J. Comp. Sci. Comm., 1(2)1-11, 2015
© 2015, TextRoad Publication
ISSN 2356-8844
Journal of Computer Sciences and Communication
www.textroad.com

Multiple Image Processing Chain Analysis for Membrane Detection – Optimization Outcome

Rajeswari Raju[a,c,]* , Tomas Maul[a], and Andrzej Bargiela[b]

[a]University of Nottingham, Faculty of Science, School of Computer Science, Malaysia Campus, Semenyih, 43500, Selangor, Malaysia
[b]University of Nottingham, Faculty of Science, School of Computer Science, Jubilee Campus, Nottingham,NG81BB, United Kingdom
[c]University of Technology MARA, Faculty of Computer and Mathematical Sciences, 23000 Terengganu, Malaysia

ABSTRACT

This paper is part of the ongoing effort to improve automated neuronal membrane detection, where the core challenge consists of distinguishing membranes from organelles. Earlier we did some work on single sequence (or chain) Image Processing Optimization (IPCO). In this paper, we optimized multiple sequences (or chains) of image processing (MIPCO) using a global stochastic optimization approach by combining elements of Genetic Algorithms (GA), Differential Evolution (DE) and rank-based uniform crossover (RBUC). MIPCO performedwith an average F1 score of91.8%,which is slightly higher than the average performance of our previous method, IPCO. Our approach is both efficient and interpretable, and facilitates the generation of new insights. In this paper too, we highlight an observation pertaining to morphological operators and their appearance in an unorthodox position in image processing chains, and suggest a new set of pipelines for image processing.
KEYWORDS: Image processing, Optimization, Segmentation, Morphological Operators, Membrane detection, Neuronal structures, Transmission Electron Microscopy data

1.0 INTRODUCTION

1.1 Main Research Goal

The main goal of our research is to optimize Image Processing chains, and analyze the best approaches with the highest accuracy levels. We compare the differences and similaritiesbetween different single (Single Image Processing Chain (IPCO) [1]) and multiple (Multiple Image Processing Chain (MIPCO)) chains, and extract several useful insights for image processing users. This paper highlightsthe main insights we have gathered so far.

2.0 Background Study

2.1 Genetic Algorithm (GA) and Global Stochastic Optimization

GA is a method to solve both constrained optimization problems, which optimize an objective function with respect to some variables in the presence of constraints on those variablesand unconstrained optimization problems, which consider the problem of minimizing an objective function that depends on real variables with no restrictions on their values. This method solves the problem based on a natural selection process, and repeatedly modifies a population of individual solutions. Global optimization algorithms are typically good at avoidinglocal minima and can be further sub-divided into deterministic and stochastic variants [2]. According to Uryasav[3], simulated annealing, GA, evolutionary strategies and programming are examples of stochastic search algorithms. Theoretically, this stochastic global approach is good at exploring the potential solution space widely, but isgenerally slow at finding the local optimum, once the algorithm has found a good area of the solution space. So, to get the best result by using this approach, it is common to combine local search with global search by using the weights obtained from the global search as starting values for the local search [4,5].

2.2 Differential Evolution (DE)

DE is favored because of its' 3 main (arguable) advantages, i.e.:(1) ability to find the true global minimum(regardless of initial parameter values), (2) limited use of control parameters, and (3) fast convergence. DE uses operators which are related to those of Genetic Algorithms (GA), i.e.: crossover, selection and mutation. According to Nurhan and Bahadir [6], when considering global optimization methods for filter design, GA is a good choice. Filters designed by GA have the potential of obtaining near global optimality[7].However, in terms

Abbreviations: ISBI, international symposium on biomedical imaging; IPCO, image processing chain optimization, MIPCO, multiple image processing chain optimizations.

***Corresponding author:** Rajeswari Raju, University of Nottingham, Faculty of Science, School of Computer Science, Malaysia Campus, Semenyih, 43500, Selangor, Malaysia Tel.: +6-013-398-5678; e-mail: khyx1rru@nottingham.edu.my

of convergence speed, it hasdisadvantages which can be partly addressed by differential evolution, which is a simple and yet powerful evolutionary algorithm first introduced by Storn and Price [8]. Early in the literature, according to Nurhan Karaboga and Bahadir Cetinkaya, the DE algorithm was not as common as GA[6], but it has picked up tremendously over the years partly due to its effectiveness and partly due to its relative simplicity. DE has been convincingly successful in solving single-objective optimization problems [9], and several researchers are trying to match this success in the domain of multi –objective optimization problems.

