

# **Genetic Programming for Supervised Figure-ground Image Segmentation**

by

Yuyu Liang

A thesis  
submitted to the Victoria University of Wellington  
in fulfilment of the  
requirements for the degree of  
Doctor of Philosophy  
in Computer Science.

Victoria University of Wellington  
2018



## Abstract

Figure-ground segmentation is a process of separating regions of interest from unimportant backgrounds. It is essential to various applications in computer vision and image processing, e.g. object tracking and image editing, as they are only interested in certain regions of an image and use figure-ground segmentation as a pre-processing step. Traditional figure-ground segmentation methods often require heavy human workload (e.g. ground truth labeling), and/or rely heavily on human guidance (e.g. locating an initial model), accordingly cannot easily adapt to diverse image domains.

Evolutionary computation (EC) is a family of algorithms for global optimisation, which are inspired by biological evolution. As an EC technique, genetic programming (GP) can evolve algorithms automatically for complex problems without pre-defining solution models. Compared with other EC techniques, GP is more flexible as it can utilise complex and variable-length representations (e.g. trees) of candidate solutions. It is hypothesised that this flexibility of GP makes it possible to evolve better solutions than those designed by experts. However, there have been limited attempts at applying GP to figure-ground segmentation.

In this thesis, GP is enabled to successfully address figure-ground segmentation through evolving well-performing segmentors and generating effective features. The objectives are to investigate various image features as inputs of GP, develop multi-objective approaches, develop feature selection/construction methods, and conduct further evaluations of the proposed GP methods. The following new methods have been developed.

Effective terminal sets of GP are investigated for figure-ground segmentation, covering three general types of image features, i.e. colour/brightness, texture and shape features. Results show that texture features are more effective than intensities and shape features as they are discriminative for different materials that foreground and background regions normally belong to (e.g. metal or wood).

Two new multi-objective GP methods are proposed to evolve figure-ground segmentors, aiming at producing solutions balanced between the segmentation performance and solution complexity. Compared with a reference method that does not consider complexity and a parsimony pressure based method (a popular bloat control technique), the proposed methods can significantly reduce the solution size while achieving similar segmentation performance based on the Mann-Whitney U-Test at the significance level 5%.

GP is introduced for the first time to conduct feature selection for figure-ground segmentation tasks, aiming to maximise the segmentation performance and minimise the number of selected features. The proposed methods produce feature subsets that lead to solutions achieving better segmentation performance with lower features than those of two benchmark methods (i.e. sequential forward selection and sequential backward selection) and the original full feature set. This is due to GP's high search ability and higher likelihood of finding the global optima.

GP is introduced for the first time to construct high-level features from primitive image features, which aims to improve the image segmentation performance, especially on complex images. By considering linear/non-linear interactions of the original features, the proposed methods construct fewer features that achieve better segmentation performance than the original full feature set.

This investigation has shown that GP is suited for figure-ground image segmentation for the following reasons. Firstly, the proposed methods can evolve segmentors with useful class characteristic patterns to segment various types of objects. Secondly, the segmentors evolved from one type of foreground object can generalise well on similar objects. Thirdly, both the selected and constructed features of the proposed GP methods are more effective than original features, with the selected/constructed features being better for subsequent tasks. Finally, compared with other segmentation techniques, the major strengths of GP are that it does not require pre-defined problem models, and can be easily adapted to diverse image domains without major parameter tuning or human intervention.



# List of Publications

1. Yuyu Liang, Mengjie Zhang, and Will N. Browne. "Image feature selection using genetic programming for figure-ground segmentation." *Engineering Applications of Artificial Intelligence* 62 (2017): 96-108. DOI: 10.1016/J.ENGAPPAI.2017.03.009.
2. Yuyu Liang, Mengjie Zhang, and Will N. Browne. "Figure-ground Image Segmentation Using Feature-based Multi-objective Genetic Programming Techniques." *Neural Computing and Applications* (2017): 1-20. DOI: 10.1007/s00521-017-3253-8.
3. Yuyu Liang, Mengjie Zhang, and Will N. Browne. "Genetic programming for evolving figure-ground segmentors from multiple features." *Applied Soft Computing* 51 (2017): 83-95. DOI: 10.1016/J.ASO-C.2016.07.055.
4. Yuyu Liang, Mengjie Zhang, and Will N. Browne. "Learning Figure-ground Image Segmentors by Genetic Programming." In *Proceedings of The Genetic and Evolutionary Computation Conference (GECCO 2017)*. Berlin, Germany, 2017. pp. 239-240.
5. Yuyu Liang, Mengjie Zhang, and Will N. Browne. "Wrapper Feature Construction for Figure-Ground Image Segmentation Using Genetic Programming." In *Proceedings of Australasian Conference on Artificial Life and Computational Intelligence (ACALCI 2017)*. Melbourne, Australia, 2017. pp. 111-123.

6. Yuyu Liang, Mengjie Zhang, and Will N. Browne. "Feature Construction Using Genetic Programming for Figure-Ground Image Segmentation." In Proceedings of The 20th Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES 2016). Canberra, Australia, 2016. pp. 237-250.
7. Yuyu Liang, Mengjie Zhang, and Will N. Browne. "Figure-ground image segmentation using genetic programming and feature selection." In Proceedings of IEEE Congress on Evolutionary Computation (IEEE CEC 2016). Vancouver, Canada, 2016. pp. 3839-3846.
8. Yuyu Liang, Mengjie Zhang, and Will N. Browne. "Multi-objective genetic programming for figure-ground image segmentation." In Proceedings of Australasian Conference on Artificial Life and Computational Intelligence (ACALCI 2016). Canberra, Australia, 2016. pp. 134-146.
9. Yuyu Liang, Mengjie Zhang, and Will N. Browne. "A supervised figure-ground segmentation method using genetic programming." In Proceedings of European Conference on the Applications of Evolutionary Computation (EvoApplications 2015). Copenhagen, Denmark, 2015. pp. 491-503.
10. Yuyu Liang, Mengjie Zhang, and Will N. Browne. "Image segmentation: a survey of methods based on evolutionary computation." In Proceedings of Asia-Pacific Conference on Simulated Evolution and Learning (SEAL 2014). Dunedin, New Zealand, 2014. pp. 847-859.

# Acknowledgments

I would like to express my sincere gratitude to all those who helped and supported me during my PhD study.

I am thankful to my dear supervisors, Prof. Mengjie Zhang and Assoc. Prof. Will N. Browne, for providing me instructions to my research, writing skills, presentation skills and my personal life as well. I really enjoyed the discussions with them, which inspired me to bring up new ideas and helped me to develop critical thinking. In addition, I also appreciated that they spent long hours revising my papers, and their valuable feedback helped me improve my writing skills. Without their help and support, I could not finish my PhD study so smoothly. I feel lucky to have them as my supervisors.

My officemates, Binh, Deepak, Ibrahim, Qi and Samaneh, accompanied and supported me in the last three years. We discussed, attended lectures and had lunch together, which are great memories to me. In addition, I would like to thank all ECRG members, with whom I made progress together. Specially, I received many valuable suggestions from Harith, Bing, Soha, Saud and Yi.

In addition, thank to my friends (Abigail, Alvina, Huiqi, Qihao, Shima, Tony, Yanyan, Ying, Yu, Yuxin, etc.), my life in Wellington became so interesting and unforgettable. I also would like to thank Abigail, Tony and Yiming, who spent their valuable time supervising me learning to drive.

I also greatly appreciate the support of my dear parents and family members in China, who motivated and brightened my life. I am also

thankful to the joint scholarship of Victoria University of Wellington and Chinese Scholarship Council, which provided me financial support for over three years.

# List of Tables

2.1	Koza's GP Settings. . . . .	67
3.1	Terminal sets of GP based figure-ground segmentation. . . .	76
3.2	Description of features. . . . .	77
3.3	Function set of GP based figure-ground segmentation. . . . .	78
3.4	Four types of images. . . . .	80
3.5	Test Results of segmentors evolved from intensities on bitmap images. . . . .	82
3.6	Test Results of segmentors evolved from different features on a Brodatz texture image. . . . .	83
3.7	Result examples on Weizmann images. . . . .	84
3.8	Test Results of segmentors evolved from different features on Weizmann images. . . . .	85
3.9	Result examples on Pascal images. . . . .	86
3.10	Test Results of segmentors evolved from different features on Pascal images. . . . .	87
3.11	Average training performance of GP evolved segmentors from Gabor features. . . . .	92
4.1	Function set of GP based figure-ground segmentation methods. . . . .	107
4.2	Information on the test sets. . . . .	109
4.3	Results of multi-objective GP evolved segmentors. . . . .	117

4.4	Comparison of ISPGP against INSGP. . . . .	120
5.1	Function set of GP based feature selection methods. . . . .	141
5.2	Training performance of PGP-FS on the Pascal dataset with different coefficients in its fitness function. . . . .	143
5.3	Statistical results of the proposed feature selection methods.	145
5.4	Statistical results on Weizmann test set. . . . .	149
5.5	Statistical results on Pascal test set. . . . .	150
5.6	Comparison of NSGP-FS against SPGP-FS in the number of front solutions . . . . .	152
6.1	Function set of GP based feature construction methods. . . .	172
6.2	GP parameter settings of GP based feature construction meth- ods. . . . .	172
6.3	Results of FSFC and WSFC. . . . .	174
6.4	Results of CoevoMFC and SubtMFC. . . . .	176
7.1	Two new datasets containing various classes of objects. . . .	189
7.2	Function set of GP based figure-ground segmentation meth- ods. . . . .	191
7.3	Result examples using pixel classification based ISPGP. . . .	192
7.4	Results on datasets with various classes of objects. . . . .	193
7.5	Training results of transfer validation experiments. . . . .	196
7.6	Test results of transfer validation experiments. . . . .	196
7.7	Function set of GP based feature selection/construction meth- ods. . . . .	200
7.8	Results of GP based feature selection/construction methods.	203

# List of Figures

1.1	The overall structure of contributions. . . . .	19
2.1	Computer vision tasks (adapted from Wu's work [172]). . . . .	28
2.2	An example of image segmentation from BSDS500. . . . .	30
2.3	An example of figure-ground segmentation. . . . .	31
2.4	A general feature selection process (adapted from Kumar's work [92]). . . . .	38
2.5	A filter model for feature selection. . . . .	38
2.6	A wrapper model for feature selection. . . . .	39
2.7	Branches of EC algorithms. . . . .	41
2.8	The framework of EC algorithms. . . . .	42
2.9	Pareto dominance and Pareto front. . . . .	43
2.10	Fitness evaluation of an individual in cooperative coevolution. . . . .	45
2.11	The framework of GP . . . . .	46
2.12	An individual in tree structure. . . . .	48
2.13	Genetic operators . . . . .	49
2.14	Filter Feature Selection/Construction based on GP. . . . .	57
2.15	Wrapper Feature Selection/Construction based on GP. . . . .	58
2.16	Example images. . . . .	65
3.1	The overall structure of contributions. . . . .	71
3.2	The window classification based segmentation using GP . . . . .	75

