAL CONTRAST HOLE-FILLING ALGORITHM

This chapter presents the key results of the experiments conducted and the contribution of the research towards the creation of the algorithms, based on the methods described in the Methodology chapter. The research contributes three algorithms. This chapter discusses the first algorithm, called the Local Contrast Hole-Filling algorithm. Further, the corresponding results obtained from experiments conducted are analysed and interpreted. In general, the results are presented in tables and figures.

The segmentation results below are the outputs obtained using Local Contrast Hole-Filling (LCHF), Image Processing Chain Optimisation (IPCO) chain, and Multiple Image Processing Chain Optimisation (MIPCO) network.



Original Image LCHF output IPCO output MIPCO output Ground truth

Figure 4.1: Segmentation result obtained using the LCHF algorithm, IPCO, and MIPCO network compared with the original image and corresponding ground-truth image.

#### **4.1 Rationale of introducing LCHF algorithm when its F1 score was 71% which is way below IPCO and MIPCO F1 scores?**

LCHF is reported as the first step algorithm that contributes to the idea of the creation of the next step IPCO and MIPCO algorithm. It is a crucial historical step of the research. The LCHF algorithm helps with manual parameter tuning. This stage allows researchers to get a feel for the underlying methods and the

way the research was organized. Moreover, it helped the research in getting information on criteria to be used for the evaluate function. This algorithm was included in the research chapters, since it is a crucial step of the history of my research and at this stage of the research, most work was done on selecting the best fitted image processing function for solving the research problem out of a pool of image processing functions by using the knowledge gained through reading the literature. So highlighting it as a chapter was crucial for the research. This research also has published work related toLCHF.

## **4.2 Initial Startup**

This research deals with the problem of neuronal membrane detection in which the core challenge consists of distinguishing membranes from organelles. The methodological focus of the research is to select the most effective method of tuning a predefined processing pipeline and determine the ranges of the effective values of parameters in the processing pipeline.

LCHF satisfies the main aim and some of goals of the research. As regards the main aim, it rapidly detects the membrane (21 seconds) at a low cost (with no specialised hardware), and is easily implementable for adoption by new researchers in the area of Image Segmentation and Classification. LCHF is also a simple and efficient approach based on several basic processing steps, including local contrast enhancement, thresholding, denoising, hole-filling, watershed segmentation, and morphological operations. Because the component methods are critically dependent on some parameters, LCHF serves also to determine the ranges of the effective values of parameters in the processing pipeline for the detection of cell membranes which are simultaneously capable of ignoring organelles. The overall process engages with exhaustive search for the most effective tuning of a predefined processing pipeline.

As the aim of this research is to design and implement a simple, efficient, and easily adopted method for membrane detection, at this early stage of the research, LCHF, which is a non-learning approach, was suggested and adopted.

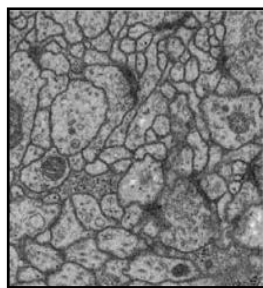
Other simple non-learning methods such as Edge Detection, Simple Thresholding, Intensity Thresholding (on enhanced membrane features), Diffusion, and Graph Cuts tend to be inadequate for membrane detection and organelle elimination.

The experimental results show that these simple methods cannot solve the problem of membrane detection and organelle elimination by themselves. This is an important early step of the research that needs to be highlighted in this chapter.

### **4.3 Experiments using existing simple segmentation methods**

#### **4.3.1 Edge Detection**

In a greyscale image, edge detection detects the outline or edges of structures and it is a fundamental tool in image processing, in the area of feature detection and extraction. However, this method results in many unwanted edges given the presence of intracellular structures (*e.g.*, organelles). It recognises many unwanted structures that lead to a high proportion of false positives, which results in error metrics calculation, and low accuracy. The disadvantages of the method outweigh its reputation for speed and easy to use capability. Figure 4.2 shows a microscopic image of neuronal structures (left) and outputs generated by different edge detection methods (namely, Canny (Canny, 1986), Laplacian, Sobel, Prewitt (Prewitt & Mendelsohn, 1966), Roberts, and Log).



Original Image

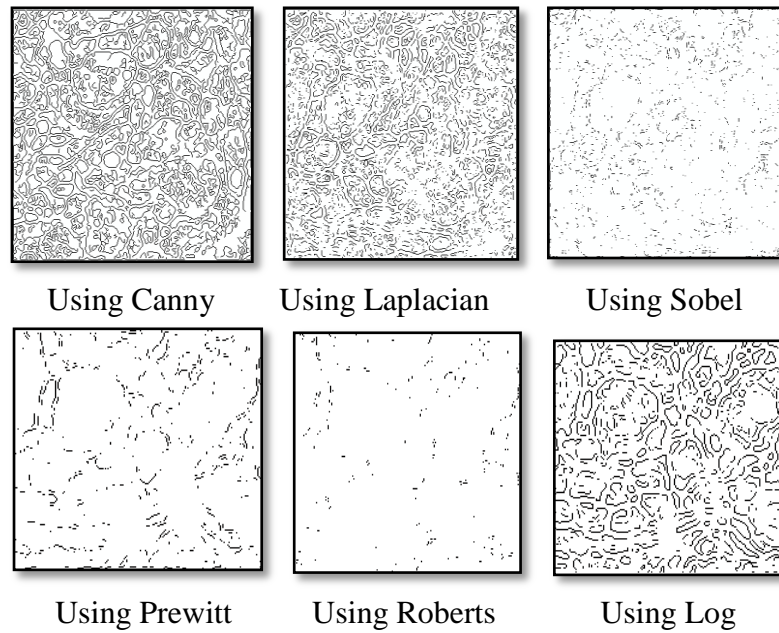


Figure 4.2: Simple comparison of different edge detection methods for *Droshopila* dataset (greyscale image).

The above figure shows that standard edge detection methods do not perform well on the *Droshopila* dataset. They not only detect the membranes for this dataset, but also detect other intracellular structures. Thus, it is clear that standard edge detection methods on their own are not suitable for the *Droshopila* dataset. However, when combined with other functions they may provide better results.

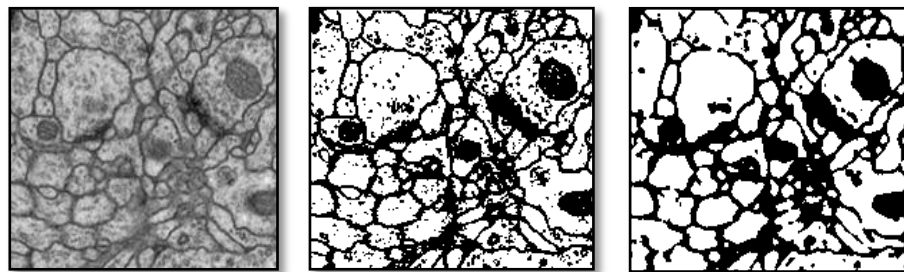
#### 4.3.2 Simple thresholding with enhanced membrane features

Thresholding is well-known as the simplest method of image segmentation. It can create a binary image from a greyscale image. However, when further separation of information is required, thresholding will not suffice by itself. On the other hand, this method can be combined with other functions to give excellent results. In this research, thresholding is used with other enhanced functions such as contrast enhancement, denoising, hole-filling, morphological operations, and watershed. The use of these functions in combination results in improved accuracy in membrane detection and unwanted information elimination.



Adaptation of thresholding with additional functions is adequate for some datasets. However, for the *Drosophila* dataset, its performance is the same as that of thresholding when used by itself. The experimental results of thresholding with extra enhancements on *Drosophila*, *C.Elegans*, and Rabbit Retina datasets are shown below. Two examples of the thresholding techniques with added extra enhancements are shown: thresholding with anisotropic smoothing and thresholding with gradient magnitude.

**a) Thresholding (TH) and Anisotropic Smoothing (AS)**



Original Image

Thresholding alone

TH with AS

*Drosophila* dataset

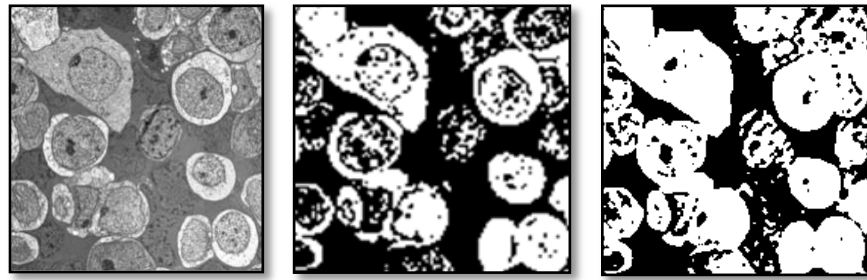


Original Image

Thresholding alone

TH with AS

*C.Elegans* dataset



Original Image

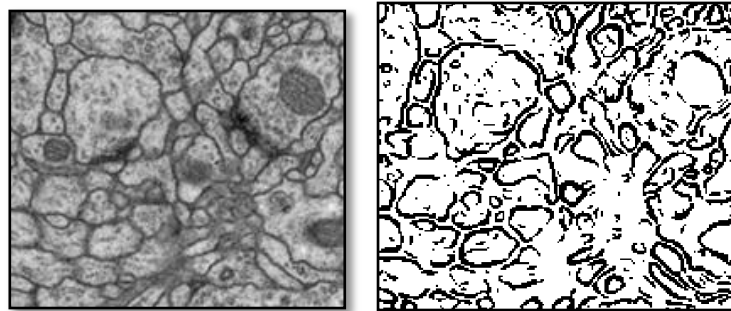
Thresholding alone

TH with AS

Rabbit Retina dataset

Figure 4.3: Output using thresholding alone and thresholding with anisotropic smoothing (TH with AS) for three different datasets.

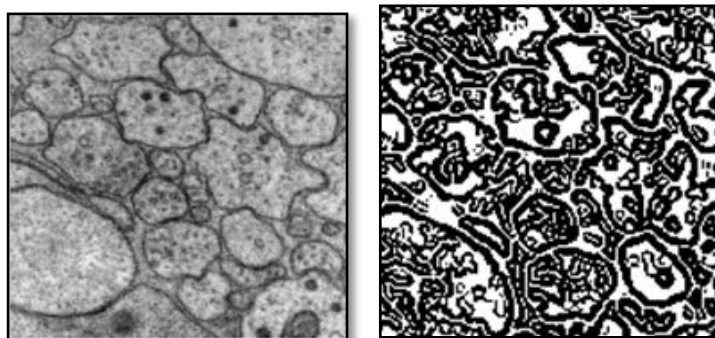
### b) Thresholding with Gradient Magnitude



Original Image

Thresholding with Gradient Magnitude

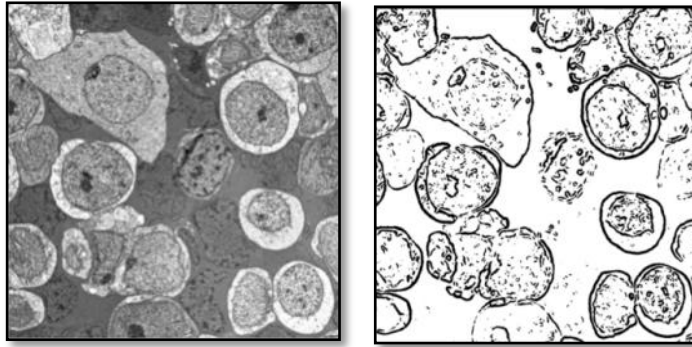
*Drosophila* dataset



Original Image

Thresholding with Gradient Magnitude

*C.Elegans* dataset



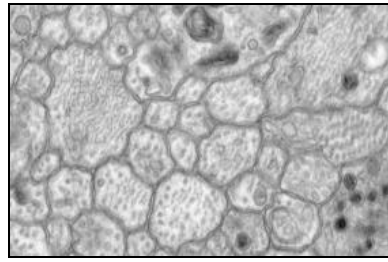
Original Image      Thresholding with Gradient Magnitude

Rabbit Retina dataset

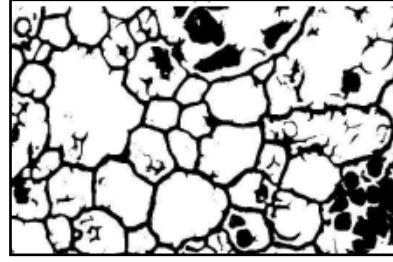
Figure 4.4: Output obtained using thresholding with gradient magnitude for three different datasets.

### 4.3.3 Diffusion

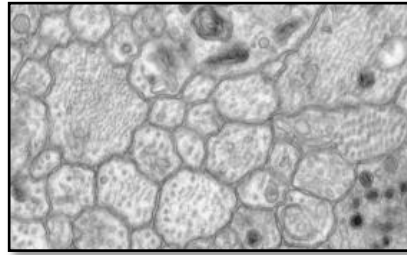
The basic Partial Differential Equation (PDE) method is suitable for denoising images composed of homogeneous intensity regions (Jones, 2005), but is unsuitable for filtering Transmission Electron Microscopy (TEM) images and textured images. Modification of PDE using Weickert's coherence enhances diffusion by replacing the information from the structure tensor by information from Hessian Based Diffusion (Jones, 2005). In spite of this modification, the method still does not fulfil the goal of organelle elimination.



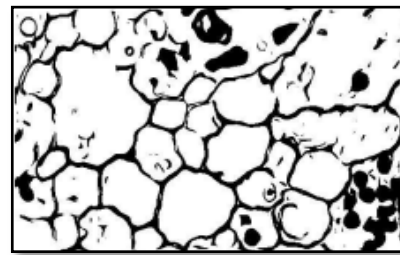
TEM Image of Rabbit Retina



Weickert's PDE



TEM Image of Rabbit Retina



Hessian Based Diffusion

Figure 4.5: Comparison of diffusion-based approaches.

However, when tested with LCHF many dark black spots (organelles) are removed from the dataset.

#### 4.3.4 Graph cuts

Diffusion and the other techniques described above are inadequate mainly because of the presence of intracellular structures. Researchers such as Ciresan, Giusti, Gambardella, & Schmidhuber (2012) suggest that 2D graph cuts can be used to segment images and separate intracellular structures from membrane. However, when graph cuts alone are used specific energy minimisation functions dependent on the type of cell under consideration are required. Further, they are highly dependent on adequate initialisation. The error in the initialisation can lead to confusion in segmentation. The aim and goal of this research is to develop an efficient algorithm that is easily adaptable with existing hardware. This simplicity can be found in LCHF.

## **4.4 First stage of the experiments**

In the first stage of the experiments, many different functions were utilised in a trial-and-error manner. Consequently, upon discovery of successful functions, the chosen functions underwent exhaustive parameter tuning to find the best fit parameter for the used dataset. This stage is considered manual because there is need for human intervention in parameter tuning and function selection. Because this is the preliminary stage of the research and experiments, much of the work carried out here was on a trial-and-error basis.

### **4.4.1 Explanation of Functions Selected and Tuned for LCHF**

#### **a) Image denoising**

As with Laptev, Vezhnevets, Dwivedi, & Buhmann (2012), the *Drosophila* dataset provided by the Cardona *et al.* were highly anisotropic. This occurred as a result of the visualisation technique of ssTEM images which results in highly anisotropic volumes of images. TEM is a currently popularly available microscopy technique that can provide sufficient resolution for medical images. The technique depicts the observed volume as a stack of images and, in 3D viewing, the images are viewed as x, y, and z. In essence, x and y have high resolution, whereas the z direction information (pixels) cannot be viewed clearly, and has a poor resolution. Thus, the next step needed is to use a denoising method to improve these highly anisotropic images.

From experiments conducted, we discovered that to enhance the image quality (anisotropic filtering), combinations of denoising and contrast enhancement provide better results for image pixels with this issue because of its oblique viewing angles.

Two sets of experiments were conducted:

1) In the preliminary set of experiments, the processing pipelines were fixed to three functions: denoising, thresholding, and hole-filling. Result: With this particular sequence, the Wiener filter is the best denoising method, giving a resulting F1 score of 0.6592 and Median filter with F1 score of 0.6569 (as shown in Table 1).

2) Next, the sequence of steps was expanded by incorporating a local contrast enhancement function, in which the processing pipelines were set to four functions: denoising, contrast enhancement, thresholding, and hole-filling.

Result: F1 scores of 0.7107 from using the median filter, which is better than that resulting from using the Wiener filter (*i.e.*, 0.7091). Because of this advantage the median filter was used in the final configuration of the LCHF algorithm.

Table 4.1: Measurement values for denoising filter for three experimental functions (denoising, thresholding, and hole-filling).

| <b>Measures</b>          | <b>Median</b> | <b>Gaussian</b> | <b>Wiener</b> | <b>Average</b> | <b>Laplacian</b> |
|--------------------------|---------------|-----------------|---------------|----------------|------------------|
| <b>Average F1</b>        | 0.6569        | 0.6501          | 0.6592        | 0.6503         | 0.3588           |
| <b>Average Precision</b> | 0.6265        | 0.6333          | 0.6367        | 0.6324         | 0.2194           |
| <b>Average Recall</b>    | 0.7281        | 0.7092          | 0.7232        | 0.7073         | 0.9925           |

Figure 4.6 shows the results of applying different types of denoising filters. The two best denoising methods (as per F1 score), median filter (b) and Wiener filter (c), are shown.

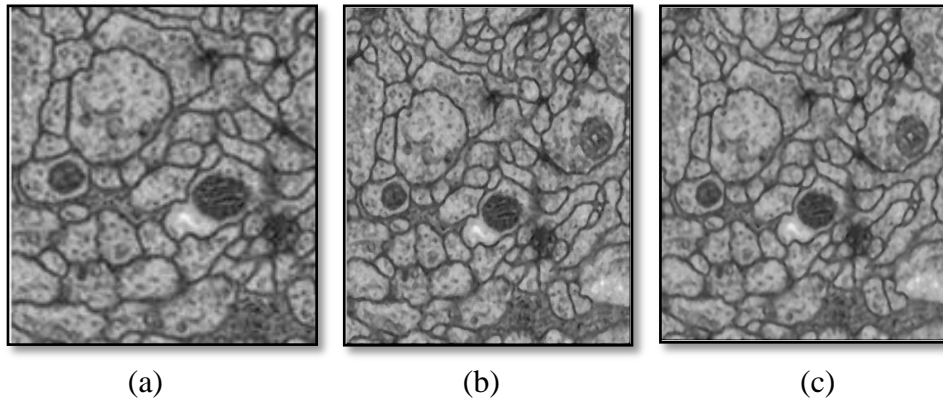


Figure 4.6: (a) Original noisy image, (b) result of median denoising, (c) result of denoising with Wiener filter.

Figure 4.6(a) shows the original *Drosophilalarvae* image before application of denoising filters. The original image dataset contains noise and the image is anisotropic, as explained above. To improve the quality of the image, the denoising technique is applied, which actually improved the quality of the image. The middle image (b) in Figure 4.6 is the resulting image obtained using median denoising, and 4.6(c) is the resulting image obtained using the Wiener denoising techniques.

### **b) Contrast Enhancement**

The algorithm operates on small regions in the image, called tiles, rather than the entire image and enhances the contrast of each (Prathibha & Sadasivam, 2012). The histogram of the output is matched with a specified histogram shape of the set parameter. Then, the neighbouring tiles are combined using an extension of linear interpolation, known as bilinear interpolation, used to interpolate the functions of two variables on a 2D grid.

Algorithm Steps:

- a) Input the image (number of regions in row and columns)
- b) Pre-process the input image: Determine the type of parameters to be used. Pad the image if necessary.
- c) Process the image on a tiles basis. A single image region is extracted, and using specified bins, the histogram is generated.

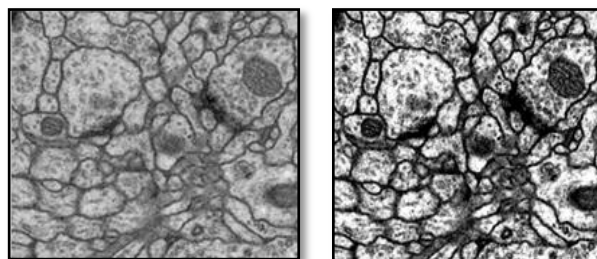
d) Next, the grey-level mappings are interpolated to assemble the final output image after contrast enhancement using Contrast Limited Adaptive Histogram Equalisation (CLAHE). Clusters are then extracted for the four neighbouring mappings.

e) The results are interpolated to obtain the output pixel. This step is repeated for the entire image.

In general, various contrast enhancement functions available in MATLAB were tested. Three main functions—Histogram Equalisation (Histeq), Intensity Adjustment (Imadjust), and CLAHE—were used on the *Drosophila* dataset.

In MATLAB(MatLab, 2012):

a) Histeq performs histogram equalisation by transforming the intensity values of images to match the histogram of the output image by a specified histogram, which helps to enhance the image contrast. Histeq was tested with various discrete grey levels.



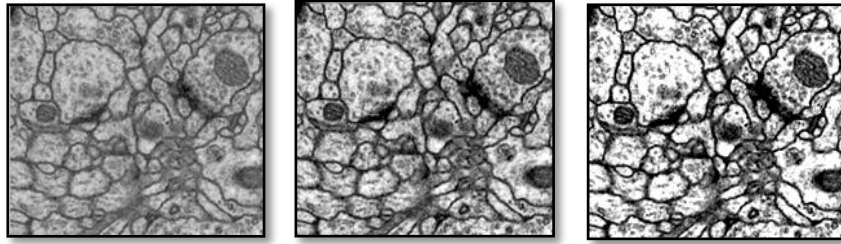
Original Image before  
Contrast Enhancement

Image after Contrast  
Enhancement

Figure 4.7: Original image before contrast enhancement and image after contrast enhancement (Histogram Equalisation).

b) Imadjust increases the contrast of the input image by mapping the input intensity image values to the new counted values, which have a range of low and high values. Imadjust was tested with various image intensity values and the values then mapped in such a manner that 1% of the data were saturated at low and high.

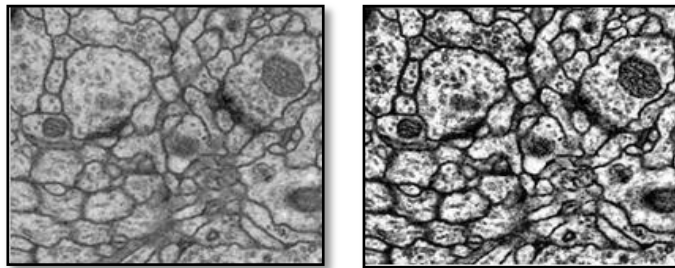




Original            Using default value    Prescribed high and low contrast limit

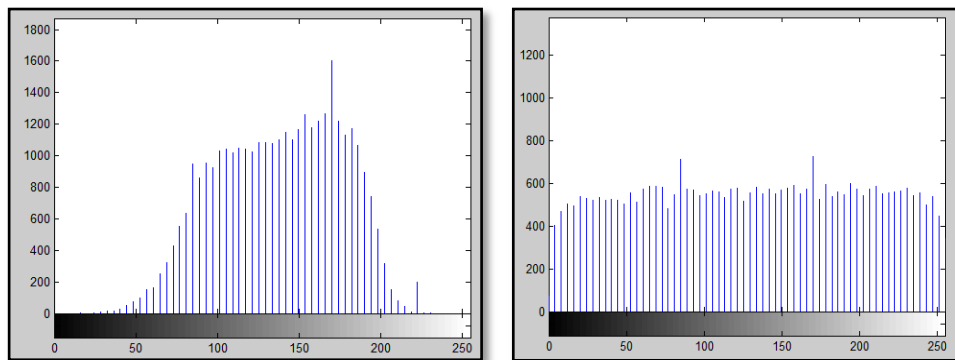
Figure 4.8: Output obtained using default and prescribed high and low values for contrast limit.

c)     Adaptive histogram equalisation concentrates on smaller tiles (region) than the entire image.



Original Image before Contrast Enhancement (CLAHE)            Image after Contrast Enhancement

Figure 4.9: Original image before application of CLAHE and after application.



Histogram of the Original Image vs. Histogram of the Processed Image (for above output)  
(Examples using 64 bins—the default value)

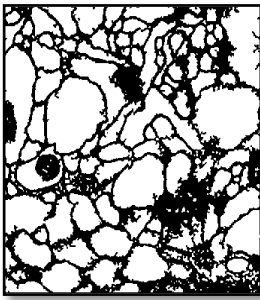

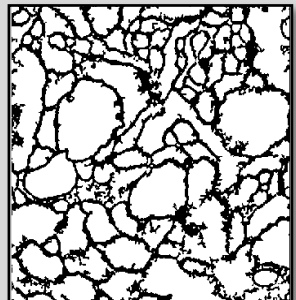
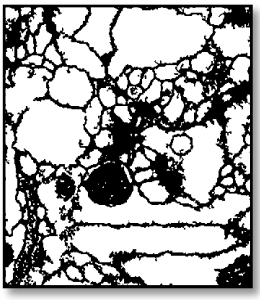
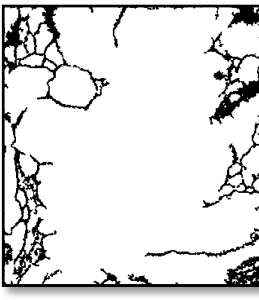
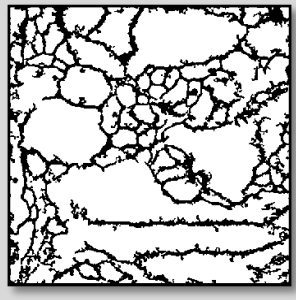
Figure 4.10: Histogram of original image vs. histogram of the processed image using 64 bins.

Various parameters, such as Number of Tiles in Image (NumTiles), Contrast Enhancement Limit (ClipLimit), and Number of Bins for Histogram Used (NBins), with various output ranges and specific histogram shapes were used for CLAHE.

- a. NumTiles [M N]: Represents the tile rows and columns, default [8 8].
- b. ClipLimit: Limits contrast enhancement. Values from zero to one, with higher values giving more contrast.
- c. NBins: Sets number of bins for contrast transformation; higher values result in slower speed, default: 256.

The Table 4.2 below shows the results of experiments conducted using the three methods: Histogram Equalisation (Histeq), Intensity Adjustment (Imadjust), and CLAHE. They were used for examples of randomly picked slices from the *Drosophila* dataset.

Table 4.2: Randomly picked slices from the 30 stacks dataset.

| Img      | Enhance contrast using Histeq   | Global using | Enhance contrast using Imadjust  | Global using | Using Contrast Enhancement Method -CLAHE  | Local |
|----------|---|--------------|--|--------------|---|-------|
| Slice 7  |  |              |  |              |  |       |
| Slice 22 |  |              |  |              |  |       |

From the images, it can clearly be seen that organelles are still being falsely detected with the histogram equalisation method and that the membranes are erroneously eliminated when the image intensity values are adjusted using *Imadjust*, in both global contrast enhancement conditions. When a local contrast enhancement is adopted with CLAHE, there is no major elimination of membranes and no significant false detection of organelles. Because of this, CLAHE was chosen as the contrast enhancement algorithm for the algorithm. Table 4.3 shows the average performance values for both global and local contrast enhancement methods.

Table 4.3: Average performance values after using different contrast enhancement techniques.

| Measures              | Using Global Contrast –Histogram Equalization after De-noising | Using MatLab’s <i>imadjust</i> after de-noising | Using Local Contrast-CLAHE after De-noising |
|-----------------------|--|---|---|
| Average F1            | 0.6778   | 0.6861  | 0.7107                                      |
| Average Precision     | 0.5515   | 0.6301  | 0.6429                                      |
| Average Recall        | 0.8838   | 0.7660  | 0.7974                                      |
| Elapsed Time (second) | 14.6197  | 15.0907   | 21.0894                                     |

**i) The CLAHE Parameter of Choice - NumTiles**

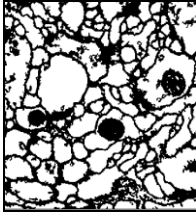


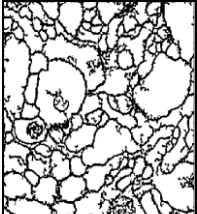
The tiles [25 25] were used for the algorithm. This type of tiles was used for the following reasons:

- From the findings, the image after denoising and execution of the CLAHE function gives a better line (less jutting out).
- Using less tiles [5 5] affects detection of organelles when hole-filling is applied.
- Using more tiles [200 200] results in too many jutting lines, with no clear lines. Further, when incorporating the thresholding function with the contrast function, if more tiles and low threshold values are used, a blank white picture results (all lines are cleared).

- Using tiles [100 100], the result is acceptable, but was time-consuming, for one slice approximately 13 seconds, compared to one second for tiles using [25 25]. Further, it does not meet the objective of the research for creation of high speed algorithm.

The Table 4.4 shows an example output obtained using various NumTilesparameter values.

Table 4.4: Showing the F1 result using various NumTiles parameter values.

| Num Tiles    | [5 5]  | [25 25]  | [100 100] using the same threshold as [5 5] and [25 25]                             | [100 100] using different threshold  |
|--------------|--|--|---|--|
| Output Image |  |  |  |  |
|              | F1 score - 0.6635<br>Elapsed Time : 20.675   | F1 score - 0.7107<br>Elapsed Time : 21.089   | F1 score - 0.4010<br>Elapsed Time : 428.413   | F1 score - 0.6425<br>Elapsed Time : 441.711  |

## ii) Experiments Using Various CLAHE parameters

Table4.5: Explanation and F1 results for various CLAHE parameters.

| Parameter              | Explanation  |
|------------------------|--|
| Cliplimit              | By specifying a lower contrast number ( <i>e.g.</i> , 0.2), fewer jutting lines occur and organelles are detected. Higher contrast ( <i>e.g.</i> , 0.9) results in more jutting lines, because a higher number results in more contrast.<br>The highest F1 score recorded for a cliplimit of 0.2 was 0.6215.<br>The highest F1 score recorded for a cliplimit of 0.9 was 0.5995. |
| Histogram Bins (NBins) | Higher values result in greater dynamic range at the cost of slower processing speed.<br><br>In the latter part of the algorithm, using Lower Bins, results in fewer jutting lines, but when run with the algorithm, the F1 score was lower than that using NumTiles [25 25].<br><br>The highest F1 score recorded was 0.6648.   |

For all the parameters, the F1 scores for the experimental results (Table 4.3) show that thus far (at this stage) only NumTiles with [25 25] results in the best F1 scores on the *Drosophila* dataset. The Figure 4.11 below shows the different values for NumTiles and their outputs.

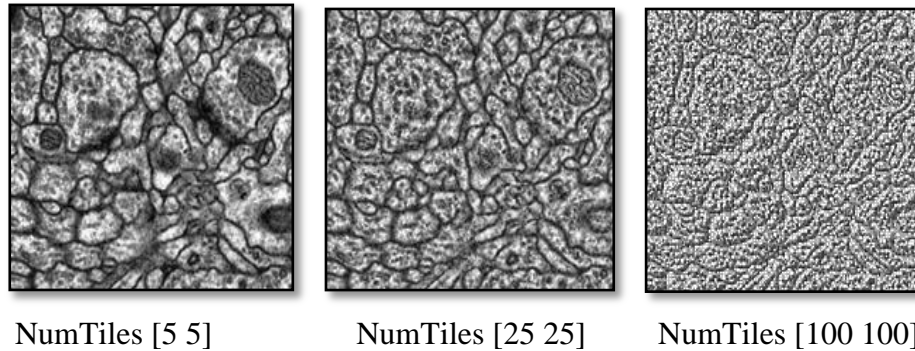


Figure 4.11: Values for NumTiles and their corresponding output.

Using parameters that result in fewer jutting lines aids detection of both membrane and organelles. When organelles are not detected, the lower F1 scores are lower. Thus, for the algorithm, we decided to choose the parameter that contributes higher F1 scores. The jutting line issue is considered and solved in the next stage of the algorithm (enhanced advance stage).

### c) Thresholding Functions

We used the Global image threshold by adopting Otsu's method, in which the threshold is chosen to minimise the variance of black and white pixels. It is a nonparametric and unsupervised method for automatic threshold selection in image segmentation that operates on grey-level histogram (256 bins) (Taghadomi *et al.*, 2014). The underlying idea is to find a threshold that can minimise the weighted within-class variance or maximise the between-class variance. In short, it is used to extract an object from its background by assigning a thresholding intensity value for each pixel in the image, so that the pixel is classified either to object or background of the image (Pandey, Gamit, & Naik, 2014).

For multilevel thresholding, the greyscale image needs to be converted to an indexed image,  $Z$ , using the intensity image and the value of  $V$ , the vector of values between zero and one. The thresholded image with a colourmap (for illustration purposes) is shown below.

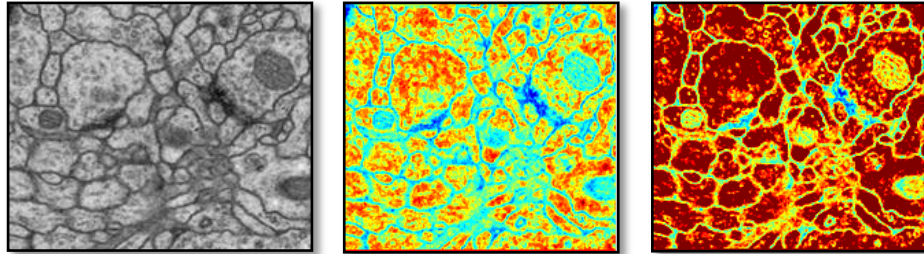
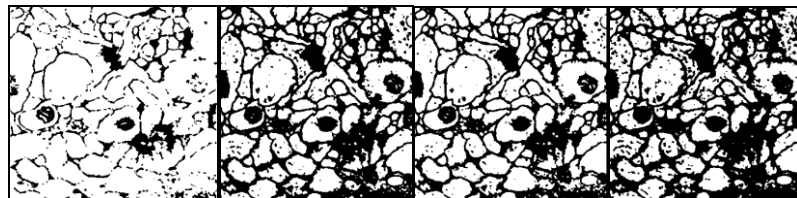


Figure 4.12: Original Image vs. Thresholded Image ( $v = 16, v = 25$ ) with colourmap.

In the experiments, an exhaustive search was conducted to find the best threshold for the *Drosophila* dataset. The Figure 4.13 below shows the effects and results obtained using various threshold values.

#### **Effects of various Threshold values on *Drosophila* dataset**



Threshold= 80    Threshold = 100    Threshold = 104 (best)    Threshold = 120

Figure 4.13: Effect of using various threshold values on *Drosophila* dataset.

Figure 4.13 shows how various threshold values affect membrane and organelle detection. On the basis of an exhaustive search procedure (using F1 scores) it was found that a threshold of 104 was the best choice for the *Drosophila* dataset.

#### d) Hole-Filling Functions

The algorithm is based on morphological reconstruction. It fills the holes in the greyscale image; here, hole is defined as an area of dark pixels surrounded by lighter pixels. It filled the objects that had a complete unbroken outline. Figure 4.14 shows examples of hole-filling function performance for both greyscale and binary images.