2.3 Rank Based Uniform Crossover

Uniform crossover was first proposed by Ackley [10]. The operator has been successfully used in several different applications (e.g. (Duarte-Mermoud et al., in his 2013 publication, [11])) and has been studied theoretically at some length (e.g. (Chicano et al., in his recent publication, 2014, [12])). 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 in (Semenkin and Semenkina, [13])), in our study, we bias P towards the stronger solution, and therefore set P = 0:75. This bias towards the stronger parent, is reflected in the "rank based" half of the term rank-based uniform crossover (RBUC).

3.0 METHODOLOGY

3. 1 Image Processing Platform - MatLab and the Image Processing Toolbox

Our approach is based on a sequence of basic image processing steps, most of which we adopted from MatLab's image processing toolbox by MathWorks. This toolbox is useful for the processing, visualization and analysis of images, whilst MatLab is convenient for rapid prototyping.

3.2 Data

The experiments were performed on data provided by the ISBI 2012 (IEEE International Symposium on Biomedical Imaging) challenge: "Segmentation of neuronal structures in Electron Microscopy (EM) stacks"[14]. Albert Cardona and team provide public access to 30 slices of Transmission Electron Microscopy (TEM) images with corresponding ground-truth images for training, and a second set of 30 TEM images for testing [15].

3.3 Performance Measures

The performance of the algorithm was measured in terms of Precision, Recall and F1 score.

$$\text{Precision} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Positive (FP)}} \qquad (2)$$

Where:

TP : The number of pixels correctly labeled as belonging to the positive class
FP :The number of pixels incorrectly labeled as belonging to the positive class

$$\text{Recall} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Negative (FN)}} \qquad (3)$$

Where:
FN :The number of pixels incorrectly labeled as belonging to the negative class

$$\text{F1} = 2\left(\frac{\text{Precision} * \text{Recall}}{(\text{Precision} + \text{Recall})}\right) \qquad (4)$$

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 where an F1 score reaches its best value at 1 and worst score at 0.

For each slice, a confusion matrix was computed followed by corresponding precision(2), recall(3) and F1 scores(4). The final performance values were averaged from the results corresponding to each one of the 30 slices.

3.4 Multiple Image Processing Chain Optimization (MIPCO)

3.4.1 Motivation

The core aim underlying IPCO (our earlier successful approach) consists of the design and implementation of a simple, computationally efficient and easily adopted method for cellular membrane detection. To further enhance our approach for accuracy, we here introduce MIPCO. In our previous research, we divided our effort into 4 main stages:

i) Semi-automated stage to obtain a decent sequence of functions and parameterizations for membrane detection –algorithm known as Local Contrast Hole Filling (LCHF) Algorithm [16].
ii) Automated stage - We optimized sequences (or chains) of image processing functions using a global stochastic optimization approach, the overall process of which we refer to as Image Processing Chain Optimization (IPCO) [17]
iii) Analysis stage – Through the analysis of relatively large sets of optimal chains, we discoveredseveral interesting and useful facts pertaining to pre and post processing [18]
iv) In order to further boost performance, we created ensembles from several high-scoring IPCO chains.

MIPCO is the result of the effort to further boost the performance of IPCO.

3.4.2 Conceptual Summary of MIPCO

MIPCO, is essentially the direct application of global stochastic optimization to multiple image processing chains.These chains run in parallel and can exchange intermediate information. MIPCO is fully automated and its optimization process incorporates elements of Genetic Algorithms (GA), Differential Evolution (DE) and rank-based uniform crossover (RBUC) to obtain a more robust approach. The optimization algorithm has several basic image processing functions available to it [16] which it configures in different sequences and with different parameter settings, in response to the behaviour of the cost function, defined as the F1 score relative to a subset of the ISBI training images.

Below are fluxogramspertaining to the general framework underlying IPCO and MIPCO chains.

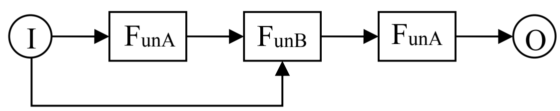


Figure1: Flowchart showing the overall computational flow in a specific **IPCO** chain consisting of three functions. I: input image. O: output image. F_{unA}: single-input function such as denoising. F_{unB}: multiple-input function such as image blending.

Figure 1 shows the flowchart of an IPCO chain consisting of three functions. In our experiments IPCO chains were typically allowed to use a maximum of 8 functions [1][16].

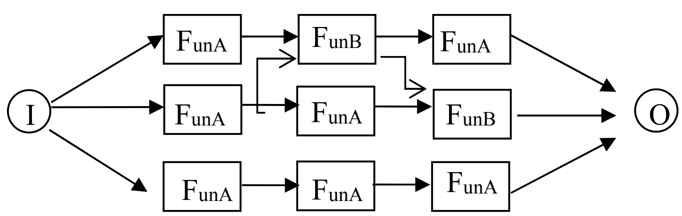


Figure 2: Flowchart showing the overall computational flow in a specific **MIPCO** chain consisting of three functions, and 3 layers of chains (for illustration purposes). I: input image. O: output image. F_{unA}: single-input function such as denoising. F_{unB}: multiple-input function such as image blending.