3.3	Comparison of GP evolved segmentors from different features	88
3.4	Comparisons of different methods on Weizmann and Pascal images. . . . .	90
3.5	Average training performance of GP evolved segmentors from Gabor features . . . . .	91
3.6	A GP tree solution evolved from histogram statistics for texture image "D24vs34". . . . .	92
3.7	GP tree solutions evolved for bitmap/texture images. . . . .	94
4.1	The overall structure of contributions. . . . .	97
4.2	The complexity measure . . . . .	101
4.3	Gabor features extracted from one subimage . . . . .	106
4.4	Training performance of NSGP and SPGP. . . . .	111
4.5	ISPGP training performance on Pascal dataset with different dominance thresholds . . . . .	113
4.6	Training performance of proposed methods and the reference method. . . . .	114
4.7	Test performance of proposed methods and the reference method. . . . .	116
4.8	The comparison of ISPGP and SPGP on Weizmann dataset. .	118
4.9	Result examples of INSGP and ISPGP. . . . .	119
4.10	Tree analysis on an image from the Pascal dataset. . . . .	121
4.11	Analyses of small solutions non-dominated on Pascal test set	122
4.12	Analyses of large solutions non-dominated on Pascal test set.	124
4.13	Solutions (evolved from a rotated Pascal training set) non-dominated on Pascal test set. . . . .	125
5.1	The overall structure of contributions. . . . .	127
5.2	Framework of the pixel classification based figure-ground segmentation with feature selection . . . . .	132
5.3	Feature subset generation . . . . .	133
5.4	Distinguishable image information. . . . .	133



5.5	Extracted features and their dimension indexes . . . . .	134
5.6	Penalty measure . . . . .	136
5.7	Training performance on the Weizmann and Pascal sets. . .	144
5.8	Performance ( $F_1$ score) of feature selection methods on the Weizmann test set. . . . .	147
5.9	Performance ( $F_1$ score) of feature selection methods on the Pascal test set. . . . .	148
5.10	Selected features based on the proposed methods. . . . .	154
6.1	The overall structure of contributions. . . . .	157
6.2	Framework of the pixel classification based figure-ground segmentation with feature construction. . . . .	161
6.3	Feature construction using an evolved solution of single fea- ture construction methods . . . . .	162
6.4	Pixel distribution in the feature space . . . . .	163
6.5	Fitness evaluation of an individual in the evolution process of WSFC. . . . .	165
6.6	Feature construction using evolved solutions of SubtMFC . .	167
6.7	Fitness evaluation of one individual in the evolution pro- cess of SubtMFC. . . . .	168
6.8	Feature construction using a group of evolved solutions of CoevoMFC . . . . .	168
6.9	Fitness evaluation of an individual in the evolution process of CoevoMFC . . . . .	170
6.10	Terminal set . . . . .	171
6.11	Distribution of class <i>object/background</i> pixels in different fea- ture spaces. . . . .	178
6.12	Example solutions (feature construction functions) evolved by CoevoMFC . . . . .	180
6.13	Distribution of class <i>object/background</i> pixels in different fea- ture spaces. . . . .	181

7.1	The overall structure of contributions. . . . .	185
7.2	. . . . .	194
7.3	Result examples of transfer validation test. . . . .	198
7.4	Training performance (solutions with high classification accuracies and low number of features are desired). . . . .	201
7.5	Test performance (solutions with high classification accuracies and low number of features are desired). . . . .	202

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Problem Statement . . . . .	1
1.2	Motivations . . . . .	3
1.2.1	Limitations of Current Figure-ground Segmentation Methods . . . . .	4
1.2.2	Why GP Is Hypothesised to Be Suited to Figure-ground Image Segmentation . . . . .	5
1.2.3	Limitations of Current GP based Figure-ground Segmentation Methods . . . . .	6
1.2.4	GP for Feature Selection/Construction . . . . .	7
1.2.5	Summary of Research Questions . . . . .	11
1.3	Thesis Goals . . . . .	13
1.4	Thesis Contributions . . . . .	18
1.5	Thesis Organisation . . . . .	23
<b>2</b>	<b>Literature Review</b>	<b>27</b>
2.1	Computer Vision and Figure-ground Image Segmentation . . . . .	27
2.1.1	Image Segmentation . . . . .	30
2.1.2	Figure-ground Image Segmentation . . . . .	31
2.1.3	Edge Detection . . . . .	34
2.2	Machine Learning . . . . .	35
2.2.1	Feature Selection . . . . .	37
2.2.2	Feature Construction . . . . .	40

2.3	Evolutionary Computation and Genetic Programming . . .	40
2.3.1	Evolutionary Computation Techniques . . . . .	40
2.3.2	Genetic Programming (GP) . . . . .	45
2.4	GP for Figure-ground Segmentation . . . . .	50
2.4.1	GP based Figure-ground Segmentation Methods . . .	51
2.4.2	Bloat Control in GP . . . . .	54
2.4.3	GP based Feature Selection/Construction Methods .	57
2.5	Standard Image Datasets, Features, Segmentation Measures and GP Settings . . . . .	62
2.5.1	Standard Image Datasets . . . . .	62
2.5.2	Image Features . . . . .	64
2.5.3	Evaluation Methods . . . . .	65
2.5.4	Koza's GP Settings . . . . .	66
2.6	Chapter Summary . . . . .	67
<b>3</b>	<b>GP for Evolving Segmentors from Diverse Image Features</b>	<b>71</b>
3.1	Introduction . . . . .	72
3.1.1	Chapter Goals . . . . .	72
3.1.2	Chapter Organisation . . . . .	73
3.2	GP based Figure-ground Segmentation . . . . .	73
3.2.1	Window Classification based Segmentation Frame- work . . . . .	73
3.2.2	Terminal Set . . . . .	76
3.2.3	Function Set . . . . .	77
3.2.4	Other GP Settings . . . . .	78
3.3	Experiment Preparation . . . . .	79
3.3.1	Training and Test Images . . . . .	79
3.3.2	Experiment Design . . . . .	80
3.4	Results and Discussions . . . . .	81
3.4.1	Results on Bitmap Images . . . . .	81
3.4.2	Results on Brodatz Texture Images . . . . .	82

3.4.3	Results on Weizmann Horse Images . . . . .	82
3.4.4	Results on Pascal Aeroplane Images . . . . .	84
3.5	Analyses of Evolved Segmentors . . . . .	85
3.5.1	Comparison between Evolved Segmentors . . . . .	86
3.5.2	Comparison with Traditional Methods . . . . .	87
3.5.3	Example Segmentors . . . . .	92
3.6	Chapter Summary . . . . .	93
<b>4</b>	<b>Multi-objective GP for Evolving Segmentors</b>	<b>97</b>
4.1	Introduction . . . . .	98
4.1.1	Goals for Multi-objective GP Evolved Segmentors . .	98
4.1.2	Chapter Organisation . . . . .	99
4.2	Proposed Methods . . . . .	99
4.2.1	Weighted Sum GP Method . . . . .	100
4.2.2	Multi-objective GP Methods . . . . .	101
4.3	Experiment Preparation . . . . .	102
4.3.1	Terminal Set . . . . .	102
4.3.2	Function Set . . . . .	106
4.3.3	Parameters . . . . .	107
4.3.4	Experiment Design . . . . .	108
4.3.5	Performance Measures . . . . .	109
4.4	Results . . . . .	109
4.4.1	Determination of Parameter <i>DominanceThreshold</i> in INSGP and ISPGP . . . . .	110
4.4.2	Proposed Methods vs Reference Method . . . . .	112
4.4.3	INSGP vs ISPGP . . . . .	118
4.4.4	Further Analysis . . . . .	120
4.5	Chapter Summary . . . . .	124
<b>5</b>	<b>GP for Feature Selection in Figure-ground Segmentation</b>	<b>127</b>
5.1	Introduction . . . . .	128
5.1.1	Chapter Goals . . . . .	128

5.1.2	Chapter Organisation . . . . .	129
5.2	Overall Structure of GP based Feature Selection . . . . .	130
5.2.1	Pixel Classification Based Figure-ground Segmentation . . . . .	131
5.2.2	Feature Selection using GP based Methods . . . . .	131
5.2.3	Feature Extraction . . . . .	131
5.3	Proposed Feature Selection Methods . . . . .	134
5.3.1	The Parsimony Coefficient GP Method . . . . .	135
5.3.2	The Multi-objective GP Methods . . . . .	137
5.4	Experiment Preparations . . . . .	138
5.4.1	Benchmark Methods for Comparison . . . . .	138
5.4.2	Image Datasets . . . . .	138
5.4.3	GP Settings . . . . .	141
5.4.4	Experiment Design . . . . .	142
5.5	Results and Discussions . . . . .	142
5.5.1	Determining Coefficient $c$ in Fitness Function of PGP-FS . . . . .	142
5.5.2	Training Results . . . . .	143
5.5.3	Test Results . . . . .	146
5.5.4	NSGP-FS vs SPGP-FS . . . . .	151
5.5.5	Analysis on Selected Features . . . . .	153
5.6	Chapter Summary . . . . .	153
<b>6</b>	<b>GP for Feature Construction in Figure-ground Segmentation</b>	<b>157</b>
6.1	Introduction . . . . .	158
6.1.1	Chapter Goals . . . . .	159
6.1.2	Chapter Organisation . . . . .	159
6.2	Overall Structure of GP based Feature Construction . . . . .	160
6.3	Proposed Single Feature Construction Methods . . . . .	161
6.3.1	Filter Single Feature Construction Method . . . . .	162
6.3.2	Wrapper Single Feature Construction Method . . . . .	164

6.4	Proposed Multiple Feature Construction Methods . . . . .	165
6.4.1	Subtrees-based Multiple Feature Construction . . . . .	166
6.4.2	Coevolution Multiple Feature Construction . . . . .	167
6.5	Experiment Preparations . . . . .	170
6.5.1	Terminal Set . . . . .	170
6.5.2	Function Set . . . . .	171
6.5.3	Other GP Settings . . . . .	171
6.5.4	Experiment Design . . . . .	173
6.6	Results and Discussions . . . . .	173
6.6.1	Single Feature Construction Methods . . . . .	174
6.6.2	Multiple Feature Construction Methods . . . . .	175
6.6.3	Single Methods vs Multiple Methods . . . . .	177
6.7	Further Analyses . . . . .	177
6.7.1	Analyses of A Single Feature Construction Solution . . . . .	178
6.7.2	Analyses of Multiple Feature Construction Solutions . . . . .	179
6.8	Chapter Summary . . . . .	180
<b>7</b>	<b>Further Analyses of Proposed Methods</b>	<b>185</b>
7.1	Introduction . . . . .	186
7.2	Performance on Datasets with Various Foreground Objects . . . . .	187
7.2.1	Design of Experiments . . . . .	187
7.2.2	Datasets . . . . .	188
7.2.3	Experiment Preparation . . . . .	189
7.2.4	GP Settings . . . . .	189
7.2.5	Results . . . . .	190
7.3	Transfer Validation . . . . .	194
7.3.1	Design of Experiments . . . . .	195
7.3.2	Results . . . . .	195
7.4	Selected Features Versus Constructed Features . . . . .	198
7.4.1	Design of Experiments . . . . .	199
7.4.2	Results . . . . .	200

7.5	Further Discussion . . . . .	204
7.6	Chapter Summary . . . . .	206
<b>8</b>	<b>Conclusions and Future Work</b>	<b>209</b>
8.1	Achieved Objectives . . . . .	211
8.2	Main Conclusions . . . . .	213
8.2.1	GP for Evolving Segmentors from Diverse Image Features . . . . .	213
8.2.2	Multi-objective GP for Evolving Segmentors . . . . .	214
8.2.3	GP for Feature Selection in Figure-ground Segmentation . . . . .	215
8.2.4	GP for Feature Construction in Figure-ground Segmentation . . . . .	217
8.2.5	Holistic Considerations . . . . .	218
8.3	Future Work . . . . .	220
8.3.1	A Challenge for The Proposed GP Methods . . . . .	221
8.3.2	Filter GP Methods . . . . .	222
8.3.3	Datasets with Various Types of Objects . . . . .	222
8.4	Chapter Summary . . . . .	223
	<b>Bibliography</b>	<b>225</b>



# Chapter 1

## Introduction

Figure-ground image segmentation is a process of separating foreground objects or regions of interest from unimportant backgrounds. In computer vision and image processing, many tasks, e.g. object tracking, object recognition and image editing, are only interested in certain regions of images and use figure-ground segmentation as a pre-processing step to separate these regions [184]. As the results produced by figure-ground segmentation have important influence on the subsequent application tasks, it is important to achieve accurate segmentation results. However, current segmentation methods have limitations to perform well, especially on complex images.