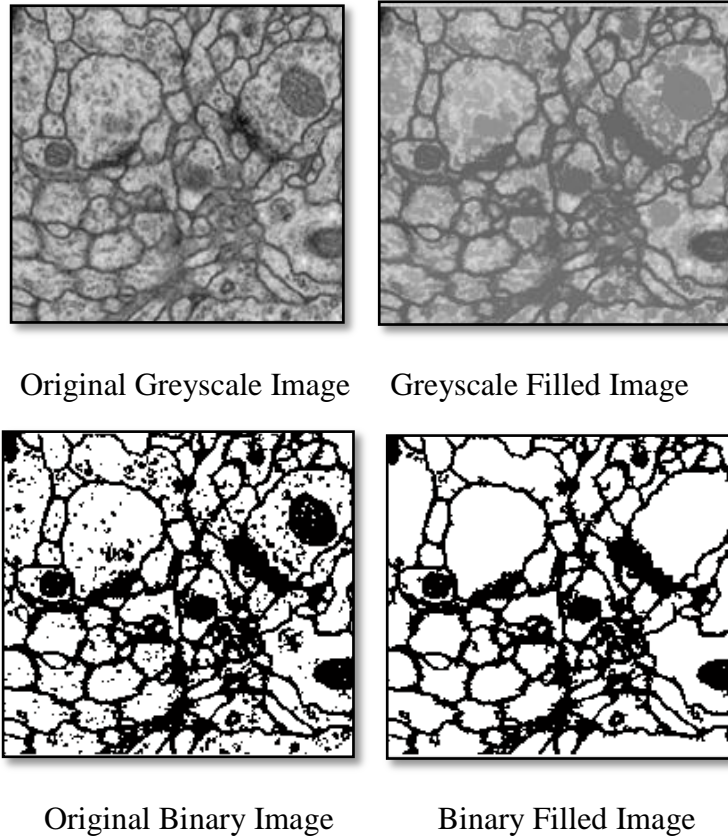


Figure 4.14: Hole Filling for Greyscale and Binary Image.

Although the hole-filling function can perform organelle elimination, from experiments conducted (as per Figure 4.15 below), it was found that the hole-filling technique did not remove all detected organelles for the *Drosophila* dataset. Figure 4.15 shows the appearance of the organelle in black patches. If adequate preprocessing is implemented, with combination of other segmentation functions, then the hole-filling technique could perform give a better output. This was proven from experiments conducted in this research; combining the operations of multiple functions results in the elimination of organelles Table 4.6 shows the performance measures for the hole-filling

technique in collaboration with the denoising and thresholding technique. The average F1 score in this combination is 65.69%.

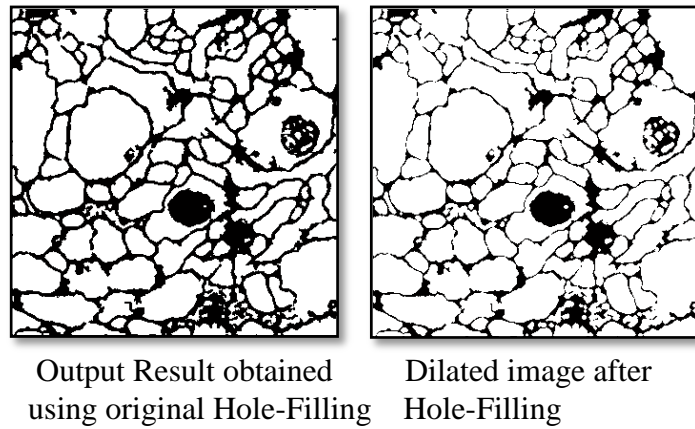


Figure 4.15: Effects of Hole-filling.

- **Performance Measures for Basic hole-filling technique results**

Table 4.6: Measurement values for basic Hole-Filling.

| Measures              | Denoising + Thresholding + Hole-Filling |
|-----------------------|---|
| Average F1            | 0.6569                                  |
| Average Precision     | 0.6265                                  |
| Average Recall        | 0.7281                                  |
| Elapsed Time (second) | 14.4165                                 |

### e) Morphological Operators

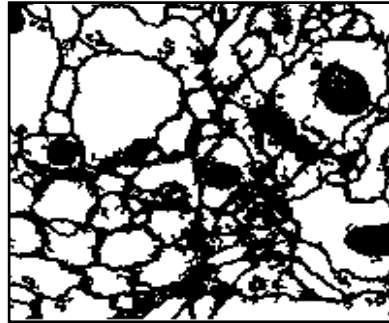
The morphological operation counts the value of the corresponding pixel and compares with the neighbouring pixels of the original (input) image.

### f) Dilation

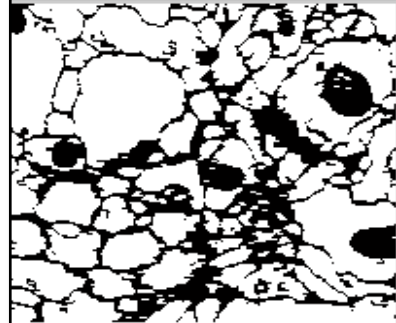
In the early stage of the experiments, this step was considered an optional step and was only conducted for better visual inspection. However, in later stages of the research, this step played a larger role and does lead to significant improvements.



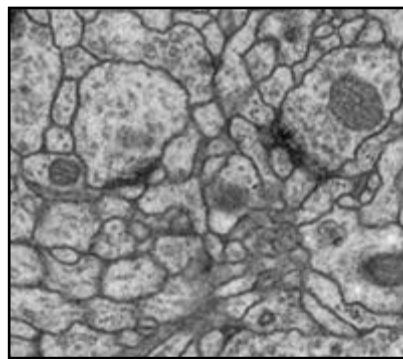
If the image is logical and the structuring element has a flat value, then the dilation operator performs binary dilation; otherwise, it performs greyscale dilation.



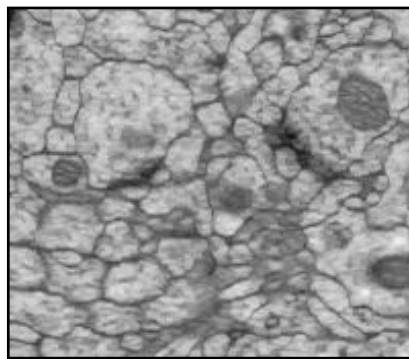
Original Binary Image



Dilation using  
Flat Structuring Element



Original Greyscale Image

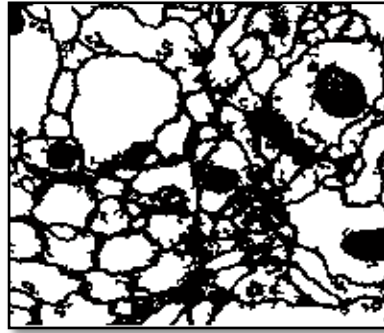


Dilation using  
Flat Structuring Element

Figure 4.16: Effects of Dilation.

### **g) Eroding**

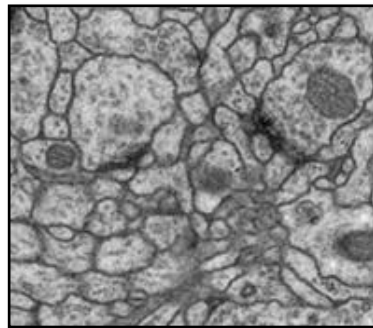
Like dilation, eroding is a morphological operator. However, it performs opposite function to dilation. Eroding reduces the boundary of the region so it shrinks in size. It works both on binary and greyscale images.



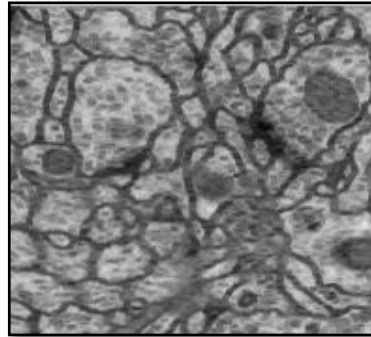
Original Binary Image



Erosion using  
Flat Structuring Element



Original Greyscale Image



Erosion using  
Flat Structuring Element

Figure 4.17: Effects of Eroding.

#### 4.5 Local Contrast Hole-Filling Algorithm (LCHF)

Using the ISBI (*Drosophila*) dataset, experiments were carried out and the findings recorded. In the initial stage, using standard segmentation algorithms, as described above, in the output presented, there appeared to be compromises between cell membrane and organelle detection. The more organelles that were ignored, the more cell membranes that were poorly detected. Consequently, the next step was used to rectify the problem, such that the outcome detected cell membranes and ignored organelles. Experiments were carried out using various processing and classification steps in segmentation, such as Contrast Adjustment, Denoising, Thresholding, Watershed, Eroding and Dilation. Many experiments were conducted to test the sensitivity and consistency of the

functions used. Precision, Recall, and F1 score were used as error measure metrics. The best approaches were chosen via the F1 score. From the experiments, optimal parameters were estimated for the used functions.

#### **4.5.1 LCHF Outcome**

In this research, we proposed a simple, efficient, non-learning approach based on several basic processing steps, including LCHF. LCHF was found to be capable of efficiently detecting membranes in the TEM Drosophila dataset (downloaded from IEEE International Symposium on Biomedical Imaging) with an average F1 score of 71% and an average processing time of 21 seconds for 30 slices (Raju, R., Maul, T., & Bargiela, A., 2014).

As this area of research has existed for many years, a number of algorithms have been developed, with most of them developed solely depending on ground truth to train the dataset, to learn, and to output results. In such cases, if no ground truth is available, then the algorithm is not usable because the corresponding dataset cannot be trained. An expert in the needed area needs to engage in the preparation of the ground-truth image manually, and this engagement does cost in terms of time and money. In some cases, it can be achieved semi-automatically; that is, by running some existing standard segmentation algorithm, but still needs human involvement to correct the wrongly labelled pixels. LCHF is a non-learning approach that does not depend on ground truth to perform its output. In this research, at this stage, the ground-truth data were used only for error measurement metric, to compare the output and measure it in comparison with the ground truth.

#### 4.5.2 Summary of functions and associated parameters

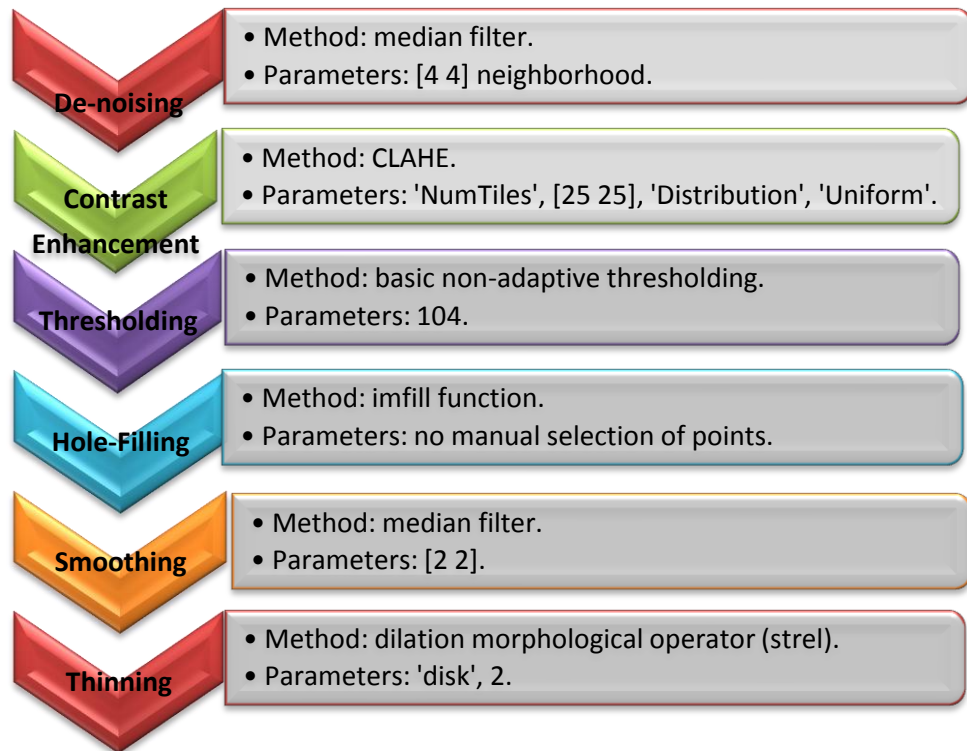


Figure 4.18 : Summary of LCHF Functions.

As explained above, many parameters are supplied to their corresponding functions via the error metrics measurement, with the most optimal parameter for each function chosen. This function is displayed in Figure 4.18, above.

#### 4.5.3 Comparison of LCHF algorithm with simple segmentation methods

##### a) Comparison with Edge Detection method

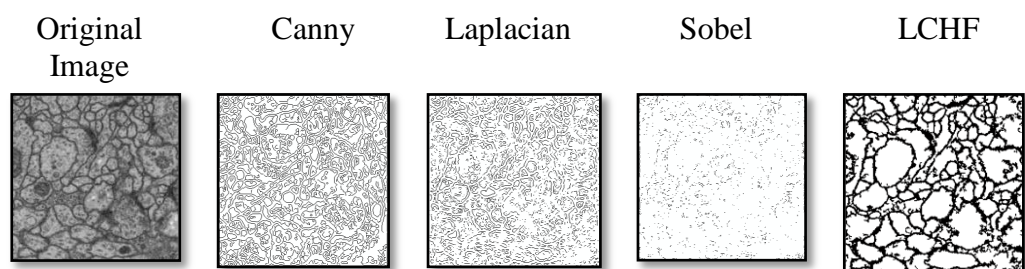
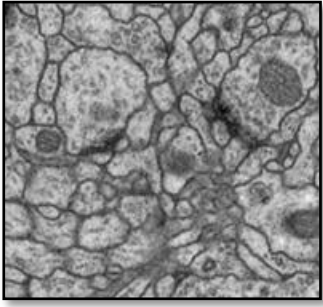
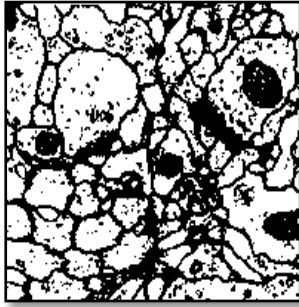
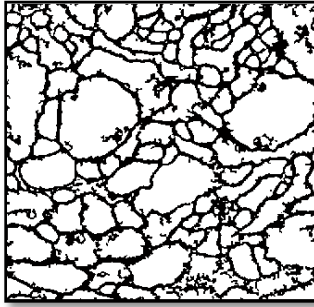
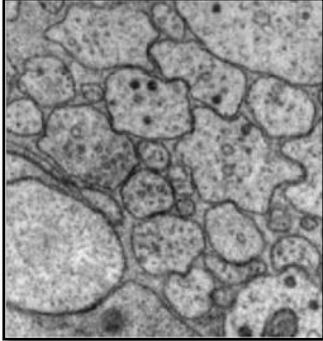

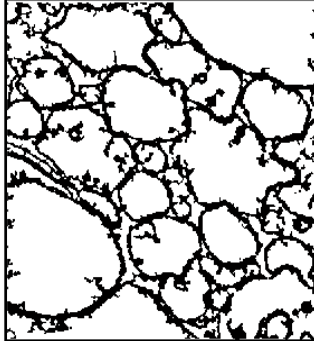
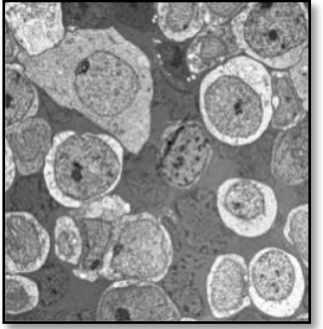

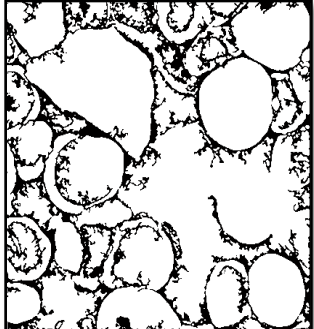


Figure 4.19: Comparison using LCHF with the other edge detection method.

Figure 4.19: Compares the output obtained using the LCHF method to those obtained using other edge detection methods (introduced earlier in this chapter). The output obtained using LCHF shows that it can clearly detect membranes and eliminate organelles better than the three edge detection methods compared (Canny, Laplacian, and Sobel).

**b) Comparison of Simple Thresholding with enhanced features**

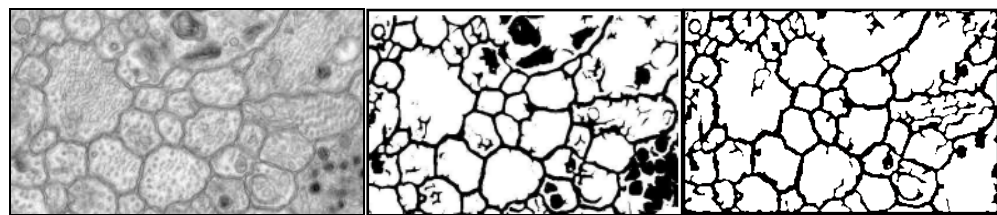
Table 4.7: Images and corresponding output obtained using the LCHF algorithm.

| Original Image  | Simple Thresholding  | LCHF  |
|---|--|---|
|   |   |   |
|  |  |  |
|  |  |  |

Figures 4.3 and 4.4 show the output results obtained using thresholding with AS and gradient magnitude. Both methods highlight the membrane boundaries (Figures 4.3 and 4.4) but fail to remove internal structures. In contrast, LCHF (Table 4.7) not only highlights the membrane boundaries, but also removes internal structures relatively successfully.

Figure 4.20 shows the performance of LCHF on Rabbit Retina TEM images. The performance of LCHF is better than Weickert's PDE, and Hessian Based Diffusion. It detects fewer organelles than these two methods. Not many black spots are detected using the LCHF method. Further, took less than four seconds to output results.

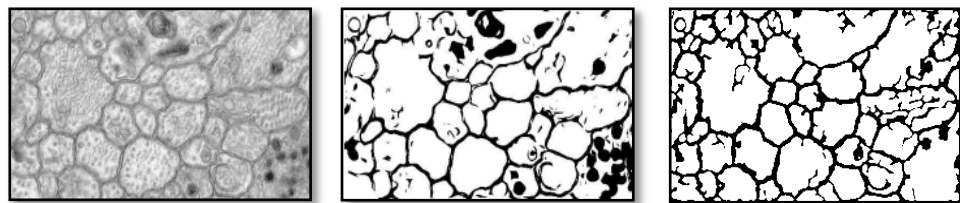
**c) Comparison with Weickert's PDE and Hessian Based Diffusion**



TEM Image of Rabbit Retina

Weickert's PDE

LCHF



TEM Image of Rabbit Retina

Hessian Based Diffusion

LCHF

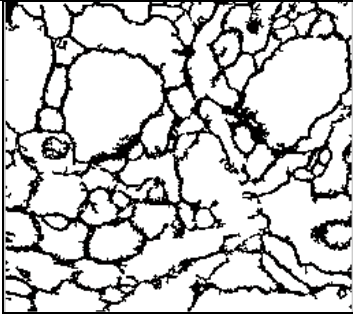

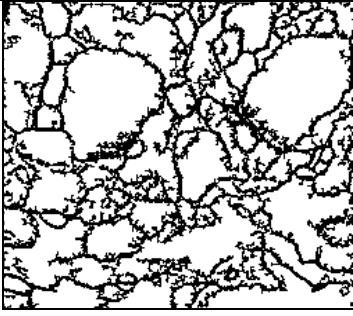
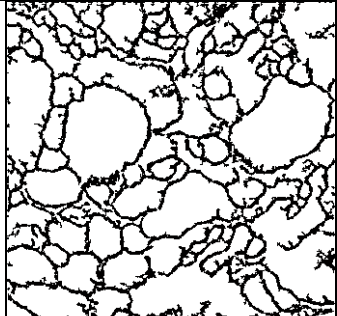
Figure 4.20: Comparison of diffusion-based approaches to LCHF.

**d) Comparison with and without Denoising function**

The Table 4.8 shows the result of the algorithm with and without application of the Denoising function. As can be seen, when the denoising function is not integrated, the output, shown in column 1, of the LCHF without denoising

appears to still comprise detected organelles, whereas the output in row 1 of LCHF with denoising does not show any organelles. Furthermore, the output in row 2 of LCHF without denoising show many ‘jutting lines’ (present in the detected noise), whereas row 2 of LCHF with denoising is much cleaner.

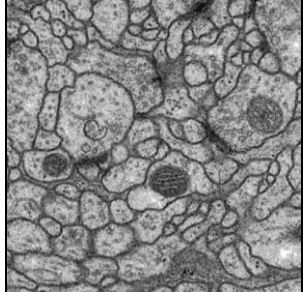
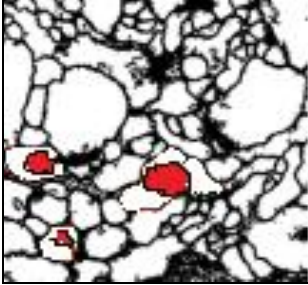
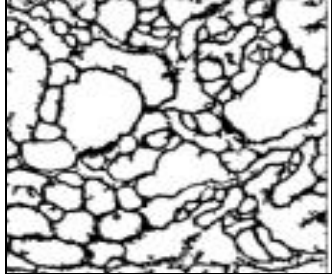
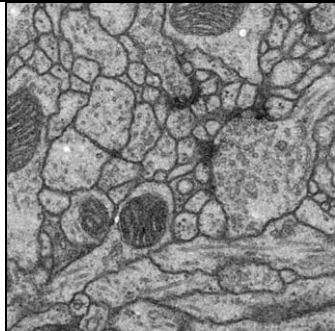
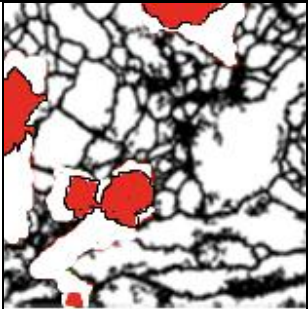
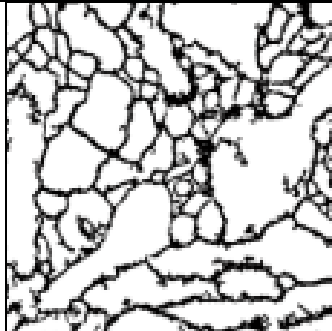
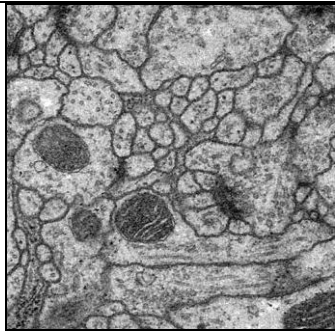
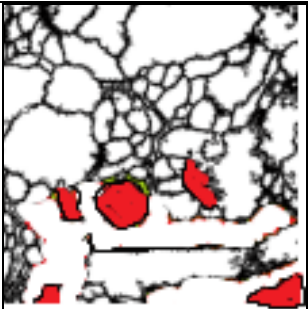
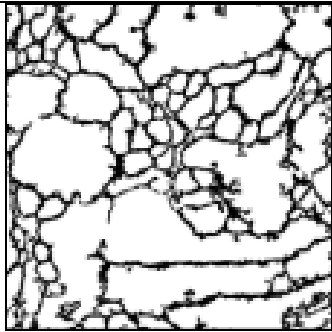
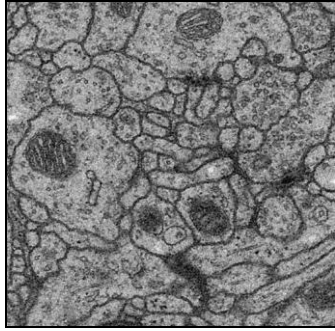
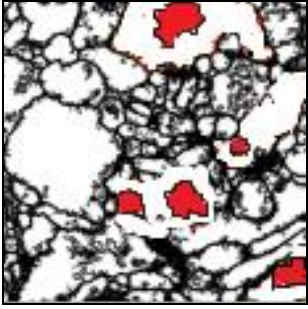
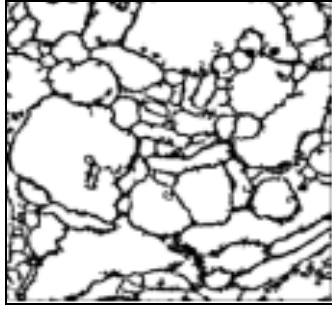
Table 4.8: LCHF with and without the denoising function.

| No. | LCHF without Denoising   | LCHF with Denoising   |
|-----|--|---|
| 1   |   |   |
| 2   |  |  |

**e) Comparison with output of Hole-Filling**

The experimental results obtained using thresholding and the hole-filling method are compared with those obtained using the LCHF algorithm in the Table 4.9. The results clearly show that the LCHF algorithm detects membranes and eliminates organelles more successfully than thresholding and hole-filling only—which erroneously detected organelles, coloured in red in the middle figure. Further, for data slice in Table 4.9, a much better result is shown in terms of F1 scores in Table 4.11.

Table 4.9: Results of experiments conducted using thresholding with hole-filling only and using LCHF.

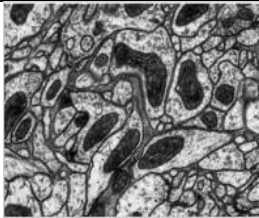
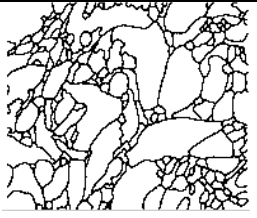

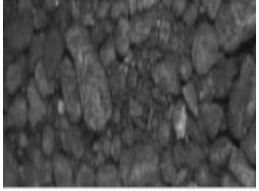
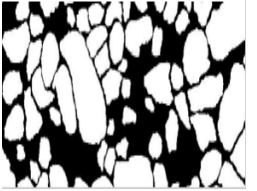

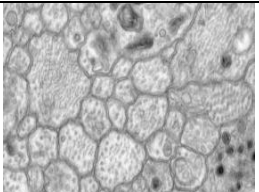
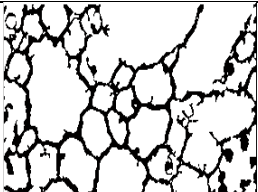
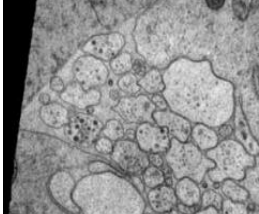

| Original Image  | Experimental Results Using Thresholding + Hole-Filling Method Only                   | Experimental Results Using LCHF   |
|---|--|---|
|    |    |    |
|   |   |   |
|  |  |  |
|  |  |  |



#### 4.6 Experiments with other datasets

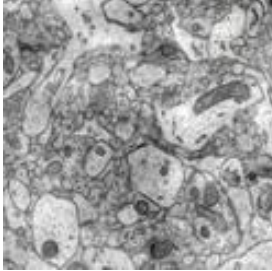


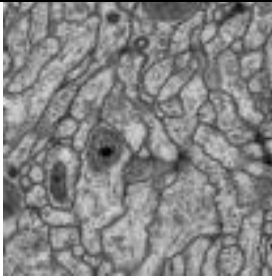

Several experiments were also carried out to test the LCHF algorithm with other datasets, including medical and non-medical images. The results of the experiments and their corresponding ground-truth images are given below. These results show that LCHF is versatile and generalisable. Note that only minimal parameter tuning was required in terms of threshold values and CLAHE parameters.

Table 4.10: Experimental results obtained using other datasets (published in a conference paper presented at IEEE Symposium).

| Dataset                                     | Original Image  | Ground-Truth Image   | Output Using LCHF   |
|---|---|--|---|
| Lamina and medulla neuropiles of optic lobe |   |   |   |
| Mineral ore                                 |  |  |  |
| TEM image of Rabbit Retina                  |  | Not available  |  |
| C.Elegans                                   |  | Not available  |  |

Continue...

...continued

|                                       |   |  |   |
|---------------------------------------|---|--|---|
| EM image of mouse cortical Neurons    |  |  |  |
| <i>Drosophila</i> Test Image Slice 16 |  | Not available (the test set ground-truth dataset are not released online)          |  |

#### 4.7 Strength of the Proposed LCHF Algorithm

**What exactly does the LCHF algorithm contribute to the overall research?**

One of the main advantages of LCHF is that training (*i.e.*, fine-tuning) is very rapid, it does not require much training (except for some parameter tuning), which translates into ease and efficiency of use, whereas most algorithms applied in the ISBI challenge required extensive training. It is a non-learning approach with a small sequence of processing steps, each with a small set of parameters, which are not time-consuming to fine-tune for different types of datasets. LCHF can also be considered fast at pixel classification, where the task of detecting membranes in 30 TEM images (each with a resolution of  $343 \times 343$  pixels) can be done in approximately 21 seconds on an average personal computer (*i.e.*, 1.60 GHz processor with 1.48 GB of RAM).

Another LCHF advantage is that it does not require specialised hardware, in contrast to the approach adopted by the winner of the ISBI 2012 challenge. Based on the hardware constraints of today, training classifiers with a very large number of free parameters (*e.g.*, deep neural networks) can require weeks

of computation, even when using high performance machines with high data transfer rates. This involves significant monetary and energy costs.

Next, LCHF not only saves time, but is also very easy to use and deploy. With little effort, it was possible to get reasonable results. Although the best F1 score thus far is approximately 71%, the algorithm does indeed do a reasonably good job at distinguishing membranes and organelles, thus satisfying the original goal.

Moreover, LCHF is easy to adopt by researchers with limited experience of computer vision, machine learning, and even programming. The algorithm involves only a few simple programming steps with basic functions which can typically be found in standard image processing libraries such the MATLAB Image Processing Toolbox (by MathWorks). LCHF has also been shown to be effective with other types of datasets.

#### **4.8 Weakness of LCHF**

**(\*solved in the next algorithm in this research)**

LCHF is simple, efficient, usable, and can effectively distinguish membranes and organelles. However, in comparison with the competitor's algorithm, the overall accuracy of the algorithm has not yet reached state-of-the-art levels. The recorded highest score is 94% using the test dataset. One particular artefact of LCHF being addressed in the next step of the enhancement of the research with the new proposed algorithm is the presence of 'squiggly lines' jutting out from the membrane, as can be seen, for example, in Figure 4.21.

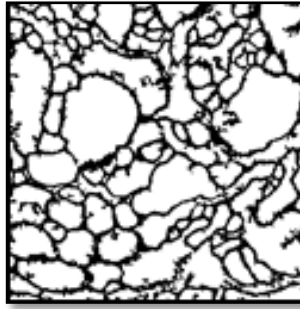


Figure 4.21: Output result obtained using LCHF (with squiggly lines).

Figure 4.22 (below) shows a randomly picked image from a stack of 30 images, with its corresponding output using the LCHF method. The Figure 4.22 shows detected cell membranes (in black), eliminated organelles, and its corresponding ground truth. The method is simple and can easily be adopted by beginners in the field of medical imaging.

The aim of the research in creating this algorithm is to detect membranes and ignore organelles with minimal effort and less processing time with minimal loss of undetected membranes.

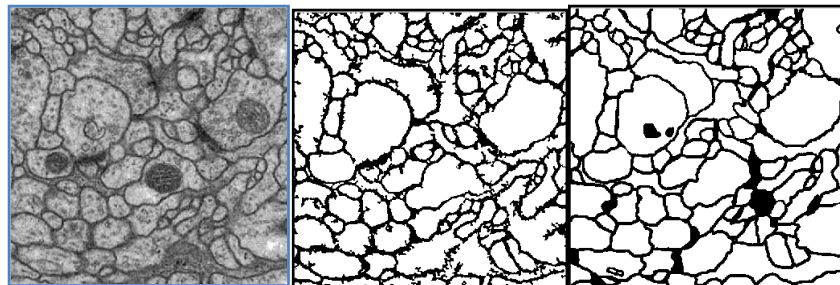


Figure 4.22: Microscope image (left) with the corresponding LCHF result (middle), and corresponding Ground truth (right).

Table 4.11: Measured values of final LCHF method as per the above visual output.

| Average Scores | F1 | Average Precision Scores | Average Scores | Recall | Elapsed Time (seconds) |
|----------------|----|--------------------------|----------------|--------|------------------------|
| <b>0.7107</b>  |    | <b>0.6429</b>            | <b>0.7974</b>  |        | <b>21.0894</b>         |

## 4.9 Conclusion

The overall message of the LCHF algorithm is the following:

‘Even a very simple algorithm consisting of a short sequence of basic processing steps can be relatively competitive’.

The LCHF algorithm is non-learning, simple, easily adopted, and can recognise membranes and eliminate organelles using a very simple algorithm that consists of short sequences of basic processing steps, yet it is relatively competitive. In simple tests over various datasets (medical and non-medical) it helped to segment the dataset in a meaningful way. It generously classified the pixels into membrane/non-membrane for medical images, highlighted membrane boundaries, and also removed internal structures. In non-medical images, it highlighted the outer line of the objects in the image. It took a matter of seconds to produce the comparable result.

LCHF achieved the set goal and indeed did a reasonably good job at distinguishing membranes and organelles. However, in terms of accuracy, the highest recorded accuracy was 71% (Average F1 Score). This accuracy issue is solved in the next proposed algorithm which recorded more than 91% individual F1 score. Moreover, the next proposed algorithm, which is a logical extension of the initial work in LCHF also addresses the presence of ‘squiggly lines’ jutting out from the membranes. Furthermore, it is based on the knowledge that image processing pipelines are useful for the membrane detection problem (as demonstrated in this chapter), and the assumption that the space of pipelines is in fact too large to search manually, and therefore requires an optimisation procedure. The next proposed algorithm, called the IPCO algorithm, begins at this point.

\*A paper titled ‘Local Contrast Hole-Filling Algorithm For Neural Slices Membrane Detection’ was presented in the 2014 IEEE Symposium on Computer Applications & Industrial Electronics covering this stage. The symposium is fully sponsored by the IEEE Malaysia Section and IEEE Industrial Electronics and Industrial Applications of Malaysia Joint Chapter. It is currently indexed in IEEE explorer, and its print ISBN No. is 978-1-4799-4352-4.

## CHAPTER 5

### RESULTS

#### IMAGE PROCESSING CHAIN OPTIMISATION (ICPO) ALGORITHM

As mentioned earlier in Chapter 4, the research contributes three algorithms. This chapter discusses the second algorithm called the Image Processing Chain Optimisation (IPCO) algorithm. The key results of experiments conducted using IPCO and the contribution towards the creation of the algorithms, based on the methods described in the Methodology chapter are also discussed. The output and results obtained throughout the experiments are analysed and interpreted. Summaries of the results obtained are presented in figures and tables.

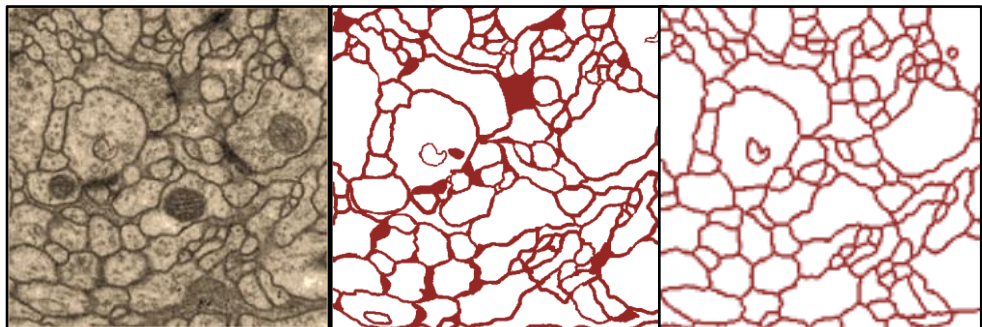


Figure 5.1: Left: ssTEM section from *Drosophila* first instar larvae; Middle: corresponding ground-truth maps for cell membrane (maroon); Right: segmentation result using IPCO.

This chapter is about the IPCO algorithm. This algorithm is not a pre-existing algorithm; it is a new algorithm that uses pre-existing image processing functions as a basis of the IPCO algorithm. The new function that was introduced in this algorithm is the ‘combiner’ function. This function is designed so that it can encourage ‘image blending’ (merging between 2 outputs, where the chain was designed in such a manner that the functions can receive input from earlier functions and this capability of the processing chain enables it to be regarded as a processing network). 4 different types of

blending were used, *i.e.*: averaging, scaled multiplication, addition and minimum/maximum. These functions are specifically designed to encourage chains to form different representations and transformations. The existence of these functions can be considered a contribution to the processing network which results in a better performance score. All functions that were introduced in the processing chain were meant to integrate with one another for better accuracy and speed.