Figure 2 shows the flowchart of a MIPCO chain consisting of three functions and multiple layers of chains (3 layers). In our experiments MIPCO chains were allowed to use a maximum of 8 functions per chain, and 5 parallel chains.

In the 3rd stage of our earlier experimental designwith IPCO, we confirmed that ensembles of IPCO chains (where each chain is optimized separately) perform better than single chains. This finding partly motivated the simultaneous optimization of multiple and interacting chains (i.e. MIPCO).

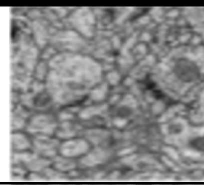
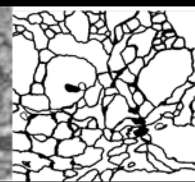
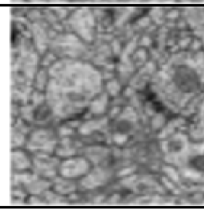
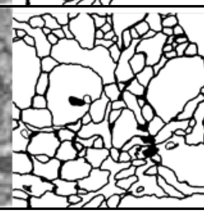
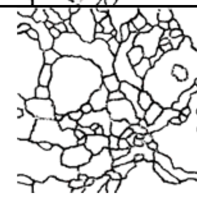
Algorithm	Score (Individual)	Original / Ground Truth	Output
IPCO	91.67 with max 8 function	 	
MIPCO	91.80 with max 8 function and 5 chains	 	

Figure 3: Shows the visual comparison and score of IPCO and MIPCO

Figure 3 shows the visual comparison and score of IPCO vs MIPCO. MIPCO performs generallybetter than IPCO with a higher F1 score (0.13%), as of the time of writing.

3.4.2 Functions

MIPCO consists of chains or simple networks of image processing functions optimized via a global stochastic optimization algorithm, which combines elements of genetic algorithms, differential evolution and rank based uniform crossover.The optimization algorithm has several basic image processing functions available to it, which are typically found in standard image processing libraries such as the MatLab Image Processing Toolbox (by MathWorks). These 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 3 broad categories, i.e.: pre-processing, classification and post-processing. The two main types of pre-processing functions currently being used consist of denoising and contrast enhancement. The three main types of classification functions consist of thresholding, hole-filling and watershed. Post-processing functions include smoothing via combining functions and morphological operators. Note that the categorization of function types into pre-processing, classification and post-processing, is based on their typical usage and interpretation, and that optimization often finds unexpected ways to use functions (e.g. in some chains, denoising operators have been found in the middle of said chains)[17][18].The functions used in MIPCO are the same as those used in IPCO. 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. So, a layer must complete all computation before the next layer can initiate its own computation. Please refer to the diagram in Figure 2 for a simple view of the computational flow in MIPCO.

3.4.2.1 Training MIPCO

In order to compute the cost function, we typically take a subset of slices 1 and 2, which accelerates the optimization process dramatically without excessively deteriorating accuracy(after optimization, all chains typically have F1 scoreslarger than 90%). MIPCO's optimization process runs continuously until a target cost of 0 has been reached or a maximum of 10,000 generations has been completed, whichever occurs first. MIPCO can lead to a diverse set of useful chains, 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 our research, chains were allowed to have a maximum number of eight basic functions, and a maximum number of 5 chains, although the total pool of functions was much larger. In general, functions can appear in any order, and there is no restriction in order, and can even repeat several times in a chain. Each function typically comes along with a small set of parameters which also undergoes optimization (e.g. tile size for the contrast function). Generally speaking, it doesn't take long to optimize a chain for different types of data (typically less than 1000 optimization generations). MIPCO 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 x 343 pixels can be done in about 20 seconds per image on an average personal computer (i.e. 1.60 GHz processor and 1.48 GB of RAM) for 3 chains with 8 functions. Moreover, there is no requirement for specialized hardware.

3.4.3 Measuring Performance

For most experimental designs involving MIPCO, we focus on analyzing the properties of good quality chains. In general, we define "good quality" chains as those that obtain F1 scores larger or equal to 91%. For this purpose, we used publicly available training/test datasets (Droshopila TEM Images from ISBI2012)[14]. In order to more efficiently test our chains (since the ground truth of the ISBI2012 test images are not public), we have separated some of the ISBI2012 training set images and labels and used them for testing/validation purposes. In the above Section, we have given the calculation method for the performance measurement.