### 1.1 Problem Statement

Existing figure-ground segmentation approaches mainly include model-based methods [62, 103, 105] and fragment-based methods [23, 24, 25]. Model-based methods commonly match a deformable model to an image by minimising the image energy, which is calculated by a function of image features [79]. Fragment-based methods extract a fragment set in the training stage, which is a set that captures shape and appearance information of the common object parts. Then each fragment's figure-ground

segmentation is generated and used to match objects in the test stage. In general, model-based methods require much human intervention to locate the initial models, and only when the initial models are located close to the target objects are they likely to obtain accurate segmentation results. Fragment-based methods require much work from researchers to collect informative fragments.

As figure-ground segmentation is task-driven and uses prior knowledge of objects of interest to guide the segmentation process, figure-ground segmentation methods are normally supervised [180]. As supervised learning algorithms can learn solutions based on patterns in the input features [101, 116], it is crucial to extract a suitable feature set that contains effective patterns to separate objects from backgrounds. However, features extracted by existing feature descriptors can be irrelevant, redundant or even noisy [101], which have negative impacts on the segmentation performance. Feature selection and construction can help solve this problem by providing more effective features than the original features.

*Feature selection* is a process of selecting a subset of features from a large number of original features [51, 92, 181], thus removing/reducing irrelevant and redundant features, lowering computational cost and improving performance of the later tasks, e.g. classification, data clustering or image segmentation. It is beneficial to select useful features from a wide range of image features to form an effective feature set for figure-ground image segmentation. However, feature selection tasks often face the problem of large search space [159], so powerful search methods are necessary to select the most informative features. In contrast, *feature construction* is a process of transforming a given set of features to a new set of more powerful features [159]. Since the constructed features consider the interactions between the original features, they can be more meaningful and effective. As a good feature representation is important for any machine learning task to obtain good performance, it is worthwhile constructing features for image segmentation tasks. However, existing feature construction methods

are restricted to assumptions and the constructed features are difficult to interpret [127].

Evolutionary computation (EC) is a family of algorithms for global optimisation, including evolutionary algorithms (e.g. genetic algorithms, genetic programming), swarm intelligence (particle swarm optimisation, ant colony optimisation) and others (e.g. learning classifier systems) [13, 119]. EC techniques are stochastic search algorithms, yet additional heuristics are incorporated for evaluating the fitness of individual solutions. Specifically, candidate solutions are called individuals, which are coded by a structure (e.g. binary string, tree or graph). A fitness function is defined to evaluate each individual's performance. Initially, a population of individuals is randomly created. During the evolutionary process, the population is transformed by guided genetic operators (selection, crossover and mutation), thus increasing the quality of solutions/individuals.

As one of the main EC techniques, genetic programming (GP) is considered a powerful and global search method, and can evolve algorithms automatically for complex problems [16, 83, 135]. Therefore, it is promising for GP to solve the problems related to image segmentation. However, there are only limited attempts at using GP in the field of image segmentation. Hence, this thesis will make contributions to applying GP for this important task.

## 1.2 Motivations

This section analyses the limitations or problems in the existing methods of figure-ground segmentation, GP based figure-ground segmentation and feature selection/construction. Moreover, the potentials of GP to handle the limitations or problems of the existing works are also described.

### 1.2.1 Limitations of Current Figure-ground Segmentation Methods

Popularly-used figure-ground segmentation methods have two branches: model based methods [62, 103, 105] and fragment-based methods [23, 24, 25]. Model-based approaches aim to use shape properties or constraints to recover predefined object shapes/structures [105]. Deformable models, active shape models and active contour models are commonly used models for image segmentation. However, these approaches require researchers to guide the segmentation process by providing an indication of the object location in an image. The indication is often a bounding box around each object. Then by minimising an energy function, objects are segmented [88]. In addition, the initial indication of the object location must be close to the target, otherwise it is difficult to obtain accurate results [103].

Fragment-based methods utilise image fragments (or patches) to handle the variability of shape and appearance within a certain class [25]. There are two stages: the training stage and the segmentation stage. In the training stage, a fragment set is built from training images to capture possible shapes and appearance variances of the common object parts within a given class. Then the figure-ground segmentation of each fragment is generated manually or learned automatically. When segmenting a new image, these fragments are used to match the objects, thus obtaining the segmentation result. However, fragment-based methods require much work to collect informative fragments.

Based on the above analyses, three specific problems exist in model-based and fragment-based approaches.

1. They require a heavy workload from users.
2. As they rely heavily on human guidance/work and require fine-tuning to segment specific objects of interest, it makes them difficult to be applied to images in diverse domains.

3. The more involvement of human work, the higher the probability of introducing human bias. As involving too much human bias may prevent machine learning techniques from obtaining better performance, it should be avoided.

### 1.2.2 Why GP Is Hypothesised to Be Suited to Figure-ground Image Segmentation

GP can evolve computer programs to solve problems automatically, which does not require users to specify the form or structure of solutions [135]. If GP can be introduced to evolve segmentation algorithms, then GP based methods will involve less human work than the model-based and fragment-based approaches.

Note that other EC techniques (e.g. differential equation, partial swarm optimization and especially genetic algorithms) have been applied for image segmentation. These works [15, 19, 74, 78, 145, 150, 154] mainly combine the EC techniques with the classic segmentation algorithms (e.g. threshold, region growing and clustering), where the EC techniques are used to optimise parameters for classic segmentation algorithms. Compared with these techniques, GP can evolve segmentors automatically to avoid the problems in the classic segmentation algorithms [135], such as the need of heavy workload or human guidance (e.g. in setting the parameters). In addition, GP is more flexible [52, 167], as it can utilise complex and variable-length representations, such as trees, in which various kinds of functions can be used to combine input features in *linear* or *non-linear* forms. The flexibility of GP makes it possible to evolve better solutions than those designed by experts [133]. Moreover, as an evolutionary computation technique, GP is considered as a powerful global search algorithm, which helps handle large search spaces for the related tasks on images with high variations.

### 1.2.3 Limitations of Current GP based Figure-ground Segmentation Methods

Existing works [47, 133, 140, 157, 160, 161] (to be described in Section 2.4.1) have two approaches to the use of GP for image segmentation, i.e. numerical operators<sup>1</sup> based methods and image processing operators<sup>2</sup> based methods. Numerical operators based methods transform segmentation problems to window (subimages) classification based tasks, where GP is employed to evolve binary classifiers from image features (or pixel intensities) and numerical operators [47, 96, 100, 133, 160, 161]. The evolved classifiers can classify subimages as class *foreground* or *background*. In contrast, image processing operators based methods evolve segmentation algorithms from raw images and image processing operators directly [140, 157]. Instead of working on classifying sub-images, image processing operators based methods take whole images as inputs. As they need to process images, these methods use image processing operators (e.g. thresholding and filters) in the function set.

Even though existing GP based segmentation works produce promising results, there are still several limitations listed as follows.

1. They have only been tested on simple images in a limited number of domains, including texture images [160, 161] and medical images with relatively uncluttered backgrounds [133, 140, 157]. Therefore, whether GP-evolved segmentors<sup>3</sup> can deal with a wide range of images, especially complex images with cluttered backgrounds, is still not clear.
2. Only intensities (pixel gray levels) [140, 157, 160, 161], intensity statistics [133] and local binary patterns (LBP) [47] have been utilised by

---

<sup>1</sup>Operators that process numeric values, e.g. add and subtract.

<sup>2</sup>Operators that process raw images, e.g. image filters.

<sup>3</sup>Segmentors refers to segmentation algorithms evolved by GP based methods in this thesis.

GP to evolve segmentors. As the inputs of GP have a great influence on the performance of evolved solutions [83], it is anticipated that if the terminal set contains advanced features, such as histogram statistics and Gabor features, better performing segmentors will be evolved.

3. GP based methods tend to evolve large/complex solutions that are computationally expensive for evolution and later use, difficult to interpret, and are likely to have low generalisation performance. This phenomenon is often caused by bloat, where in an evolutionary process, an increase in the average solution size and depth does not lead to a corresponding increase in the solution fitness [112]. However, existing methods have not addressed this problem for GP in figure-ground segmentation.
4. It is still challenging to separate foreground objects on complex images with high variations (e.g. cluttered backgrounds), which requires effective feature sets to capture the distinguishing information between objects and backgrounds.

#### 1.2.4 GP for Feature Selection/Construction

Feature selection (FS) and feature construction (FC) are crucial stages for many image analysis tasks, such as image recognition, classification and segmentation [32, 90, 142]. The goal of feature selection is to select a subset of features, removing features that are redundant, irrelevant or have little predictive information. In addition, feature construction is the process of transforming a set of original features to a new set of more powerful features.

### Problems in Current Feature Selection Methods

Till now it is still challenging to select an effective subset of features from the original ones. As the number of possible subsets can be  $2^N$  for a problem domain of  $N$  dimensions, effective search methods are crucial for feature selection [104]. FS algorithms can be divided into three groups based on the search method, i.e. exhaustive, sequential and random [104].

Feature selection based on the exhaustive search methods, e.g. breadth first search, evaluates all possible combinations of the original features exhaustively, and then determines the best feature subset. The exhaustive search has a high computational cost and may cause the overfitting problem [166]. Sequential search methods, such as forward selection and backward selection [48], aim to produce a good solution in a reasonable time by trading off accuracy and optimality for speed. Random search methods start with a randomly selected feature subset. Then one of two ways is applied to proceed to obtain an optimal subset [92]. One is to use the completely random manner to generate the next subset, such as the Las Vegas algorithm [125]. The other is to include heuristic knowledge in the search process, such as evolutionary computation techniques. The Las Vegas algorithm is restricted to one assumption that the run time is always finite, which may be not practical. However, evolutionary computation techniques, e.g. GP, have potential to solve problems with large search spaces efficiently, and can be applied to a wide range of global optimisation problems [177] (see Section 2.6).

### GP Potentials and Challenges for Feature Selection

As GP can handle large search spaces by providing automated solutions [167], it is promising to use GP as a search method for feature selection. Note that other EC techniques have also been used for feature selection by existing works: ant colony optimization (ACO) [31], genetic algorithm [102, 175] and particle swarm intelligence [93]. As the focus of this thesis



is to solve existing problems in figure-ground segmentation and improve the segmentation performance, e.g. by discovering effective features for complex images with high variations, it does not aim to investigate all possible feature selection methods. Therefore, other EC techniques are not investigated for feature selection in this thesis.