As stated in earlier chapters, the focus of this research is on the problem of neuronal membrane detection in which the core challenge consists of distinguishing membranes from organelles. The methodological focus of IPCO is to run the algorithm in an automated manner by adopting a hybrid global stochastic optimisation method, with a combination of Genetic Algorithm, Differential Evolution, and Rank-Based Uniform Crossover (RBUC). F1 measures are also used in IPCO, for error measurement.

## **5.1 IPCO's training and testing**

A dataset containing 30 slices of *Drosophila* and corresponding ground truth were obtained from the ISBI team. The chains were trained on small sections of a small subset of the training dataset (slices of data) obtained from ISBI. Then, the chains were tested on the remaining unseen slices of data, *i.e.*, the slices that were not used to optimise the chains. IPCO scored a performance value of 91.67% for the test set at a speed of only 10 seconds per image. The chain took only 280 seconds to optimise, which is approximately less than five minutes for typically less than 500 optimisation generations, for the recorded highest IPCO score. This score satisfies the main aim of the research by detecting membranes and eliminating organelles with high accuracy and high speed. No specialised hardware is needed, and IPCO leads to chains consisting of short sequences of basic processing steps which are efficient and easy to interpret. Moreover, IPCO is flexible and can be applied to many different types of datasets.

As a continuation from the early LCHF algorithm, IPCO still maintains the simple and efficient approach, based on several basic processing steps, including local contrast enhancement, thresholding, denoising, hole-filling, watershed segmentation, and morphological operations.

## **5.2 The Novelty**

The novelty of the neuronal membrane detection algorithm lies in optimisation, the type of dataset used, and the new set of chains found. The work in this stage of the research differs from that of other compared researchers (Chapter 2) in terms of the set of functions used, the parameterisations allowed, the optimisation methods adopted, the combination framework, and the testing and analyses conducted. A new category of special-purpose ‘combiner’ functions are included in IPCO, in comparison with the previous LCHF algorithm. It is specifically designed to encourage chains to form various representations and transformations. As mentioned earlier, IPCO adopts a hybrid global stochastic optimisation method, which includes an element of genetic algorithms, differential evolution, and RBUC. Moreover, systematic analyses of the statistics of optimised chains revealed several interesting and unconventional insights pertaining to preprocessing, classification, post-processing, and speed. The types of analyses conducted are novel and reveal interesting insights pertaining to denoising and its appearance in unorthodox positions in image processing pipelines.

## **5.3 IPCO Processing functions**

IPCO is a continuing effort from LCHF. In essence, all the LCHF functions are used and other needed functions are added to further enhance the capability of IPCO.

Table 5.1 and Figure 5.2 give a list of all the functions used in IPCO.



Table 5.1: List of all the functions used in IPCO.

| <b>Main Processing Functions</b> | <b>Pre-experiment using Built-in Function Subtypes</b>                  | <b>Parameter Available for Each Function Subtype</b>  |
|----------------------------------|---|---|
| Denoising                        | MedianFilter<br>Wiener Filter   | Size of the neighbourhood used for filtering is from a minimum of one to a maximum of 10.   |
| Contrast Enhancement             | CLAHE<br><br><br><br><br><br><br><br><br><br>HISTEQ<br><br><br>ImAdjust | <p>NumTiles<br/>Minimum Tile (rectangular contextual region) for row and column is set to 25 and Maximum Tile is 50. The function divides the image using this tile value.</p> <p>ClipLimit<br/>Contrast enhancement limit is set to min 0 and max 1, the scalar value has to be in the range [0 1].</p> <p>Nbins<br/>Number of bins for histogram is set at a min 2 and max 256.</p> <p>Intensity values, min of 0 and max of 255.<br/>Contrast Limit is set to min of 0 and max of 1.</p> |
| Thresholding                     | Single Value<br>Double Value  | Min of 70 and Max of 130.<br>Min of 70 and Max of 130.  |
| Hole-Filling                     | 2D<br>Connectivity  | Clears the image border using four connected neighbourhoods for 2D connectivity.  |
| Watershed                        | 2D Inputs   | Specifies the connectivity (4 and 8) to be used in watershed computation. The results are complements (zeroes become ones and ones become zeroes)   |
| Morphological Operators          | Eroding<br><br><br>Opening  | <p>Min is set to 1 and Max to 10; uses <i>disk</i>, <i>ball</i>, <i>diamond</i>, and <i>octagon</i> as structuring elements (strel).</p> <p>Min number of pixels that defines 'small objects' is set to one and max set to 100; connectivity are set to 4 and 8 scalar values.</p>  |
| Combination Function             | MinMax,<br>Average,<br>Addition,<br>Multiply                            | For the Multiply Function's Scaling factor: MinScale is set to 0 and MaxScale is set to 2.  |

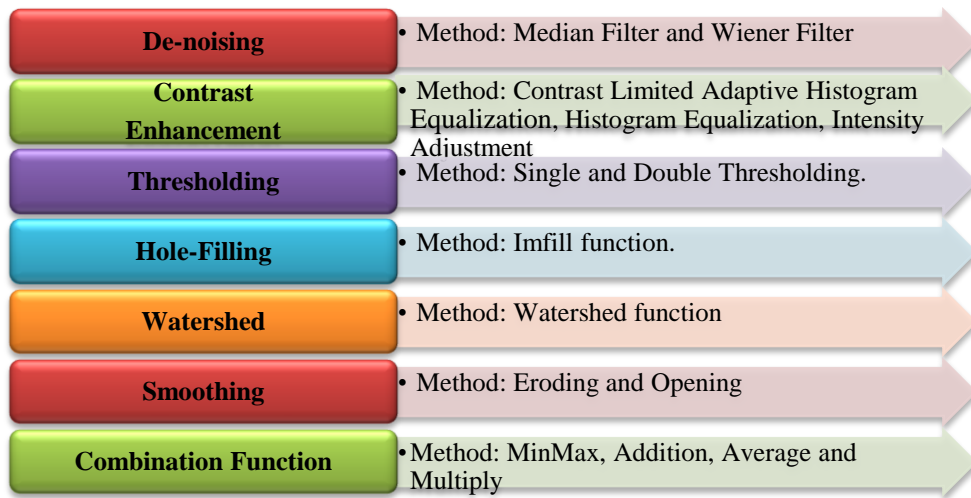


Figure 5.2: List of all the functions used in IPCO (represent in Diagram).

Table 5.1 and Figure 5.2: Main categories of processing functions available to IPCO (there is no order restriction, functions can appear in any order).

The other enhanced functions included in IPCO that are not in LCHF are the watershed and combination functions.

### 5.3.1 Watershed Function

This function returns a labelled image (different segments will have different pixel values). This method was adopted to eliminate ‘jutting line’ artefacts. The algorithm used in the Image Processing Toolbox is adapted from Meyer’s flooding algorithm.

In IPCO, two post-processing functions are available to the optimisation process: *opening* and *eroding*. Note that although these functions are typically categorised as post-processing functions, optimised chains often show them in unorthodox positions (even in early stages) which calls for caution in the categorisation of functions.

### 5.3.2 Simple Combination Functions

The following four combination functions are used in IPCO:

- i) **Combine-Average:** This function computes the average between the output of the previous processing step and the output of any random previous processing step;

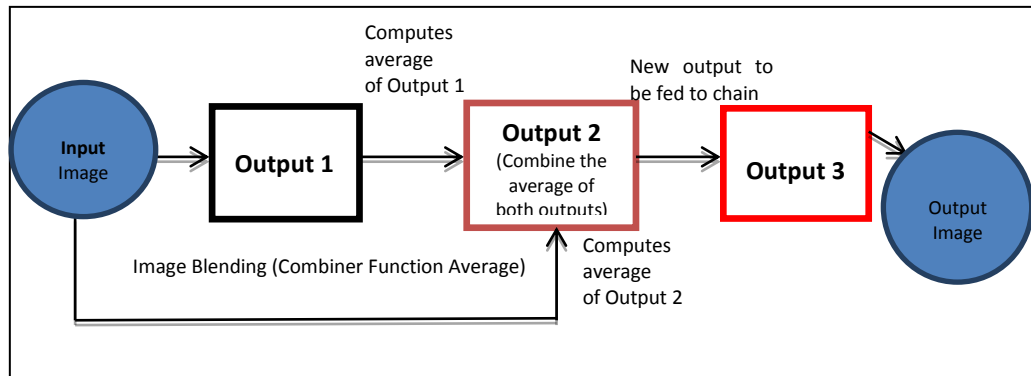


Figure 5.3: Flowchart showing the combiner Average processing for illustration.

- ii) **Combine-Addition:** This function adds the output of the previous processing step to the output of any random previous processing step;

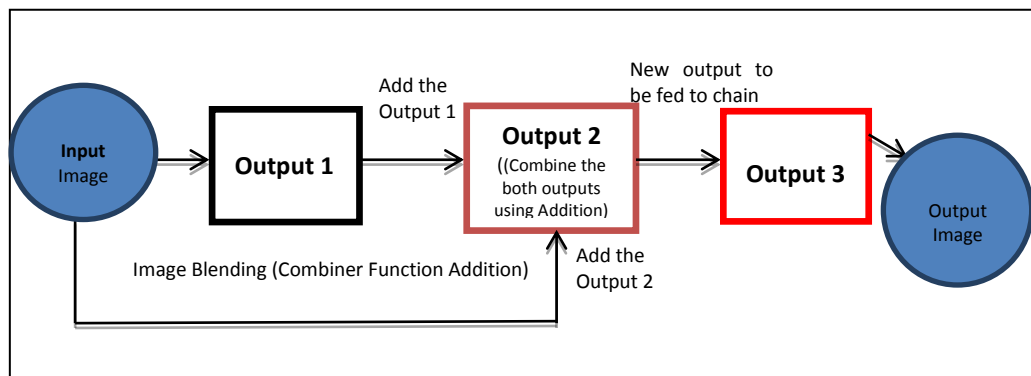


Figure 5.4: Flowchart showing the combinerAddition processing for illustration.

- iii) **Combine-Multiply:** This function computes the product of the output of the previous processing step and the output of any random previous processing step and multiplies the result by a scaling factor;

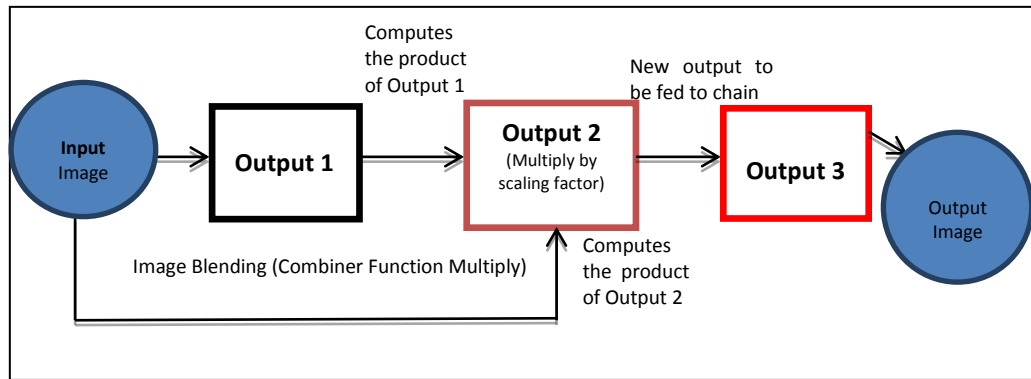


Figure 5.5: Flowchart showing the combinerMultiply processing for illustration.

iv) **Combine-MinMaxTwo**: This function compares the output of the previous processing step to the output of any random previous processing step, pixel by pixel, and takes either the minimum or the maximum (depending on which function is selected).

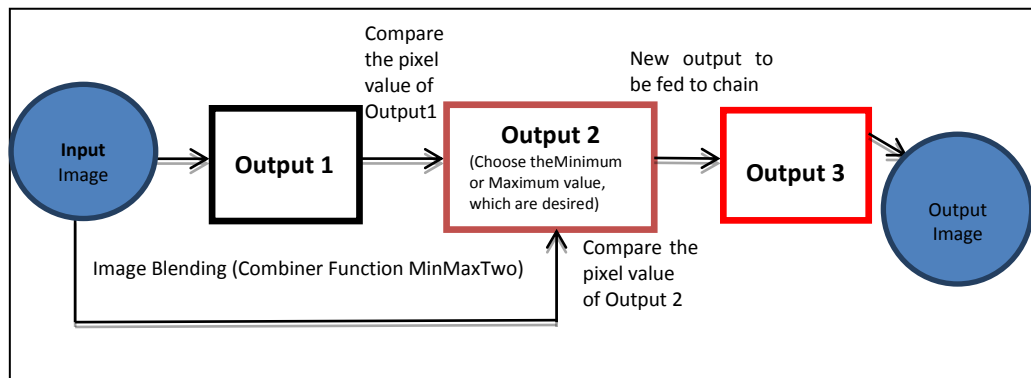


Figure 5.6: Flowchart showing the combiner MinMaxTwo processing for illustration.

Table 5.2 summarises the main categories of image processing functions available to IPCO for the experiments conducted in this research. The algorithm consists of several preprocessing, classification, and post-processing steps.

- The two preprocessing categories consist of denoising and contrast enhancement.
- The three classification functions consist of thresholding, hole-filling, and watershed segmentation. Thresholding is primarily responsible for membrane detection whereas hole-filling and watershed are primarily responsible for organelle elimination.

- The chains typically proceed to smooth the results via combining functions and morphological operators such as erosion and dilation.

Table 5.2 summarises the choices of processing functions and their general purposes.

Table 5.2: Main classes of IPCO functions with their corresponding image processing phases and general purposes.

| <b>Choices of Processing Functions</b>               | <b>Traditional Image Processing Phase</b> | <b>Typical Purpose</b> |
|--|---|------------------------|
| Denoising  | Pre-processing                            | Cleaning               |
| Contrast Enhancement                                 | Pre-processing                            | Enhancing              |
| Thresholding   | Classification                            | Classifying            |
| Hole Filling   | Classification                            | Classifying            |
| Watershed  | Classification                            | Classifying            |
| Combination Function of MinMax, Average and Multiply | Classification                            | Hybrid                 |
| Morphological Operators                              | Post-processing                           | Cleaning               |

Using IPCO, the algorithm executes automatically to reach the target cost of zero or a maximum of 10000 generations, whichever occurs first. Section 5.4.2 discusses the best result obtained thus far and how IPCO can lead to a diverse set of useful chains, many of which consist of unorthodox sequences and choices of functions.

#### **5.4 IPCO Computational Flow**

In the experiments conducted in this research, chains were allowed to have a maximum number of eight basic functions. In general, functions can appear in any order and can even repeat several times in a chain. Each function typically comes along with a small set of parameters which also undergoes optimisation (*e.g.*, window size for the median function). IPCO can also be considered fast

at pixel classification, where the task of detecting membranes in Transmission Electron Microscopy (TEM) images with a resolution of  $512 \times 512$  pixels can be accomplished in about 10 seconds per image on an average personal computer (*i.e.*, 1.60 GHz processor and 1.48 GB of RAM). Moreover, there is no requirement for specialised hardware.

IPCO was executed in automated manner, with the end result of processing being image pixels classified as ‘membrane’ being labelled ‘1’ and pixels classified as ‘non-membrane’ being labelled ‘0’. The 0-labelled pixels include various organelles that are eliminated from the image. These binary 0-1 images are compared with the binary images of the ground truth to find pixels that are correctly and incorrectly identified. Pixels that are falsely identified as a boundary in the IPCO output, but are classed as the cell interior pixels in the ground-truth image are referred to as False Positives. Conversely, pixels that are identified as interior in the IPCO output, but are classed as a boundary in the ground-truth image are referred to as false negatives. The flowchart below shows the overall computation flow in a specific IPCO chain consisting of three functions (for illustration purposes).

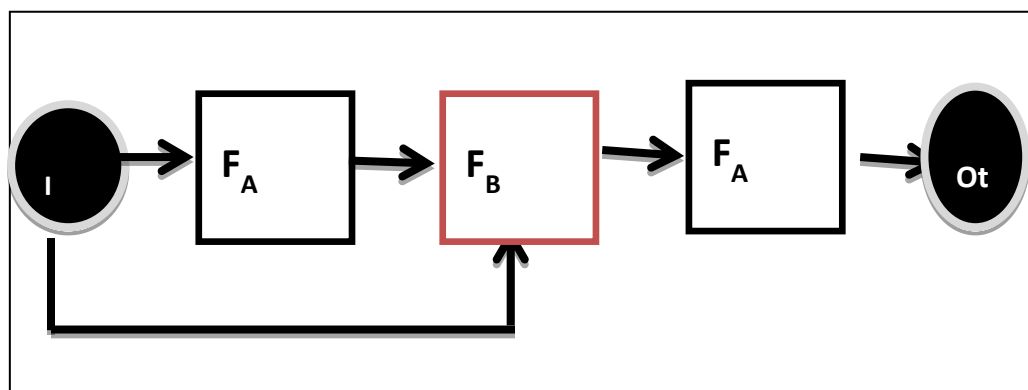


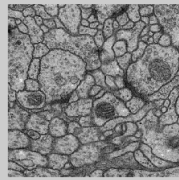


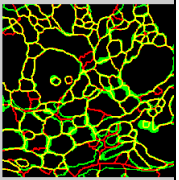
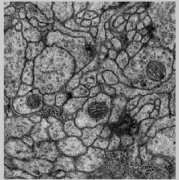
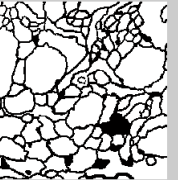

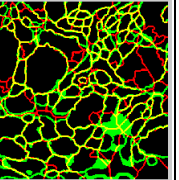
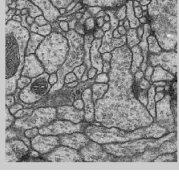
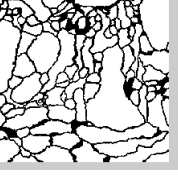
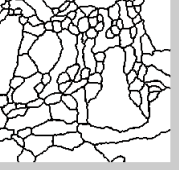
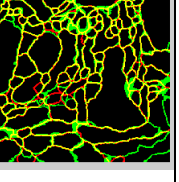
Figure 5.7: Flowchart showing the overall computational flow in a specific IPCO chain consisting of three functions. I: input image. O: output image.  $F_A$ : single-input function such as denoising.  $F_B$ : multiple-input function such as image blending (*e.g.*, Combiner MinMax).

Figure 5.7 shows the flowchart for an IPCO chain consisting of three functions. In the experiments, IPCO chains were allowed to use a maximum of eight functions.

### 5.4.1 Outputs obtained using IPCO

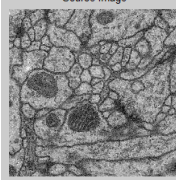
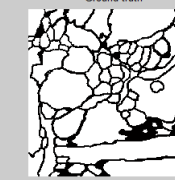
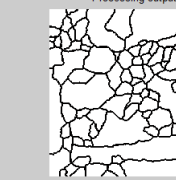
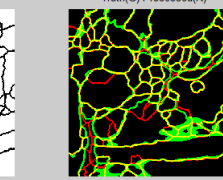
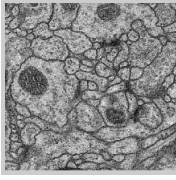
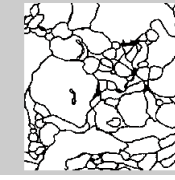
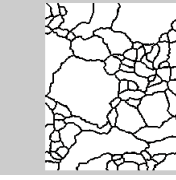
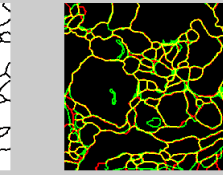
Below are several sample output images obtained using an IPCO chain. At a glance, the outputs appear to be virtually identical to the corresponding ground-truth images. Most of the 8.33% average error is possibly due to missing black patches (false negatives; coloured green) and some extra lines, possibly due to the watershed function (false positives; coloured red), as well as non-identical thickness of the lines. True positives are coloured in yellow, whereas true negatives are coloured in black.

Figure 5.8: Sample output images obtained using an IPCO chain.(More output is given in the Appendix section).

| Img | Sample output obtained using IPCO   |   |  |   |
|-----|---|---|--|---|
| 2   |   |   |   |   |
| 5   |  |  |  |  |
| 10  |  |  |  |  |

Continue...

...continued

|    |   |   |  |   |
|----|---|---|--|---|
| 20 |  |  |  |  |
| 30 |  |  |  |  |

#### 5.4.2 Results obtained using IPCO (For average F1 scores > 90%)

Several optimisation trials were conducted, in order to obtain the best image processing chain and to study the statistics of ‘good’ chains. ‘Good’ chains are chains that have high performance value with low error measurement, typically with F1 scores greater than 90%.

The 10 best chains (‘good’ chains), which show an average F1 score greater than 90%, are given below.

As shown in Table 5.3, iteration count is the number of times the generations are optimised. For experimental purposes, the maximum number of iterations was set at 10,000. Thus, it would automatically stop at either 10,000 or when a zero error measure was met (whichever occurred first). The average F1 is the error measurement calculation which takes the error measurement of every set of 30 data slices (the *Drosophila* dataset, for this research) performance, averages it, and outputs it as the average F1 score. The combination of IPCO chains shows the list of functions and its position in each output chain. As can be seen in the table below, the choice of function and its appearance changes for some functions, whereas there are also functions that are favoured and continuously appear in the early stage of the chain (*e.g.*, the Contrast function). This may occur because the chains typically prefer to enhance the data images contrast for better image enhancement for further processing. There are also



some functions that appear in multiple positions (e.g., Denoising, Morphological Operator, and Thresholding function) but with different parameter usage. There is a significant diversity of ‘good’ chains; this shows that many different sets of combinations of functions can appear and there is no rule in the appearance of functions. This paves the way for many interesting findings and observations.

Table 5.3: IPCO chains with F1 score greater than 90%, averaged over all training images.

| <b>Iteration Count</b> | <b>Scores (Average F1)</b> | <b>Combination of IPCO chains</b>   |
|------------------------|----------------------------|---|
| 3414                   | 91.67                      | Contrast→Thresh→Morph(Opening)→Denoise→Watershed→Hole-Fill→Morph(Eroding) |
| 200                    | 91.64                      | Contrast→Thresh→HoleFill→Denoise→Morph(Opening)                           |
| 96                     | 91.43                      | Contrast→Denoise→Denoise→Combine→Thresh→HoleFill                          |
| 7003                   | 91.35                      | Contrast→Denoise→HoleFill→Denoise→Combine→Thresh→HoleFill→Denoise         |
| 487                    | 91.27                      | Contrast→Combine(Average)→Thresh→HoleFill→Denoise                         |
| 200                    | 91.15                      | Contrast→Thresh→HoleFill→Thresh→Morph(Opening)→Denoise→Morph(Opening)     |
| 2548                   | 91.12                      | Contrast→Denoise→Combine(MinMax)→Denoise→Thresh→HoleFill                  |
| 200                    | 91.05                      | Contrast→Combine→HoleFill→Morph(Opening)→Watershed→Thresh→Morph(Erode)    |
| 70                     | 91.01                      | Contrast→Denoise→Thresh→Thresh→HoleFill                                   |
| 324                    | 90.33                      | Contrast→Thresh→Watershed→Morph(Erode)→HoleFill                           |

#### 5.4.2.

##### a) **Results obtained using IPCO (For Best Single Chain—as per F1 score)**

If each function is classified in terms of its general purpose, as per Table 1 (i.e., enhance, classify, and clean), the following simplified chains can arguably be obtained:

- First Best Chain:  
Enhance→Classify→Clean→Clean→Classify→Classify→Clean
- Second Best Chain:  
Enhance→Classify→Classify→Clean→Clean

As can be seen, in both chains, cleaning only takes place after enhancement and classification. This arguably runs contrary to common expectation. This is something that needs to be taken into account by image processing users, that one should not always clean images at an early stage because this will remove important information that may be needed by other component functions.

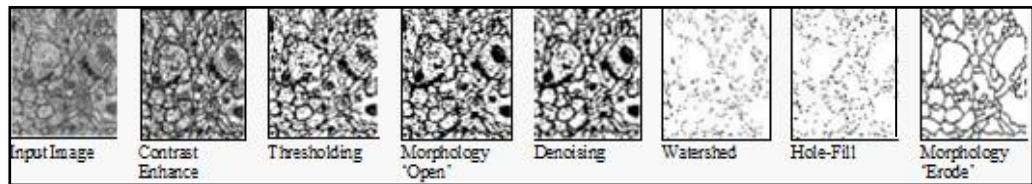


Figure 5.9: Sequence of functions and processed images of a chain with an F1 score of 91.67%.

Figure 5.9 shows the sequence of functions and processed images of the best chain (at the time of writing). It is a visual summary of the behaviour of the best IPCO chain.

## 5.4.2.

### b) Optimisation dynamics

The figures below on optimisation dynamics show the representation of chains and corresponding iteration counts. The X axis shows the number of iterations that occur versus the best cost (shown on the Y axis). The best costs are the error measurement scores for the chain, with the lower the score, the better it is in terms of its F1 score. From the figure, it is clear that the lower the best cost (on the training dataset) is, the higher the F1 score (on the test dataset) tends to be; however, overfitting is occasionally observed. Earlier, the stopping condition for the iteration was set as cost = 0 or generation set to maximum 10,000. However, after conducting the experiments several times, it was

observed that running the optimisation for longer than 1000 generations does not lead to any significant and advantageous difference in average F1 score.

Figure 5.10 shows the best cost for the five chains with iterations lower than 200. These five chains were chosen from among the listed 10 chains, which can be seen in Figure 5.11. These chains were randomly chosen from among the best chains which have recorded F1 scores greater than 90%.

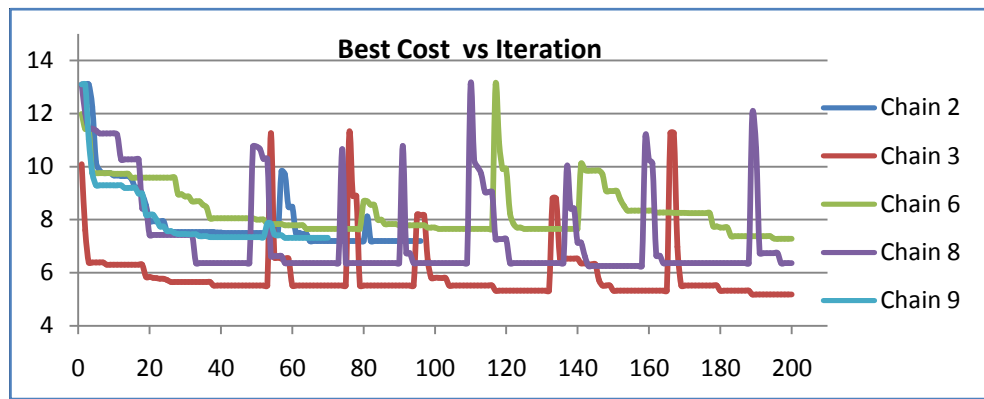


Figure 5.10: Best Cost versus iteration count for five chains out of the 10 best IPCO chains (as per Table 5.2).

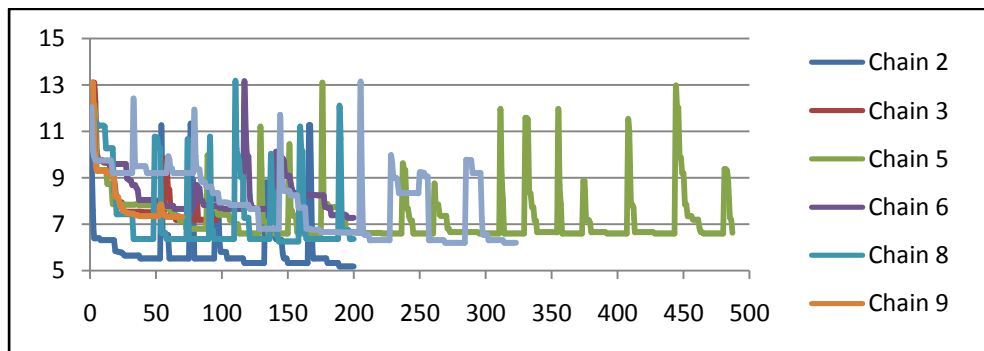


Figure 5.11: Average F1 value versus iteration count for the 10 best IPCO chains (as per Table 5.2).

An example of an iteration (or generation) graph constructed from the results of the experiments is shown below.

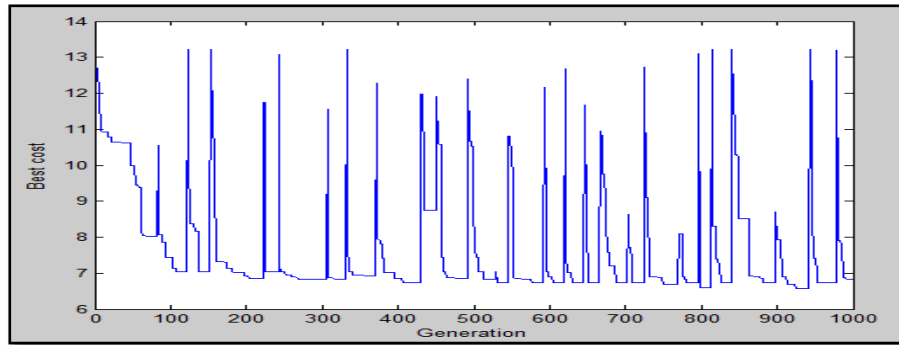


Figure 5.12: Iteration Graph for 1000 generations.

On conducting the experiments several times, we observed that running the optimisation longer than 1000 generations did not tend to lead to any significant and advantageous difference in average F1 scores. The spikes in the above figure are caused by the optimisation algorithm. When the optimisation process is considered as stagnating it creates a new random population. The best solution in this new population tends to be worse than the current best solution, which is why a spike occurs. In the creation of new solutions via recombination operators, the optimisation algorithm considers the current and all old populations probabilistically.

### 5.5 Observations, New Findings, and Suggestions

a) As shown in Table 5.3 regarding observations made as to the composition of chains, it can be seen that the Hole-filling function appears in all processing chains, and so does the Contrast and Thresholding functions.

b) From Table 5.3, it appears that the contrast function is the first choice in all the 10 displayed chains. Eight out of 10 chains (as displayed in Table 5.3) appear to choose Denoising as one of the functions of choice. The named functions were considered the core IPCO functions. The other functions such as watershed and morphological functions only appear to give better output for the appearance of the membrane lines. Analysis of the top two chains in Table 5.3 shows that the main difference between the 91.67% and 91.64% chain is the watershed and morphological functions. Thus, it can be said that when the watershed function is carried out the membrane line is excessively thinned (as

shown in the figure below), which affects the F1 score detrimentally because the compared gold-standard membrane line is slightly thicker. The application of morphological erode helps to thicken the watershed thin membrane line. This can be the reason why in most chains, these two functions seem to appear one after the other. This is analogous to the difference between the top two chains and may explain the F1 difference of 0.03%.

c) The Denoising function is associated with another interesting observation. It is well-known that the main purpose of denoising is to filter out image noise in order to minimise detrimental effects in subsequent processing. Denoising is mostly carried out as an early preprocessing stage before application of other core functions. However, this experiment revealed many interesting findings. For this membrane segmentation problem, denoising typically appears later in the chain. According to the above table, 10 out of 10 cases (including the top scoring chain) found that contrast enhancement appears before denoising. Using contrast enhancement enhances and preserves image information, whereas in denoising, the unwanted information or noises are filtered out. For this dataset, the appearance of contrast in the early stage and denoising in the later stage suggests that details need to be enhanced before being cleaned, which can be encapsulated by the heuristic *enhance it before you lose it*.

d) In the other chains with scores below 90% (not depicted), the denoising component tends to appear early in the chain before classification.

As a small conclusion, can say that 100% of the ‘good’ chains adopted all the main components of IPCO (*i.e.*, contrast enhancement, thresholding, and hole-filling), 90% of chains adopted denoising as one of their components, and 50% of chains preferred to include the watershed function.

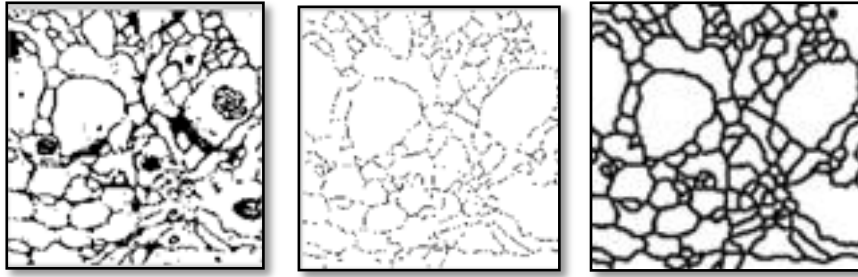
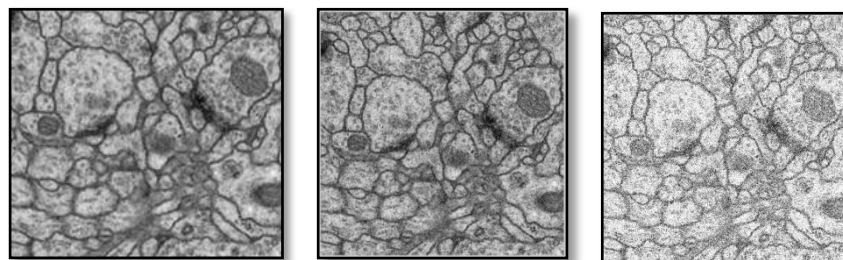


Figure 5.13: (a) Binary image, (b) Watershed Image, (c) Result of application of Morphological Eroding.

There is also another doubt that the early suggested ‘enhancement before cleaning’ may work for the *Drosophila* dataset because the dataset may be contaminated with low degree of noise, hence the order of processing it may not matter (incorporate contrast enhancement before denoising). But the question arises is that: If the dataset are contaminated with higher degree of noise, then does the suggest manner still workable?

A way to determine whether denoising can be used in later stage is to carry out an experiment by adding sufficient Gaussian noise to the images to be seen as noisy images. Then proceed denoising at the later stage in IPCO.

Experiments were carried out by adding extra noise to the dataset. Below are the results.



(a) (b) (c)

Figure 5.14: (a) Original Image, (b) Image with added ‘Salt and Pepper’ Noise, (c) Image with added Gaussian noise

Experiment 1: Adding Salt and Pepper noise to the image.

Result: The result is consistent with my suggestion to enhance first, before denoising. In other words, the denoising appears in later stages after contrast enhancement. This ordering also contributes

to high scoring chains, *e.g.* F1 score of 91.10%. The diagram below summarizes this result. From this result, the significance of denoising appearing at middle or late stages is clear, even when the input image is supplied with extra noise.

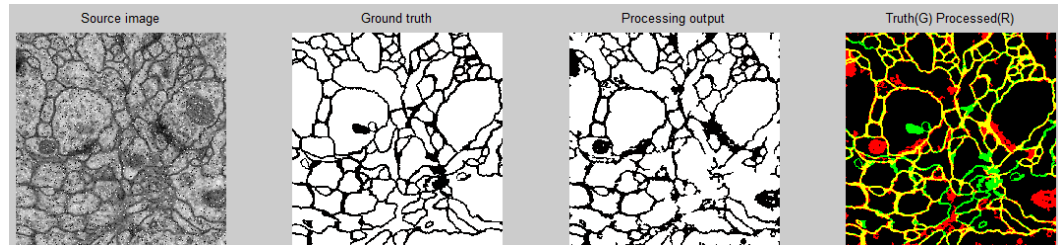


Figure 5.15: Shows an example output after adding Salt and Pepper noise.

Experiment 2: Adding Gaussian noise to the image.