4.0 EXPERIMENTAL RESULTS

4.1 Best Shortest MIPCO functions

We ran several experiments testing different chain sizes and different numbers of chains. We discovered that even with a small number of chains (3 chains), MIPCO could still perform very well, consistently reaching F1 scores larger than 91%. Table 2shows the smallest MIPCO cases with F1 scores larger than 91%.

4.2 Interesting Observation – Morphological Operators

An interesting observation that can be made pertaining to morphological functions,consists of the appearance ofmorphological 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 a post-processing phase. In our experiments, it seems that although Morphological Operators, are frequently encountered at a post- processing phase, they do also appear in various other positions in the MIPCO chain.Moreover the appearance of this operator in atypical positions doesseem to contribute to better performance. Also note that morphological operators are not the only type of function to be found in post-processing smoothing. This is also the case with denoising functions [17,18]which we have also found to exist in unorthodox positionsas reported in our earlier publication.In general, optimization often finds unexpected ways to use functions (e.g. morphological operators have been found performing classification in some chains).

Two basic morphological operators used in the experimentsreported here are morphological 'open' (erosion followed by a dilation process), and erosion.

Below is an example ofan image being processed with morphological operators 'open' and 'erode'.

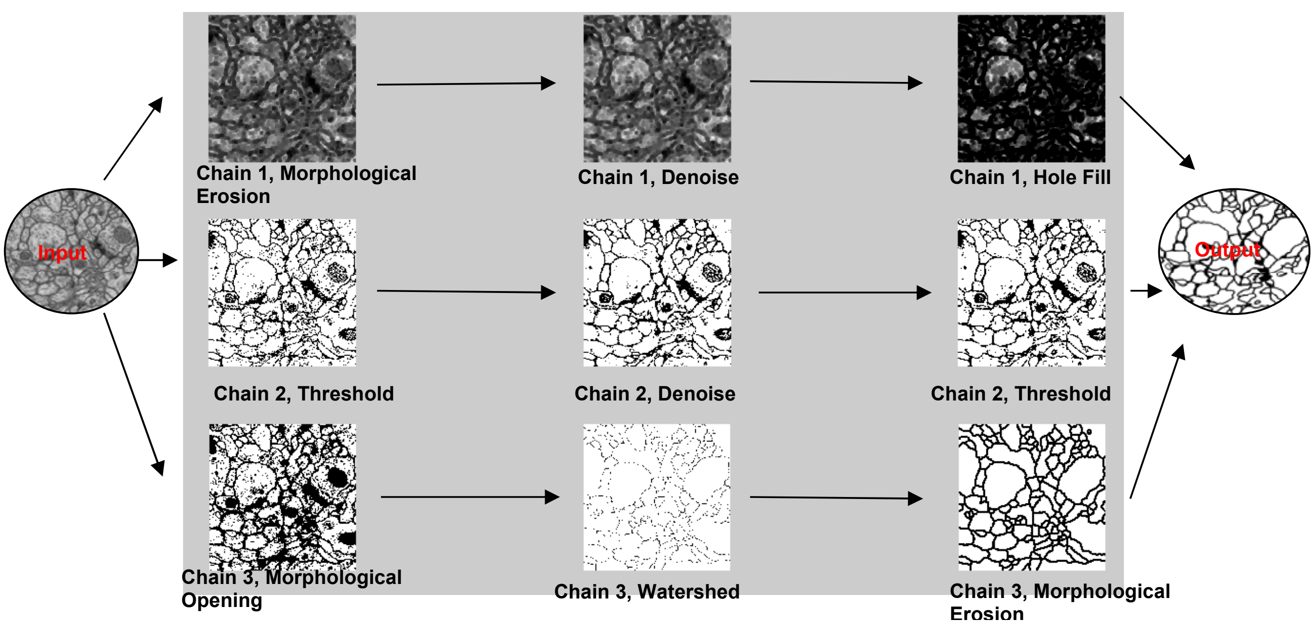


Figure 4: The selection of Morphological Operators at different chain positions

Our experiments show that MO can appear in unorthodox chain positions. As we can see in Figure 4, our MIPCO experiments show that, at least for this membrane segmentation problem, MO appear in early (in Chain 1, as Morphological Erosion, in Chain 3, as Morphological Opening) and final stages (in Chain 3 as Morphological Erosion. As we can see in Figure 4, the morphological operator erosion seems to appear early in the chain (as a 1st function), which arguably runs contrary to common expectation, that morphological operators are used typically for post-processing. The insight that morphological operators can often perform useful computations in atypical positions of image processing pipelines,is something that needs to be taken into account by image processing users. In other words, we should not always restrict morphological operatorsto the final stages of our pipelines. According to our experiments, the utilization of morphological operators in early stagescan have a positive effect on accuracy. We discovered that the chains with morphological operatorsat early or middle regions of pipelines do tend to show higher F1 scores. Below in Table 1, we show the scores and chain positionsfor the "earliest morphological operators", dividing networks into three categories characterized by scores (>91%, between 90% to 91% and <90%). These resultsdepictnetworks that exhibited a maximum of 3 chains with a maximum of 8 functions each.