Not all features in the feature set have to be used in a GP tree solution, which is the basis of feature selection using GP. Recent works [3, 41, 123, 158] of GP based feature selection perform well, as they produce feature subsets with a smaller number of features and better performance for the given tasks than the original feature sets.

However, there are still several issues that are listed as follows.

1. Most existing works on GP based feature selection address classification problems, whereas other tasks (e.g. image segmentation) have not been sufficiently investigated.
2. Due to GP's characteristic that it can implicitly select features in the evolutionary process, most existing works conduct embedded feature selection. However, GP's ability for explicit feature selection, i.e. filter<sup>4</sup> and wrapper<sup>5</sup> feature selection, has not been sufficiently exploited.
3. Feature selection is a multi-objective task, maximising the performance (e.g. classification accuracy and segmentation accuracy) based on the selected features and minimising the number of selected features. However, there are very few works that use multi-objective GP to conduct feature selection for image segmentation.

Therefore, it is worth investigating feature selection using GP for figure-ground segmentation tasks.

---

<sup>4</sup>Filter methods evaluate candidate solutions based on the general characteristics of the training data rather than the feedback of an inductive algorithm [146].

<sup>5</sup>Wrapper methods employ an inductive algorithm to evaluate the goodness of candidate solutions [165].

### **Problems in Current Feature Construction Methods**

Commonly-used feature construction methods include principal component analysis (PCA) [75] and linear discriminant analysis (LDA) [90], which are linear statistical techniques. In PCA, an orthogonal transformation is used to convert a set of possibly correlated variables into a set of linearly uncorrelated variables (principal components) [75]. LDA (or Fisher's discrimination analysis) is a method to minimise the interclass scattering matrix measure while maximising the intraclass measure, thus making features belonging to different classes more discriminative [90].

Even though these methods are simple and nonparametric, they often involve certain assumptions and constraints [127]. For example, the restrictions can be in terms of the discriminant functions or relationships between features. In addition, it is difficult to interpret the constructed features [127]. Take PCA as an example, the constructed features are linear constructions of the original features, making it difficult to interpret the constructed features. Moreover, as the scales of original features are not comparable and the first principal component will be dominated by those with high absolute variances [90], PCA often requires data normalisation, which makes it even harder to interpret the new features.

### **GP's Potential and Challenges for Feature Construction**

For feature construction, GP can combine input features [20] to build linear and/or non-linear feature transformations without predefining any models [4, 128]. It does not have the constraints of PCA/LDA, e.g. orthogonal or linear transformations only. Moreover, to a limit of human patience for large tree sizes, analysing the evolved programs by GP makes it possible to deconstruct and interpret the new features. Therefore, GP is a well-suited technique for feature construction.

Note that there are several works that use other EC techniques for feature construction, e.g. PSO [173] and GA [156]. Compared with these EC

techniques, GP can utilise complex and variable-length representations, such as trees, so it is more flexible [52] to combine primitive features in linear or non-linear forms. Moreover, this thesis aims to solve existing problems in figure-ground segmentation, e.g. the need for effective features for complex images with high variations, so it does not intend to investigate all possible feature construction methods. Therefore, other EC techniques for feature construction are beyond the scope of this thesis.

The existing GP based feature construction works have the following problems:

1. The constructed features of existing methods are mainly used for image detection and classification tasks, whereas other problems (e.g. segmentation) have not been extensively investigated.
2. It is still challenging to design an effective fitness function to measure the goodness of the constructed features, especially for filter methods.
3. Existing works construct features by mixing different types of image features homogeneously, which may lead to unnecessary interactions of non-related features, lowering the effectiveness of constructed features.

Therefore, it is worth investigating feature construction using GP for figure-ground segmentation.

### 1.2.5 Summary of Research Questions

Based on the analyses of this section, the issues/questions that deserve investigation are described as follows, along with the corresponding goals.

1. As limited types of features (terminal sets) have been used as the inputs to GP to evolve segmentors, it is undetermined what kind of image information is necessary for GP to evolve capable segmentors.

Therefore, one goal of this thesis is to analyse diverse image features by using them as the inputs of GP to evolve segmentors, aiming to determine the necessary information for GP to evolve capable segmentors.

2. As GP based methods tend to evolve solutions with large sizes, the issue “how to control the solution size in GP based methods for figure-ground segmentation?” leads to another goal that is to design multi-objective GP based methods for figure-ground segmentation to balance the evolved solutions’ performance and sizes.
3. GP based feature selection is not sufficiently investigated for figure-ground segmentation. Moreover, feature selection is a multi-objective task (maximising the segmentation performance and minimising the number of selected features), but few existing works consider this issue. Therefore, it needs investigation: how to select effective features using GP for figure-ground segmentation while considering the number of selected features? Based on this question, another goal of this thesis is to develop new feature selection methods using GP for figure-ground segmentation that consider feature selection as a multi-objective task with two objectives (i.e. the segmentation performance and the solution size).
4. GP based feature construction has not been extensively investigated for figure-ground segmentation. Moreover, existing methods mix different types of features homogeneously, which may cause unnecessary interaction of features. The issues, “how to construct effective features using GP for figure-ground segmentation?” and “how to avoid interaction of different types of features?”, deserve investigation. Therefore, one goal of this thesis is to develop novel feature construction methods using GP for figure-ground segmentation, including methods that can construct new features separately from different types of features simultaneously.

Note that detailed thesis goals will be provided in the following section.

## 1.3 Thesis Goals

The overall goal of this thesis is to investigate and improve GP's capacity for figure-ground image segmentation on diverse image domains, especially on complex images with high variations, through evolving well-performing segmentors and generating effective features. This goal is to be investigated based on the following objectives and their sub-objectives:

1. Investigate diverse image features to determine what image information is necessary for GP to evolve capable segmentors for a wide range of image domains. It is anticipated that the evolved segmentors can perform consistently well on various image domains, e.g. binary images, texture images and object images.

To achieve this objective, the following research sub-objectives have been set.

- (a) Develop a GP based method for figure-ground segmentation by transforming segmentation tasks to window classification problems.

Existing GP based segmentation methods fall into two categories, i.e. numerical operators based methods (or window classification based methods) and image processing operators based methods. Specifically, numerical operators based methods transform the segmentation tasks to window/subimage classification tasks. They take features or pixel values of subimages as inputs to evolve classifiers from numerical operators, e.g. arithmetic and conditional operators [47, 133, 160, 161]. The evolved classifiers can classify subimages as class *foreground* or *background*. In contrast, image processing operators based methods evolve segmentation algorithms from raw images directly [140, 157].

As they need to process images, image processing operators based methods use image processing operators (e.g. thresholding and filters) in the function set. Since this objective is to investigate the effectiveness of image features for GP to evolve capable segmentors, the window classification based methods are suitable as they take image features as inputs of GP. Therefore, the window classification framework will be employed to develop a segmentation method using GP.

- (b) Determine effective terminal sets for GP to evolve capable segmentors by investigating a wide range of image features as inputs of GP.

A terminal set comprises inputs to GP based methods [16]. Poli [133] claims that good terminal sets for image analysis should be able to capture all image information at different scales. Therefore, pixel values may not be sufficient for complex images (e.g. real images with clustered background). Hence, a wide range of terminal sets, which cover three general kinds of image features (colour/brightness, texture and shape features), will be investigated to determine effective terminal sets.

2. Design multi-objective GP based methods for figure-ground segmentation to balance the evolved solutions' performance and sizes. GP based methods often have bloat problems, which tend to evolve complex segmentors. Therefore, multi-objective GP based methods will be designed by optimising the two objectives, i.e. improving the segmentation performance of the derived solutions and minimising the solution size. The proposed methods are expected to evolve a set of non-dominated segmentors that are trade-offs between the segmentation performance and the solution complexity<sup>6</sup>.

---

<sup>6</sup>The term "solution complexity" refers to complexity in terms of solution size in this thesis.

To achieve this objective, the following research sub-objectives have been set.

- (a) Develop new parsimony pressure based methods to set as a benchmark comparison technique.

A popular approach for bloat control is the use of parsimony pressure [163]. It incorporates the solution size as an additional, but hidden, objective as a penalty term [163]. This will be combined with standard GP to form a benchmark method to compare with the proposed multi-objective approaches.

- (b) Improve standard multi-objective techniques, i.e. NSGA-II and SPEA2, and assimilate them into GP to create new multi-objective GP based methods.

NSGA-II and SPEA2 are selected as the basic multi-objective techniques since they are commonly-used techniques [37]. For NSGA-II and SPEA2, a new Pareto dominance mechanism is to be designed to incorporate preference information, helping produce a more fine-grained approximation of the most relevant areas on the Pareto front and/or reduce the computation time. By applying the improved NSGA-II and SPEA2 to GP respectively, two novel multi-objective GP methods will be developed.

3. Develop new feature selection methods using GP for figure-ground image segmentation that consider feature selection as a multi-objective optimisation task, i.e. maximising the segmentation performance based on the selected features and minimising the number of selected features. Compared with the original features, the feature subsets produced by the proposed methods are anticipated to improve segmentation performance while reducing the number of features.

To achieve this objective, the following research sub-objectives have been set.

- (a) Develop a new single-objective feature selection method by combining multiple objectives linearly to form a single objective. Specifically, the parsimony coefficient technique<sup>7</sup> is incorporated with GP to include a penalty term based on the number of selected features in the original fitness.
  - (b) Design and develop novel multi-objective feature selection methods by applying multi-objective techniques in GP. Compared with the parsimony coefficient technique, which aggregates two objectives together, the multi-objective methods treat objectives independently. The evolved solutions of the multi-objective methods are trade-offs along all the objectives.
4. Develop novel feature construction methods using GP to construct single or multiple features from primitive image features for figure-ground segmentation. Both single and multiple feature construction methods will be investigated, which construct an individual feature and multiple features respectively in a single GP run. As the constructed features are combinations of the original features generated by linear or non-linear constructive operators (e.g. addition and division), they consider the interactions of original features. Therefore, it is anticipated that the constructed features are more effective for segmentation problems than the original features.

To achieve this objective, the following research sub-objectives have been set.

- (a) Develop new single feature construction methods using GP. Both explicit feature construction approaches (i.e. filter and wrapper approaches) will be investigated. An entropy based fitness function will be designed for filter methods, since it can measure the impurity of two classes separated based on the con-

---

<sup>7</sup>This is one parsimony pressure technique, and is originally used for bloat control by decreasing the raw fitness by an amount dependent on the solution size [97].



structed features, which can reflect the effectiveness of the constructed features.

- (b) Design and develop novel multiple feature construction methods using GP.

Techniques, i.e. subtrees and cooperative coevolution techniques, will be introduced in GP to design multiple feature construction methods. As subtrees rooted on the non-terminal nodes of an evolved tree solution produce useful features for classification tasks [4], subtrees will be used to construct features for segmentation tasks. Subtrees-based methods conduct homogeneously combination of varied types of features, which leads to the interactions of non-related features and may lower the effectiveness of constructed features. In contrast, the cooperative coevolutionary technique allows GP to construct features from different types of image features separately.

5. Further investigate the effectiveness and applicability of the proposed GP methods in the field of figure-ground segmentation.