Result: The result shows that in most cases, the ‘Denoising’ function is not even selected by the optimization process. The Morphological and Watershed functions seem to be favoured for this modified dataset. In cases where Denoising appears in middle or later stages, the scores are not promising (less than 91% of F1 score). From the result, it is clear that the image processing rule of thumb, whereby denoising appears early in the processing chain, is not empirically supported, even when images are corrupted with additional noise.

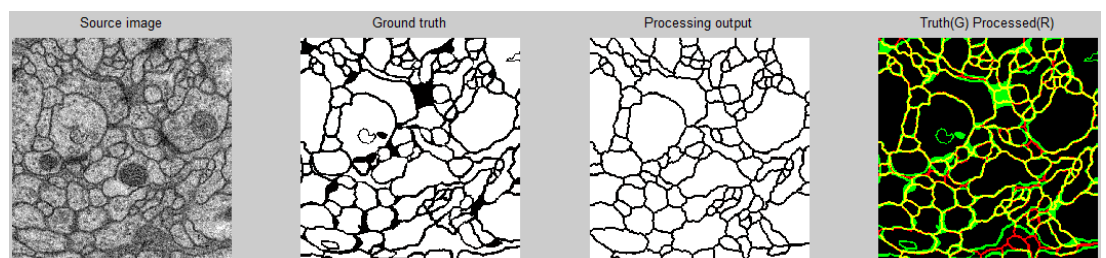


Figure 5.16: Shows an example output after adding extra Gaussian noise.

Experiment 3: Adding Poisson noise to the image.

Result: The result of this experiment is closer to the result of Experiment 1, which involved ‘Salt and Pepper’ noise. The output images result in high scores (F1 score > 91%), and the denoising appears in later stages after contrast enhancement. From the result, the significance of denoising appearing in

middle or late stages is evident, although the image is being supplied with extra noise.

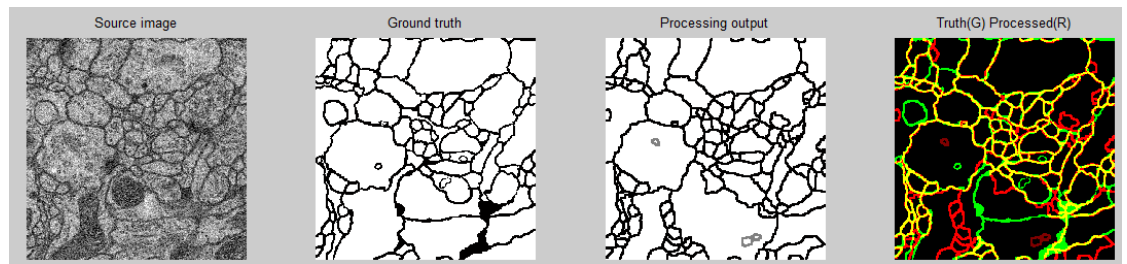


Figure 5.17: Shows an example output after adding extra Poisson noise.

## 5.6 Appearance of the Used Function in the Chain

From the analysis conducted, it was discovered that some functions are favoured and almost always appear in all chains generated by IPCO. In this list, the Thresholding function seems to score more than 90% of appearance in all chains generated by IPCO.

When the function is forced and maximum functions to be used are set, some interesting outcomes were noticed. Initially, only one function was allowed, then two functions, and so on in that order until the maximum of eight functions. Each of these functions were utilised 50 times in experiments carried out to collect information about the nature of function appearance and its frequency and repetition manner.

Table 5.4 summarises the frequency of function appearance and repetition.

Table 5.4: Frequency of Function appearance in 50 trial experiments.

| Functions                    | 1 Func | 2 Func | 3 Func | 4 Func | 5 Func | 6 Func | 7 Func | 8 Func | Average |
|------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|---------|
| Thresholding (Simple/Double) | 100%   | 96%    | 72%    | 76%    | 92%    | 92%    | 92%    | 96%    | 90%     |
| Hole Filling                 | 0%     | 36%    | 32%    | 44%    | 64%    | 80%    | 80%    | 76%    | 52%     |
| Morphology                   | 0%     | 4%     | 44%    | 52%    | 48%    | 48%    | 60%    | 76%    | 42%     |
| Denoising                    | 0%     | 52%    | 16%    | 48%    | 48%    | 52%    | 52%    | 52%    | 40%     |
| Combination                  | 0%     | 0%     | 4%     | 32%    | 32%    | 60%    | 92%    | 100%   | 40%     |
| Contrast Enhancement         | 0%     | 4%     | 44%    | 24%    | 32%    | 48%    | 52%    | 60%    | 33%     |
| Watershed                    | 0%     | 0%     | 36%    | 28%    | 36%    | 44%    | 60%    | 60%    | 33%     |
| Edge                         | 0%     | 0%     | 0%     | 0%     | 8%     | 24%    | 24%    | 32%    | 11%     |



When only a choice for optimiser was given to choose only one function, 100% of the chains chose the Thresholding function.

A conclusion can thus be drawn:

- i. The Thresholding and Denoising functions are the functions with the highest appearance percentage in a chain. The the maximum number of functions supplied to the process is eight, but the optimisation can returns a minimum number of functions in a chain (*e.g.*, two or three functions). This scenario can be termed the ‘Shortest Chain’ scenario.
- ii. The optimisation also returns a chain that uses all eight functions supplied. Occasionally, the functions appear to repeat themselves, in both a consecutive and non-consecutive manner. However, they resemble different parameter usages. This scenario can be termed the ‘Longest Chain’ scenario.
- iii. ‘Combination’ functions (*e.g.*, MinMax, Average, Multiply) seem to appear in all ‘Longest Chain’.

## **5.7 Comparison with state-of-the-art approaches**

To compare the method with current state-of-the-art approaches, the ISBI challenge results were used (with approval from the organiser). Table 5.5 shows the comparison with the ISBI 2012 challenge workshop Leaders Group and the approach called ICOS.

### **a) Evaluation Metrics Used**

To evaluate and rank the performances of the participant methods, 2D topology-based segmentation metrics were used as the challenge. The proposed approach, called ‘ICOS’, was sent to get a result for benchmarking purposes. Table 5.5 gives the details of scores and the participating groups.

The metrics were as follows (Jain *et al.*, 2010):

- Mergers Warping Error: A segmentation metric that penalises topological disagreements (*i.e.*, object splits and mergers).

- Rand Error: Defined as one—the maximal F-score of the Rand index, also known as foreground restricts Rand error. It is a measure of the similarity between two clusters/segmentations.
- Pixel Error: Defined as one—the maximal F-score of pixel similarity, or squared Euclidean distance between the original and the result labels.

Table 5.5: Participating groups for first challenge on 2D segmentation of neuronal processes in electron microscopy images in the International Symposium on Biomedical Imaging (ISBI) 2012 challenge workshop (Leaders Group).

| Group Name                 | Rand Error<br>[.10 <sup>-3</sup> ] | Warping Error<br>[.10 <sup>-6</sup> ] | Pixel Error<br>[.10 <sup>-3</sup> ] |
|----------------------------|------------------------------------|---------------------------------------|-------------------------------------|
| IDSIA – SCI                | 18                                 | 652                                   | 102                                 |
| Connectome                 | 59                                 | 577                                   | 64                                  |
| MLL-ETH                    | 63                                 | 581                                   | 79                                  |
| R1D                        | 67                                 | 539                                   | 67                                  |
| CRVI_T                     | 69                                 | 742                                   | 67                                  |
| Blackeagles                | 73                                 | 592                                   | 67                                  |
| CellProfiler               | 86                                 | 1049                                  | 85                                  |
| UofU                       | 87                                 | 2710                                  | 155                                 |
| Coxlab                     | 89                                 | 659                                   | 79                                  |
| CoMPLEX                    | 90                                 | 2188                                  | 134                                 |
| GVI                        | 91                                 | 915                                   | 99                                  |
| Vision Science UCL         | 98                                 | 1523                                  | 93                                  |
| sdu                        | 100                                | 998                                   | 71                                  |
| ml                         | 118                                | 1503                                  | 86                                  |
| <b>ICOS (The Approach)</b> | <b>141</b>                         | <b>2520</b>                           | <b>101</b>                          |
| CLP                        | 144                                | 1725                                  | 101                                 |
| IMMI                       | 144                                | 2959                                  | 104                                 |
| TSC+PP                     | 146                                | 885                                   | 87                                  |
| mla                        | 148                                | 1280                                  | 83                                  |
| MGUCC                      | 167                                | 2563                                  | 113                                 |
| Freiburg                   | 173                                | 1538                                  | 99                                  |
| CVJ                        | 229                                | 2895                                  | 124                                 |
| NIST                       | 230                                | 5246                                  | 140                                 |
| PurpleMatter               | 231                                | 1819                                  | 87                                  |
| ComputerVision Jena        | 280                                | 5116                                  | 135                                 |
| Bar-Ilan                   | 306                                | 2346                                  | 112                                 |
| ** threshold **            | 449                                | 17141                                 | 225                                 |

**Note:** These results are based on evaluation the methods conducted on the **full** test dataset, which remains private in order to keep the challenge open to new contributions. Participants can sent as many entries as they wish, but the table will only reflect the highest score of the same participants.

### 5.8 Comparison of IPCO score with Other Method scores

This comparison was conducted to show IPCO performance in comparison with other image processing collaboration techniques; for example, with Partial Differential Equation and the Watershed Merge Tree techniques (Liu *et al.*, 2012). The result was published by Tasdizen *et al.* (2014) (in Table 5.6 below). This method was chosen because it uses the same dataset; that is, the *Droshopila* dataset, to do the testing, thus the comparisons are more significant.

Table 5.6: Testing performance score of IPCO vs. Post-Processing for *Droshopila* dataset (Tasdizen *et al.*, 2014).

| Method  | Testing Result<br>Pixel Error Measurement      |
|---|--|
| <b>Post-processing (PDE + Watershed Merge Tree)</b> | 0.1026 (using test dataset as per ISBI result) |
| <b>IPCO (using test dataset)</b>                    | 0.1010 (using test dataset as per ISBI result) |
| <b>IPCO (using training dataset)</b>                | 0.0833 (using training dataset)                |

The testing performance of post-processing result with Partial Differential Equation + Watershed Merge tree for the *Droshopila* ssTEM dataset achieved an F1 score of 89.74% for the test dataset. IPCO scored marginally higher (0.16%) than PDE + Watershed Merge Tree which had a recorded score of 89.90% (as per ISBI record).

## 5.9 Ensembles of IPCO chains

A simple way to improve the generalisation capabilities of any classification approach is to combine several different classifiers in an ensemble. In this research, several IPCO chains were also combined (by manual selection) in ensembles and improvements in F1 scores obtained. As of the time of writing, the best ensemble of IPCO chains obtained an average F1 score of 92.11%.

Table 5.7 depicts the IPCO chains that were used in the best ensemble so far.

Table 5.7: IPCO chains used in the best ensembles (as of the time of writing).

| Chain-ns | IPCO Chains Used   |
|----------|--|
| 1        | Contrast→Thresh→Morph(Open)→Denoising→Watershed→HoleFill→Morph (Erode) |
| 2        | Contrast→Denoising→Denoising→Combining(MinMax)→Thresh→HoleFill         |
| 3        | Contrast→Denoising→Combine(MinMax)→Denoising→Thresh→Hole File          |
| 4        | Contrast→Thresh→HoleFill)→Denoising→Morph(Opening)                     |

A flowchart representation of the ensemble classification is shown below.

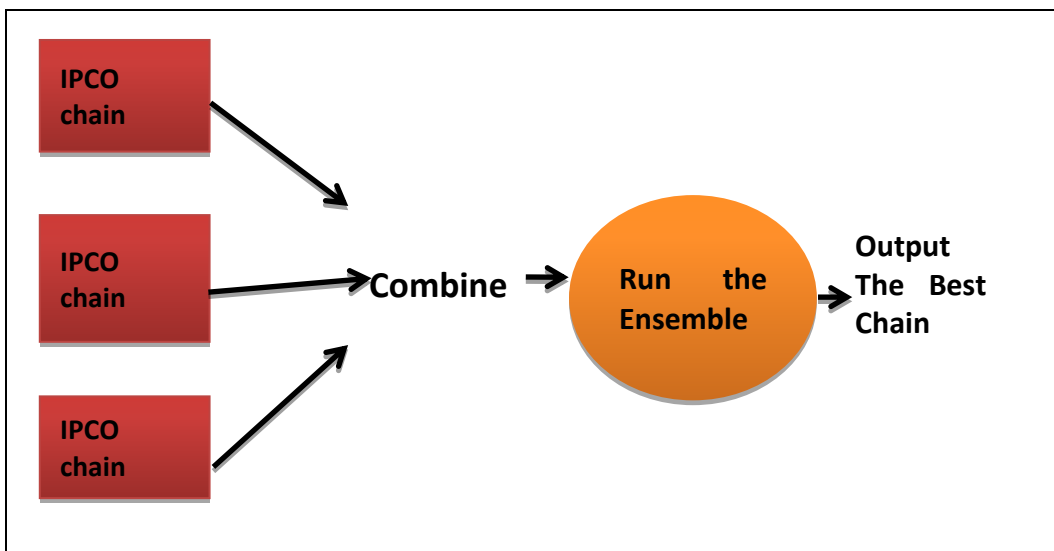
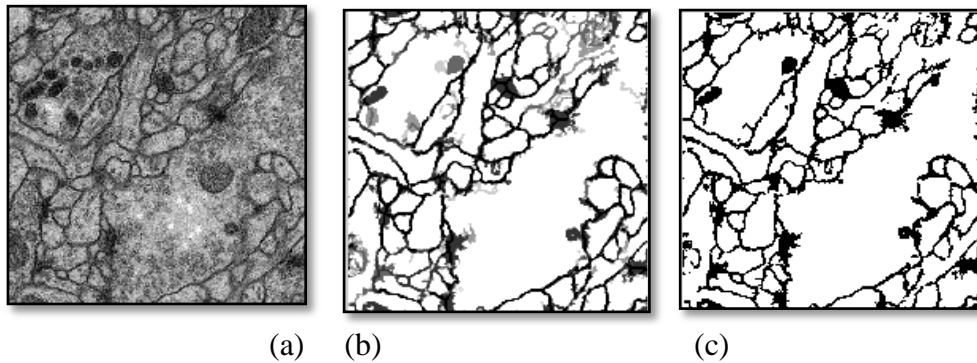


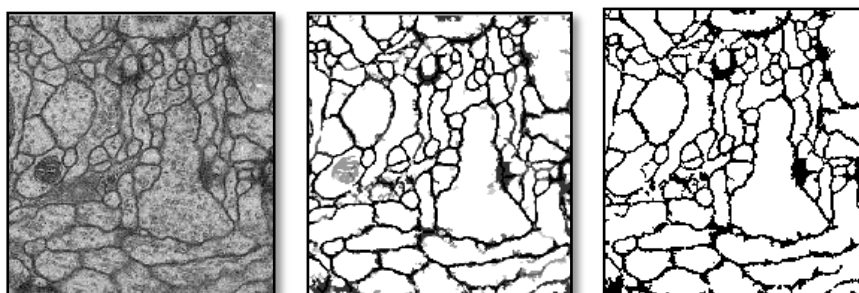
Figure 5.18: Ensemble Flowchart.



(a) Original Test Image (b) Average Output using IPCO (c) Classified Output using IPCO ( More output is given in the Appendix section).

Figure 5.19: Shows the output image obtained using IPCO method on the ISBI Test Image (Slice 2).

Figure 5.19(a) shows the average output, which was retrieved by combining all the outputs and then averaged. Classifying output 5.19(b) was also retrieved using a set threshold. The test was conducted using slice 2 of Test Image, which was provided by Cardona et al (2010). Figure 5.20 shows the average output and classified output for IPCO using Training Image.



Original Training Image Average Output Classified Output  
(Slice 10)

Figure 5.20: Ensemble output for the training Image.

Ensemble—The Highest Performance Score (as of time of writing)

- ✓ Ensemble Average of Precision: 91.85%
- ✓ Ensemble Average of Recall: 92.45%
- ✓ Ensemble Average of F1 score: 92.11%

## 5.10 Conclusion

IPCO and ensembles of IPCO chains not only highlight membrane boundaries, but also remove internal structures (eliminate organelles) successfully. To enhance F1 scores while preserving the simplicity and efficiency, global stochastic optimisation and ensemble methods were incorporated. The implementation of IPCO chains was found to be capable of efficiently detecting membranes in the ISBI 2012 challenge dataset with an average F1 score of 92.11%. IPCO implies simplicity and efficiency of simple sequences of image processing functions and involves the automated fine-tuning of an algorithm relative to a dataset. IPCO met the goals and objectives:

- Relatively fast and consistent optimisation process
- Does not require specialised hardware
- Low cost compared to high-end hardware needed for some compared approaches
- Fast classification
- Easy to use and deploy
- High ensemble accuracy: 92.11%
- Can distinguish membranes and organelles and also remove internal structures

Two papers associated with these findings were published:

- A paper was presented at the 2<sup>nd</sup> International Conference on Intelligent Systems and Image Processing 2014, in Kitakyushu, Japan.
- A paper was published in the Journal of the Institute of Industrial Application Engineers (JIAE).

A paper:

- ‘Image Processing Chain Optimization for Membrane Detection in Neural Slices’ is ready for publication.

The next chapter explains the parallel design algorithm, called Multiple Image Processing Optimisation Network, which operates in parallel to perform the classification.

## CHAPTER 6

### RESULTS

#### MULTIPLE IMAGE PROCESSING CHAIN OPTIMISATION (MIPCO) NETWORK

This chapter discusses the third algorithm, called the Multiple Image Processing Chain Optimisation Network (MIPCO). The key results of experiments conducted using MIPCO parallel network and the contribution of the research towards the creation of the algorithm, based on the methods described in the Methodology chapter are presented. The output and results obtained throughout the experiments are also analysed and interpreted.

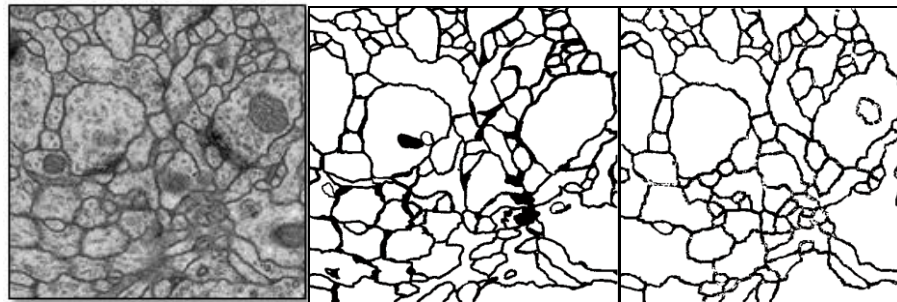


Figure 6.1: Left: ssTEM section from *Drosophila first instar larvae*; Middle: corresponding ground-truth maps for cell membrane (black); Right: segmentation result obtained using Multiple Image Processing Chain Optimisation Network (MIPCO).

MIPCO was introduced as a simple logical next step of IPCO with some biological motivation of neural system parallelisation. MIPCO is the result of efforts to further boost the performance of IPCO. IPCO is based on a representation consisting of a single chain of image processing functions, whereas MIPCO is a generalized representation which allows for multiple interacting chains in a network. MIPCO can be referred to as image processing network optimisation. MIPCO computes layer by layer and there is no dependency of functions in the same layer. Functions in a layer can receive input from any other function in previous layers. IPCO does not have this



capability since it is a single chain algorithm. Functions in a MIPCO layer can receive input from any other function in previous layers, including those from different chains.

MIPCO is referred to as network optimisation. According to Network or Graph Theory, a single chain is a special type of network. MIPCO performs better than the feed-forward approach and the nodes interconnect not just in sequential mode. As the focus of this research is the problem of neuronal membrane detection, in which the core challenge consists of distinguishing membranes from organelles, the methodological focus of MIPCO is execution in automated parallel manner by adopting a hybrid global stochastic optimisation method, with the combination of Genetic Algorithm, Differential Evolution, and Rank-Based Uniform Crossover (RBUC) and further enhance the performance of the algorithm. F1 measures are also used in MIPCO for error measurement.

MIPCO performed with an average F1 score of 91.80% on the test set, which is slightly higher than the average performance of the previous method, IPCO. Further, it took 20 seconds per image. This score satisfies the main aim of the research by detecting membranes and eliminating organelles with high accuracy and high speed. MIPCO is both efficient and interpretable, and facilitates the generation of new insights. No specialised hardware is needed, and MIPCO leads to a network consisting of short sequences of basic processing steps which are efficient and easy to interpret. MIPCO is also flexible and can be applied to many different types of datasets.

## **6.1 Processing by MIPCO**

As per continuation from the early LCHF and IPCO algorithms, MIPCO still maintains the simple and efficient approach, based on several basic processing steps, including local contrast enhancement, thresholding, denoising, hole-filling, watershed segmentation, morphological operators, and integrating with combination functions. The difference here is that MIPCO executes in parallel and is able to exchange intermediate information. For experimental purposes,

and for this research purpose, a maximum of five chains and eight functions were supplied to MIPCO network for its optimisation process.

MIPCO computes layer by layer and there is no dependency of functions in the same layer. Functions in a layer can receive input from any other function in previous layers. Therefore, a layer must complete all computation before the next layer can initiate its own computation.

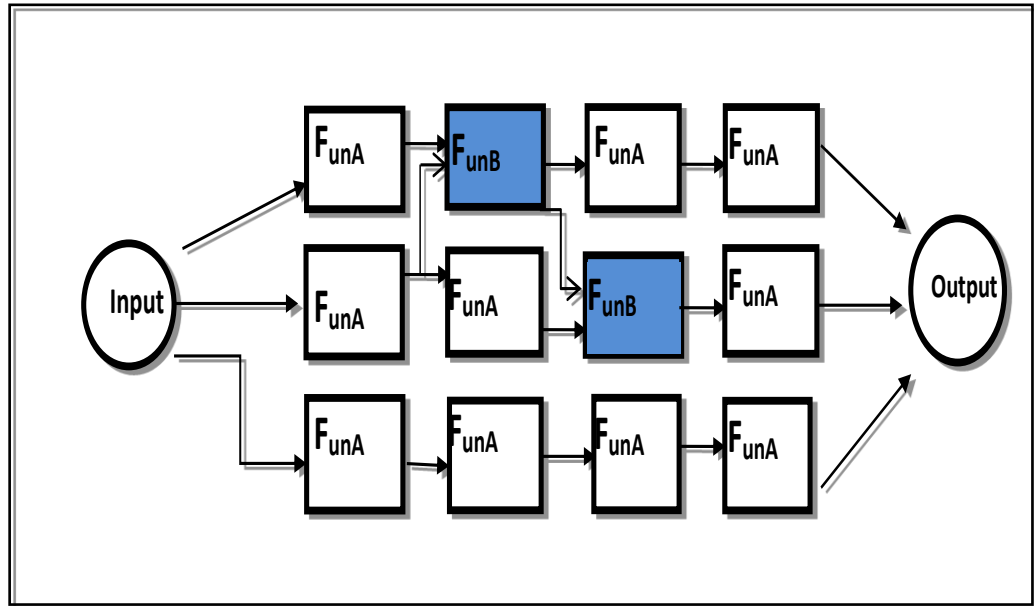


Figure 6.2 shows a flowchart of a MIPCO network consisting of three chains and four layers. In the experiments, MIPCO networks were allowed to use a maximum of eight functions per chain, and five parallel chains.

As with IPCO, a simple and efficient approach is proposed based on several basic processing steps, including local contrast enhancement, thresholding, denoising, hole-filling, watershed segmentation, and morphological operations. The MIPCO functions are classified into different types (*e.g.*, contrast modulation vs. denoising) and sub-types (*e.g.*, median vs. Wiener). Types are further classified into three broad categories: preprocessing, classification, and post-processing. The two main types of preprocessing functions currently being used are denoising and contrast enhancement. The three main types of classification functions are thresholding, hole-filling, and watershed. Post-processing functions include smoothing by combining functions and morphological operators. Note that the categorisation of function types into

preprocessing, classification, and post-processing is based on their typical usage and interpretation. Further, optimisation often finds unexpected ways to use functions (*e.g.*, in some chains, denoising operators have been found in the middle of said chains). MIPCO network is fully automated, the sequences (or chains) of image processing functions are optimised by using a global stochastic optimisation approach to implement the framework. As stated above, the enhancement to the algorithm was done using parallelisation. The MIPCO network consists of one or more chains that interact and optimise together.

## 6.2 Training MIPCO

The *Drosophila* dataset obtained from the ISBI team consists of 30 slices with ground truth. The MIPCO network was trained on small sections of a small subset of the training dataset (slices of data) obtained from ISBI. Then, the network was tested on the remaining unseen slices of data, *i.e.*, slices that were not used to optimise the network.

To compute the cost function, typically a subset of slices 1 and 2 are taken, which accelerates the optimisation process significantly without excessively deteriorating accuracy (after optimisation, the network typically has F1 scores greater than 90%). MIPCO's optimisation process runs continuously until a target cost of zero has been reached or a maximum of 10,000 generations have been completed, whichever occurs first. MIPCO can lead to a diverse set of useful networks, many of which consist of unorthodox sequences and choices of functions. The functions are configured in different sequences and with different parameter settings, in response to changes in the cost function, defined as the F1 score relative to a subset of the training images. In the experiments conducted in the research, chains were allowed to have a maximum of eight basic functions and five chains, although the total pool of functions was much larger. In general, functions can appear in any order, and there is no restriction on order. Further, they can even repeat several times in a chain. Each function typically comes with a small set of parameters which also undergo optimisation (*e.g.*, tile size for the contrast function). In general, optimisation of a network for different types of data is not time-consuming

(typically less than 1000 optimisation generations). MIPCO network can also be considered fast at pixel classification, where the task of detecting membranes in Transmission Electron Microscopy (TEM) images with a resolution of  $343 \times 343$  pixels can be accomplished in about 20 seconds per image on an average personal computer (*i.e.*, 1.60 GHz processor and 1.48 GB of RAM) for five chains with eight functions. Moreover, there is no requirement for specialised hardware.

### **6.3 Findings**

MIPCO networks involve a relatively fast (*e.g.*, 20 second per image) and consistent optimisation process, which leads to a variety of useful and easily interpretable solutions. In experiments conducted using MIPCO, observation made pertaining to morphological operators and their appearance in unorthodox positions in image processing chains suggest a new set of pipelines for image processing. Some experiments showcase that MIPCO network can even have a high F1 score ( $> 90\%$ ) even with fewer chains (*e.g.*, three chains). The occurrence of some partner functions (*e.g.*, Watershed with Morphological Operator) is also observed. In addition, several interesting observations are made at the functions frequency level, its repetition manner, and the result generated in a table (*e.g.*, Table 6.3) to support the experimental data (which are further discussed in ensuing sections). Visual outputs are also presented in this chapter for visual inspection of the findings.

#### **6.3.1 Best Shortest MIPCO functions**

Several experiments were conducted to test various chain sizes and numbers of chains. Consequently, it was discovered that even with a small number of chains (three chains), MIPCO could still perform very well—consistently attaining F1 scores greater than 91%. Table 6.1 shows the smallest MIPCO cases with F1 scores greater than 91%.

a) **Best shortest MIPCO network**

Table 6.1: Best shortest network (with three chains) for MIPCO with score >91%.

| No | F1 Scores | Chain 1 –<br>Arranged in<br>1<br>2<br>3              | Chain 2 –<br>Arranged in<br>4<br>5<br>6            | Chain 3 –<br>Arranged in<br>7<br>8<br>9             |
|----|-----------|--|--|---|
| 1  | 91.43     | MorphOpen<br>Denoise Median<br>HoleFill              | DoubleThresh<br>Denoise Median<br>Double Thresh    | MorphOpen<br>Watershed<br>MorphErode                |
| 2  | 91.38     | MorphErode<br>Denoise Median<br>HoleFill             | Double Thresh<br>Thresh Simple<br>Denoise Median   | MorphOpen<br>Watershed<br>MorphErode                |
| 3  | 91.23     | Combine<br>MinMax<br>Denoise Median<br>Double Thresh | Combine<br>MinMax<br>Denoise Median<br>Morph Open  | MorphOpen<br>Watershed<br>MorphErode                |
| 4  | 91.21     | Double Thresh<br>Denoise Median<br>Thresh Simple     | MorphOpen<br>Watershed<br>MorphErode               | Double Thresh<br>Morph Erode<br>Combine<br>Multilpy |
| 5  | 91.16     | Double Thresh<br>Denoise Median<br>Thresh Simple     | MorphOpen<br>Denoise Wiener<br>Combine<br>Subtract | MorphOpen<br>Watershed<br>MorphErode                |
| 6  | 91.14     | MorphOpen<br>Watershed<br>MorphErode                 | Double Thresh<br>Denoise Median<br>Hole Fill       | Denoise Median<br>Double Thresh<br>Denoise Median   |
| 7  | 91.09     | Denoise Median<br>Morph Erode<br>Hole Fill           | MorphOpen<br>Watershed<br>MorphErode               | Combine<br>MinMax<br>Double Thresh<br>Morph Erode   |
| 8  | 91.01     | Double Thresh<br>Denoise Median<br>MorphErode        | MorphOpen<br>DoubleThresh<br>Denoise Median        | MorphOpen<br>Watershed<br>MorphErode                |

As stated earlier, in this research, the optimisation was set to accept a maximum of five chains and eight functions. However, in Table 6.1, there are

only three chains with three functions for each network. This scenario is used for the purpose of documenting this research and is known as ‘Shortest Chain’. During the experiments, it was discovered that even a combination of a few chains (*e.g.*, three) can result in good performance with high F1 scores (>91%). This also has an effect on the time factor because fewer chains definitely reduces the time spent on the optimisation process.

Table 6.1 shows the eight best shortest chains for MIPCO with scores >91%. In these eight optimisation chains, morphological operators are evident, and in unorthodox positions. In some cases, they are found at the beginning of the chains, whereas in other cases, they are found in the middle or at the ends of chains.

The experiments revealed that 100% of the ‘good’ networks adopted morphological operators, watershed, denoising and thresholding.

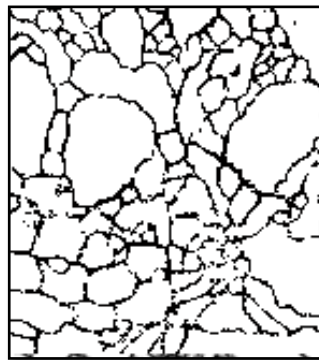
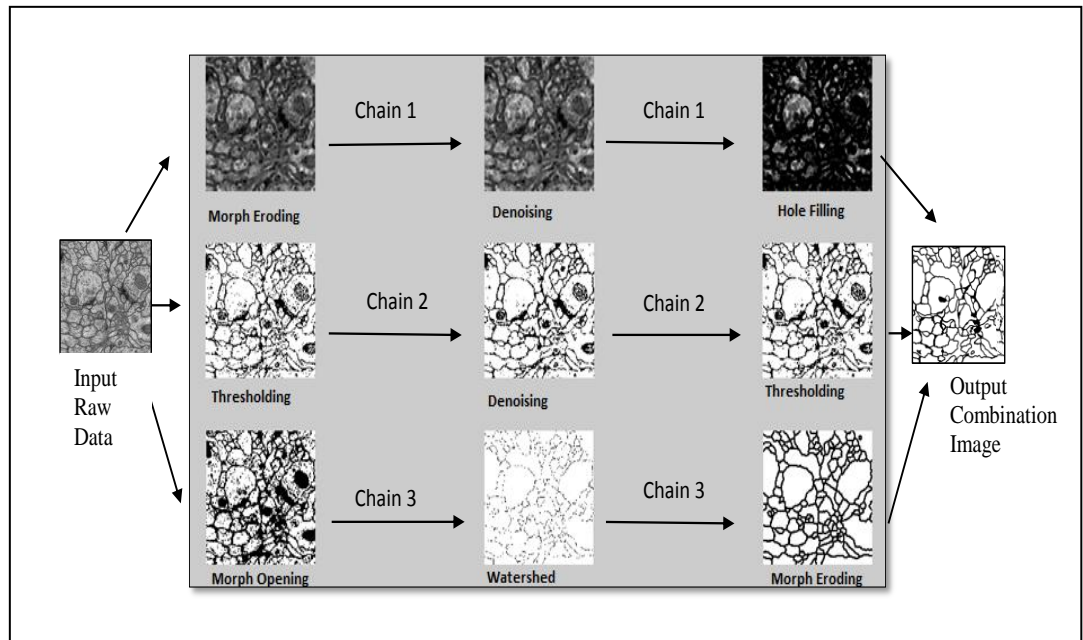
### **6.3.2 Interesting Observation: Morphological Operators**

An interesting observation pertaining to morphological functions is that morphological operators appeared in all of the best chains and in unorthodox positions. As is commonly known, one of the main purposes of morphological operators is to provide a smoothing effect, which typically occurs in a post-processing phase. In the experiments, it appears that although Morphological Operators are frequently encountered in the post-processing phase, they do also appear in various other positions in the MIPCO network (as seen in Table 6.1). Moreover, the appearance of these operators in typical positions does appear to contribute to better performance. Also note that morphological operators are not the only type of function found in post-processing smoothing; denoising functions have also been found in unorthodox positions, as reported by Raju Raju, R., Maul, T.H, & Bargiela, A.(2014, 2015). In general, optimisation often finds unexpected ways to use functions (*e.g.*, morphological operators have been found performing classification in some networks).

The results of experiments show that a Morphological Operator (MO) can appear in unorthodox chain positions. As can be seen in Figure 6.3, MIPCO experiments show that, at least for this membrane segmentation problem, MO appears in early (in Chain 1, as Morphological Erosion; in Chain 3, as Morphological Opening) and final stages (in Chain 3 as Morphological Erosion). As can be seen in Figure 6.3, the morphological operator erosion seems to appear early in the chain (as the first function), which arguably runs contrary to common expectations—that morphological operators are used typically for post-processing. The insight that morphological operators can often perform useful computations in an atypical position of image processing pipelines is a fact that needs to be taken into account by image processing users. In other words, one should not always restrict morphological operators to the final stages of pipelines. The experimental results show that utilisation of morphological operators in early stages can have a positive effect on accuracy. It has been discovered that the networks with morphological operators in the early or middle regions of pipelines do tend to show higher F1 scores. This can be seen in the following Table 6.2, which shows the position of morphological operators and their average scores.

The information flow in Figure 6.3 is as follows:

- i) The arrows below show the information flow. All chains have the same input.
- ii) At the bottom of Figure 6.3, there is no combination function to carry the combinations.
- iii) The final combinations are performed in averaging mode. The value of the output of every layer of Figure 6.3(a) is averaged to get the final output presented in Figure 6.3(b).