Table 1: Position of earliest morphological operators(grey row: average scores and positions)

Scores >91%	Position of Appearance of The Earliest Morphological Operator	Scores >90 but< 91%	Position of Appearance of Morphological Operator	Scores < 90%	Position of Appearance of Morphological Operator
91.43	1st	90.20	5th	89.64	6th
91.38	1st	90.00	4th	89.00	7th
91.23	3rd	90.99	3rd	88.22	7th
91.21	2nd	90.50	4th	89.03	6th
91.16	2nd	90.01	3rd	89.33	5th
91.14	1st	90.91	4th	89.56	6th
91.09	2nd	90.96	4th	88.20	8th
91.01	2nd	90.72	3rd	89.91	5th
91.33	1st	90.63	3rd	88.87	8th
91.10	2nd	90.03	5th	89.23	6th
91.20	1.7	90.50	3.8	89.10	6.4

As per Table 1, we can say that, 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 a middle stage and 89.1% denotes the average accuracy of those chains that have at least one morphological operator (MO) at a final stage. From Table 1, we can clearly see that having atleast one MO at an early stage has a positive impact on performance, compared to having MOsat later stages.

4.2.1 Insight of Morphological Operators

Morphological operators(MO) rely on the relative ordering of pixel values, and are well suited to the processing of binary images. They probe an image with a small translated shape called 'structuring element'[19]. The 'structuring element', is positioned at all possible locations and is then compared with its neighbor (whether it fits or hits or intersects the neighbor). Figure 5represents the 3 positions of the operation and Figure 6illustrates several examples of simple structuring elements, (e.g.5x5).

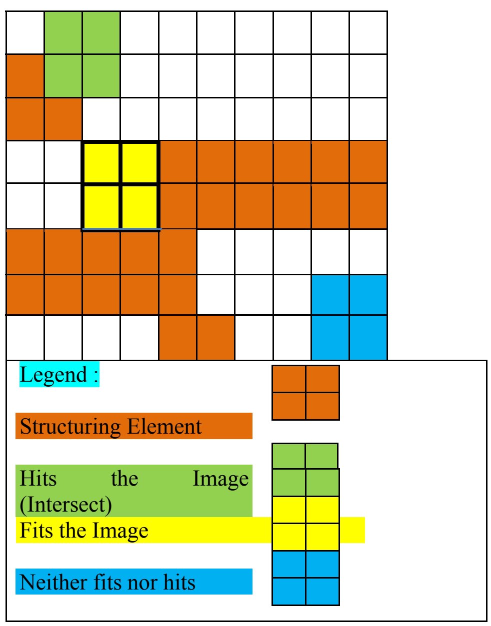


Figure 5:Probing an image with a structuring element.
(Brownand White pixels have Non zero and Zero values respectively

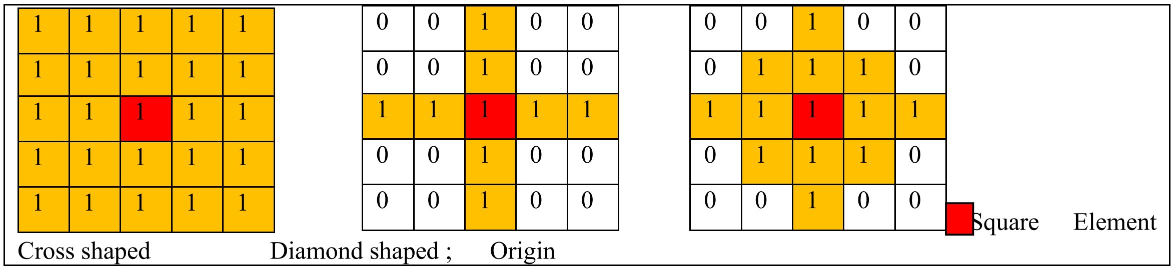


Figure 6: Examples of simple structuring elements (5x5)

Morphological operations such as erosion, dilation, opening, and closing are usedto perform morphologicalimage analysis [20, 21]. Morphological operations apply structuring elements to an input image, creating an output image of the same size. Morphological operators, thus, can directly deal with shape information with the help of a structuring element, which may be viewed as a probe. Morphological algorithms closely resemble the human strategy of image understanding, as both of them are neither fully subjective nor fully objective, but a judicious combination of the two. In mathematical morphology, the operations are precisely defined, but the selection of the structuring element is an ad-hoc process and depends on the application and the data [22].