To achieve this objective, the following research sub-objectives have been set.

- (a) Investigate whether the proposed GP methods can perform well on datasets containing more varied types of foreground objects. Each benchmark dataset used in this thesis (to be described in Section 2.5.1) is partitioned to contain one type of foreground object. After GP is thoroughly investigated on these datasets by the previous objectives, it is considered whether the proposed GP methods can perform well on datasets containing multiple types of foreground objects.
- (b) Investigate whether the segmentors evolved from one type of foreground objects can generalise well on other types of objects by conducting “transfer validation tests”.

It is hypothesised that GP can capture common characteristics across a class of foreground objects, which will be tested on datasets containing various foreground objects.

- (c) Investigate whether the selected feature subsets or the constructed features are more effective for figure-ground segmentation.

Feature selection and feature construction using GP will be investigated by the previous objectives. It is then necessary and meaningful to conduct comparison between the selected features and the constructed features for use in figure-ground segmentation tasks.

- (d) Investigate the proposed GP methods by comparison with other non-EC segmentation techniques.

Traditional segmentation techniques (e.g. the thresholding method, K-means clustering and region growing methods) and deep learning techniques, will be analysed to compare with the proposed methods. The strengths and drawbacks of them in the field of figure-ground segmentation will be discussed, as it is beyond the scope of this thesis to experimentally verify all types of alternatives.

## 1.4 Thesis Contributions

This thesis contributes to the following important aspects in the fields of computer vision and evolutionary computation, specifically in GP based figure-ground image segmentation. The overall structure of the contributions is shown in Figure 1.1.

1. This thesis conducted the first work on analysing the three general types of image features, i.e. colour/brightness, texture and shape features, which were used as the inputs of GP to evolve segmentation algorithms (segmentors). It aimed to determine what kind of

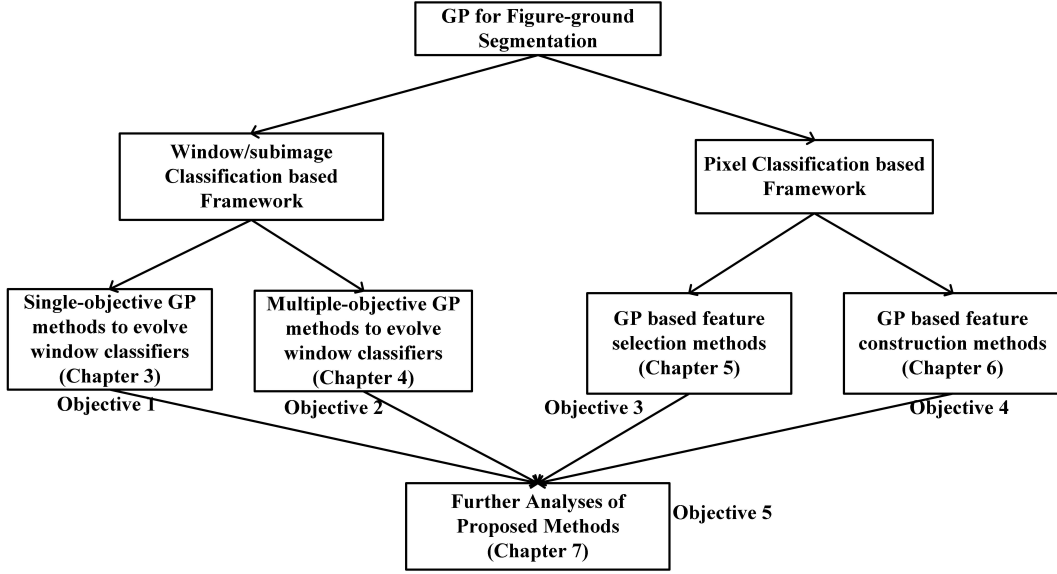


Figure 1.1: The overall structure of contributions.

image information is necessary for GP to evolve capable segmentors, especially for complex images with high variations.

Results showed that intensity based features were not sufficient for complex images, as they are not discriminative when foreground and background patches have similar brightness levels. Shape features were better than intensities, as they provide edge information; however, only utilising edge information was still not enough to distinguish the foreground/background patches with similar pixel distributions. In contrast, texture features were the most effective, as they are discriminative for different materials that foreground and background regions normally belong to (e.g. metal or wood). Moreover, with the input of discriminative image features, the proposed methods can be successfully applied to diverse image domains.

Part of the work has been published in the following papers.

- (a) Yuyu Liang, Mengjie Zhang, and Will N. Browne. “A supervised figure-ground segmentation method using genetic pro-

gramming.” In Proceedings of European Conference on the Applications of Evolutionary Computation (EvoApplications 2015). Copenhagen, Denmark, pp. 491-503. 2015.

- (b) Yuyu Liang, Mengjie Zhang, and Will N. Browne. “Genetic programming for evolving figure-ground segmentors from multiple features.” *Applied Soft Computing* 51 (2017): 83-95. DOI: 10.1016/J.ASOC.2016.07.055.

2. This thesis proposed two new multi-objective GP methods to evolve segmentors. Specifically, two Pareto dominance based multi-objective techniques (NSGA-II [44] and SPEA2 [183]) are modified to incorporate preference information, which are then combined respectively with GP to form the new multi-objective GP methods. Both methods had two objectives, one of which was related to the segmentation performance and the other was based on the solution complexity, aiming to achieve good trade-offs between solution functionality and complexity.

By optimising both the solution performance and complexity, the proposed methods can significantly reduce the solution size while achieving similar segmentation performance to a reference method that did not consider complexity. Moreover, the proposed methods were compared with a single-objective method based on a parsimony pressure technique (a popular bloat control technique, which aggregates the objectives into a scalar fitness function based on predefined weights/coefficients). The proposed methods can produce smaller solutions while performing similarly to the parsimony pressure based method, as they are based on the Pareto dominance technique that is more flexible in the search for trade-offs between objectives.

Part of the work has been published in the following papers.

- (a) Yuyu Liang, Mengjie Zhang, and Will N. Browne. “Multi-objective

genetic programming for figure-ground image segmentation.” In Proceedings of Australasian Conference on Artificial Life and Computational Intelligence (ACALCI 2016). Canberra, Australia, pp. 134-146, 2016.

- (b) Yuyu Liang, Mengjie Zhang, and Will N. Browne. “Figure-ground Image Segmentation Using Feature-based Multi-objective Genetic Programming Techniques.” *Neural Computing and Applications*. (accepted with minor revision)

3. This thesis introduced GP for the first time to build feature selection methods that aimed to improve the segmentation performance, especially on complex images. As feature selection is a multi-objective problem in nature (i.e. maximising the segmentation performance and minimising the number of features), the parsimony coefficient technique [163] and Pareto dominance techniques [44, 183] were introduced in GP to design novel feature selection methods that can optimise both objectives.

Note that parsimony pressure techniques are popular single-objective techniques for bloat control, which aggregates two objectives into a scalar fitness function. Specifically, the parsimony coefficient technique is incorporated with GP to include a penalty term based on the number of selected features in the original fitness (e.g. segmentation accuracy). The Pareto dominance techniques treat two objectives independently, which can lead to good trade-offs between two objectives. With high search ability and being likely to find the global optima, the proposed methods produced feature subsets that lead to solutions achieving better segmentation performance with lower numbers of features than those of two traditional feature selection methods and the original feature set.

Part of the work has been published in the following papers.

- (a) Yuyu Liang, Mengjie Zhang, and Will N. Browne. “Figure-ground

image segmentation using genetic programming and feature selection.” In Proceedings of IEEE Congress on Evolutionary Computation (IEEE CEC 2016). Vancouver, Canada, pp. 3839-3846. 2016.

- (b) Yuyu Liang, Mengjie Zhang, and Will N. Browne. “Image feature selection using genetic programming for figure-ground segmentation.” *Engineering Applications of Artificial Intelligence* 62 (2017): 96-108. DOI: 10.1016/J.ENGAPPAI.2017.03.009.

4. This thesis introduced GP for the first time to construct high-level features from primitive image features, which aimed to improve the image segmentation performance, especially on complex images. Both single and multiple feature construction methods were designed based on GP, termed as SFC (single feature construction) and MFC (multiple feature construction). Specifically, SFC methods combine different types of primitive features together to produce a single new feature. The subtrees and cooperative coevolution techniques were introduced in GP to develop new MFC methods. As the outputs of subtrees can also provide useful information [4], the subtrees based MFC method utilises the subtree outputs to produce multiple new features that are also combined from different types of features.

Note that it was still unknown whether the homogeneous combination of different types of features in SFC and subtrees based methods are useful to construct effective features. Therefore, the cooperative coevolution technique was employed, which enables GP to construct features from different types of features separately, and conduct combination of the constructed features simultaneously. By considering interactions of the original features, the constructed features of the proposed methods achieved better segmentation performance than the original features yet were fewer in number.

Part of the work has been published in the following papers.

- (a) Yuyu Liang, Mengjie Zhang, and Will N. Browne. "Feature Construction Using Genetic Programming for Figure-Ground Image Segmentation." In *Proceedings of The 20th Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES 2016)*. Canberra, Australia, pp. 237-250. 2016.
  - (b) Yuyu Liang, Mengjie Zhang, and Will N. Browne. "Wrapper Feature Construction for Figure-Ground Image Segmentation Using Genetic Programming." In *Proceedings of Australasian Conference on Artificial Life and Computational Intelligence (ACALCI 2017)*. Melbourne, Australia, pp. 111-123. 2017.
5. This thesis further investigated the effectiveness of the proposed GP methods in the field of figure-ground segmentation. This investigation has shown that GP is suited for figure-ground image segmentation for the following reasons. Firstly, the proposed GP methods can evolve segmentors with useful class characteristic patterns to segment various types of objects. Secondly, the segmentors evolved from one type of foreground objects can generalise well on visually similar objects. Thirdly, both the selected and constructed features of the proposed GP methods are more effective than original features, with the selected features being better. Finally, compared with other segmentation techniques, the major strengths of GP are that it does not require pre-defined problem models, and can be easily adapted to diverse image domains without major parameter tuning or human intervention.

## 1.5 Thesis Organisation

The rest of this thesis is organised as follows. Chapter 2 presents the background information and related works. Chapter 3 to Chapter 7 present major contributions of this thesis, aiming to achieve the corresponding

objectives listed in this chapter. Note that Figure 1.1 provides the overall structure of these contribution chapters. In addition, Chapter 8 presents conclusions and future directions.

Chapter 2 provides an introduction of figure-ground image segmentation, feature selection/construction and genetic programming. Existing GP based works for figure-ground image segmentation and feature selection/construction are reviewed, which form the motivation of the work in this thesis. In addition, background information about multi-objective techniques, coevolution techniques and parsimony pressure techniques is also introduced, which is used in the following chapters.

In Chapter 3, seven terminal sets that can cover three general types of image features, i.e. colour/brightness, texture and shape features, are employed as inputs of GP to evolve segmentation algorithms. Results show that texture features (e.g. Gabor features) are more effective than intensities and shape features. Moreover, the evolved segmentors can perform consistently well on various image domains.