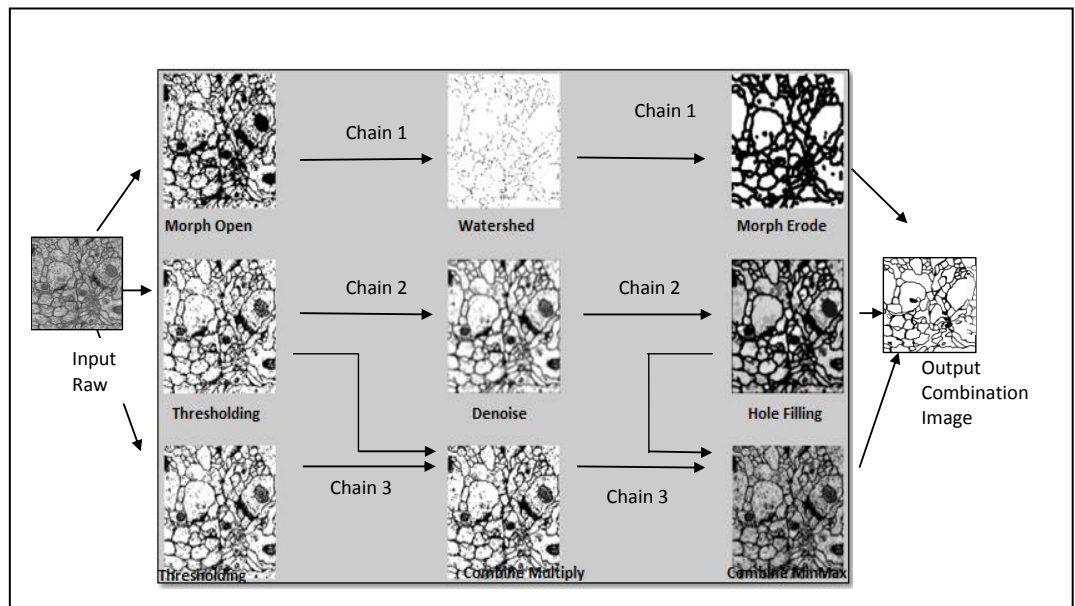
(b)

Figure 6.3: (a) MIPCO example output with three layers, flows and the selected functions, (b) Final output of the three chosen chains.

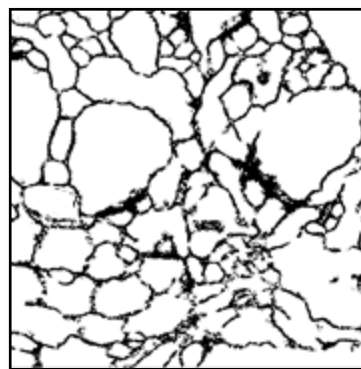
The information flow in Figure 6.4 is as follows:

- i) The arrows below show the information flow. All chains have the same input.
- ii) At the bottom of Figure 6.4, there are two combination functions: Combine-Multiply (Layer 3, Function 2) and Combine MinMax (Layer 3, Function 3). The arrow shows the choice of combination for the combining function.
- iii) The final combinations are performed in averaging mode. The value of the output of every layer of Figure 6.4(a) is averaged to get the final output presented in Figure 6.4(b).





(a)



(b)

Figure 6.4(a): Another example of MIPCO chosen functions, function flows for single, and combination function to perform its final output 6.4(b).

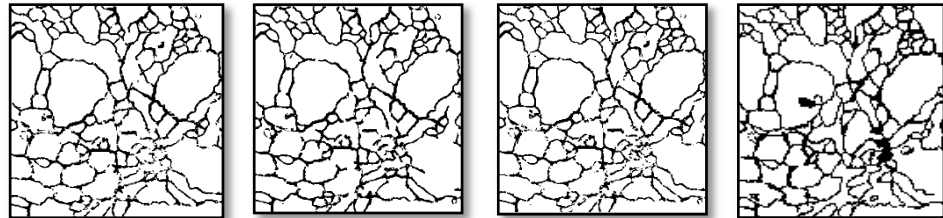
Table 6.2 shows the scores and chain positions for the ‘earliest morphological operators’, dividing the networks into three categories characterised by scores (>91%, between 90% to 91%, and <90%). These results depict networks that exhibited a maximum of three chains with a maximum of eight functions each.

Table 6.2: Appearance position of earliest morphological operators (row shows average scores and positions).

| Scores >91%           | Appearance Position of the Earliest Morphological Operator | Scores >90 but < 91%  | Appearance Position of Morphological Operator | Scores < 90%          | Appearance Position of Morphological Operator |
|-----------------------|--|-----------------------|---|-----------------------|---|
| 91.43                 | 1st  | 90.2                  | 5th   | 89.64                 | 6th   |
| 91.38                 | 1st  | 90.0                  | 4th   | 89.00                 | 7th   |
| 91.33                 | 1st  | 90.63                 | 3rd   | 88.87                 | 8th   |
| 91.23                 | 3rd  | 90.99                 | 3rd   | 88.22                 | 7th   |
| 91.21                 | 2nd  | 90.5                  | 4th   | 89.03                 | 6th   |
| 91.16                 | 2nd  | 90.01                 | 3rd   | 89.33                 | 5th   |
| 91.14                 | 1st  | 90.91                 | 4th   | 89.56                 | 6th   |
| 91.1                  | 2nd  | 90.03                 | 5th   | 89.23                 | 6th   |
| 91.09                 | 2nd  | 90.96                 | 4th   | 88.2                  | 8th   |
| 91.01                 | 2nd  | 90.72                 | 3rd   | 89.91                 | 5th   |
| Average F1 score 91.2 | Average Ranking 1.7  | Average F1 score 90.5 | Average Ranking 3.8                           | Average F1 score 89.1 | Average Ranking 6.4                           |

As shown in Table 6.2, 91.20% denotes the average accuracy of those chains that have at least one morphological operator (MO) at an early stage, 90.5% denotes the average accuracy of those chains that have at least one morphological operator (MO) at the middle stage, and 89.1% denotes the average accuracy of those chains that have at least one morphological operator (MO) at the final stage (the bottom row of the table corresponds to the average of the rows above). From Table 6.1, it can clearly be seen that having at least one MO at an early stage has a positive impact on performance, compared to having MOs at later stages.

**a) Visual Results using MIPCO network—Chains with morphological operators in unorthodox positions**



F1 score : 91.43% (1<sup>st</sup>, 7<sup>th</sup> and 9<sup>th</sup>)    F1 score : 91.38% (1<sup>st</sup>, 7<sup>th</sup> and 9<sup>th</sup>)    F1 score : 91.21% (4<sup>th</sup>, 6<sup>th</sup> and 8<sup>th</sup>)    Ground Truth  
 → Position of appearance

Figure 6.5: Example of final output images of network using MIPCO that have morphological operators in various positions (specified in the figure), in the front, middle, and end portions of the chains with F1 scores > 91% vs. the Ground-Truth image (rightmost).

Figure 6.5 depicts several sample output images using different MIPCO networks. As mentioned earlier, for the experiments, the algorithm was allowed to generously choose a maximum of eight functions and five chains. Consequently, there were 40 outputs ( $8 \times 5$ ) in the processing stages. Figure 6.5 shows the final output and the morphological operator appearance position. The figure shows the visual output of the network having Morphological Operator in various positions. Of the 40 repeatable functions, 10 are morphological operators which appear early and in the middle of the five chains.

**6.3.3 MIPCO Function Repetition**

The MIPCO implementation reported here consisted of a maximum of five chains, each with a maximum of eight functions. From the analysis, it was found that some functions repeat themselves in the same and neighbouring chains. Table 6.3 summarises these function repetitions. For example, for the length 2 case, thresholding exhibits a repetition of 4%, which means that 4% of chains (out of 50 trials (or optimal chains)) exhibit repetitions of the thresholding function. On closer inspection of the processing outputs of each

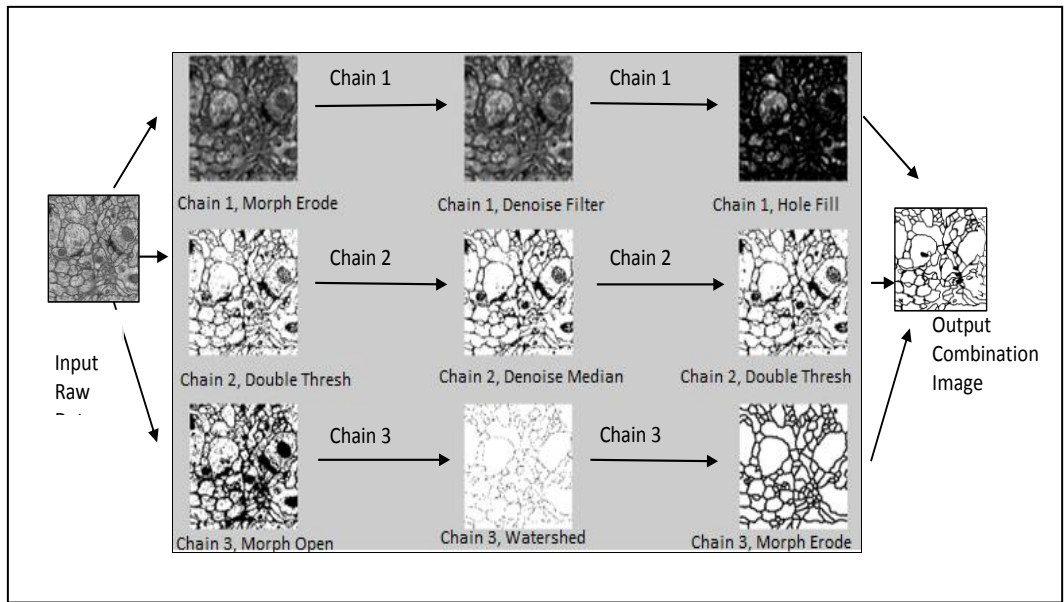
repeated function, it was confirmed that the outputs of repeated functions are indeed distinct from each other and therefore that the repetitions are performing useful computations and not just copying or relaying information.

Table 6.3: Function repetitions for 50 trials using MIPCO network.

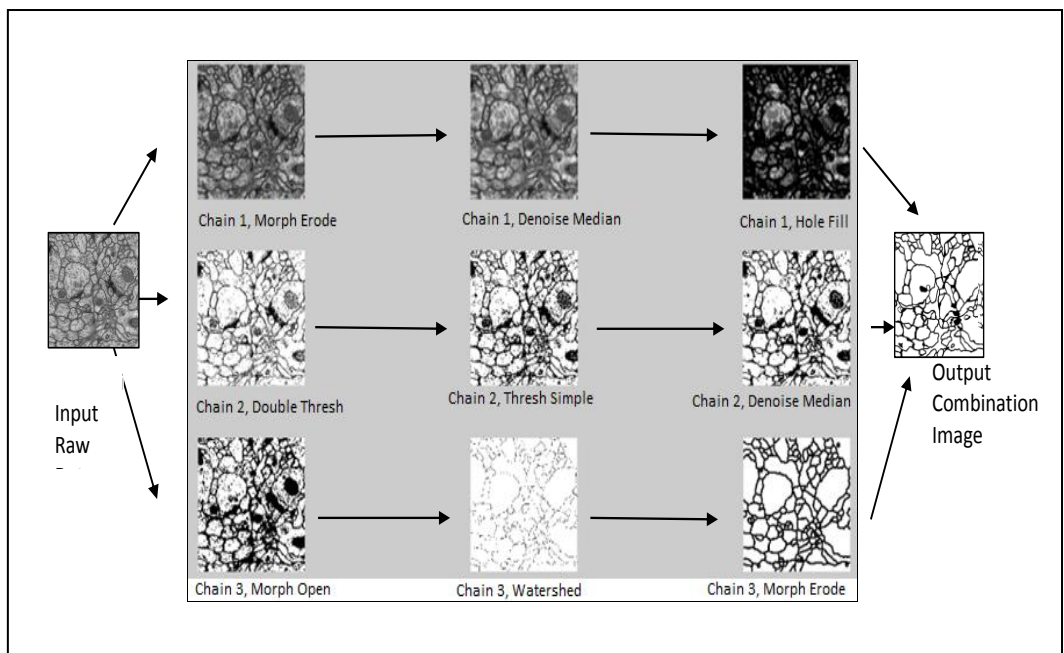
| Functions Supplied            | Func Allowed 1 | Func Allowed 2 | Func Allowed 3 | Func Allowed 4 | Func Allowed 5 | Func Allowed 6 | Func Allowed 7 | Func Allowed 8 |
|-------------------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Contrast Enhancement          | -              | -              | -              | -              | -              | 4%             | 20%            | 32%            |
| Denoising                     | -              | -              | -              | -              | 4%             | 12%            | 20%            | 32%            |
| Thresholding (Simple /Double) | 100%           | 4%             | 8%             | 8%             | 12%            | 20%            | 60%            | 92%            |
| Hole Filling                  | -              | -              | -              | 4%             | 8%             | 12%            | 24%            | 32%            |
| Watershed                     | -              | -              | -              | -              | 4%             | 8%             | 20%            | 28%            |
| Combination                   | -              | -              | -              | -              | 4%             | 4%             | 12%            | 12%            |
| Morphology                    | -              | -              | -              | 4%             | 12%            | 40%            | 48%            | 76%            |
| Edge                          | -              | -              | -              | -              | -              | -              | 8%             | 20%            |

Table 6.4: Examples of chains with function selection and repetition (repeatable functions are in bold).

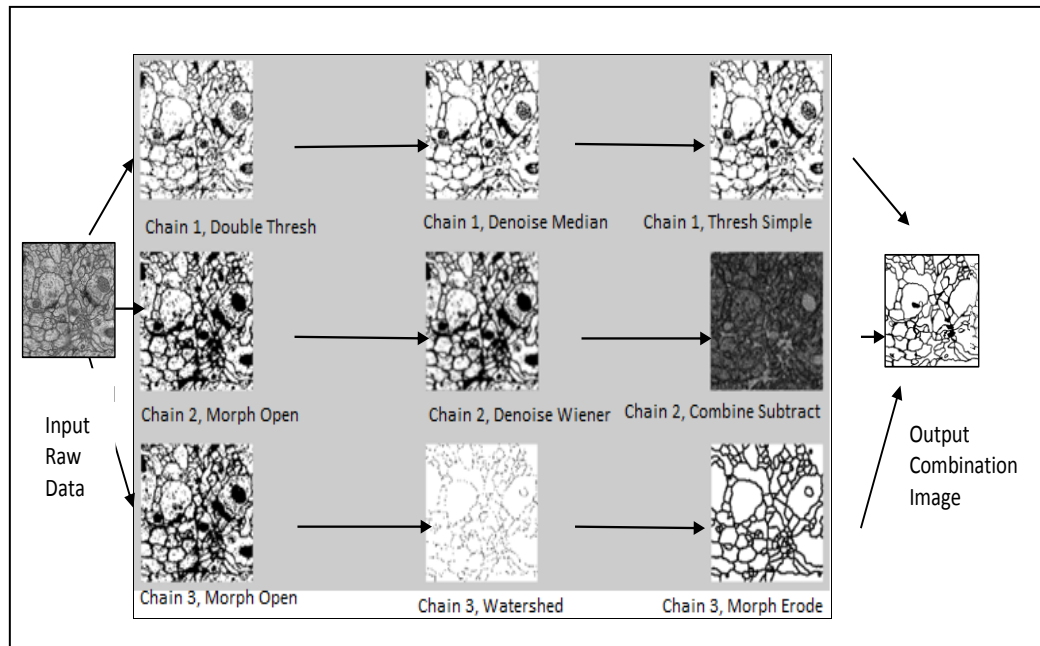
| Network Scores | Functions Used and Chains   |
|----------------|---|
| 91.43          | Chain 1 – <b>Morph Erode</b> , Denoise Filter, HoleFill               |
|                | Chain 2 – <b>Double Thresh</b> , Denoise Median, <b>Double Thresh</b> |
|                | Chain 3 – Morph Open, Watershed, <b>Morph Erode</b>                   |
| 91.37          | Chain 1 – <b>Morph Erode</b> , <b>Denoise Median</b> , HoleFill       |
|                | Chain 2 – Double Thresh, Thresh Simple, <b>Denoise Median</b>         |
|                | Chain 3 – Morph Open, Watershed, <b>Morph Erode</b>                   |
| 91.16          | Chain 1 – Double Thresh, Denoise Median, Thresh Simple                |
|                | Chain 2 – <b>Morph Open</b> , Denoise Wiener, Combine Subtract        |
|                | Chain 3 – <b>Morph Open</b> , Watershed, Morph Erode                  |



(a) Output Image for Network Scores of 91.43 (as per Table 6.4).



(b) Output Image for Network Scores of 91.37 (as per Table 6.4).



(c) Output Image for Network Scores of 91.16 (as per Table 6. 4).

Figure 6.6: (a), (b), (c) Examples of chains with function selection and repetition for visual inspection as per Table 6.4.

From the output images in Figure 6.6, it can clearly be seen that ‘combiners’ functions such as ‘combine subtract’, do not just copy images but actually perform useful combinations. It is also interesting (unusual) to see Morph Erode applied to a greyscale image, and a post-processing function such as a morphological operator appearing in an early stage.

### 6.3.4 Mandatory functions that always appear in chains

In the experimental analysis, it was discovered that there are sets of mandatory functions that always seem to appear together in chains. It was found that the functions listed below are very frequently selected. Thus, it is also believed that the selection of these functions contributes to overall better performance for the dataset used in this research. These combinations were identified and measured through the performance of their F1 scores.

- i) Morphological Operator: Opening
- ii) Watershed

It seems that at least for this membrane detection problem, all the ‘good chains’ selected watershed as one of the preferred functions, and this function seems to always appear together with its co-partner, namely, the morphological operator ‘open’.

- iii) Morphological Operator: Eroding
- iv) Denoising
- v) Thresholding

#### 6.4 Comparison between IPCO and MIPCO

Figure 6.7(a) shows that the single best algorithm for IPCO consists of seven functions with F1 score of 91.67%.

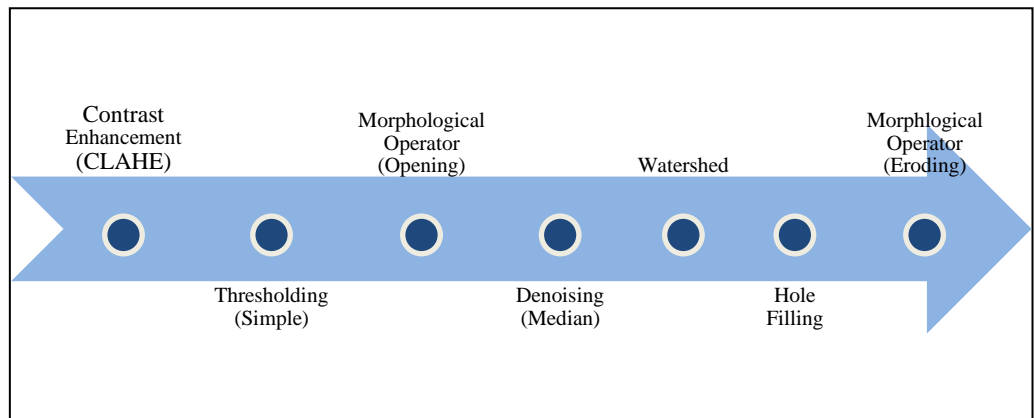


Figure 6.7(a): IPCO output function for single best chain.

|                |                |                |               |                |                |                |               |
|----------------|----------------|----------------|---------------|----------------|----------------|----------------|---------------|
| Denoise Median | Thresh Double  | Thresh Simple  | Morph Open    | Watershed      | Denoise Wiener | Thresh Double  | Morph Erode   |
| Denoise Median | Thresh Double  | Thresh Simple  | Morph Erode   | Holefill       | Combine Divide | Denoise Wiener | Morph Open    |
| Morph Open     | Holefill       | Watershed      | Morph Erode   | Thresh Double  | Watershed      | Holefill       | Watershed     |
| Morph Erode    | Thresh Double  | Morph Erode    | Holefill      | Combine Median | Combine Median | Morph Open     | Thresh Simple |
| Denoise Wiener | Denoise Median | Denoise Median | Thresh Simple | Watershed      | Morph Erode    | Watershed      | Watershed     |

Figure 6.7(b) shows that the best MIPCO network consists of eight functions and five chains with F1 score of 91.80%.

Figure 6.7(b): MIPCO network function and subtype for the best network (arranged in the same manner, five chains and eight functions).

**a) Which method achieves optimal rates?**

MIPCO achieves optimal rates; it scores higher than individual IPCO. MIPCO network scored 91.80% using maximum eight functions and five chains. There are four options for averaging the output for the final representation:

- mipco.combiner = 1 ( this is to average the final layer )
- mipco.combiner = 2 ( this is to average all layers)
- mipco.combiner = 3 (this is to apply ‘mode’ calculation to all layers)
- Finally, if there is no choice set, then the chain will by default conduct averaging according to assigned weight.

The ‘Combiner-Mode’ is the combiner selection for the chain with the highest recorded MIPCO score. Table 6.5 shows the list of functions that appear in scoring the best score of 91.80%. Figure 6.8 shows the output reflected in Table 6.5 for the best MIPCO network.



Table 6.5: MIPCO best chain and function occurrences (rearrange for better view).

| Function             | Repetition |
|----------------------|------------|
| Denoise Median       | 4 times    |
| Thresholding Double  | 5 times    |
| Morphology Erode     | 6 times    |
| Watershed            | 7 times    |
| Morphology Open      | 4 times    |
| HoleFill             | 4 times    |
| Denoise Wiener       | 3 times    |
| Thresholding Simple  | 4 times    |
| Combination Function | 3 times    |

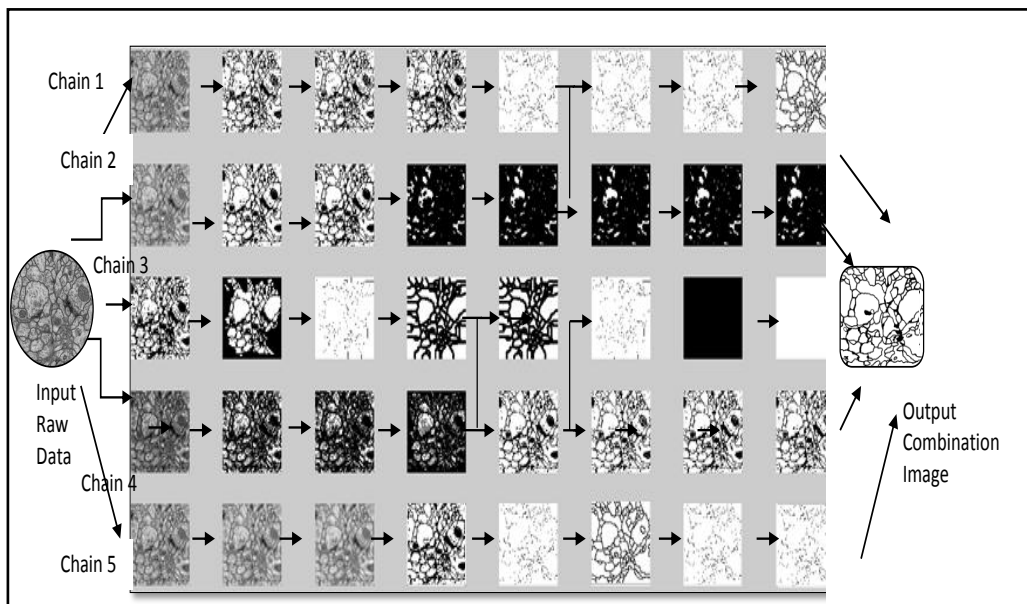


Figure 6.8 shows the output reflected in Table 6.5 for the best MIPCO network.

**b) Performance Score for Each Slice for the Best MIPCO network.**

Table 6.6: Performance score for each slice of the dataset for the score of 91.80%.

| <b>Image</b>   | <b>Precision</b> | <b>Recall</b>   | <b>F1 Score</b> |
|----------------|------------------|-----------------|-----------------|
| Slice 1        | 0.897774         | 0.952891        | 0.924512        |
| Slice 2        | 0.9011562        | 0.937588        | 0.919011        |
| Slice 3        | 0.8847477        | 0.954205        | 0.918165        |
| Slice 4        | 0.8965156        | 0.943468        | 0.919393        |
| Slice 5        | 0.8670974        | 0.929103        | 0.89703         |
| Slice 6        | 0.8487762        | 0.923174        | 0.884413        |
| Slice 7        | 0.8970235        | 0.937917        | 0.917014        |
| Slice 8        | 0.8722412        | 0.903588        | 0.887638        |
| Slice 9        | 0.841646         | 0.848421        | 0.84502         |
| Slice 10       | 0.9036902        | 0.961479        | 0.931689        |
| Slice 11       | 0.8730061        | 0.959321        | 0.914131        |
| Slice 12       | 0.8966041        | 0.929261        | 0.91264         |
| Slice 13       | 0.8702659        | 0.948021        | 0.907481        |
| Slice 14       | 0.9115428        | 0.940408        | 0.92575         |
| Slice 15       | 0.9328845        | 0.950661        | 0.941689        |
| Slice 16       | 0.9034125        | 0.96308         | 0.932293        |
| Slice 17       | 0.911486         | 0.934666        | 0.92293         |
| Slice 18       | 0.8847706        | 0.948909        | 0.915718        |
| Slice 19       | 0.8981885        | 0.954013        | 0.925259        |
| Slice 20       | 0.8881126        | 0.960401        | 0.922843        |
| Slice 21       | 0.8273795        | 0.969272        | 0.892723        |
| Slice 22       | 0.895814         | 0.970895        | 0.931845        |
| Slice 23       | 0.9044369        | 0.940577        | 0.922153        |
| Slice 24       | 0.9172298        | 0.946368        | 0.931571        |
| Slice 25       | 0.9234972        | 0.94879         | 0.935973        |
| Slice 26       | 0.8990611        | 0.967842        | 0.932184        |
| Slice 27       | 0.9039442        | 0.96301         | 0.932543        |
| Slice 28       | 0.8879244        | 0.96326         | 0.924059        |
| Slice 29       | 0.9118171        | 0.943527        | 0.927401        |
| Slice 30       | 0.9314036        | 0.956961        | 0.944009        |
| <b>Average</b> | <b>0.8927817</b> | <b>0.945036</b> | <b>0.917969</b> |

Table 6.6 shows the individual score; the Recall (how well the membrane voxels were detected by the classifier) has an average score of 94.50%. The score was not high for Precision, with just 89.28%. Precision considers the scores that provide confidence values for positive results. As true positive

(correctly labelled) and false positive (incorrectly labelled) approach zero, the precision approaches one. In the case where the denominator is zero,

- $TP + FN = 0$ : meaning that there were no positive cases in the input data
- $TP + FP = 0$ : meaning that all instances were predicted as negative

Because the actual breadth of the cell membrane in the gold standard is thick in comparison with the algorithm outputs, this may result in positional fluctuations and different classification in error measurement that provide the composite measure of successful membrane classification and successful exclusion of non-membrane voxels to differ in terms of categorisation in grouping them to True Positive (the number of pixels correctly labelled as belonging to the positive class) and False Positive (pixels that are falsely identified as a boundary in the output, but are classed as the cell interior pixels in the ground-truth image). This will affect the performance score of the chain.

## **6.5 Differences and Similarities (IPCO and MIPCO)**

Similarities:

- Neither function uses Edge Function for the Best Chain (Score).
- Denoise Median is more favoured than other filters

Differences:

- Because it only optimises a single chain, the time taken for IPCO optimisation is lower than that of MIPCO. MIPCO is a network with parallel interactions; as a result, processing time is prolonged. However, even when MIPCO is lower in terms of time (speed) than IPCO, it still manages to optimise the network in seconds manner per image, and in minutes manner for the whole network to be optimised (*e.g.*, 9 minutes for the best MIPCO network to optimise).
- Combination function is not a choice in IPCO's highest chain.

Mandatory functions in ‘shortest chain’ with F1 scores >91%.

**6.5.1 For IPCO chain:**

- Thresholding
- Denoising
- Contrast Enhancement
- Hole-Filling

**6.5.2 For MIPCO network:**

- Thresholding
- Denoising
- MorphOpen
- Watershed
- MorphErode

This shows that only Thresholding and Denoising appear as favourite functions of chains in IPCO chains and MIPCO networks.

IPCO’s ‘longest chain’ (with eight functions) and MIPCO’s ‘largest network’ (with five chains and eight functions)

The longest chain for IPCO and largest network for MIPCO differ from each other.

IPCO’s longest chains have the following characteristics:

- They appear without Hole-Filling, Watershed, and Morphology Operator.
- They are also functions with no Thresholding (about 4% of tested experiments out of 50 trials) and Denoising (48% of tested experiments out of 50 trials) in the chain as shown in Table 6.7

Table 6.7: Denoising and Thresholding Function Appearance Frequency (reproduced from Chapter 5, Table 5.4).

| Functions                    | Chain with 1 Func | Chain with 2 Func | Chain with 3 Func | Chain with 4 Func | Chain with 5 Func | Chain with 6 Func | Chain with 7 Func | Chain with 8 Func |
|------------------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| Denoising                    | 0%                | 52%               | 16%               | 48%               | 48%               | 52%               | 52%               | 52%               |
| Thresholding (Simple/Double) | 100%              | 96%               | 72%               | 76%               | 92%               | 92%               | 92%               | 96%               |

- However, the Combination function differs, as it seems to appear in all ‘longest chains’ (as can be seen in Table 6.9), but the choice of Combination function (MinMax, Average, and Multiply) differs from one chain to another.

Table 6.8: Combination Function Appearance Frequency (reproduced from Chapter 5, Table 5.4).

| Functions                               | Chain with 1 Func | Chain with 2 Func | Chain with 3 Func | Chain with 4 Func | Chain with 5 Func | Chain with 6 Func | Chain with 7 Func | Chain with 8 Func |
|---|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| Combination (MinMax, Average, Multiply) | 0%                | 0%                | 4%                | 32%               | 32%               | 60%               | 92%               | 100%              |

MIPCO’s longest networks have the following characteristics:

- There is no single function that scores 100% when eight functions are allowed in a chain (please see Table 6.3 above).
- However, thresholding still results in the highest scoring function appearing in chains in a network, but they are also layers of chains that do not collaborate with the thresholding function (about 8% out of 50 trials) in the longest network (which allows eight functions). Table 6.9 shows the percentage score.

- The Combination function differs from IPCO because this function is not favoured in MIPCO network, which only shows six times of out of 50 trials and sits in the last position in the table score.

Table 6.9: Thresholding and Combination Functions Appearance Frequency for MIPCO (reproduced from Table 6.3).

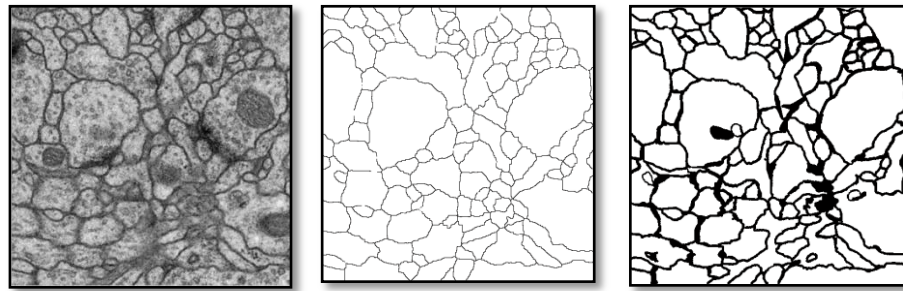
| Function<br>s<br>Supplied                | Func<br>Allo<br>w-ed<br>1 | Func<br>Allow<br>-ed<br>2 | Func<br>Allow<br>-ed<br>3 | Func<br>Allow<br>-ed<br>4 | Func<br>Allow<br>-ed<br>5 | Func<br>Allow<br>-ed<br>6 | Func<br>Allow<br>-ed<br>7 | Func<br>Allow<br>-ed<br>8 |
|--|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|
| Thres-<br>holding<br>(Simple<br>/Double) | 100<br>%                  | 4%                        | 8%                        | 8%                        | 12%                       | 20%                       | 60%                       | 92%                       |
| Combi-<br>nation                         | -                         | -                         | -                         | -                         | 4%                        | 4%                        | 12%                       | 12%                       |

Figure 6.9 shows an example of MIPCO output with a score of only 88.96%. Although, from a visual perspective one can see a clean output without any organelles and clear detection of membrane lines, it only recorded a score of 88.96%, which is about 2.84% less than the best recorded MIPCO score (at the time of writing). Comparing the output image with the benchmark ground truth, it can be seen that the ground-truth image has thicker membranes than the output image using MIPCO. This can cause a drop in the score calculation because the pixel intensity value will differ because of the thickness of the membrane (Repairs being done using Morphological Operators (MO) parameterizations, but yet it do not reach the right amount of thickening).

Figure 6.10 gives a visual representation of MIPCO network best score output versus the ground truth. There is approximately a difference of 8% to reach 100%. Although the output visual appears similar to the ground-truth image, because of the membrane thickness, a score of 100% was not achieved.

Note that the score one achieves is highly dependent on the manual reference segmentation images drawn by experts, called the gold standard or ground truth. Even a similar output does not guarantee a score of 100% because of

variants in the thickness of the displayed membrane, even when the complete membrane lines are correctly identified.



Input Image

Output using MIPCO

Groundtruth

Figure 6.9: Another visual representation of MIPCO output versus Ground truth for score of 88.96%.



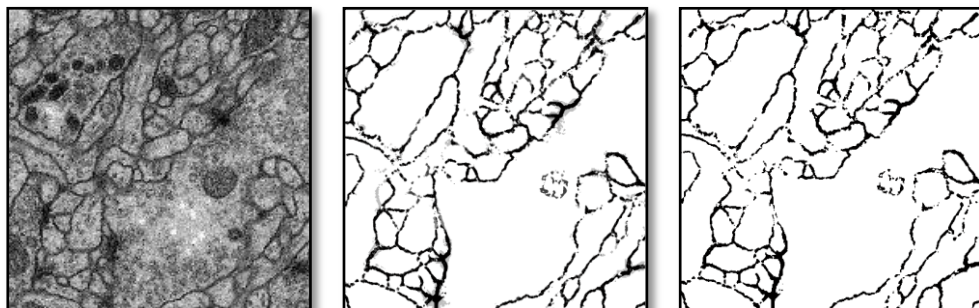
Input Image

Output using MIPCO

Groundtruth

Figure 6.10: Visual representation of example of MIPCO best score output versus Ground truth for score with 8% difference to reach 100%.

## 6.6 MIPCO Ensembles Output vs. Other Methods



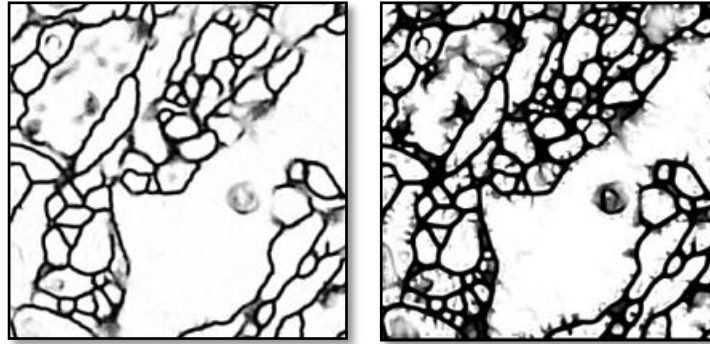
Original ISBI Test Image

Average Out

Classify Out using MIPCO

(a)

(b)



(c)

(d)

(c) Multi-scale contextual membrane detection, (d) initial Partial Differential Equation (PDE) result prior to thresholding (Tasdizen *et al.*, 2014)

Figure 6.11: MIPCO Ensembles vs. Other Methods.

Figure 6.11(a) shows the average output retrieved by combining all the outputs and then averaging it and the classifying output (6.11(b)) retrieved using a set threshold. The test was run using slice 2 of the Test Image provided by Cardona *et al.* Figures 6.11(c) and 6.11(d) show the output using other comparison methods: multi-scale contextual using a series of Artificial Neural Networks (ANNs) and Partial Differential Equation. As with the compared methods, MIPCO shows clean image with major detection of membranes and elimination of organelles.

MIPCO Ensemble—The Highest Performance Score (as of time of writing):

- ✓ **Highest Average of F1 score (Ensembles): 92.63%.**

## 6.7 Conclusion

From the experiments conducted, and given the specific membrane detection dataset adopted, it was found that the optimisation of image processing chains, when using multiple chains (MIPCO) network is generally more accurate than when using single chains (IPCO). In terms of speed, as expected, because of its larger size, and assuming a non-parallelised solution, MIPCO does perform worse (*i.e.*, approximately 10 seconds longer per optimisation epoch). However, MIPCO is still easier and faster to train than many other machine



learning approaches. Apart from sharing additional advantages with IPCO such as interpretability and retrainability, MIPCO has additional advantages such as parallelisability and the provisioning of complex interactions between chains, which opens up new opportunities for problem decomposition and solution composition. It is believed that this integrative capability of MIPCO is what allows it to perform better than individual IPCO chains.