Table 2: Best shortest function (with 3 chains and maximum of 8 functions) for MIPCO that score >91%

No	F1 Scores	Chain 1	Chain 2	Chain 3
1	91.01	Double Thresh Denoise Median MorphErode	MorphOpen DoubleThresh Denoise Median	MorphOpen Watershed MorphErode
2	91.38	MorphErode Denoise Median HoleFill	Double Thresh Thresh Simple Denoise Median	MorphOpen Watershed MorphErode
3	91.43	MorphOpen Denoise Median HoleFill	DoubleThresh Denoise Median Double Thresh	MorphOpen Watershed MorphErode
4	91.23	Combine MinMax Denoise Median Double Thresh	Combine MinMax Denoise Median Morph Open	MorphOpen Watershed MorphErode
5	91.21	Double Thresh Denoise Median Thresh Simple	MorphOpen Watershed MorphErode	Double Thresh Morph Erode Combine Multilpy
6	91.14	MorphOpen Watershed MorphErode	Double Thresh Denoise Median Hole Fill	Denoise Median Double Thresh Denoise Median
7	91.09	Denoise Median Morph Erode Hole Fill	MorphOpen Watershed MorphErode	Combine MinMax Double Thresh Morph Erode
8	91.16	Double Thresh Denoise Median Thresh Simple	MorphOpen Denoise Wiener Combine Subtract	MorphOpen Watershed MorphErode
9	91.09	Denoise Median Morph Erode Hole fill	MorphOpen Watershed MorphErode	CombineMinMax Double Thresh MorphErode
10	91.38	Morph Erode Denoise Median Hole Fill	Double Thresh Thresh Simple Denoise Median	MorphOpen Watershed MorphErode

Table 2shows the 10 best shortest functions for MIPCO that score >91%. In these 10 chains, we can observe that morphological operators appear in all chains, and in unorthodox positions. In some cases, they are foundat the beginning of chains, and in other cases they are found in the middle or ends of chains.

4.2 Function Selection

As for function selection analysis, Tables 3 and 4show the frequency of function appearance in 1 chain for 50 trials for our earlier single chain approach, Image Processing Chain Optimization (IPCO), and our new approach, multiple chain approach, Multiple Image Processing Chain Optimization (MIPCO), respectively.

4.2.1 Function Selection - IPCO

We used the data from our single image processing chain optimization. Below in Table 3, we summarizethe frequency of function appearance and repetition using the single image processing chain optimization results. We analyzed optimal chains in several conditions characterized by maximal chain length (from 1 to 8 functions), running 50 trials for each condition. We found that the most popular function to be selected was unsurprisingly Thresholding, appearing on average in 90% of chains (across chains of different lengths). Table3 shows the frequency of function appearance in chains with lengths from 1 to 8 functions. Our main observations from these results include:

i. Thresholding is consistently preferred for all chain lengths. For are lengths except 3 and 4, thresholding is selected in more than 90% of chains. For all lengths, thresholding is always the function with the highest selection preference.
ii. There is a consistently moderate selection pressure for denoising functions, whose percentage of chain appearance, hovers around 50%.
iii. As chains get longer, the probability of combination functions being selected, increases dramatically, reaching 100% for the 8 function case.
iv. The top 3 functions for chains of length 2, consist of thresholding, denoising and hole filling, in order of decreasing selection preference. In contrast to this, the top 3 functions for chains of length 3,consist of thresholding, contrast enhancement and morphological operators, in decreasing order of selection preference. In general, the order of preferred functions changes dramatically, from one length to the other, with the exception of thresholding which is consistently the number one preferred function for all lengths.

Table 3: Frequency of Function appearance in 1 chainfor 50 trials for single chain image processing optimization.

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

4.2.2 Function Selection - MIPCO

The Multiple Image Processing Optimization (MIPCO), implementation reported here, consisted of a maximum of 5 chains, each one with a maximum of 8 functions. From our analysis, we found that some functions repeat themselves in the same and neighbouringchains. Table4 summarizesthese functionrepetitions. 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, we 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 4: Function repetitions for 50 trials using multiple chain image processing optimization.