Since bloat problems are common for GP based methods that tend to evolve complex segmentors (have large sizes, and are computationally expensive and are difficult to interpret), multi-objective GP based methods are designed for figure-ground segmentation (described in Chapter 4). The aim is to balance the evolved solutions' functionality and complexity. Based on the findings of Chapter 3, Gabor features are used as inputs of GP. The proposed multi-objective GP methods can evolve a set of non-dominated segmentors that are trade-offs between the segmentation performance and the solution complexity. In addition, since the evolved segmentors have smaller sizes, the fitness evaluation of individuals (segmentors) in the evolutionary process is less time-consuming, leading to the training and test time being reduced.

For complex images with high variations (e.g. containing cluttered backgrounds), it is challenging to separate objects from the unimportant backgrounds, which requires effective features to capture the discrimi-



native information between object and background regions. Therefore, Chapter 5 develops new feature selection methods using GP for figure-ground image segmentation. Feature selection tasks are multi-objective tasks: maximising the segmentation performance and minimising the feature numbers, thus the parsimony coefficient technique and multi-objective techniques are introduced in GP to balance the solution performance and the feature number. Results show that compared with the original features, the proposed GP based methods can improve the segmentation performance while reducing the feature numbers.

In addition to feature selection, feature construction is another type of feature manipulation, which is a process of transforming original features to a new set of more powerful features. Considering that GP is a well-suited technique for feature construction, Chapter 6 investigates feature construction using GP for figure-ground segmentation. Both single and multiple feature construction methods are developed, which produce discriminative/effective features, helping improve the segmentation performance.

To further investigate the effectiveness of the proposed GP methods, the following four issues are addressed in Chapter 7. Firstly, whether the proposed methods can perform well on datasets that contain various foreground objects deserves investigation. Secondly, it deserves investigation whether the segmentors evolved from images with one type of foreground objects can generalise well on images with visually similar/different types of foreground objects. Thirdly, the selected features are compared with the constructed features of the proposed feature selection/construction methods. This aims to determine whether the selected or constructed features are more effective for the given segmentation problems. Eventually, the strengths and weaknesses of GP are discussed by comparing with alternative segmentation techniques.

Chapter 8 provides major conclusions to the objectives achieved in this work along with several avenues for future research enabled by this thesis.



## Chapter 2

# Literature Review

This chapter reviews the literature regarding the concepts that form the background and support the motivations of this thesis. Specifically, the basic concepts of figure-ground image segmentation, feature selection, feature construction, evolutionary computation techniques and genetic programming (GP) are briefly discussed. In addition, existing GP methods for figure-ground segmentation and feature selection/construction are discussed. Moreover, the experiment preparations, e.g. standard image datasets, image features and evaluation methods, are also presented.

### 2.1 Computer Vision and Figure-ground Image Segmentation

Computer vision is a major research area of artificial intelligence. A widely accepted definition of computer vision is that “computer vision is an interdisciplinary field that deals with how computers can be made for gaining high-level understanding from digital images or videos” [162]. Computer vision tasks cover methods for acquiring, processing, analysing and understanding digital images [81]. Figure 2.1 shows major computer vision tasks that are categorised to three levels (i.e. low-level, middle-level and

high-level vision tasks), along with connections of different vision tasks.

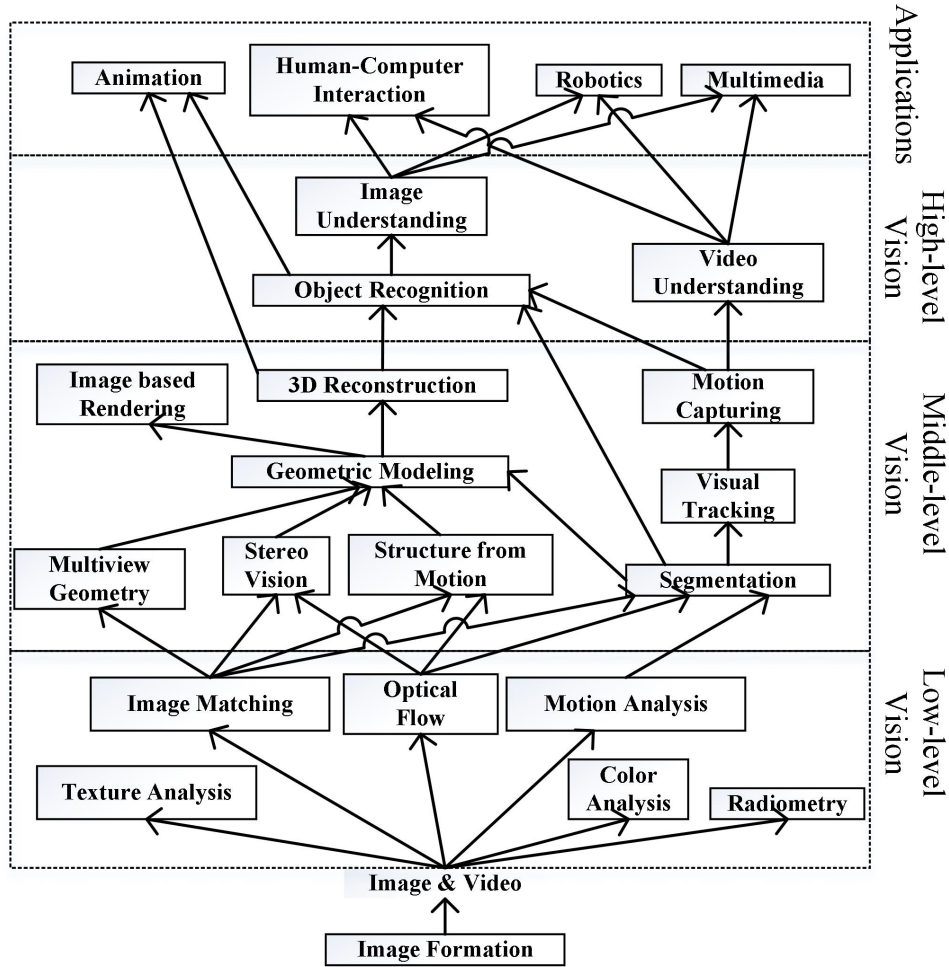


Figure 2.1: Computer vision tasks (adapted from Wu's work [172]).

- Low-level vision tasks include basic processing tasks on images and videos. Figure 2.1 presents six low-level tasks, i.e. image matching, optical flow computation, radiometry, motion analysis, texture analysis and colour analysis. Specifically, image matching is to establish correspondences between two or more images [137]. Optical flow is a type of image observation of motion, which can be used to estimate camera motion or object motion [54]. Additionally, radiometry

is a system of language, mathematics and instrumentation, aiming to describe and measure the propagation of electromagnetic radiation [117]. The images generated by low-level vision tasks can be used as inputs to higher-level tasks, e.g. image segmentation.

- Middle-level vision has two major aspects, i.e. inferring geometric and inferring motion information [172]. Firstly, multiview geometry, stereo vision and structure from motion (SfM) are fundamental geometric vision tasks that infer three-dimensional scene information from two-dimensional images. Based on these fundamental geometric tasks, geometric modelling aims to develop mathematical description (models) of shapes for objects and scenes [63], which can be used for 3D (dimension) reconstruction and image-based rendering. Secondly, to infer motion information, image segmentation is a fundamental vision task, which can separate regions of interest (e.g. moving objects) from images [80]. The results of segmentation can then be used by subsequent vision tasks, e.g. visual tracking and motion capturing, to estimate two-dimensional and three-dimensional motions.
- High-level vision tasks aim to infer semantic information. A basic high-level task is object recognition that aims to identify specific objects in digital images or videos [139]. Higher level tasks are image understanding and video understanding that deal with analysis and interpretation of scenes based on images and videos respectively [71, 172].

Real-world applications of computer vision tasks include animation, human-computer interaction, robotics and multimedia analysis. For example, analyses of multimedia contents rely on image/video understanding tasks. In addition, computer vision tasks enable computers to recognise, interpret and understand human actions/behaviours based on visual inputs for human-computer interaction [172].

In computer vision, many tasks, e.g. object tracking and image editing, are only interested in certain regions of images. Therefore, a pre-processing step to separate these regions from unimportant backgrounds is often required, which is termed as figure-ground image segmentation [184]. Figure-ground image segmentation can be regarded as a special case of image segmentation, as it only targets two well-defined parts: foreground and background [138].

### 2.1.1 Image Segmentation

Image segmentation is a process of partitioning pixels of an image to perceptually coherent regions (shown in Figure 2.2) [18, 80, 150]. For a segmented image, pixels in one region are similar with each other according to a homogeneity criterion, yet pixels in different regions are heterogeneous. Since this process divides an image to several homogeneity regions and helps to find regions of interest, images become easier to manipulate and more meaningful for following higher-level tasks, e.g. feature extraction, object detection, image recognition and classification [80].

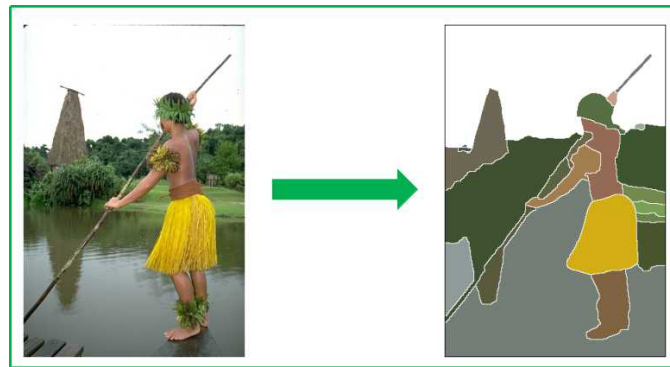


Figure 2.2: An example of image segmentation from BSDS500 (Berkeley Segmentation Data Set and Benchmarks 500 [10]).

### 2.1.2 Figure-ground Image Segmentation

Figure-ground image segmentation is a process of separating foreground objects or regions of interest from the unimportant backgrounds [184]. A target image is labelled into two complementary sets of pixels, one of which contains the pixels corresponding to foreground objects while the other contains the background pixels [131] (shown in Figure 2.3). Figure-ground image segmentation is an application-specific problem, as what should be identified as foreground objects is dependent on the given applications. For example, moving objects, e.g. people, cars and horses, are often regions of interest and regarded as foreground objects [149].

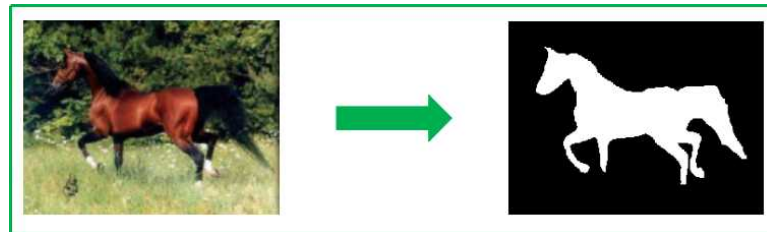


Figure 2.3: An example of figure-ground segmentation (the horse is the region of interest, labelled as colour white; the background is labelled as colour black).

Existing figure-ground segmentation methods mainly include model-based and fragment-based approaches [176]. Both of them are task-driven and use prior knowledge of objects of interest to guide the segmentation process.

#### Model-based Figure-ground Segmentation Methods

Model-based segmentation methods aim to use shape properties or constraints to recover predefined object shapes/structures [105]. They normally match a deformable model to an image by minimizing the image energy that is a function of image features. Deformable models can be curves or surfaces of target objects [17]. Commonly-used deformable models

are active contour models (snakes) [179], level sets [169] and active shape models (ASM) [33]. Specifically, when segmenting a test image, users are required to guide the segmentation process by locating the initial model on the test image [88]. The evolution of models is guided by the internal and external energies. The internal energy controls the curves' smoothness, while the external energy aims to attract the model towards object boundaries.