### **Novelty of MIPCO**

MIPCO uses existing image processing functions, but optimises the way in which they are configured and combined. It uses a larger set of functions and the combination framework is less rigid in structure, and provides reordering flexibility with no ordering constraints. This process revealed several interesting and unconventional insights pertaining to morphological operators, and the appearance of functions in unorthodox positions and repetition of functions that perform useful computations that contribute to better performance. Therefore, MIPCO challenges the existing image processing pipelines. These algorithms also capable to have single and multiple input functions such as ‘image blending’, and it uses special purpose ‘combiner functions’ specifically designed to encourage chains to form different representations and transformations.

MIPCO research is in its infancy; more improvement and work needs to be done to further enhance the capability of the algorithm. In the carried experiments, MIPCO networks were allowed to use a maximum of eight functions per chain, and five parallel chains. This research has also published work related to MIPCO.

## CHAPTER 7

### DISCUSSION

The focus of this research is the problem of neuronal membrane detection in which the core challenge consists of distinguishing membranes from organelles. The aim/goal is to propose a speedy and highly accurate algorithm that can detect membranes and eliminate organelles at low cost (hardware) and is easy to adopt by new researchers (who are not computer scientists) in the area of Image Segmentation and Classification. The literature shows that the series of functions and methods (*e.g.*, use of genetic algorithm) have been adopted in this research area, but there are limitations on the capabilities of these previously suggested methods. For example, whereas the method proposed in this research has reordering flexibilities of the functions, in other methods there is a need to determine the sequences of filters for adequate transformation or there are limitations on speed. Moreover, from the literature reviewed, many developments and contributions have been made in the area of interest. In comparison with work by other researchers in this area, the proposed algorithms differ in various ways in using existing functions by optimising the manner in which they are configured and combined. This research is not about introducing new functions for image segmentation, but rather about manipulating existing functions to get maximum results. The designed framework encourages chains to form different representations and transformations, which suggest a new set of pipelines, with minimum software and hardware requirements. Hardware and time constraints should be considered when conducting research. This research outcome is both hardware and time friendly.

The research results are useful for new researchers, especially non-computer scientists, and contribute to the advancement of knowledge, with many interesting findings that suggest a set of pipelines for image segmentation. Moreover, the framework of algorithms adapts simple segmentation functions,

as the objective of the research is to use several basic processing steps to propose a simple and efficient approach.

## **7.1 Research Outcome**

This study proposed three algorithms: Local Contrast Hole-Filling (LCHF), Image Processing Chain Optimisation (IPCO) chain, and Multiple Image Processing Chain Optimisation (MIPCO) network. The first algorithm, LCHF, precipitated the creation of the second automated algorithm, the IPCO chain, and IPCO precipitated the creation of the third automated algorithm, the MIPCO network. The major hypothesis tested was determination of whether an adaptation and combination of simple image processing functions, with less hardware requirement can speedily and accurately detect membranes and eliminate organelles. The result according to the calculation of the error measurement score supports the findings, in which the proposed algorithms successfully detected neuronal membranes and eliminated organelles, using simple combinations of image processing functions, with high accuracy and speed, and low hardware requirements at low cost. The algorithm has also been tested using a lower specification personal computer with 1.60 GHz CPU and 1.48 GB of RAM. The algorithms differ in terms of its score (error measurement). The score improves in order of the first to the third proposed algorithms, and the output results are satisfied.

## **7.2 Implication of Findings**

### **How the results support the targeted general activity of the research**

A summary of the contributions of the research with general target research activity is given in next page.

Table 7.1: Contribution of the research with general target research activity.

| Category               | Typical Activity  | Contribution of the research  |
|------------------------|---|---|
| Problem Identification | <p><sup>1</sup>Identify and specify a novel problem.</p> <p><sup>2</sup>Standard segmentation algorithms tend to over- or under-segment microscopic images of neuronal membranes, mainly because of the similarity between membrane and non-membrane (<i>e.g.</i>, organelles) materials.</p> <p><sup>3</sup>Sample-based training approaches are generally difficult and time-consuming, partly because a sufficiently large number of labelled training samples need to be provided in order to achieve a desirable outcome.</p> <p><sup>4</sup>In order to solve the task, there are often requirements for specialised hardware, initialisation and calibration procedures, prior knowledge of the medical domain under consideration, advanced programming skills, <i>etc.</i></p> | <p><sup>1</sup>Achieved</p> <p><sup>2</sup>A maximum F1 score of 92.63% was recorded (using ensembles).</p> <p><sup>3</sup>Easy to adapt, speedy, low time-consumption.</p> <p><sup>4</sup>No requirements for specialised hardware, no initialisation and calibration procedures.</p> <p><sup>4</sup>No prior knowledge of medical domain needed.</p> <p><sup>4</sup>Moderate level of programming skills is sufficient.</p> |
| Design                 | Design novelty  | The novelty of the neuronal membrane detection algorithm lies in its optimisation, type of dataset used, new set of chains found, and new set of pipelines suggested. The work in this research differs from other compared studies in terms of the set of functions used, the parameterisations allowed, the optimisation methods adopted, the combination framework, and the testing and analyses conducted.                |

Continue...

continued...

|                    |   |   |
|--------------------|---|---|
| Framework          | Implement the framework for the first time  | The framework included a new category of special-purpose ‘combiner’ functions specifically designed to encourage chains to form various representations and transformations. This is the first time this approach has been applied in the context of the <i>Drosophila</i> first instar larva ventral nerve cord. |
| Comparison         | Compare several models, designs, and frameworks or implementations in a novel way | The performance of the algorithm was compared to that of participants of the International Biomedical Symposium who used the same dataset.  |
| Empirical Analysis | Study the performance of an implemented approach in a novel way                   | The algorithm was sent for benchmark testing using an unpublished dataset, and achieved a Rand score of 90% on unseen test dataset.   |

### 7.3 Comparison with existing knowledge (literature)

The focus of this research is the problem of neuronal membrane detection in which the core challenge consists of distinguishing membranes from organelles. Tasdizen *et al.* (2014) used the same *Drosophila* dataset with image segmentation functions and collaborating Partial Differential Equation (PDE) with watershed merge tree to achieve a score of 89.74% (0.1026) using the ISBI test dataset. On the same dataset, IPCO scored 89.90% (0.1010), which is slightly better by 0.16%. In comparison with the ISBI competitors, the ISBI 2012 winner Ciresan, Giusti, Gambardella, & Schmidhuber (2012), adopted the Deep Neural Network (DNN), which is an early Artificial Neural Network idea that gained popularity around 2006. The study revealed that this method has issues in that discrimination is difficult, it does not learn to sequentially attend to the most informative parts of objects, it is weak in handling perceptual invariances, *etc.* It has also been published that DNN is

also slow to train (days) and test (hours) and requires specialised hardware for the whole dataset. Therefore, it is much more difficult to apply in real-world scenarios. IPCO and MIPCO are fast in both training and testing, with no specialised hardware requirement. The need for long hours of training and specialised hardware can be seen to counterbalance the advantage of the method.

Burget, Uher, & Masek (2012), a listed participant of ISBI 2012, succeeded in removing small objects, but they failed to remove some large objects because the objects are connected to the membrane. IPCO and MIPCO succeed in removing both smaller and larger objects. According to Burget, Uher, & Masek, (2012), their method cannot connect broken lines and other promising enhancements needed to reconnect the broken (membrane) lines. They also suggest that an extended set for better feature extraction to give better results for pixel error criteria is needed. Seyedhosseini *et al.* (2012) use Contextual Hierarchical Model (CHM) for scene labelling. CHM only uses patch information and needs to learn hundreds of parameters. According to researchers, CHM can be prone to error owing to absence of any global constraints. They suggest that some other post-processing should accompany CHM to enforce consistency and global constraints. Furthermore, according to Seyedhosseini *et al.* (2012) the CHM needs 30 hours of training time on the CPU. In contrast, IPCO and MIPCO are fast in both the training and testing phases, and do not need a huge number of parameters. Moreover, the functions chosen in IPCO and MIPCO do collaborate with post-processing functions.

Researchers such as Qi (2005), Iftikhar & Godil (2012), Tan & Sun (2012) use Support Vector Machine as a classifier. Lucchi *et al.* (2010, 2012) used SVM to segment mitochondria. However, according to Burges (1998), SVM has limitations in terms of speed, size of the training and testing data, test phase (slow), choice of appropriate kernel, selection of the kernel function parameters, algorithmic complexity (high), and extensive memory requirements for large-scale tasks. IPCO and MIPCO have no problems with speed, choice of kernels, or algorithmic complexity. Furthermore, there is no special hardware initialisation or calibration needed for IPCO and MIPCO to

execute, and memory requirement is low. The limitations of some of the compared methods show that IPCO and MIPCO can be favoured for their ‘good job’ in detecting membranes and eliminating organelles accurately and speedily at low cost (hardware) and for adoption simplicity.

## **7.4 The Three Proposed Algorithms for Membrane Detection**

### **7.4.1 LCHF (1<sup>st</sup>) Algorithm**

As stated above, this research contributed three algorithms. The first algorithm, the Local Contrast Hole-Filling (LCHF) algorithm, is the initial algorithm created. It recorded an F1 score of 71% for detecting membranes in 30 slices in just 44.42 seconds, each with a resolution of  $343 \times 343$  pixels on an average personal computer (*i.e.*, 1.60 GHz processor and 1.48 GB of RAM). This algorithm adequately detected membranes and eliminated organelles at a fast speed, as reported by Rajeswari *et al.* (IEEE ISCAIE 2014). Detection of 30 slices of membrane in 21 seconds equates to less than 1.5 second for each slice. This fast process happened because LCHF does not need training except for some parameter tuning. LCHF is a non-learning approach, which is an advantage in the initial state because it is difficult to obtain a representative training set. Further, for LCHF, the gold standards are used only for the purpose of error measurement against the benchmark data to record the score. Thus, in a scenario where there is no availability of training sets and benchmarking datasets, algorithms such as LCHF are favoured because they can still produce results. Based on the hardware constraints of today, training classifiers with a very large number of free parameters can require weeks of computation, even when using high performance machines with high data transfer rates. This involves significant monetary and energy costs. Thus, the LCHF kind of algorithm can be useful for small-scope researchers. The LCHF algorithm also did a good job in detecting membranes and boundaries with other types of datasets. LCHF is currently being tested with both medical and non-medical images and non-TEM images.

LCHF has a speed advantage but its overall accuracy has not yet reached state-of-the-art levels. Its score is only 71%, but the visual output of the algorithm showed that the algorithm output did a good job in detecting membrane lines and eliminating organelles.

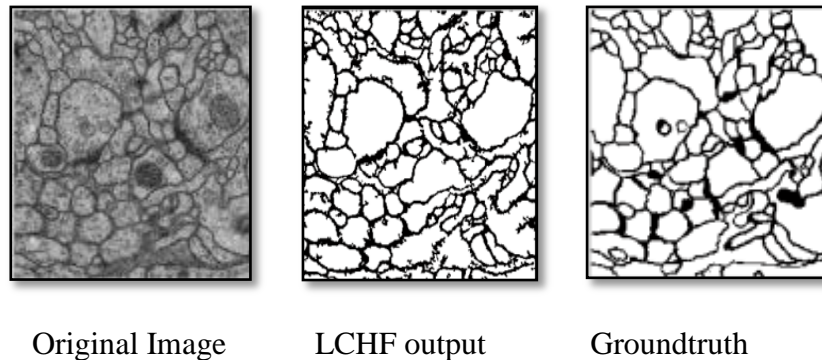


Figure 7.1: LCHF Output.

In essence, the low scoring is as a result of the thickness of the detected membrane pixel. The LCHF output images show many ‘squiggly lines’ jutting out from the membrane, which makes the membrane lines look thicker in comparison with its moderate thickness of membrane lines in the benchmark image. This may make the pixels fall under incorrect classification, resulting in the low F1 score, although the algorithm detected most membranes and eliminated most organelles. It is suspected that the preprocessing of the images with contrast enhancement which divides the images into small tiles contributes to this scenario. A test being done without incorporating the function, but the detection of the membrane lines, has poorer results, thus it is better to include contrast enhancement than to ignore it. At this stage of the research, post-processing functions are not included although the belief is that it can help to solve the ‘squiggly lines’ problem. However, as the research continued to its second phase of automated algorithm generation, the ‘squiggly lines’ problem was taken care of at the 2<sup>nd</sup> stage algorithm, called the Image Processing Chain Optimisation (IPCO) algorithm.



#### 7.4.2 IPCO chain (Automated Algorithm)

The second (automated) algorithm, IPCO, detects membranes and eliminates organelles with a recorded performance value score of 91.67% for the test set. This is an approximate 20.67% increase in score in comparison with the firststage LCHF algorithm. The algorithm took only 10 seconds per image, which equates to 280 seconds per chain, which is approximately less than 5 minutes for typically less than 500 optimisation generations, for the recorded highest IPCO score. The IPCO framework adopts a hybrid global stochastic optimisation method, with a combination of Genetic Algorithm, Differential Evolution, and Rank-Based Uniform Crossover (RBUC). Moreover, a minimum requirement in hardware needs enhances the capability of IPCO. A small scale researcher with a small amount of capital can adapt IPCO because a personal computer with its minimum requirement is sufficient to run IPCO. Hardware constraint because of cost constraint is not an issue for IPCO.

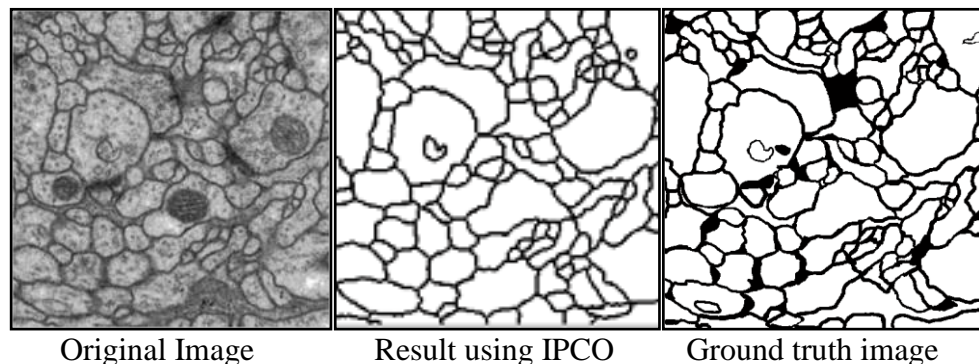
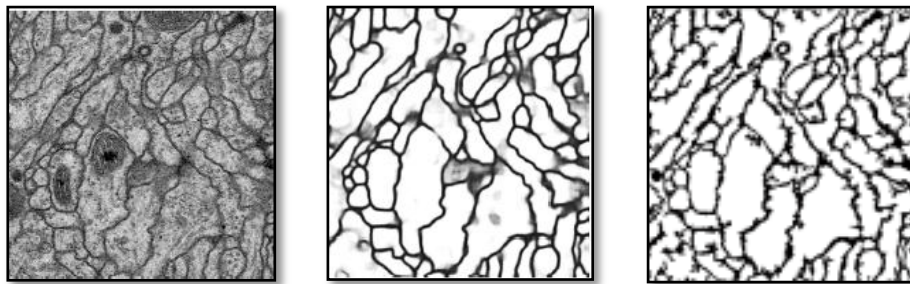


Figure 7.2: IPCO Output.

As can be seen in the above output image, the missing 8.33% score is most probably due to missing black patches (as can be seen in the ground truth image, in comparison with the IPCO output) and some extra lines possibly due to the watershed function as well as non-identical thickness of the lines. The highest recorded score using the ISBI test dataset, as published in the international competition was 94%. However, the algorithm reportedly required long hours of training, which leads to a total training time of several days with high-end hardware requirement. Even after training the network, 0.5

hour was required on 4 GPUs to conduct the testing of the entire stack of the dataset. The IPCO algorithm scored 90% on the unseen test dataset. This is about 4% below this score. Using the *Drosophila* testing dataset, IPCO recorded less than 5 minutes to present its best output for the whole stack of the dataset, with no special hardware initialisation.

Figure 7.3 shows the ISBI Test dataset with the international competition winner's output. For comparison purposes, the LCHF output is also shown. Because the ground truth image is not available for the ISBI Test dataset, the IPCO algorithm could not be tested with this ISBI test dataset on its own (the organiser only revealed the score and did not present the algorithm output visually), so they are not available for visual comparison. However, as the algorithm was submitted for benchmarking purposes to the competition, the revealed score can be used as a comparison.



ISBI Test Slice 1 Competition Winner's Output<sup>2</sup> Output using LCHF  
<sup>2</sup>reproduced with permission from Ciresan, Giusti, Gambardella, & Schmidhuber(2012) (winner of the ISBI 2012 challenge).

Figure 7.3: LCHF output using ISBI Test Slice.

IPCO implies simplicity and efficiency of simple sequences of image processing functions and involves automated fine-tuning of an algorithm relative to some dataset. IPCO met the stated goals and objectives, which are listed as follows:

- Relatively fast and consistent optimisation process
- Does not require specialised hardware
- Low hardware cost (no specialised hardware)
- Fast in training and classification
- Easy to use and deploy
- High accuracy
- Can distinguish membranes and organelles and also remove internal structures

The novelty of the neuronal membrane detection algorithm lies in its optimisation, the type of dataset used, and the new set of chains found. As stated earlier, the work in this research at this stage differs from other compared studies (Chapter 2) in terms of the set of functions used, the parameterisations allowed, the optimisation methods adopted, the combination framework, and the testing and analyses conducted. A new category of special-purpose ‘combiner’ functions are included in IPCO, in comparison with the previous LCHF algorithm. It is specifically designed to encourage chains to form various representations and transformations. As mentioned earlier, IPCO adopts a hybrid global stochastic optimisation method. Moreover, systematic analyses of the statistics of optimised chains conducted revealed several interesting and unconventional insights pertaining to preprocessing, classification, post-processing, and speed. The types of analyses conducted are novel and revealed interesting insights pertaining to denoising and its appearance in unorthodox positions in image processing pipelines, as reported by Raju, R., Maul, T.H, & Bargiela, A.(2014, 2015).

IPCO’s outstanding performance in both average and speed boosted the research capability to experiment further. This encouraged introduction of parallel processing which is the next stage algorithm created in this research, called the Multiple Image Processing Chain Optimisation Network (MIPCO) algorithm.

### 7.4.3 MIPCO network (Automated Algorithm, Parallel Network)

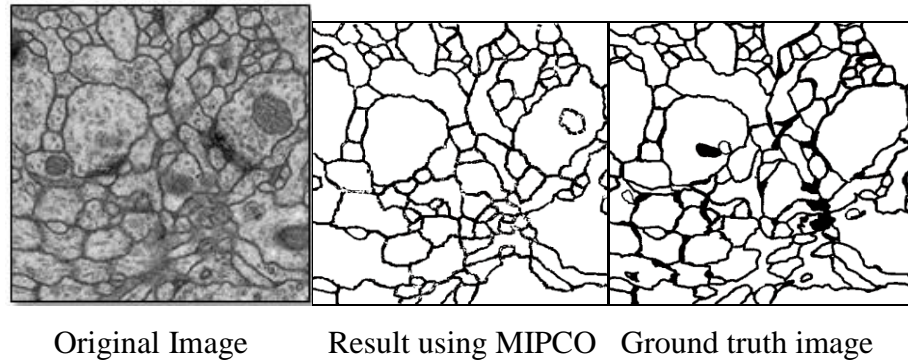


Figure 7.4: MIPCO Output.

MIPCO is the result of efforts to further boost the performance of IPCO. MIPCO is referred to as network optimisation and executes in an automated parallel manner by adopting a hybrid global stochastic optimisation method, with the combination of Genetic Algorithm, Differential Evolution, and Rank-Based Uniform Crossover (RBUC) for further enhanced performance. MIPCO computes layer by layer and there is no dependency of functions in the same layer. Functions in a layer can receive input from any other function in previous layers. MIPCO also detects membranes and eliminates organelles with a recorded individual average performance score of 91.80% for the test set, which is slightly higher than the average performance of the previous method, IPCO. MIPCO took 20 seconds per image, which equates to approximately 9 minutes for the network for typically less than 500 optimisation generations. This score satisfies the main aim of the research by detecting membranes and eliminating organelles with high accuracy and high speed. MIPCO is both efficient and interpretable, and facilitates the generation of new insights. There is no specialised hardware needed, and MIPCO leads to a network consisting of short sequences of basic processing steps which are efficient and easy to interpret. In addition to boosting the performance score, MIPCO also reveals some interesting observations pertaining to morphological operators and mandatory and repeatable functions which are elaborated further in the next section.

To add further to the advantages, IPCO and MIPCO can still be used without ground truth. They can be tweaked by a subjective notion of what a good segmentation is, and one can create a cost function for IPCO and MIPCO which measures the ‘general goodness’ of a segmentation based on concepts such as entropy.

## **7.5 Suggestion of new findings**

In the past, image processing categorised sets of functions that belong to preprocessing group, classification group, and post-processing group. This research revealed many interesting results showing that preprocessing functions which always come early in the segmentation process, appear in the middle or late in the process. The Denoising function is proof of this interesting observation. As is well-known, the main purpose of denoising is to filter out image noise in order to minimise detrimental effects in subsequent processing. Denoising is mostly carried out as an early preprocessing stage before application of other core functions, such as contrast enhancement and classification. However, in this research, the experiments conducted revealed many interesting findings: at least for this membrane segmentation problem, denoising typically appears later in the chain and the contrast enhancement function appear before the denoising function. Generally, the contrast enhancement function is used to enhance and preserve image information. The enhancement not only enhances the signal but also the noise, which makes subsequent denoising and classification harder. This is why this function is generally not favoured before filtering the unwanted noises. However, in this research, it was observed this was not the case. This is the interesting finding of the research that runs contrary to the general belief. It was discovered that by preserving the information at the early stage and enhancing the information, this information may be used in the next stage of the process. Further, experiments show that this organisation of functions (enhance then filter) does contribute to a higher score for the chain. For this dataset, the appearance of contrast in the early stage and denoising in the later stage suggests that details need to be enhanced before being cleaned, which can be encapsulated by the heuristic *enhance it before you lose it*.

In this research also, other observations revealed that there are some mandatory functions that seem to always appear in all processing chains. These mandatory functions that always appear are Thresholding, Contrast Enhancement, and Hole-Filling. Among these functions, the contrast enhancement function seems to be the first choice in most of the output chains, and appears in the early stage of the chain to preserve and enhance the information before losing it through filtering action. As for smoothing and better output appearance, the morphological operator and watershed seems to play a role. However, for this dataset, the thin line affects the F1 score detrimentally, because the compared gold-standard membrane line is slightly thicker. The appearance of the functions in the chain in the experiments conducted shows that 100% of the 'good' chains (F1 score more than 91%) adopted all the main components of IPCO (*i.e.*, contrast enhancement, thresholding, and hole-filling), 90% of chains adopted denoising as one of their components, and 50% of chains preferred to include the watershed function. The combination function appeared mostly in chains with more functions (maximum supplied with eight functions), and appeared less in 'short chains' (three to four function chains).

Another interesting observation pertaining to morphological functions is the appearance of morphological operators in all of the best chains and in unorthodox positions. As is commonly known, one of the main purposes of morphological operators is to provide a smoothing effect, which typically occurs in the post-processing phase. In the experiments carried out in this research, as per the IPCO algorithm's output observation, although the Morphological Operators are frequently encountered at the post-processing phase, they do also appear in various other positions in the MIPCO network. Moreover, the appearance of this operator in atypical positions does seem to contribute to better performance. In addition, note that morphological operators are not the only type of functions to be found in post-processing smoothing. In general, optimisation often finds unexpected ways to use functions (*e.g.*, morphological operators have been found performing classification in some networks). Experiments also show that Morphological Operators (MOs) can appear in unorthodox chain positions (early, middle) which arguably runs contrary to common expectations that morphological operators are typically

used for post-processing. The insight that morphological operators can often perform useful computations in an atypical position of image processing pipelines is a fact that needs to be taken into account by image processing users. In other words, one should not always restrict morphological operators to the final stages of pipelines. From the experiments, the utilisation of morphological operators in early stages can have a positive effect on accuracy. It was discovered that networks with morphological operators at the early or middle regions of pipelines do tend to show higher F1 scores. A score of 91.20% denotes the average accuracy of those chains that have at least one morphological operator at an early stage, 90.5% denotes the average accuracy of those chains that have at least one morphological operator at the middle stage, and 89.1% denotes the average accuracy of those chains that have at least one morphological operator at the final stage. It can clearly be seen that having at least one MO at an early stage has a positive impact on performance, compared to having MOs at later stages. This appearance of function and its score output reveal that functions cannot be classified as pre- or post-processing because they do appear in unorthodox positions, and their appearance in this manner does give good output results that boost the performance score.

In addition, to obtain higher scores, chains do repeat the functions used, which shows that the processing outputs of each repeated function are indeed distinct from each other and therefore that the repetitions perform useful computations and are not just copying or relaying information. In the experimental analysis of IPCO and MIPCO, it was discovered that there are sets of mandatory functions that always seem to appear together in MIPCO networks. It was found that the MO opening, Watershed, MO eroding, Denoising, and Thresholding functions are very frequently selected. It appears that at least for this membrane detection problem, all the ‘good chains’ select watershed as one of the preferred functions, and this function always appears together with its co-partner, namely, the morphological operator *open*. It is also believed that the selection of these functions contributes to overall better performance for the dataset used in this research. These combinations were identified and measured through the performance of its F1 scores.

## 7.6 Similarities and differences of IPCO and MIPCO

Both algorithms use similar functions and do not favour the Edge Detection function for the generation of ‘best chain’ which have F1 score >90%. In filtering noise, both algorithms favoured Median filtering, in comparison with other provided filters (*e.g.*, Wiener). In observation made for ‘short chain’, the set of mandatory functions that always appear comprise Thresholding and Denoising functions. The other favoured functions of MIPCO such as Morphological Operators (*e.g.*, opening and eroding) and Watershed function seem to not be much favoured in IPCO selection of the ‘shortest chain’ function. On the other hand, observation of IPCO’s and MIPCO’s longest chains (with eight functions chosen for IPCO and five chains for MIPCO) shows that the IPCO’s longest chain does appear without Hole-Filling, Watershed, or Morphology Operator. Analysis also showed that in IPCO, there are also functions with no Thresholding (about 4% of tested experiments out of 50 trials) and Denoising (48% of tested experiments out of 50 trials) in the chain. This contrasts with the Combination function, which appears in all of IPCO’s longest chains. On the other hand, analysis results show that there is no single function that scores 100% for MIPCO’s longest network if eight functions are allowed in a chain. The Combination function differs from IPCO, because this function is not favoured in the MIPCO network, which only shows 12% of the networks out of 50 trials and sittings in last position in the table score.

## 7.7 Algorithm Performance and Improvement

Both the second(IPCO) and third (MIPCO) algorithms developed in this research, performed well to give average F1 scores of 91.67% and 91.80%, respectively. MIPCO achieved optimal rates (accuracy) by scoring higher than the individual IPCO algorithm. Because IPCO is only optimising a single chain the average IPCO solutions use fewer functions (*e.g.*, eight functions), so the time taken for IPCO optimisation is much less than MIPCO. MIPCO is a network in which the average MIPCO solution uses more functions and chains



(*e.g.*,  $5 \times 3$ ). However, even though MIPCO is slower than IPCO, it still manages to optimise the network in ‘seconds’ per image and in ‘minutes’ for the whole network (*e.g.*, 9 minutes for the best MIPCO network to optimise).

Consequently, a user that needs an algorithm that can perform well and must also be speedy should choose IPCO, whereas a user who desires more accuracy than speed should choose MIPCO. At present, MIPCO has a maximum individual average score of 91.80%, but the performance of MIPCO and even IPCO can be enhanced by incorporating machine learning ideas such as neural network components that evolve together with IPCO and the MIPCO network. However, it is quite difficult to achieve a 100% score, as the score is highly dependent on the ground truth. A slight difference in the thickness of membrane lines can cause a drop in the score calculated because the pixel intensity value will differ as a result of the thickness of the membrane. Even a similar output is also not guaranteed a 100% score because of variations in the thickness of the displayed membrane, although it correctly identifies the complete membrane lines.

## **7.8 Empirical Analysis: Reliability of the Proposed Algorithm**

IPCO was tested in the public domain, and against a benchmark comprising the leaders of the International Symposium in BioMedical Imaging, receiving a resulting Rand score of 90% for an unseen dataset. As mentioned earlier, the recorded highest score for this competition was 94%, but according to the literature, this algorithm was obtained through long hours of training and specialised hardware requirement.

As stated above, it can clearly be seen that all three algorithms developed in this research performed well and achieved the set goals and objectives of the research. Further, the results relate to expectations and to the literature studied in the area of image processing to provide an efficient, low cost (hardware), and reliable algorithm which is easy to adapt and manipulate even by non-computer scientists with minimal knowledge of image processing. All the algorithms developed are more acceptable because each of them can contribute

to image segmentation and can reach different target audiences. LCHF can reach audiences that do not have gold standard or ground truth images but need a moderate image segmentation output at high speed. IPCO can reach audiences who need a fast and reliable automated algorithm to perform the image segmentation, and MIPCO can reach audiences who need an algorithm which is accurate in performing its task at moderate speed. The LCHF, IPCO, and MIPCO algorithms are consistent and fit in with previously published knowledge in image processing and segmentation. The findings and observations made in this research have been presented at International Conferences and have also been published in Journals in order to share the knowledge with the community.

**Generally, when a score of 92.63% it means there is some errors in the technique (in this case; 7.37%), several questions have been of interest: Do these errors in anyway affect the subsequent decision making process by medical practitioners?**

This research outcome is not only targeted at medical experts or neuroscience experts. Moreover, the primary motivation of the research is to reveal and understand the complete connectivity pattern within an organism's nervous system. In medical imaging, an error of 7.37% may have a detrimental effect on decision making, but it depends on the specific problem and anatomical structures that need to be addressed. In this research, the majority of mismatched pixels were contributed by the thickness difference of membrane lines and black patches some of which could be the result of staining procedures. Small disagreements and deviations in the boundary location are however tolerable and can be ignored for most purposes. For example, let's look at the two scenarios below.

#### Scenario 1

If a medical practitioner is examining, for example, two dendritic processes that are connected to each other, then the an error of 7.37% may have a negative effect on a subsequent decision making process.

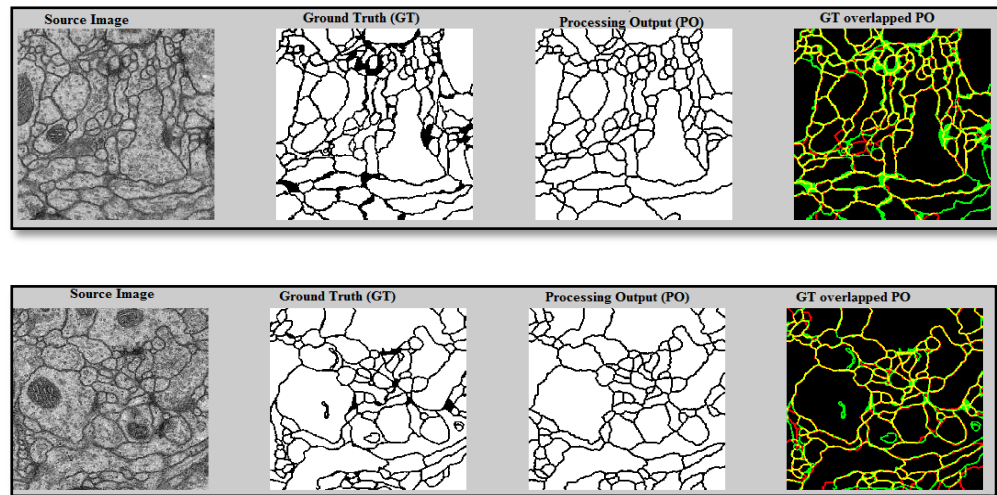
## Scenario 2

If a medical practitioner is examining the density of connections, then this type and scale of error is unlikely to be problematic. Moreover, medical practitioners often use information at a higher level of abstraction (*e.g.* density of connections rather than specific synapses), which suggests that this level of error, will not generally have major impact in medical practices.

### **7.9 Suggestions for Future work**

Some issues that need to be addressed in future work include the presence of some over-segmentation (false positives) and missing ‘black patches’ (false negatives), as can be seen for example in Figure 7.5 (GT overlapped PO). The figure shows an example of two sets of images randomly selected out of the 30 images used. The first (leftmost) image is the source image, followed by the ground truth image and the processing output. The final image is the comparison image between ground truth and the processed output. The different coloured membrane lines have the following meaning:

- Yellow line shows the matching line between the ground truth image and the processing output.
- The red line shows the over-segmentation between the ground truth image and the processing output.
- The green line shows the missing ‘black patches’ in comparison with the ground truth and the processing output image.
- The background is represented as a black background.



Source Image                      Ground Truth (GT)                      Processing Output (PO)  
 GT overlapped PO

Figure 7.5: Sample output obtained using the IPCO processing chain.

Figure 7.5 also provides an opportunity to subjectively compare the outputs with the ground truth. At a glance, in subjectively comparing the outputs with the ground truth, the outputs appear to be almost identical to the corresponding ground truth. Most of the average error is possibly due to missing black patches (false negatives) that is believed to be caused as an artefact of staining procedures and the segmentation process is complicated by the texture generated by other stained structures. The source of contrast in TEM images is the darker appearance of stained structures and noise elements. The images are representative of actual images in the real-world: there is a bit of noise; there are image registration errors; there is even a small stitching error in one section (Cardona *et al.*, 2012). Each image captured by the camera has to be ‘stitched’ to its neighbors, much like a panoramic view taken in multiple photos. But the heat of the electron beam distorts the thin sections, making the digital stitching process difficult (Tasdizen *et al.*, 2005). In preparing the image, all membranes have been highlighted as one unique object. All neurites (and glia) have been highlighted each as its own independent object. Each noise element being assigned a unique label which is believed to correspond to ‘black patches’. The noise element varies from a tile stitching error, which occurs as result of combining multiple images with overlapping fields of view to produce

a segmented high resolution image. Noise also emerges from precipitate and bubbles that form from the chemical solution that is added for dissection.

Further, in comparing ground truth images with the algorithms' output images, there are also some extra lines, possibly due to the watershed function (false positives), as well as differences in line thickness (ground truth images exhibit relatively large variations in line thickness, which contrasts with most of the processed outputs which show relatively constant line thickness). Below are the examples cumulative False Positive and False Negative pixels. Image resolution consists of 512x512 pixels. That means that there are overall 262,144 pixels in every image slice. So, basically from the Table 7.2, it can be noted that on average about 26,375 pixels are recorded to be false in this selected image processing chain.