Functions	1 Func	2 Func	3 Func	4 Func	5 Func	6 Func	7 Func	8 Func
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%

4.3 Mandatory functions that always appear in chains

In our experimental analysis, we discovered that there are sets of mandatory functions that always seem to appear together in chains. For the purpose of our analysis, out of 5 MIPCO chains (the maximum chain allowed to be used), we choose 3 MIPCO chains with higher efficiency and most accurate chains (through performance of it's F1 score >91%). We found that the functions listed below are very frequentlyselected (to be noted, not always together). We believe that the selection of these functions, contributes to overall better performance.

i) Morphological Operator – Opening
ii) Watershed
iii) Morphological Operator – Eroding
iv) Denoising
v) Thresholding

4.4 Results using MIPCO –Chains with morphological operators in unorthodox positions

F1 score : 91.80% (Appear in 10th position) | F1 score : 91.43% (1^{st}, $7^{th}$and 9^{th}) | F1 score : 91.38% (1^{st}, $7^{th}$and 9^{th}) | F1 score : 91.21% (4^{th} , 6^{th} and 8^{th}) | Ground-truth Image

Figure7: Final output images of chains using MIPCOthat have morphological operators in various positions (specified in the figure), in the front, middle and end portions of the chains which F1 scores> 91%.

Figure 7 depicts several sample output images using different MIPCO chains. As mentioned earlier, we generously let the algorithm to choose maximum of 8 functions and 5 chains. So this will contribute to 40 outputs (8x5) in processing stages. In above Figure 7, shows the final output and the position of morphological operator appearance.

One possible conclusion from this result,is that the selection of morphological operators at early or middle portions of chains doesseem to have a positive impact on F1 scores. The first figure shows scores of 91.80% (this is an example of an output using 5 chains and 8 functions (total of 40 functions). Out of 40 repeatable functions, 10 of them are morphological operators which appear early and in the middle of the 5 chains.

The figure also provides an opportunity to subjectively compare the MIPCO outputs with the ground truth. In one glance, the outputs seem to be almost identical to the corresponding ground-truth. Most of the 8.2% average error is most probably due to missing black patches (false negatives) and some extra lines,possibly due to the watershed function (false positives), as well as differences in line thickness (ground truth images exhibit relatively large variation of line thickness, which contrasts with most of the processed output which show relatively constant line thickness).

Our experiments revealed that 100% of the "good" chains adopted morphological operators, denoising and thresholding. Moreover, it seems that atleast for this membrane detection problem, all the 'good chains' seem to select watershed as one of the preferred functions, and this function seems to always appear together with its co-partner, namely: the morphological operator 'open'.

5.0 CONCLUSION AND DISCUSSION

From our experiments, and given the specific membrane detection dataset adopted, we find that the optimization of image processing chains, when using multiple chains (MIPCO) is generally more accurate than when using single chains (IPCO). In terms of speed, as expected due to its larger size, and assuming a non-parallelized solution, MIPCO does perform worse (i.e. approximately 10 seconds longer per optimization epoch). Having said that, MIPCO is still easier and faster to train than many other learning machines.Apart from sharing additional advantages with IPCO such as interpretability and re-trainability, MIPCO has additional advantages such as parallelizability and the provisioning of complex interactions between chains, which opens up news opportunities for problem decomposition and solution composition. We believe that this integrative capability of

MIPCO is what allows it to perform not only better than individual IPCO chains, but also better than ensembles of the latter.

Acknowledgments

We would like to thank ISBI (International Symposium on Biomedical Imaging) for providing the training, test and ground-truth datasets, Ministry of Higher Education (MOE) and University of Technology MARA for providing financial assistance for our research.

REFERENCES

[1] R.Raju, Tomas Maul, Andrej Bargiela, 'Image Processing Chain Optimization for Membrane Detection in Neural Slices (update revised manuscript, waiting for decision for publication.

[2] Lonnie Hamm · B. Wade Brorsen · Martin T. Hagan , "Comparison of Stochastic Global Optimization Methods to Estimate Neural Network Weights," Neural Process Lett (2007) 26:145–158 DOI 10.1007/s11063-007-9048-7

[3] Uryasev S, Pardalos PM (eds) (2001) Stochastic optimization: algorithms and applications. Kluwer Academic Publishers, Netherlands

[4] Pardalos PM, Romeijn E (eds) (2002) Handbook of global optimization vol 2: Heuristic approaches, Dordrecht, Netherlands

[5] Boender CGE, Romeijn HE (1995) Stochastic methods. In: Horst R, Pardalos PM (eds) Handbook of global optimization. Kluwer Academic Publishers, Netherlands, pp 829–869

[6] Nurhan Karaboga and Bahadir Cetinkaya, "Performance Comparison of Genetic and Differential Evolution Algorithms for Digital FIR Filter Design, T.Yakhno (Ed):ADVIS 2004, LNCS 3261, pp 482-488, 2004, Springer-Verlag Berlin Heidelberg 2004