Zhang et al. [179] present an improved geometric active contour model. A window function is introduced to obtain the mean information of each pixel's neighborhood, and a novel pressure force function is created to drive the contour towards the object boundary. Results show that the proposed method can achieve satisfactory results when tested on images with intensity inhomogeneity and fuzzy boundaries. However, prior knowledge is used to place the initial curves accurately, which helps achieve accurate results. In practice, prior knowledge is usually not available to achieve accurate initial curves.

Wang et al. [169] develop a region-based level set framework, which incorporates a local correntropy-based K-means (LCK) clustering method. The LCK clustering is proposed to solve the problem that real-world images are often distorted by noise and intensity inhomogeneity, as this method is robust to outliers. Experiments are conducted on synthetic and real images, and the proposed method outperforms several state-in-the-art methods (such as LBF+Order (a local order method based on Local Binary Fitting) [170]) in noise sensitivity and segmentation accuracy. However, there are two limitations: the proposed method is sensitive to the initialisation as the LCK model is not convex, and this method may produce inaccurate edges on images with cluttered objects (e.g. hair), as edge pixels can be easily grouped as outliers.

Chen et al. [33] use active shape models (ASM) to develop a fully automated segmentation method to segment the common carotid artery (CCA, a type of blood vessel). In the test stage, they use prior knowledge (CCA's



average diameter) to detect the centroid of CCA for placing the trained model automatically. This method can produce accurate CCA contours. However, the initialisation is based on an assumption that the best initial position is the centre of CCA.

Based on the above analyses, the mentioned model-based methods often require users to locate the initial model (generated in the training stage) for testing. Moreover, the initial model must be located close to the target object, otherwise it is difficult to obtain accurate segmentation results [103]. Therefore, it is desired to minimise human interaction in this work to save associated costs.

### **Fragment-based Figure-ground Segmentation Methods**

Fragment-based methods utilise image fragments (or patches) to handle the variability of shape and appearance within a certain class [25]. There are two stages: a training stage and a segmentation stage. In the training stage, a fragment set is built from training images to capture possible shapes and appearance variances of the common object parts within a given class. Then the figure-ground segmentation of each fragment is generated manually or learned automatically. In the segmentation stage, fragments are detected to match objects in a test image.

Borenstein et al. [23] propose a fragment-based method for class-specific segmentation. Firstly, it collects a representation of the object shapes of a given class, which aims to capture the common characteristics of the object shapes. Then it uses the pre-stored representation to guide the segmentation process by generating a consistent cover of the object on a test image. This cover determines a figure-ground map for this test image. The proposed method is shown to be better than a normalised-cuts segmentation algorithm and can handle variations in object shapes. Nevertheless, it is difficult to segment highly variable parts, such as the legs and tails of horses. In addition, the collection of fragments is complicated and labour intensive.

Kuettel et al. [88] propose an overlapping window based method. Firstly, all test and training images are decomposed into overlapping windows that are likely to cover the foreground objects. Then the segmentation masks from training windows, which are similar to the test windows visually, are transferred to test images. Next the transferred masks are used to derive the unary potential of the energy function, which represents the possibility of a pixel being background or foreground. However, the selection of windows is based on the intuition that visually similar windows often contain similar segmentation masks.

Borenstein et al. [24] also develop a fragment based method. Its major difference from other methods is that the fragment labelling can be conducted automatically. However, the automatic labelling relies on an assumption that the fragments generally have highly varied backgrounds, yet homogeneous foreground regions, which is often not the case in practice.

Based on the above analyses, the major problem of fragment-based methods is that they require fragment generation and fragment labelling, which are labor intensive and time-consuming [23, 88]. Even though there are works that can conduct automatic fragment labelling, they are restricted to certain assumptions, e.g. the foreground regions should be homogeneous [24].

### 2.1.3 Edge Detection

Edge detection is a process of identifying pixels (or edges) in an image where pixel values (or intensities) change sharply [91, 144]. The aim of edge detection is to reduce the amount of information in an image, while preserving the structural properties for subsequent image processing tasks, e.g. image segmentation and feature extraction [91].

For image segmentation, edge based methods are one type of traditional segmentation techniques [80]. Specifically, the detected edges of edge detection can be borders that separate pixels of an image into coher-

ent regions with different intensities [144].

For figure-ground image segmentation, edge detection is rarely used. Firstly, as figure-ground image segmentation aims to separate foreground regions [131], the edges detected in background regions can be misleading. Secondly, the results of figure-ground image segmentation are binary images that separate foreground and background regions on the pixel level [131]; however, the results of edge detection are edge pixels based on intensity changes [91]. Extra processes would be required to generate figure-ground segmentation results based on the detected edges of edge detection. Therefore, it is not suitable to conduct figure-ground image segmentation based on the results of edge detection directly by GP.

## 2.2 Machine Learning

To avoid the restrictions imposed by conventional approaches, the abilities of machine learning approaches are considered. Machine learning is a branch of artificial intelligence, which aims to enable computer systems to learn from data automatically [7, 122]. Two widely accepted definitions are as follows.

- “A computer program is said to learn from experience  $E$  with respect to some class of tasks  $T$  and performance measure  $P$ , if its performance at tasks in  $T$ , as measured by  $P$ , improves with experience  $E$ ” (Proposed by Tom Mitchell) [120].
- “The goal of machine learning is to develop methods that can automatically detect patterns in data, and then to use the uncovered patterns to predict future data or other outcomes of interest” (proposed by Kevin P. Murphy) [122].

The construction of a proper training set, validation set and test set is crucial for machine learning tasks. A *training set* is a set of examples used for learning solution models for given problems [46]. A *validation set* is a

set of examples used to estimate the performance of the trained models in a training process to avoid overfitting to the data. In contrast, a *test set* is a set of examples used only to assess the performance of trained solution models. Therefore, training and validation sets are used in the training process to learn solution models and control overfitting respectively, while test sets are only used in a test process, in which the solution model must not be further tuned. In addition, cross validation (used in Chapter 3) is a technique to evaluate solution models by splitting the original training set to a new training set to train a model and a validation set to evaluate it [11].

Based on the feedback mechanism in the learning process, machine learning algorithms fall into three major branches, i.e. supervised learning, unsupervised learning and reinforcement learning [143]. For unsupervised learning, training data is not labelled, which means that the desired outputs corresponding to the inputs in training data are not provided. Therefore, only the statistical properties of data itself is relied on in the learning process [35, 46]. In contrast, for supervised learning, completely labelled training data is provided, based on which solution models can be inferred by mapping inputs to the desired outputs [35, 46]. For reinforcement learning, desired outputs of the training data are not directly provided, and the feedback is in the form of rewards and punishments [164].

As figure-ground segmentation is task-driven and uses prior knowledge of objects of interest to guide the segmentation process, figure-ground segmentation methods are normally supervised [180], which learn solutions based on patterns in the input features [101, 116]. Therefore, it is crucial to extract a suitable feature set, which contains effective patterns to separate objects from backgrounds. However, features extracted by existing feature descriptors can be irrelevant, redundant or even noisy [101], leading to negative impact on the segmentation performance. Feature selection and construction can help solve this problem by providing more

effective features than the original features.

### 2.2.1 Feature Selection

Feature selection (FS) is a process of selecting a subset of features from original features [114, 126, 181]. The advantages that feature selection brings to learning algorithms are threefold [14]: feature selection can help learning algorithms reduce the computational complexity, avoid over-fitting<sup>1</sup> and be robust to noisy features, thus improving their performance.

There are four major aspects in a general procedure for feature selection (shown in Figure 2.4), i.e. generating subsets, evaluating subsets, choosing the stopping criteria and validating the results [38].

1. Generating subsets: search methods are used to generate feature subsets based on the original features.
2. Evaluating subsets: Two evaluation criteria are widely-used, i.e. independent criteria (e.g. distance measures, information measures and probability of error measures) and dependent criteria (e.g. classifiers). The former suits a filter model [24] that does not require a learning algorithm to evaluate selected features. The latter suits a wrapper model [165] that needs a predefined inductive algorithm. Investigation into effective measures to evaluate feature subsets will be conducted in this thesis (see Section 5.3).
3. Stopping criteria: general stopping criteria include that the search reaches a predefined minimum number of features or maximum number of iterations; addition or deletion to the subset of features does not obtain a significant difference in the performance.

---

<sup>1</sup>Over-fitting occurs when a solution model is over-trained to memorise the training data at the expense of predictive performance on the unseen data with the same structural regularities [9].

4. Result validation: this step aims to check whether the subset of features are valid. Even though this part does not belong to the feature selection process, it is necessary to validate the selection method.

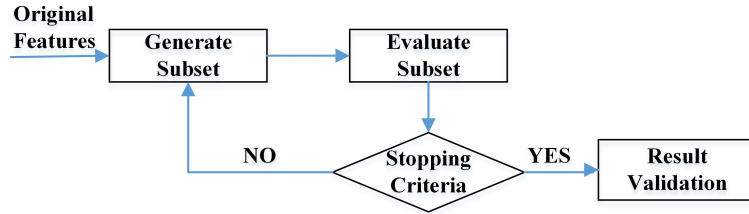


Figure 2.4: A general feature selection process (adapted from Kumar’s work [92]).

Based on whether an inductive learning algorithm is needed to evaluate selected feature subsets, feature selection methods can be divided into three major branches [92], i.e. filter, wrapper and embedded methods.

### Filter Approaches

Filter methods evaluate the selected feature subsets depending on the general characteristics of the training data rather than the feedback of an inductive algorithm [146] (shown in Figure 2.5). Without relying on a specific learning algorithm to direct the selection, filter approaches generally have low computational complexity, which makes them fast. However, the avoidance of inductive algorithms also makes it challenging to evaluate the subsets in the search process, since they are not optimised with a learning algorithm for specific tasks.



Figure 2.5: A filter model for feature selection.

### Wrapper Approaches

Wrapper methods employ an inductive algorithm to evaluate the goodness of the selected subsets [30] (shown in Figure 2.6). Even though the selected subsets tend to be more effective, the computational complexity of wrapper approaches tends to be higher than that of filter approaches.

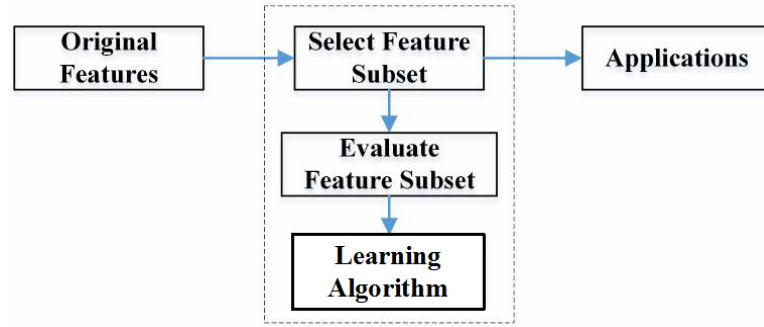


Figure 2.6: A wrapper model for feature selection.