Table 7.2: Sample cumulative pixels showing false positive and false negative..

| <b>Score/Slice</b> | <b>Cumulative of False Positive (1)</b> | <b>Cumulative of False Negative (2)</b> | <b>Error Pixels (1+2)</b> |
|--------------------|---|---|---------------------------|
| <b>Slice 1</b>     | 29948                                   | 8117                                    | 38065                     |
| <b>Slice 2</b>     | 22321                                   | 11080                                   | 33401                     |
| <b>Slice 3</b>     | 25437                                   | 8255                                    | 33692                     |
| <b>Slice 4</b>     | 24082                                   | 9843                                    | 33925                     |
| <b>Slice 5</b>     | 27727                                   | 17487                                   | 45214                     |
| <b>Slice 6</b>     | 31205                                   | 14011                                   | 45216                     |
| <b>Slice 7</b>     | 22952                                   | 11981                                   | 34933                     |
| <b>Slice 8</b>     | 27166                                   | 16821                                   | 43987                     |
| <b>Slice 9</b>     | 30622                                   | 24913                                   | 55535                     |
| <b>Slice 10</b>    | 24460                                   | 10698                                   | 35158                     |
| <b>Slice 11</b>    | 46165                                   | 7241                                    | 53406                     |
| <b>Slice 12</b>    | 22971                                   | 16020                                   | 38991                     |
| <b>Slice 13</b>    | 28633                                   | 12573                                   | 41206                     |
| <b>Slice 14</b>    | 19886                                   | 14895                                   | 34781                     |
| <b>Slice 15</b>    | 17551                                   | 12453                                   | 30004                     |
| <b>Slice 16</b>    | 25476                                   | 12817                                   | 38293                     |
| <b>Slice 17</b>    | 19097                                   | 15081                                   | 34178                     |
| <b>Slice 18</b>    | 29329                                   | 11766                                   | 41095                     |
| <b>Slice 19</b>    | 24204                                   | 10692                                   | 34896                     |
| <b>Slice 20</b>    | 27460                                   | 9671                                    | 37131                     |
| <b>Slice 21</b>    | 54487                                   | 2463                                    | 56950                     |

continue...

continued....

|                 |          |          |          |
|-----------------|----------|----------|----------|
| <b>Slice 22</b> | 29338    | 9251     | 38589    |
| <b>Slice 23</b> | 19757    | 15205    | 34962    |
| <b>Slice 24</b> | 19806    | 13912    | 33718    |
| <b>Slice 25</b> | 18959    | 14093    | 33052    |
| <b>Slice 26</b> | 30531    | 13098    | 43629    |
| <b>Slice 27</b> | 25530    | 7120     | 32650    |
| <b>Slice 28</b> | 25976    | 8965     | 34941    |
| <b>Slice 29</b> | 19885    | 9392     | 29277    |
| <b>Slice 30</b> | 20290    | 10778    | 31068    |
| <b>Average</b>  | 26375.03 | 12023.07 | 38398.10 |

We also plan to continue to emphasise the simplicity, usability, interpretability, and efficiency of IPCO and the MIPCO network whilst improving their accuracy in future work. The main priority for future work is inclusion of neural network components that evolve together with IPCO and the MIPCO network.

In the appendix section, many competitions and challenges that evolve around image segmentation are listed. Continuation of the work will also encompass application of the IPCO/MIPCO to these challenges, refinement of the approaches, and acquisition of new insights by using other benchmarks.

## CHAPTER 8

### CONCLUSION

IPCO chain and MIPCO network not only highlight membrane boundaries, but also remove internal structures (*i.e.*, eliminate organelles) successfully. To enhance F1 scores while preserving simplicity and efficiency, global stochastic optimisation and ensemble methods are incorporated into them. The implementation of IPCO chain and MIPCO network were found to be capable of efficiently detecting membranes of *Drosophila First Instar Larvae Ventral Nerve Cord* dataset with (at present) the highest recorded Ensemble average F1 score of 92.11% (for IPCO) and 92.63% (for MIPCO) and the highest recorded individual average F1 score (at present) of 91.67% (for IPCO) and 91.80% (for MIPCO).

One of the main advantages of IPCO chain and MIPCO network, as mentioned before, is that they involve a relatively fast (minutes) and consistent optimisation process, which leads to a variety of useful and easily interpretable solutions. Another IPCO advantage is that they do not require specialised hardware. Based on today's hardware constraints, training classifiers with a very large number of free parameters can require weeks of computation, even when high performance machines with high data transfer rates are used. This involves significant monetary and energy costs. Moreover, long hours of training and specialised hardware are usually not feasible for small researchers. IPCO chain and MIPCO network not only save time, but are also very easy to use and deploy. With relatively little computational cost it was possible to get reasonable results. The best individual average F1 score thus far on a specific partition of the ISBI 2012 testing set is 91.80% and the algorithm does indeed do a reasonably good job at distinguishing membranes and organelles, thus satisfying the original goal. Moreover, the simplicity, efficiency, interpretability, and usability of IPCO chain and MIPCO networks, make them

easier to adopt by researchers with limited experience of computer vision and machine learning.

## 8.1 Interesting Findings and Suggestion

1. An interesting observation pertains to denoising. As is commonly known, the main purpose of denoising is to filter out image noise in order to minimise detrimental effects in subsequent processing. Typically, denoising is carried out as an early preprocessing stage before other core functions are applied. However, through the experiments, it was found that, at least for this membrane segmentation problem, denoising typically appears later in chains. From the experimental results, it was found that in a majority of chains with F1 scores higher than 90%, cleaning only takes place after enhancing and classification. This arguably runs contrary to common expectation. This is a fact that needs to be taken into account by image processing users, that we should not always clean images at an early stage as this will remove important information that may be needed by other component functions. As for the observation regarding other chains with scores below 90%, the denoising component tends to appear early in the chain before classification. In summary, the chains tend to choose denoising as a middle to late processing component, because chains are generally trying to enhance information before losing or cleaning it.
2. Another interesting observation pertains to morphological functions, specifically, the appearance of morphological operators in all best chains (F1 score >90%) in unorthodox positions. As is commonly known, the main purpose of the morphological operator is to give a smoothing effect to the image and it is typically treated as a post-processing operator. Through the MIPCO experiments, it was found that, at least for this membrane segmentation problem, the morphological operator does appear in unorthodox positions (early, middle) in the MIPCO network, and the appearance of this operator in unusual positions does contribute to better performance and has a



significant effect on the F1 score. In the experiments, two morphological operators used were the morphological ‘open’ (erosion followed by dilation process), and standalone erosion. The results obtained showed that the morphological operator erosion appears early in the chain (as a first function), this arguably runs contrary to common expectation of its appearance in image processing pipelines. This new appearance of the morphological operator in unorthodox positions is fact that needs to be taken into account by image processing users, that we should not always use morphological operators in the final stage of processing only for smoothing purposes. Optimisation often finds unexpected ways to use functions and perform new sets of pipelines.

3. From the experiments conducted it was discovered that the most popular function to appear is the thresholding function. The second most popular is the denoising function. To analyse further, an experiment was conducted to force the number of functions chosen. Given a choice for the optimiser to choose one function, the Thresholding function was chosen 100% of the time. The experimental analyses conducted show that the ‘shortest chain’ (the optimisation returns a minimum number of functions in a chain (*e.g.*, two to three functions)) for both IPCO and MIPCO consist of a higher percentage of appearance for both the Thresholding and Denoising functions.
4. The experiments also showed that functions seem to repeat themselves in the same chains and neighbour chains. A closer look at the parameter used for the repetition function showed that the function does useful computations, and not just copy images. Although some functions do appear in consecutive positions (one after another), there are output variations. Thus, it obviously is doing useful computation.
5. In the experimental analysis conducted using MIPCO network, it was discovered that there are sets of mandatory functions that always seem to appear as partners in chains. For example, Morphological

Operator seems to always appear with its co-partner Watershed functions.

The focus of this research was on the problem of neuronal membrane detection in which the core challenge consists of distinguishing membranes from organelles. From the experiments, and given the specific membrane detection dataset adopted, it was found that optimisation of the image processing chains, using Single and Multiple Chains does result in better accuracy and speed. Time is an important factor when accuracy is a little bit less than the existing methods, thus the time factor is crucial to consider. Further, there are many advantages over comparable approaches. One of the main advantages of IPCO, as mentioned before, is that it involves a relatively fast (minutes) and consistent optimisation process, which leads to a variety of useful and easily interpretable solutions, easy customisability and re-trainability, while for MIPCO network, function can connect to any other function in the previous layer. MIPCO can optimise networks together and can interact with each other.

The research output meets and satisfies the stated aims, goals, and objectives (please see the Table 8.1).

Table 8.1: Research output satisfying the aims, goals, and objectives.

| Number | Aim, Goal, and Objectives   |
|--------|---|
| 1      | Developed a membrane detection algorithm with accuracy close to the state-of-the-art, but with additional features such as efficient training, interpretability, usability, and easy adoption by new researchers in detecting membranes and eliminating organelles. |
| 2      | Developed a membrane detection algorithm with improvements in speed close to the state-of-the-art.  |
| 3      | Low cost (money –in term of hardware used and energy).  |
| 4      | Higher interpretability (understandability) network, for example, in comparison with Neural Network.  |
| 5      | Usability, tested with a different set of images and it resulted in a reasonable output.  |
| 6      | To adopt the hybrid algorithm that combines the high-level knowledge (optimisation of networks) with low-level information (image processing functions).  |
| 6      | Developed a simple and efficient approach based on several basic processing steps, including local contrast enhancement, thresholding, denoising, hole-filling, watershed segmentation, and morphological operators.  |
| 7      | Able to obtain insights into new types of useful image processing pipelines.  |

## 8.2 Contribution of the Proposed Algorithms

In comparison with work by other researchers in this area, the proposed algorithms differ in one or more ways in using existing functions by optimising the manner in which they are configured and combined; specifically, the set of functions used, the parameterisations allowed, the optimisation methods adopted, the combination framework, and the testing and analyses conducted.

The framework included a new category of special-purpose ‘combiner’ functions specifically designed to encourage chains to form various representations and transformations. This is the first time this approach has been applied in the context of the *Drosophila* first instar larva ventral nerve cord (VNC), imaged at a resolution of  $4 \times 4 \times 50$  nm/pixel and covering a  $2 \times 2 \times 1.5$  micron cube of neural tissue. A systematic analysis of the statistics of optimised chains was conducted. The results revealed several interesting and unconventional insights pertaining to preprocessing, classification, post-processing, and speed. In other words, the types of analyses which were conducted are novel, and have, for example, revealed interesting insights pertaining to denoising, and the morphological operator, and its appearance in unorthodox positions in image processing pipelines.

The following conclusions can be stated:

- (a) The algorithms (IPCO and MIPCO) not only highlight membrane boundaries, but also remove internal structures (eliminate organelles) successfully.
- (b) The implemented IPCO and MIPCO chains were found to be capable of efficiently detecting membranes in the ISBI 2012 challenge dataset. IPCO applies the simplicity and efficiency of sequences of image processing functions and involves automated fine-tuning of the algorithm relative to some dataset. Further, MIPCO optimises the overall network, and interacts to produce the best output with the highest score.
- (c) They involve a relatively fast (minutes) and consistent optimisation process, which leads to a variety of useful and easily interpretable solutions.
- (d) They do not require specialised hardware.
- (e) They involve relatively low monetary and energy costs.

- (f) IPCO and MIPCO not only save time, but are also very easy to deploy and use.
- (g) They are feasible for a small researcher with a small amount of capital and non-computer scientists with limited knowledge of image segmentation.

## REFERENCES

- Ackley, D. (1987). A connectionist machine for genetic hillclimbing, *The Kluwer International Series in Engineering and Computer Science* (Vol 28) : Kluwer academic publishers.
- Aishwarya, S., & Anto, S. (2014). A Medical Decision Support System based on Genetic Algorithm and Least Square Support Vector Machine for Diabetes Disease Diagnosis, *International Journal of Engineering Sciences & Research Technology*, pp 4042-4046.
- Alvarez, J., Jernigan, M., & Nahmias, C. (1999). Neural network-based segmentation of magnetic resonance images of the brain. *Nuclear Science, IEEE Transactions on*, 44(2), pp 194-198.
- Antonelli, M., Lazzerini, B., & Marcelloni, F. (2005). Segmentation and reconstruction of the lung volume in CT images. Paper presented at the Proceedings of the 2005 ACM symposium on Applied computing, pp 255-259.
- Aoki, S., & Nagao, T. (1999). Automatic construction of tree-structural image transformations using genetic programming. Paper presented at the Image Analysis and Processing, Proceedings. IEEE International Conference, pp 136-141.
- Arunachalam, V. (2014). Optimization Using Differential Evolution, Water Resources Research Report, Water Resources Research Report (060).
- Astola, J., & Kuosmanen, P. (1997). Fundamentals of nonlinear digital filtering (Vol. 8): CRC pressLLC, Boca Raton, Fla, USA.
- Atli, E. (2013). The effects of three selected endocrine disrupting chemicals on the fecundity of fruit fly, *Drosophila melanogaster*. *Bulletin of environmental contamination and toxicology*, 91(4), pp 433-437.
- Bhuyan, R., & Borah, S. (2014). Image Segmentation: Computational Approaches for Medical Images. *IJCA Proceedings on National Conference cum Workshop on Bioinformatics and Computational Biology NCWBCB*(Vol 3).
- Blaschke, T. (2010). Object based image analysis for remote sensing. *ISPRS journal of photogrammetry and remote sensing*, 65(1), pp 2-16.
- Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2), pp 121-167.
- Burget, R., Uher, V., & Masek, J. (2012). *Trainable segmentation based on local-level and segment-level feature extraction*. Paper presented at the 2012 IEEE international symposium on biomedical imaging (ISBI). from nano to macro.
- Canny, J. (1986). A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*(6), pp 679-698.
- Cardona, A., Saalfeld, S., Preibisch, S., Schmid, B., Cheng, A., Pulokas, J., Hartenstein, V. (2010). An integrated micro-and macroarchitectural analysis of the *Drosophila* brain by computer-assisted serial section electron microscopy. *PLoS Biol*, 8(10), e1000502.
- Carpenter, A. E., Jones, T. R., Lamprecht, M. R., Clarke, C., Kang, I. H., Friman, O., Moffat, J. (2006). CellProfiler: image analysis software for

- identifying and quantifying cell phenotypes. *Genome biology*, 7(10), R100.
- Chen, J., Luo, M., Li, L., Li, D., Zhang, C., Huang, Y., & Jiang, Y. (2008). Comparison and analysis methods of moderate-resolution satellite remote sensing image classification. *WSEAS Transactions on Computers*, 7(7), pp 877-886.
- Chen, S. (2000). IIR model identification using batch-recursive adaptive simulated annealing algorithm, Presented at 6th Annual Chinese Automation and Computer Science Conference, UK.
- Cheng, K.-S., Lin, J.-S., & Mao, C.-W. (1996). The application of competitive Hopfield neural network to medical image segmentation. *Medical Imaging, IEEE Transactions on*, 15(4), pp 560-567.
- Chicano, F., Whitley, D., & Alba, E. (2014). Exact computation of the expectation surfaces for uniform crossover along with bit-flip mutation. *Theoretical Computer Science*, 545, pp 76-93.
- Ciresan, D., Giusti, A., Gambardella, L., & Schmidhuber, J. (2012). Neural networks for segmenting neuronal structures in EM stacks, The Swiss AI Lab IDSIA. In: ISBI Electron Microscopy Segmentation Challenge, Barcelona, Spain.
- Collette, R., Douglas, J., Patterson, L., Bahun, G., King, J., Keiser, D., & Schulthess, J. (2015). Benefits of utilizing CellProfiler as a characterization tool for U-10Mo nuclear fuel. *Materials Characterization* (105), pp 71-81.
- Darwish, A., Leukert, K., & Reinhardt, W. (2003). *Image segmentation for the purpose of object-based classification*. Paper presented at the International Geoscience and Remote Sensing Symposium.
- DeJong, E. M., & Green, W. B. (1997). Animation and Visualization of Space Mission Data, Animation and Visualization of Space Mission Data, Nasa TechnicalReports(20060041716), USA.
- Dobell, C. (1932). Antony van Leeuwenhoek and his " Little Animals." Unpublished Manuscripts, and Contemporary Records. Published on the 300th Anniversary of Anthony's Birth., Publisher: Staples; First Thus Edition.
- Dollar, P., Tu, Z., & Belongie, S. (2006). Supervised learning of edges and object boundaries. Paper presented at the Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference (Vol 2).
- Donoser, M., Urschler, M., Hirzer, M., & Bischof, H. (2009). Saliency driven total variation segmentation. Paper presented at the Computer Vision, 2009 IEEE 12th International Conference.
- Duarte-Mermoud, M., Beltrán, N., & Salah, S. (2013). Probabilistic adaptive crossover applied to chilean wine classification. *Mathematical Problems in Engineering*.
- Dzyubachyk, O., Niessen, W., & Meijering, E. (2008). *Advanced level-set based multiple-cell segmentation and tracking in time-lapse fluorescence microscopy images*. Paper presented at the Biomedical Imaging: From Nano to Macro, ISBI 2008. 5th IEEE International Symposium.
- Estellers, V., Zosso, D., Lai, R., Osher, S., Thiran, J., & Bresson, X. (2012). Efficient algorithm for level set method preserving distance function. *Image Processing, IEEE Transactions on*, 21(12), pp 4722-4734.

- Farag, A. A., El-Baz, A., & Gimel'farb, G. (2004). *Precise image segmentation by iterative EM-based approximation of empirical grey level distributions with linear combinations of Gaussians*. Paper presented at the Computer Vision and Pattern Recognition Workshop, CVPRW'04.
- Fu, K.-S., & Lu, S.-Y. (1977). A clustering procedure for syntactic patterns. *Systems, Man and Cybernetics, IEEE Transactions on*, 7(10), pp 734-742.
- Fu, K.-S., & Mui, J. (1981). A survey on image segmentation. *Pattern recognition*, 13(1), pp 3-16.
- Gonzalez, R. C., Woods, R. E., & Eddins, S. L. (2010). Morphological reconstruction. *Digital Image Processing using MATLAB, MathWorks*, Prentice Hall.
- Guo, J., Zheng, P., & Huang, J. (2015). Secure watermarking scheme against watermark attacks in the encrypted domain. *Journal of Visual Communication and Image Representation*, 30, pp 125-135.
- Hadamard, J. (1923). Lectures on Cauchy's problem in linear partial differential equations/Jacques S. Hadamard. New Haven Yale University Press.
- Haga, H., & Suehiro, A. (2012). *Automatic test case generation based on genetic algorithm and mutation analysis*. Paper presented at the Control System, Computing and Engineering (ICCSCE), 2012 IEEE International Conference.
- Haidekker, M. A. (2013). *Medical imaging technology*, pp 1-96: Springer.
- Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7), pp 1527-1554.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*: U Michigan Press.
- Hooke, R., & Gunther, R. T. (1938). *Micrographia, 1665*: Oxford, 14 vols.
- Hu, S., Hoffman, E. A., & Reinhardt, J. M. (2001). Automatic lung segmentation for accurate quantitation of volumetric X-ray CT images. *Medical Imaging, IEEE Transactions on*, 20(6), pp 490-498.
- Huang, X., & Tsechpenakis, G. (2009). Medical image segmentation. *Information Discovery on Electronic Health Records*, 10, pp 251-289.
- Iftikhar, S., & Godil, A. (2012). The Detection of Neuronal Structures using a Patch-based Multi-features and Support Vector Machines Learning Algorithm. Paper presented at the Proc. of ISBI.
- Jain, V., Bollmann, B., Richardson, M., Berger, D. R., Helmstaedter, M. N., Briggman, K. L., Abraham, W. C. (2010). Boundary learning by optimization with topological constraints. Paper presented at the Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference.
- Jones, B. (2005). *Enhancement of cell boundaries in transmission electron microscopy images*. Paper presented at the Image Processing, 2005. ICIP 2005. IEEE International Conference.
- Jurrus, E., Paiva, A. R., Watanabe, S., Whitaker, R., Jorgensen, E. M., & Tasdizen, T. (2009). Serial neural network classifier for membrane detection using a filter bank. Paper presented at the Proc. workshop on microscopic image analysis with applications in biology.



- Kamentsky, L. (2012). Segmentation of EM images of neuronal structures using CellProfiler. Paper presented at the Proc. of ISBI.
- Karaboga, N., & Cetinkaya, B. (2004). Performance comparison of genetic and differential evolution algorithms for digital FIR filter design *Advances in information systems*, pp. 482-488: Springer.
- Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: Active contour models. *International journal of computer vision*, 1(4), pp 321-331.
- Kaynig, V., Fischer, B., & Buhmann, J. M. (2008). Probabilistic image registration and anomaly detection by nonlinear warping. Paper presented at the Computer Vision and Pattern Recognition. CVPR 2008. IEEE Conference.
- Koppal, S. J., & Narasimhan, S. G. (2015). Beyond perspective dual photography with illumination masks. *Image Processing, IEEE Transactions on*, 24(7), pp 2083-2097.
- Lamprecht, M. R., Sabatini, D. M., & Carpenter, A. E. (2007). CellProfiler™: free, versatile software for automated biological image analysis. *Biotechniques*, 42(1), 71.
- Laptev, D., Vezhnevets, A., Dwivedi, S., & Buhmann, J. M. (2012). Anisotropic ssTEM image segmentation using dense correspondence across sections *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012*, pp. 323-330: Springer.
- Liu, T., Jurrus, E., Seyedhosseini, M., Ellisman, M., & Tasdizen, T. (2012). Watershed merge tree classification for electron microscopy image segmentation. Paper presented at the Pattern Recognition (ICPR), 2012 21st International Conference.
- Low, C., Hong, K., Salleh, K., & Johnny, K. (2010). Path optimization using genetic algorithm evolution. Paper presented at the Research and Development (SCORED), 2010 IEEE Student Conference.
- Lucchi, A., Smith, K., Achanta, R., Knott, G., & Fua, P. (2012). Supervoxel-based segmentation of mitochondria in em image stacks with learned shape features. *Medical Imaging, IEEE Transactions on*, 31(2), pp 474-486.
- Lucchi, A., Smith, K., Achanta, R., Lepetit, V., & Fua, P. (2010). A fully automated approach to segmentation of irregularly shaped cellular structures in EM images *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2010*, pp. 463-471: Springer.
- Malladi, R., Sethian, J. A., & Vemuri, B. C. (1993). *Topology-independent shape modeling scheme*. Paper presented at the SPIE's 1993 International Symposium on Optics, Imaging, and Instrumentation.
- Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. Paper presented at the Computer Vision. ICCV 2001. Proceedings. Eighth IEEE International Conference.
- Martin, V., & Thonnat, M. (2008). Learning contextual variations for video segmentation, *Computer Vision Systems*, pp. 464-473: Springer.
- Mathew, J. A., Khan, A., & Niranjana, U. (2011). Diagnosis of the Abnormality Extracted MRI Slice Images of a GUI Based Intelligent Diagnostic Imaging System. Paper presented at the Process Automation, Control and Computing (PACC), 2011 International Conference.

- MatLab, M. (2012). The language of technical computing. *The MathWorks, Inc.* <http://www.mathworks.com>.
- McInerney, T., & Terzopoulos, D. (1995). A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image analysis. *Computerized Medical Imaging and Graphics*, 19(1), pp 69-83.
- McInerney, T., & Terzopoulos, D. (1996). Deformable models in medical image analysis: a survey. *Medical image analysis*, 1(2), pp 91-108.
- Meijering, E. (2010). Neuron tracing in perspective. *Cytometry Part A*, 77(7), pp 693-704.
- Meijering, E. (2012). Cell segmentation: 50 years down the road [life sciences]. *Signal Processing Magazine, IEEE*, 29(5), pp 140-145.
- Meyer, F. (1994). Topographic distance and watershed lines. *Signal processing*, 38(1), pp 113-125.
- Milanova, M. (2014). Lecture Notes on Image Processing. Department of Computer Science, University of Arkansas at Little Rock.
- Mogoseanu, M., Pascut, M., Barsasteanu, F., Motoi, S., Tutelca, A., Vesa, A. M., & Socoliuc, C.(2003). Computed Tomography (CT) Versus Magnetic Resonance Imaging (MRI) in Evaluation of Head Injuries.
- Mohamed, N. A., Ahmed, M. N., & Farag, A. (1999). Modified fuzzy c-mean in medical image segmentation. Paper presented at the Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference.
- Nagao, T., & Masunaga, S. (1996). Automatic construction of image transformation processes using genetic algorithm. Paper presented at the Image Processing, 1996. Proceedings., International Conference .
- Nakano, Y., Shirakawa, S., Yata, N., & Nagao, T. (2010). Automatic construction of image transformation algorithms using feature based genetic image network. Paper presented at the Evolutionary Computation (CEC), 2010 IEEE Congress.
- Nayak, N. M., Zhu, Y., & Roy Chowdhury, A. K. (2015). Hierarchical Graphical Models for Simultaneous Tracking and Recognition in Wide-Area Scenes. *Image Processing, IEEE Transactions on*, 24(7), pp 2025-2036.
- Neubert, M., Herold, H., & Meinel, G. (2006). Evaluation of remote sensing image segmentation quality—further results and concepts. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Salzburg.
- Ng, H., Ong, S., Foong, K., Goh, P., & Nowinski, W. (2006). Medical image segmentation using k-means clustering and improved watershed algorithm. Paper presented at the Image Analysis and Interpretation, 2006 IEEE Southwest Symposium.
- Oh, J., Harman, M., & Yoo, S. (2011). Transition coverage testing for simulink/stateflow models using messy genetic algorithms. Paper presented at the Proceedings of the 13th annual conference on Genetic and evolutionary computation.
- Overington, I. (1985). A paradox-high fidelity from poor image quality'. *Proceedings of Machine Intelligence*, (Vol 85).
- Overington, I. (1988). *Vision and acquisition*. London, U.K.: Pentech Press.

- Pal, N. R., & Pal, S. K. (1993). A review on image segmentation techniques. *Pattern recognition*, 26(9), pp 1277-1294.
- Pandey, R., Gamit, N., & Naik, S. (2014). A novel non-destructive grading method for Mango (*Mangifera Indica L.*) using fuzzy expert system. Paper presented at the Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference).
- Pardalos, P. M., & E, R. (2002). Handbook of global optimization, Heuristic approaches (Vol. 2).
- Pham, D. L., Xu, C., & Prince, J. L. (2000). Current methods in medical image segmentation 1. *Annual review of biomedical engineering*, 2(1), pp 315-337.
- Pierce, B. A. (2015). Genetics essentials: concepts and connections, Just the Facts101,' Textbook Key Facts and Reviews, e-StudyGuides (Vol. 2nd).
- Pitas, I., & Venetsanopoulos, A. (1990). Nonlinear Digital Filters: Principles and Applications Kluwer Academic. Publishers Hingham.
- Pohle, R., & Toennies, K. D. (2001). Segmentation of medical images using adaptive region growing. Paper presented at the Medical Imaging 2001.
- Prathibha, B., & Sadasivam, V. (2012). Hybrid transforms domain-based mammogram analysis using C-SVM classifier. *International Journal of Medical Engineering and Informatics*, 4(2), pp 146-156.
- Preston, K. (1976). Computer processing of biomedical images. *Computer*, 9(5), pp 54-68.
- Prewitt, J., & Mendelsohn, M. L. (1966). The analysis of cell images\*. *Annals of the New York Academy of Sciences*, 128(3), pp 1035-1053.
- Punnappurath, A., Rajagopalan, A. N., Taheri, S., Chellappa, R., & Seetharaman, G. (2015). Face recognition across non-uniform motion blur, illumination, and pose. *Image Processing, IEEE Transactions on*, 24(7), pp 2067-2082.
- Qi, X. (2005). Lecture Notes on 'Computer Vision and Image Processing. Department of Computer Science, Utah State University.
- Rahnamayan, S., & Mohamad, Z. (2010). Tissue segmentation in medical images based on image processing chain optimization. Paper presented at the International Workshop on Real Time Measurement, Instrumentation and Control, Toronto.
- Raju, R., Maul, T., & Bargiela, A. (2014). Local contrast hole filling algorithm for neura slices membrane detection—LCHF. Paper presented at the Computer Applications and Industrial Electronics (ISCAIE), 2014 IEEE Symposium.
- Raju, R., Maul, T., & Bargiela, A. (2014). A new image processing heuristic suggested by optimization experiments—Enhance it before you Lose it. Paper presented at the 2nd International Conference on Intelligent Systems and Image Processing.
- Raju, R., Maul, T. H., & Bargiela, A. (2015). New image processing pipelines for membrane detection. *Journal of the Institute of Industrial Applications Engineers*, 3(1), pp 15-23.
- Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336), pp 846-850.

- Reiter, L. T., Potocki, L., Chien, S., Gribskov, M., & Bier, E. (2001). A systematic analysis of human disease-associated gene sequences in *Drosophila melanogaster*. *Genome research*, 11(6), pp 1114-1125.
- Robič, T., & Filipič, B. (2005). *DEMO*: Differential evolution for multiobjective optimization. Paper presented at the Evolutionary multi-criterion optimization.
- Roobottom, C., Mitchell, G., & Morgan-Hughes, G. (2010). Radiation-reduction strategies in cardiac computed tomographic angiography. *Clinical radiology*, 65(11), pp 859-867.
- Rosenfeld, A. (1977). Iterative methods in image analysis. *Pattern recognition*, 10(3), pp 181-187.
- Rosenfeld, A. (1984). Picture processing: 1983. *Computer Vision, Graphics, and Image Processing*, 26(3), pp 347-384.
- Russ, J. C. (2011). The image processing handbook: CRC press.
- Saha, S. K., Kar, R., Mandal, D., & Ghoshal, S. P. (2013). Bacteria foraging optimisation algorithm for optimal FIR filter design. *International Journal of Bio-Inspired Computation*, 5(1), pp 52-66.
- Scholl, I., Aach, T., Deserno, T. M., & Kuhlen, T. (2011). Challenges of medical image processing. *Computer science-Research and development*, 26(1-2), pp 5-13.
- Semenkin, E., & Semenkina, M. (2012). Self-configuring genetic programming algorithm with modified uniform crossover. Paper presented at the 2012 IEEE Congress on Evolutionary Computation.
- Seyedhosseini, M., Kumar, R., Jurrus, E., Giuly, R., Ellisman, M., Pfister, H., & Tasdizen, T. (2011). Detection of neuron membranes in electron microscopy images using multi-scale context and radon-like features Medical Image Computing and Computer-Assisted Intervention–MICCAI 2011, pp. 670-677: Springer.
- Seyedhosseini, M., Liu, T., Jurrus, E., & Tasdizen, T. (2012). Neuron segmentation in em images using series of classifiers and watershed tree. Paper presented at the Proc. of ISBI.
- Seyedhosseini, M., & Tasdizen, T. (2015). Semantic Image Segmentation with Contextual Hierarchical Models.
- Shapiro, L., & Stockman, G. C. (2001). Computer vision. 2001. ed: Prentice Hall.
- Shapiro, L. G., & Linda, G. (2002). stockman, George C. Computer Vision, Prentice Hall.
- Sharma, N., & Aggarwal, L. M. (2010). Automated medical image segmentation techniques. *Journal of medical physics*, 35(1), 3.
- Singh, K. K., & Singh, A. (2010). A study of image segmentation algorithms for different types of images. *International Journal of Computer Science*, 7(5), pp 414-417.
- Smistad, E., Falch, T. L., Bozorgi, M., Elster, A. C., & Lindseth, F. (2015). Medical image segmentation on GPUs–A comprehensive review. *Medical image analysis*, 20(1), pp 1-18.
- Sonka, M., Hlavac, V., & Boyle, R. (1998). Image Processing, Analysis, and Machine Vision. Brooks and Cole Publishing.
- Sridevi, M., & Mala, C. (2012). A Survey on Monochrome Image Segmentation Methods. *Procedia Technology*, 6, pp 548-555.

- Storn, R., & Price, K. (1995). Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces (Vol. 3): ICSI Berkeley.
- Stühmer, J., & Cremers, D. (2015). A Fast Projection Method for Connectivity Constraints in Image Segmentation. Paper presented at the Energy Minimization Methods in Computer Vision and Pattern Recognition.
- Suzuki, K. (2014). *Computational Intelligence in Biomedical Imaging*: Springer.
- Szeliski, R. (2010). *Computer vision: algorithms and applications*: Springer Science & Business Media.
- Taghadomi-Saberi, S., Omid, M., Emam-Djomeh, Z., & Ahmadi, H. (2014). Development of an intelligent system to determine sour cherry's antioxidant activity and anthocyanin content during ripening. *International Journal of Food Properties*, 17(5), pp 1169-1181.
- Tan, X., & Sun, C. (2012). Membrane extraction using two-step classification and post-processing. Paper presented at the Proc. of ISBI.
- Tasdizen, T., Perez, A. J., Seyedhosseini, M., Deerinck, T. J., Bushong, E. A., Panda, S., & Ellisman, M. H. (2014). A workflow for the automatic segmentation of organelles in electron microscopy image stacks. *Frontiers in neuroanatomy*, 8, pp 126-126.
- Tasdizen, T., & Seyedhosseini, M. (2014). Multi-class multi-scale series contextual model for image segmentation. *Image Processing, IEEE Transactions on*, 22(11), pp 4486-4496.
- Taye, W. (2011). Lithological boundary detection using multi-sensor remote sensing imagery for geological interpretation. Unpublished Master's thesis, Faculty of Geo-Information Science and Earth Observation, University of Twente.
- Tohka, J. (2002). Surface extraction from volumetric images using deformable meshes: a comparative study *Computer Vision—ECCV 2002*, pp. 350-364: Springer.
- Tohka, J. (2014). Partial volume effect modeling for segmentation and tissue classification of brain magnetic resonance images: A review. *World journal of radiology*, 6(11), 855.
- Tolles, W. (1955). Section of Biology: The Cytoanalyzer - An Example of Physics in Medical Research\*. *Transactions of the New York Academy of Sciences*, 17(3 Series II), pp 250-256.
- Unnikrishnan, R., Pantofaru, C., & Hebert, M. (2007). Toward objective evaluation of image segmentation algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6), pp 929-944.
- Uryasev, S., & Pardalos, P. M. (2013). *Stochastic optimization: algorithms and applications* (Vol. 54): Springer Science & Business Media.
- Venkataraju, K. U., Paiva, A. R., Jurrus, E., & Tasdizen, T. (2009). Automatic markup of neural cell membranes using boosted decision stumps. Paper presented at the Biomedical Imaging: From Nano to Macro, 2009. ISBI'09. IEEE International Symposium.
- Vonesch, C., Aguet, F., Vonesch, J.-L., & Unser, M. (2006). The colored revolution of bioimaging. *Signal Processing Magazine, IEEE*, 23(3), pp 20-31.

- Vovk, U., Pernuš, F., & Likar, B. (2007). A review of methods for correction of intensity inhomogeneity in MRI. *Medical Imaging, IEEE Transactions on*, 26(3), pp 405-421.
- Vu, N., & Manjunath, B. (2008). Graph cut segmentation of neuronal structures from transmission electron micrographs. Paper presented at the Image Processing, 2008. ICIP 2008. 15th IEEE International Conference.
- Wang, J., & Oliveira, M. M. (2003). A hole-filling strategy for reconstruction of smooth surfaces in range images. Paper presented at the Computer Graphics and Image Processing, 2003. SIBGRAPI 2003. XVI Brazilian Symposium.
- Withey, D., & Koles, Z. (2007). Medical image segmentation: Methods and software. Paper presented at the Noninvasive Functional Source Imaging of the Brain and Heart and the International Conference on Functional Biomedical Imaging, 2007. NFSI-ICFBI 2007. Joint Meeting of the 6th International Symposium.
- Wong, K.-P., Feng, D., Meikle, S. R., & Fulham, M. J. (2002). Segmentation of dynamic PET images using cluster analysis. *Nuclear Science, IEEE Transactions on*, 49(1), pp 200-207.
- Xu, C., & Prince, J. L. (1998). Snakes, shapes, and gradient vector flow. *Image Processing, IEEE Transactions on*, 7(3), pp 359-369.
- Zhang, H., Fritts, J. E., & Goldman, S. A. (2008). Image segmentation evaluation: A survey of unsupervised methods. *Computer vision and image understanding*, 110(2), pp 260-280.
- Zosso, D., Estellers, V., Bresson, X., & Thiran, J.-P. (2011). Harmonic active contours for multichannel image segmentation. Paper presented at the Image Processing (ICIP), 2011 18th IEEE International Conference.