[7] Chen, S.: IIR Model Identification Using Batch-Recursive Adaptive Simulated Annealing Algorithm. In: Proceedings of 6th Annual Chinese Automation and Computer Science Conference, pp. 151–155 (2000)

[8] Storn, R., Price, K.: Differential Evolution - A simple and Efficient Adaptive Scheme for Global Optimization over Continious Spaces. Technical Report TR - 95 - 012, ICSI (1995)

[9] Tea Robi and Bogdan Filipi, "DEMO : Diffential Evolution for Multiobjective Optimization, C. A. Coello Coello et al. (Eds.): EMO 2005, LNCS 3410, pp. 520–533, Springer-Verlag Berlin Heidelberg, 2005

[10] Ackley, D. H. (1987). A connectionist machine for genetic hillclimbing, volume 28. Kluwer Academic Pub

[11] Duarte-Mermoud, M., Beltr´an, N., and Salah, S. (2013). Probabilistic adaptive crossover applied to chilean wine classification. Mathematical Problems in Engineering

[12] Chicano, F., Whitley, D., and Alba, E. (2014). Exact computation of the expectation surfaces for uniform crossover along with bit-flip mutation. Theoretical Computer Science

[13] Semenkin, E. and Semenkina, M. (2012). Self-configuring genetic programming algorithm with modified uniform crossover. In Evolutionary Computation (CEC), 2012 IEEE Congress on, pages 1–6. IEEE

[14]ISBI 2012 challenge, http://bit.ly/riGDUm

[15] Cardona, A., Saalfeld, S., Preibisch, S., Schmid, B., Pulokas, A.C.J., Tomancak,P., Hartenstein, V, 'An integrated micro- and macro architectural analysis of the drosophila brain by computer-assisted serial section electron microscopy', PLoSBiol. 10, 2010

[16] R. Raju, T. Maul and A. Bargiela, "Local Contrast Hole Filling Algorithm", IEEE Symposium on Computer Applications& Industrial Electronics, April 2014.

[17] R.Raju, T.H.Maul and A.Bargiela, "New image processing pipelines for membrane detection," Journal of the Institute of Industrial Applications Engineers, Vol 3, No1 ,pp 15-23, January 2015, Online edition: ISSN 2187-8811 Print edition: ISSN 2188-1758

[18] R. Raju, T. H. Maul and A. Bargiela, "A new image processing heuristic suggested by optimization experiments - Enhance it before you Lose it", 2nd International ConferenceOn Intelligent Systems and Image Processing, Sept 2014

[19] Rafael C.Gonzales, Richard E.Woods, Steven .L.Eddins, “Digital Image Processing Using MATLAB, second edition, McGraw Hill Education Private Limited, 2010

[20] I.R. Terol-Villalobos, “Morphological image enhancement and segmentation with analysis,” P. W. Hawkes, Ed. New York: Academic, 2005, pp. 207–273.

[21] J. Serra, Mathematical Morphology Vol. I. London, U.K.: Academic, 1982.

[22] Bhabatosh Chanda (2008) Morphological Algorithms for Image Processing, IETE Technical Review, 25:1, 9-18, http://dx.doi.org/10.4103/02564602.2008.10876637

Authors Biography

Rajeswari Raju received the Bachelor (Hons.) degree in Information Technology from the University of Malaya, the M.S. degree in Computer Science also from University of Malaya, Malaysia and currently pursuing Ph.D. degree in Computer Science at The University of Nottingham, Malaysia Campus. For 3 years, she was attached with Industry, where she was a Business Release Manager with British American Tobacco (M), GSDKL. She is currently a Lecturer with The University Technology MARA (UiTM), Terengganu, Malaysia.

Tomás H. Maul received the B.Sc. (Hons.) degree in biological psychology from the University of St. Andrews, St. Andrews, U.K., the M.S. degree in computer science from Imperial College, London, U.K., and the Ph.D. degree in computational neuroscience from the University of Malaya, Kuala Lumpur, Malaysia. For two years, he was a Senior Researcher with MIMOS Berhad, where he worked in the fields of pattern recognition and computer vision. He is currently an Assistant Professor with The University of Nottingham Malaysia Campus, Semenyih, Malaysia, where he conducts research in the areas of neural computation and computer vision.

Andrzej Bargiela is Professor in the School of Computer Science at the University of Nottingham and Institute of Informatics at Krakow University of Technology. He served as President of the European Council for Modelling and Simulation (ECMS) during 2002-2006 and 2010-2012. He is Associate Editor of the IEEE Transactions on Systems Man and Cybernetics and Associate Editor of the Information Sciences. His research involves investigation into Granular Computing, human-centred information processing as a methodological approachto solving large-scale data mining.