## List of 2015 Challenges Website

**BITEWING2015:**

<http://www-o.ntust.edu.tw/~cweiwang>

**BRAIN2015:**

<http://cmic.cs.ucl.ac.uk/wmmchallenge/>

**CELL2015:**

<http://biomedicalimaging.org/2015/program/isbi-challenge>

**CHAL2015:**

<http://www-o.ntust.edu.tw/~cweiwang/ISBI2015/challenge1/index.html>

**CLEF2015:**

<http://www.imageclef.org/2015/medical>

**CSI2015:**

<http://csi2015.weebly.com/>

**CVPPP2015:**

<http://www.plant-phenotyping.org/CVPPP2015-challenge>

**CYTO2015:**

[http://cs.adelaide.edu.au/~zhi/isbi15\\_challenge/index.html](http://cs.adelaide.edu.au/~zhi/isbi15_challenge/index.html)

**DR2015:**

<https://www.kaggle.com/c/diabetic-retinopathy-detection>

**Glas@MICCAI2015:**

<http://www2.warwick.ac.uk/fac/sci/dcs/research/combi/research/bic/glascontest>

**ISLES2015:**

<http://www.isles-challenge.org/>

**OPTIMA2015:**

<http://optima.meduniwien.ac.at/optima-segmentation-challenge-1/>

**LONGITUDINAL2015:**

<http://iacl.ece.jhu.edu/MSChallenge>

**LUNG2015:**

<https://wiki.cancerimagingarchive.net>

**MICCAI2015:**

<http://endovis.grand-challenge.org/>

**NEO2015:**

<http://neatbrains15.isi.uu.nl/>

POLYP2015:

<http://www.polyp2015.com/wp/>

VISCERAL2015:

<http://biomedicalimaging.org/2015/>

VISCERALesion2015:

<http://www.visceral.eu/benchmarks/detection/>



## APPENDIX 1

Examples of Biomedical Imaging Competitions from the past 10 years:

Table Appendix 1: 2014 – 2007 example of competitions.

| <b>Year</b> | <b>Competition</b>   |
|-------------|--|
| <b>2007</b> | Brain Caudate Nucleus Segmentation<br>Liver CT Scan Segmentation   |
| <b>2008</b> | MS Lesion Segmentation Challenge<br>3D Liver Tumor Segmentation Challenge  |
| <b>2009</b> | Automatic Detection of Pulmonary Nodules in Chest CT Scans<br>Extract airway tree from thoracic CT scans<br>Rotterdam Coronary Artery Algorithm Evaluation Framework<br>Carotid Bifurcation Algorithm Evaluation Framework<br>Head & Neck Auto –Segmentation (Mandible and Brainstem)<br>Cardiac MR Left Ventricle Segmentation<br>Prostate Segmentation<br>Volume Change Analysis of Nodules<br>Retinopathy Online Challenge<br>Segmentation Validation Engine for Brain vs Non-brain MRI |
| <b>2010</b> | Accurate Registration of Thoracic CT<br>Knee Cartilage Segmentation<br>Automated Neuronal Reconstruction<br>Head & Neck Auto-Segmentation of the Parotid Glands  |
| <b>2011</b> | Lumen and External Elastic Laminae Border Detection in IVUS<br>Lobe and LUNG Analysis<br>Motion Tracking Challenge<br>4D Left-Ventricular (LV) Segmentation Challenge<br>Diffuser Tensor Imaging Tractography for Neurosurgical Planning Overview  |

|                    |   |
|--------------------|---|
| <p><b>2012</b></p> | <p>Mitosis Detection in Breast Cancer Histological Images<br/> Multi-Atlas Labeling<br/> Automatic segmentation algorithms for MRI of the prostate<br/> Alzheimer’s disease: A challenge to access measurement reliability and bias<br/> Cardiac MRI Segmentation Challenge<br/> Diffusion Tensor Imaging Tractography Challenge<br/> Multimodal Brain Tumor Segmentation<br/> Automatic Segmentation algorithm for neonatal brain tissues<br/> Coronary Artery Stenoses Detection and Evaluation Framework<br/> Lung vessel segmentation<br/> High angular resolution diffusion imaging<br/> Cardiac Delayed Enhancement MRI Segmentation<br/> Segmentation of Neuronal structures in Electron Microscopy stacks<br/> Particle tracking challenge<br/> Biometric measurements from fetal ultrasound images</p> |
| <p><b>2013</b></p> | <p>Chest Radiograph Anatomical structure segmentation<br/> Subsolid lung nodule segmentation<br/> Whole body labeling in 3D medical imaging data<br/> Multiparametric Brain Tumor Segmentation<br/> Automated Segmentation of Prostate Structures<br/> Localization Microscopy Challenge-Diffusion Tensor Imaging<br/> 3D Segmentation of Neurites in Electron Microscopy Images<br/> Computer Aided Detection of Pulmonary Embolism<br/> 3D Deconvolution Microscopy challenge<br/> High angular resolution diffusion MRI reconstruction challenge</p>   |
| <p><b>2014</b></p> | <p>Multiclass classification for Alzheimer’s disease<br/> Detection of mitosis and evaluation of nuclear atypia on breast cancer<br/> Spine and Vertebra Segmentation Challenge<br/> Lesion Detection Benchmark in Anatomical Regions<br/> Brain Tumor Image Segmentation</p>   |

|  |  |
|--|--|
|  | <p>Subthalamic Nucleus Segmentation Challenge</p> <p>Brain Tumor Digital Pathology Challenge</p> <p>Leaf Segmentation Challenge</p> <p>Automatic Segmentation of Coronary Artery Calcium Scoring in Cardiac CT</p> <p>Automated tracking of anatomical landmarks in liver ultrasound</p> <p>Diffusion Tensor Imaging Tractography Challenge – Peritumoral Anatomy</p> <p>Automatic Liver CT annotation of 3D liver data</p> <p>3D Cardiac Ultrasound Segmentation Challenge</p> <p>Machine Learning Challenge on Brain Neuroimaging</p> <p>Statistical Shape Model Challenge for Liver</p> <p>3D Deconvolution Microscopy for image reconstruction</p> <p>Cell Tracking Challenge for 2D and 3D time lapse microscopy videos</p> <p>Cephalometric X-Ray Landmark Detection</p> <p>Bone Texture Characterization to identify Osteoporosis</p> <p>Overlapping Cervical Cytology Image Segmentation Challenge</p> <p>Left Atrial Segmentation Challenge</p> |
|--|--|

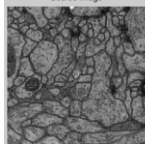
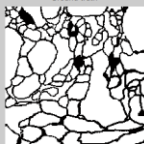

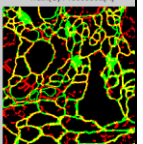
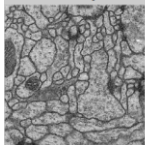
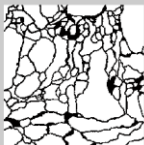

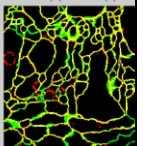
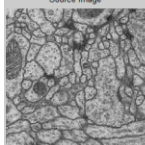
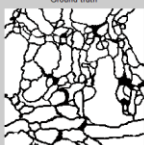

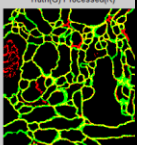
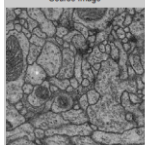

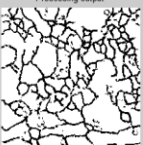
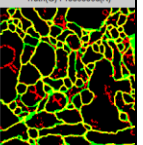
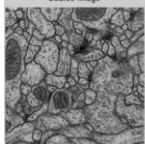
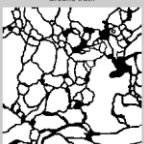
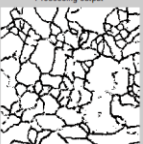
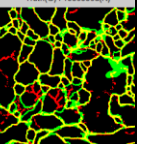
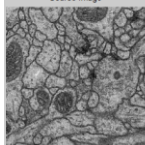

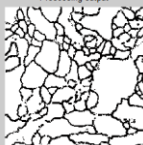
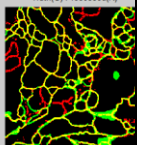
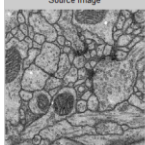
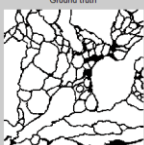

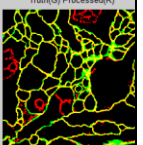
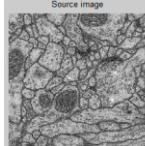

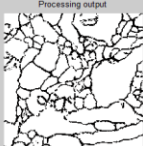
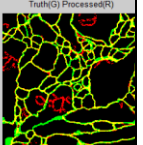
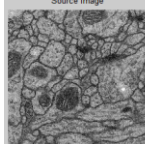
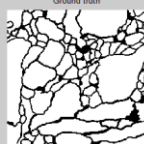

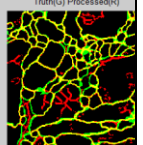
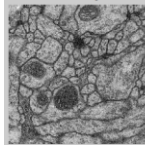
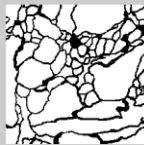
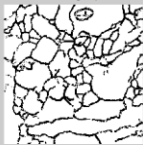
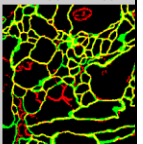
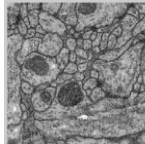
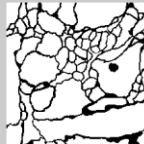

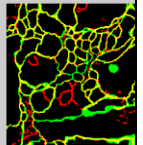
## APPENDIX 2

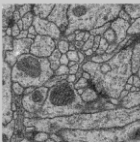
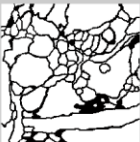

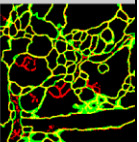
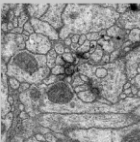

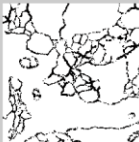
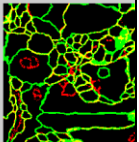
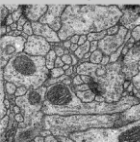


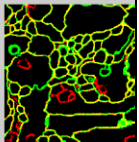
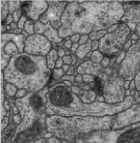


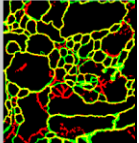
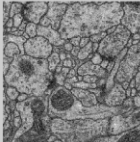


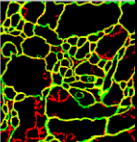
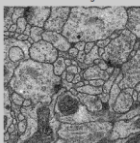

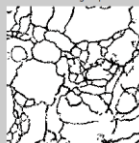
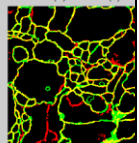
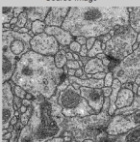

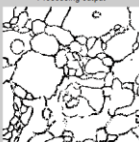
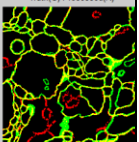
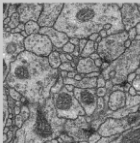


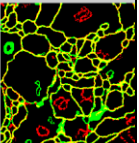
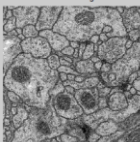
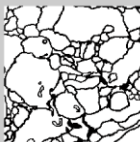

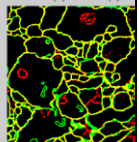
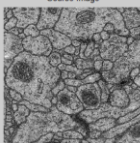
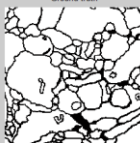

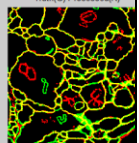
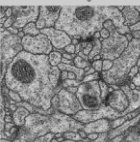
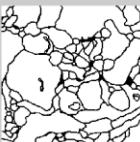
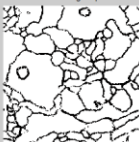
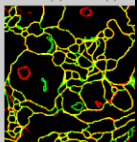
Examples of IPCO Output Result for 30 slices with its corresponding performance.

Outputs are depicted for each one the slices ( best chain), together with the colored true/false +/- maps, and F1 scores.

Table Appendix 2: Examples of IPCO Output Result for 30 slices with its corresponding performance.

| Img | Precisio | Recall  | F1 Score | Output Result |
|-----|----------|---------|----------|---------------|
| 1   | 0.89777  | 0.95289 | 0.92451  |               |
| 2   | 0.90116  | 0.93759 | 0.91901  |               |
| 3   | 0.88475  | 0.95421 | 0.91817  |               |
| 4   | 0.89652  | 0.94347 | 0.91939  |               |
| 5   | 0.8671   | 0.9291  | 0.89703  |               |
| 6   | 0.84878  | 0.92317 | 0.88441  |               |
| 7   | 0.89702  | 0.93792 | 0.91701  |               |
| 8   | 0.87224  | 0.90359 | 0.88764  |               |

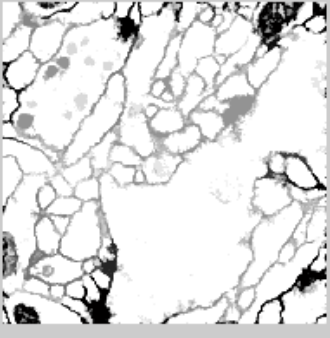
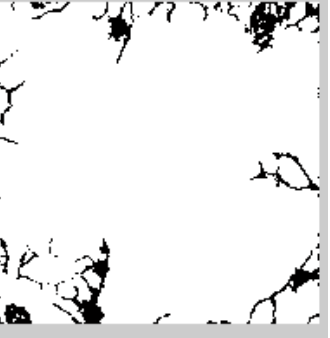
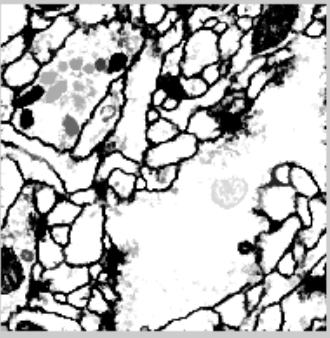
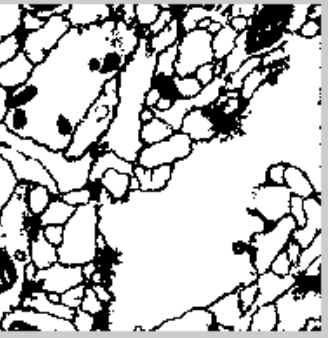
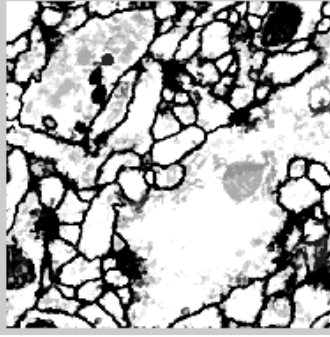

|    |         |         |         |  |
|----|---------|---------|---------|--|
| 9  | 0.84165 | 0.84842 | 0.84502 |             |
| 10 | 0.90369 | 0.96148 | 0.93169 |             |
| 11 | 0.87301 | 0.95932 | 0.91413 |             |
| 12 | 0.8966  | 0.92926 | 0.91264 |             |
| 13 | 0.87027 | 0.94802 | 0.90748 |             |
| 14 | 0.91154 | 0.94041 | 0.92575 |         |
| 15 | 0.93288 | 0.95066 | 0.94169 |     |
| 16 | 0.90341 | 0.96308 | 0.93229 |     |
| 17 | 0.91149 | 0.93467 | 0.92293 |     |
| 18 | 0.88477 | 0.94891 | 0.91572 |     |
| 19 | 0.89819 | 0.95401 | 0.92526 |     |

|    |         |         |         |   |  |   |   |
|----|---------|---------|---------|---|--|---|---|
| 20 | 0.88811 | 0.9604  | 0.92284 |    |    |    |    |
| 21 | 0.82738 | 0.96927 | 0.89272 |    |    |    |    |
| 22 | 0.89581 | 0.9709  | 0.93185 |    |    |    |    |
| 23 | 0.90444 | 0.94058 | 0.92215 |    |    |    |    |
| 24 | 0.91723 | 0.94637 | 0.93157 |    |    |    |    |
| 25 | 0.9235  | 0.94879 | 0.93597 |   |   |   |   |
| 26 | 0.89906 | 0.96784 | 0.93218 |  |  |  |  |
| 27 | 0.90394 | 0.96301 | 0.93254 |  |  |  |  |
| 28 | 0.88792 | 0.96326 | 0.92406 |  |  |  |  |
| 29 | 0.91182 | 0.94353 | 0.9274  |  |  |  |  |
| 30 | 0.9314  | 0.95696 | 0.94401 |  |  |  |  |

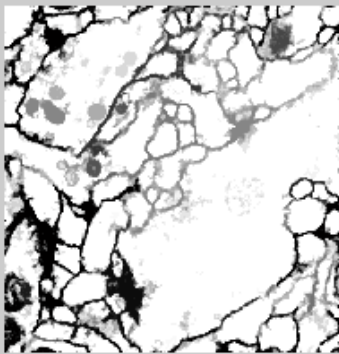
### APPENDIX 3

Ensembles output result for TestImage:

Table Appendix 3: Ensembles output result for TestImage.

| Test Images | Average Out   | Classify Out  |
|-------------|---|---|
| Slice 1     |    |    |
| Slice 2     |   |   |
| Slice 3     |  |  |

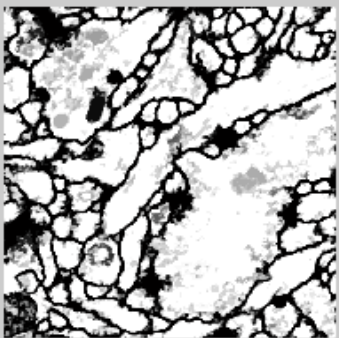
Slice 4



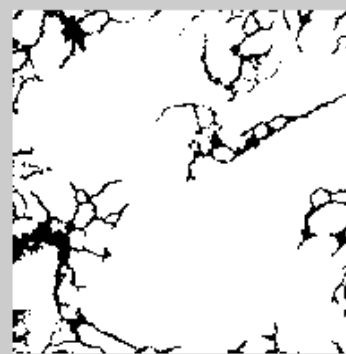
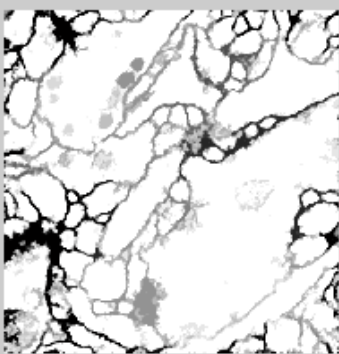
Slice 5



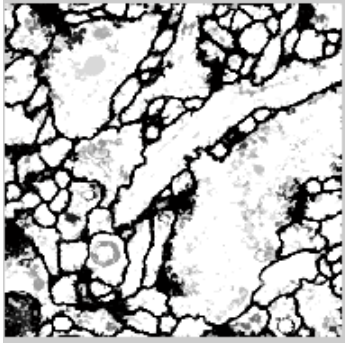
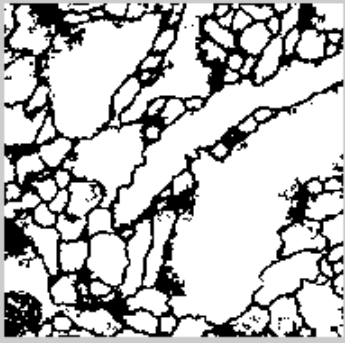
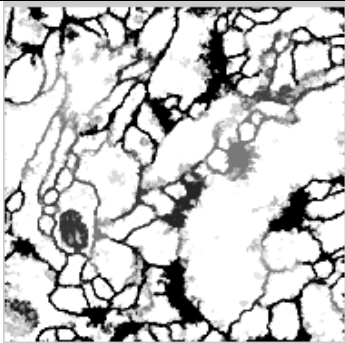
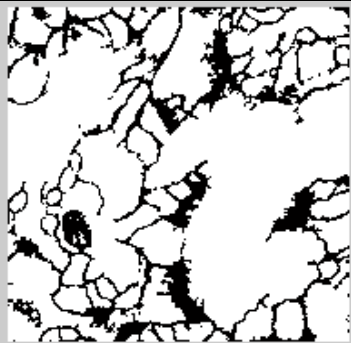
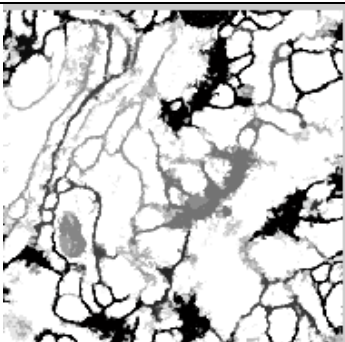
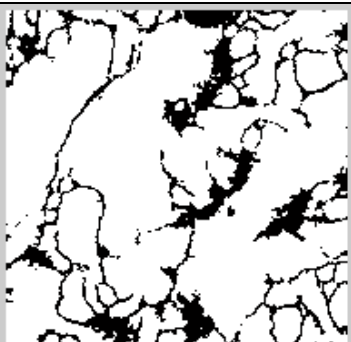
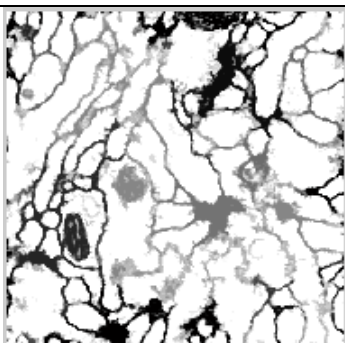
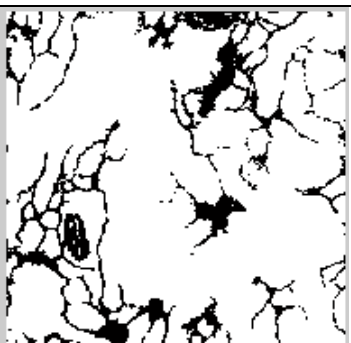
Slice 6



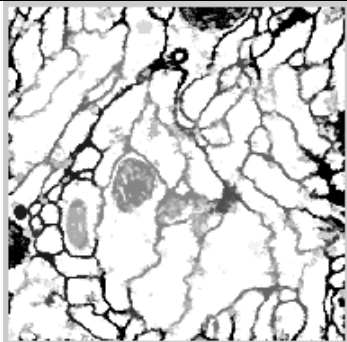
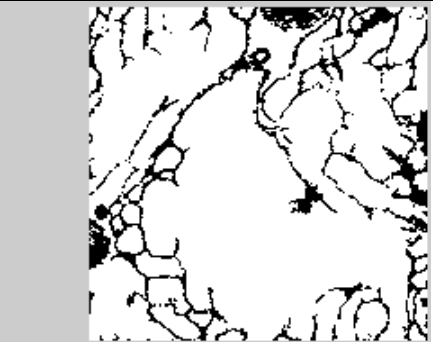
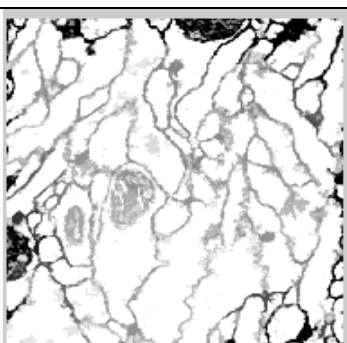
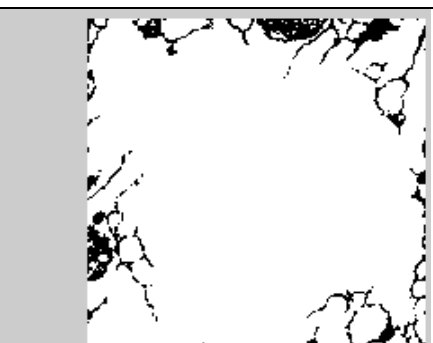
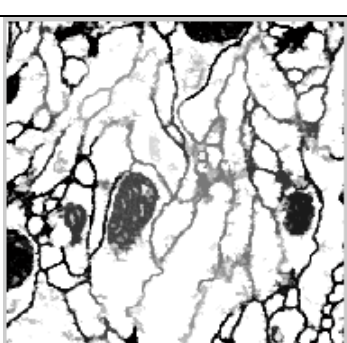
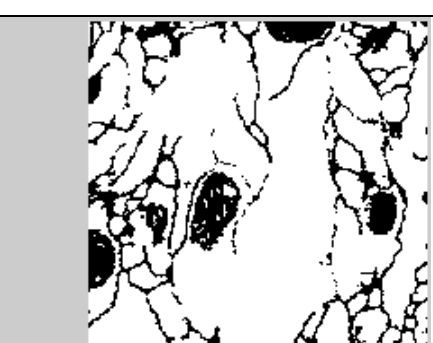


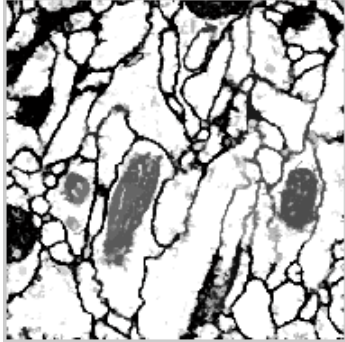

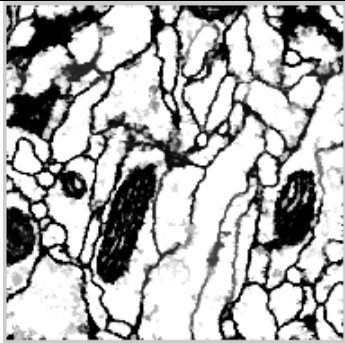

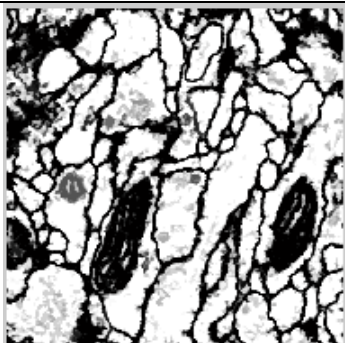

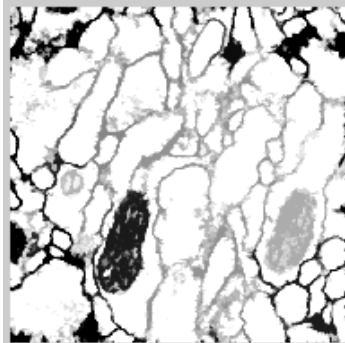
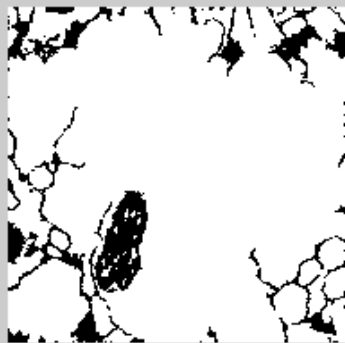
Slice 7

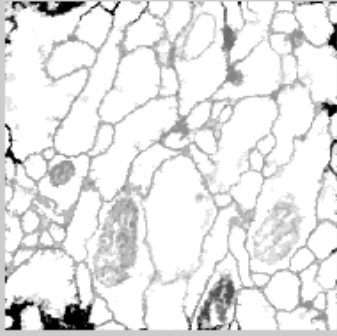

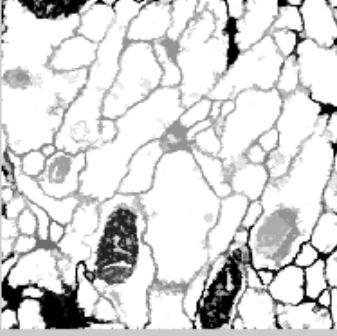
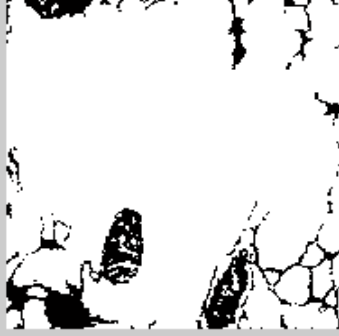
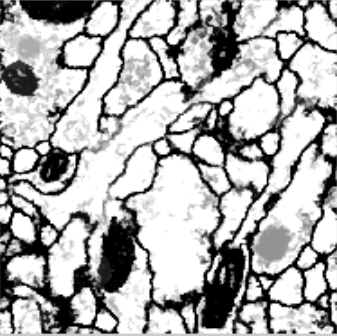

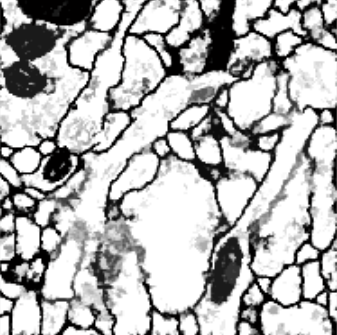



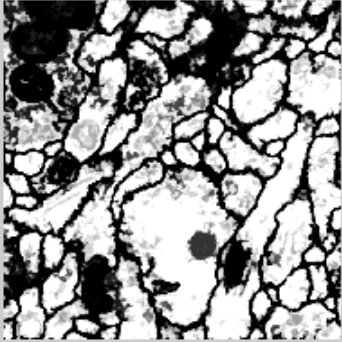
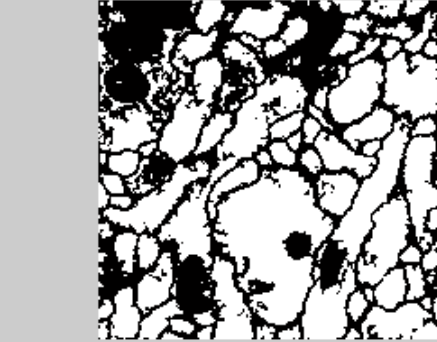
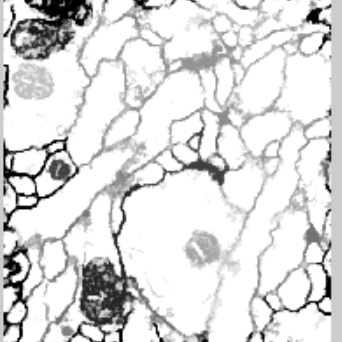
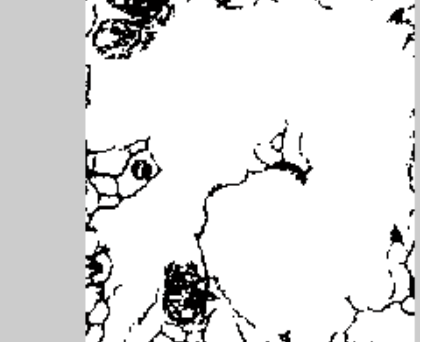
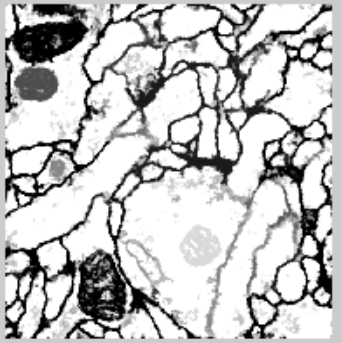

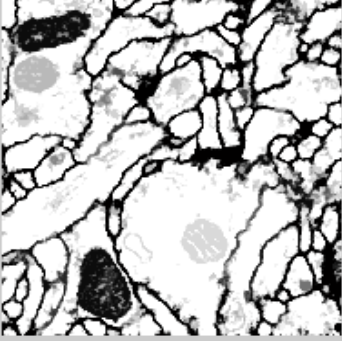
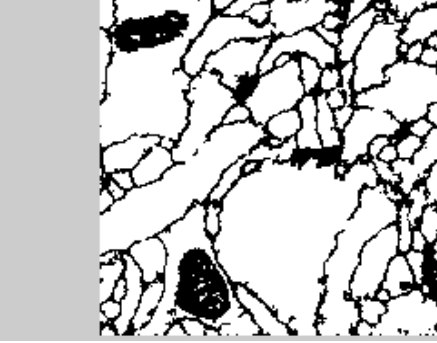


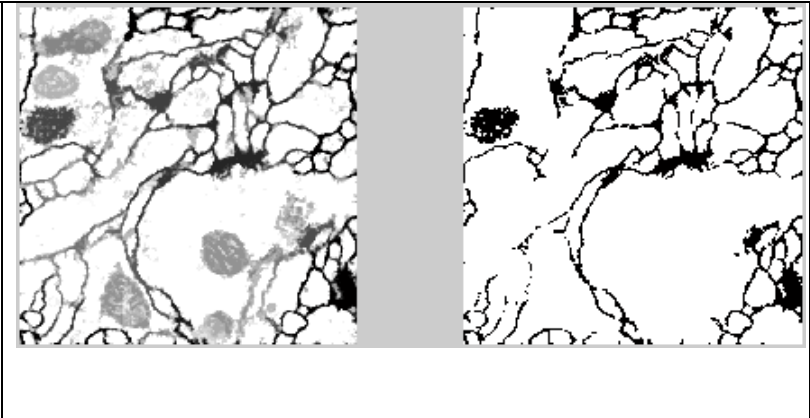
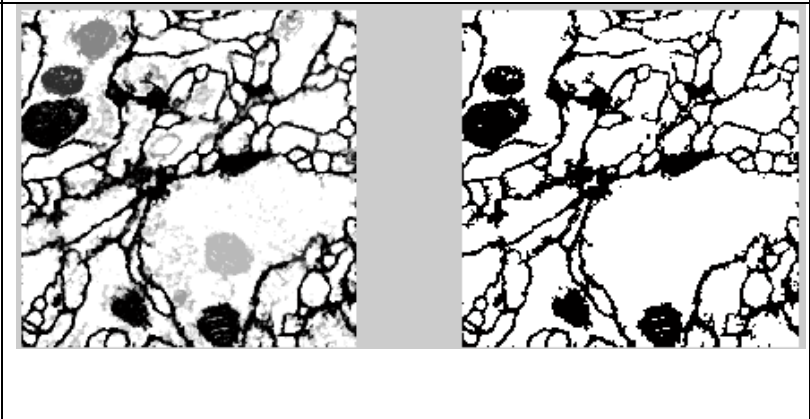
|                 |   |   |
|-----------------|---|---|
| <p>Slice 8</p>  |    |    |
| <p>Slice 9</p>  |    |    |
| <p>Slice 10</p> |  |  |
| <p>Slice 11</p> |  |  |

|          |   |  |
|----------|---|--|
| Slice 12 |    |    |
| Slice 13 |    |    |
| Slice 14 |  |  |
| Slice 15 |  |  |

|                 |   |   |
|-----------------|---|---|
| <p>Slice 16</p> |    |    |
| <p>Slice 17</p> |    |    |
| <p>Slice 18</p> |  |  |
| <p>Slice 19</p> |  |  |

|          |   |   |
|----------|---|---|
| Slice 20 |    |    |
| Slice 21 |    |    |
| Slice 22 |  |  |
| Slice 23 |  |  |

|                 |   |  |
|-----------------|---|--|
| <p>Slice 24</p> |    |    |
| <p>Slice 25</p> |    |    |
| <p>Slice 26</p> |   |   |
| <p>Slice 27</p> |  |  |

|          |  |
|----------|--|
| Slice 28 |  The image for Slice 28 consists of two side-by-side panels. The left panel is a grayscale micrograph showing a cross-section of a vascular bundle. A large central vessel is surrounded by smaller cells, some of which are stained dark. The right panel is a binary (black and white) version of the same image, where the cell walls and dark-stained cells are rendered as black shapes against a white background. |
| Slice 29 |  The image for Slice 29 also consists of two side-by-side panels. The left panel is a grayscale micrograph showing a cross-section of a vascular bundle, similar to Slice 28 but with a different arrangement of cells and staining. The right panel is a binary (black and white) version of the same image, highlighting the cell walls and dark-stained cells.   |