### Embedded Approaches

Similar to wrapper methods, the selection of feature subsets for embedded methods is directed by an inductive method, which is the learning algorithm itself for a given task [60]. They select features in the training process of building a learning model for the task [92]. In other words, embedded methods select features and build a learning model in one step, while filter and wrapper methods realise these in two steps: firstly select features, then conduct the model learning.

They have the advantages of wrapper methods that the selection of feature subsets has an interaction with the learning tasks, so the selected features tend to be more effective than those generated by filter methods. Furthermore, they are less computationally complex than wrappers, as the feature selection is directly included in the construction of a learning model during the training process [60]. However, they are more complex conceptually, and modifications to the learning algorithm may cause poor performance [113].

### 2.2.2 Feature Construction

Feature construction (FC) is the process of transforming a set of original features to a new set of more powerful features [20]. The same as feature selection, feature construction methods can also be grouped into three categories: filter, wrapper and embedded approaches [4]. Filter approaches do not use inductive algorithms for the evaluation of constructed features, while wrapper approaches rely on inductive algorithms. Embedded approaches conduct FC in the process of building a learning model for a problem at hand. The explicit description of filter, wrapper and embedded approaches can be found in Section 2.2.1.

In addition, based on whether a single feature or multiple features is/are constructed, there are single feature construction (SFC) [28] and multiple feature construction (MFC) methods [129]. The constructed features are linear or non-linear transformation of original features, which takes the interactions of original features into consideration, thus making the constructed features more effective than original ones. However, it is likely that a single constructed feature does not contain sufficient information for complex tasks and is not able to outperform the original feature set [99]. In contrast, MFC methods generate multiple high-level features, which is likely for them to contain more information than a single constructed feature.

## 2.3 Evolutionary Computation and Genetic Programming

### 2.3.1 Evolutionary Computation Techniques

Evolutionary computation (EC) is a subfield of artificial intelligence, which is inspired by biological evolution. EC techniques are commonly used for continuous optimisation or combinatorial optimisation problems [49, 180]. In addition, they also have been successfully applied to many image processing and computer vision problems, e.g. edge detection, image segmentation, object detection, classification and recognition. The main



advantage of evolutionary algorithms is that they are not hindered by “preconceptions that limit human problem-solving to well-trodden paths” [82]. EC techniques can mainly be categorised into three groups [49, 180], i.e. evolutionary algorithms, swarm intelligence and others (as shown in Figure 2.7).

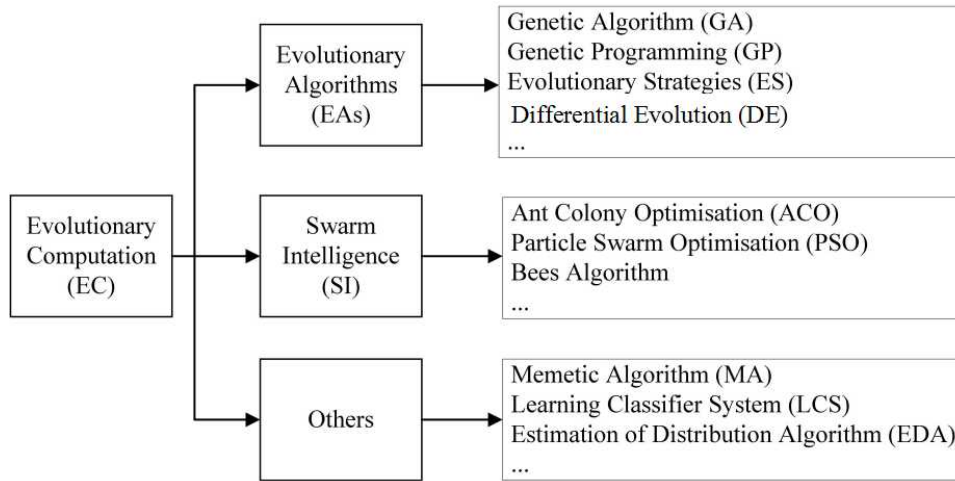


Figure 2.7: Branches of EC algorithms.

EC algorithms have a similar framework, which is presented in Fig. 2.8. An EC algorithm generally includes four major steps [36]. Firstly, an initial population of candidate solutions are created randomly. Secondly, termination criteria (e.g. whether the ideal solution has been found) are checked. Thirdly, a fitness function is pre-defined to evaluate the performance of each candidate for a given problem, and certain candidates are selected based on their fitness values. Fourthly, the selected candidates are employed for reproduction to build a new, yet normally better, generation. Finally, when the termination criteria are met, the best-performing solution in the population is output.

When applying EC algorithms to solve optimisation problems that have multiple objectives to be optimised simultaneously, multi-objective techniques are incorporated into EC algorithms. In addition, for complex

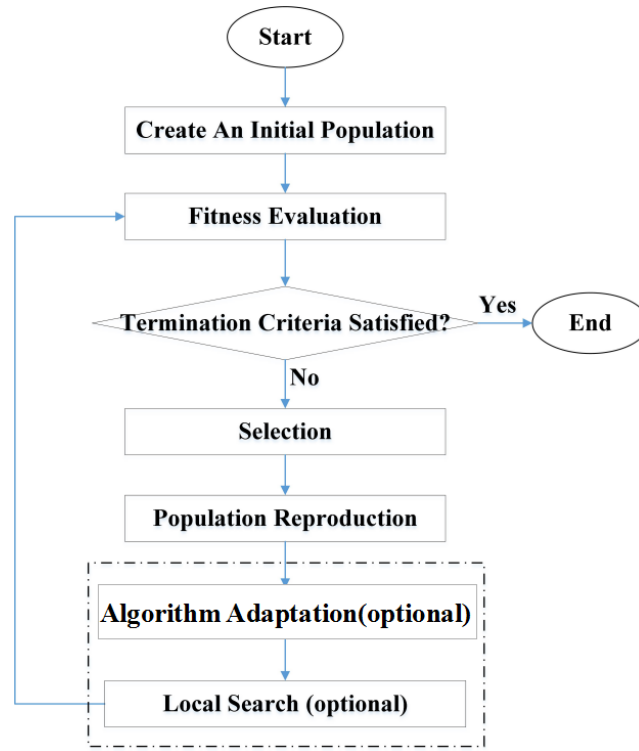


Figure 2.8: The framework of EC algorithms (adapted from Zhang's work [180]).

problems, decomposing the original problems into several subproblems is a possible way to solve them. Coevolutionary techniques employ this divide-and-conquer strategy to solve complex optimisation problems [109]. Therefore, another engagement of EC algorithms is to be combined with coevolutionary techniques to solve complex problems.

### Multiobjective Techniques

Pareto dominance is a popular method to deal with multi-objective problems [115]. In Pareto based optimisation techniques, whether one solution is better than another relies on the notion called dominance, which is defined as follows. Given a set of objectives and two solutions (S1 and S2), solution S1 is said to dominate solution S2, when S1 is not inferior to S2 in

all objectives and S1 is better than S2 in at least one objective. As shown in Figure 2.9, where there are two objectives in this example, solutions that are not dominated by any others are called non-dominated solutions, which form the Pareto front. The result of a Pareto based optimisation method is the Pareto front, which are trade-offs along all the objectives rather than one best solution.

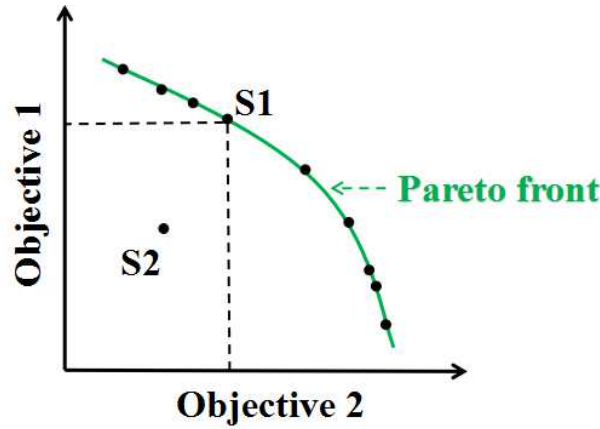


Figure 2.9: Pareto dominance and Pareto front (for both objectives, the higher the better; S1 and S2 refer to two candidate solutions).

There are many existing Pareto based multi-objective optimisation techniques, e.g. niched-Pareto genetic algorithm (NPGA) [69], multi-objective genetic algorithm (MOGA) [55], nondominated sorting genetic algorithm (NSGA, NSGA-II [44]), strength Pareto evolutionary algorithm (SPEA and SPEA2 [183]) and Pareto archived evolutionary strategy (PAES) [37]. According to Coello's work [34], NSGA, NPGA and MOGA are considered as the first generation multi-objective evolutionary algorithms that emphasize simplicity, while SPEA, SPEA2, PAES and NSGA-II are regarded as the second generation algorithms that emphasise efficiency. The similarities and differences of SPEA2, PAES<sup>2</sup> and NSGA-II from the second generation algorithms will be discussed in the following, because they are

<sup>2</sup>Note that there are different versions of PAES, the description of PAES in this thesis refers to (1+1)PAES as it is the basic version.

more advanced than other Pareto based multi-objective techniques. Note that NSGA-III [43, 72] is proposed in 2014, which introduces the reference-point-based nondominated sorting into the NSGA-II framework to solve many-objective (having four or more objectives) problems. As the related multi-objective problems in this thesis have only two objectives, NSGA-III is not considered.

In addition to the Pareto dominance concept, all the three techniques (SPEA2, PAES and NSGA-II) employ a secondary measure for the fitness assignment to keep the diversity of the nondominated solutions [27]. The major differences between these techniques lie in the way of preserving the elite solutions and the way of keeping diversity in the evolutionary process [34]. In terms of preserving the elite solutions, PAES uses a historical archive to store non-dominated solutions. The archive is used as a reference set, against which each newly created solution is compared to determine the current solution for reproduction [37]. Similarly, SPEA2 also uses an external archive elite solutions (mainly non-dominated solutions) [183]. The archive is combined with the current population to generate the next archive, which is then used to produce the next generation. In contrast, instead of creating an archive to hold elite solutions, NSGA-II combines the previous population and current population to ensure elitism [44]. The combined population is sorted based on nondomination to select parent solutions for the creation of the next generation. In terms of keeping diversity, PAES utilises a crowding procedure that divides the objective space recursively, and NSGA-II introduces a crowding distance to estimate the solution density around a specific solution. In contrast, SPEA2 employs a strength value that is portional to the number of solutions around a specific solution.

### **Coevolutionary Techniques**

Coevolutionary techniques are different from other evolutionary algorithms in the methods of evaluating individuals, as individuals work in groups rather than separately [110]. The fitness evaluation of an individual re-

lies on its performance in the context of other individuals by competing or cooperating with them, based on which co-evolution techniques fall into two groups, i.e. competitive co-evolution and cooperative co-evolution. In competitive co-evolution, an individual plays against other individuals within the same population (for one-population competitive coevolution) or in another opponent population (for two-population competitive coevolution) [110]. In cooperative coevolution, a problem is split into multiple sub-problems, and multiple sub-populations are used with each sub-population handling one sub-problem. As shown in Figure 2.10, an individual is evaluated by combining it with one individual from each of the other populations to form a complete solution. As the evaluation context affects an individual's fitness, coevolution techniques typically evaluate an individual multiple times with a different set of other individuals (a different context) each trial, and then computes the fitness based on the results of multiple trials. Therefore, coevolutionary techniques tend to have high computational cost.

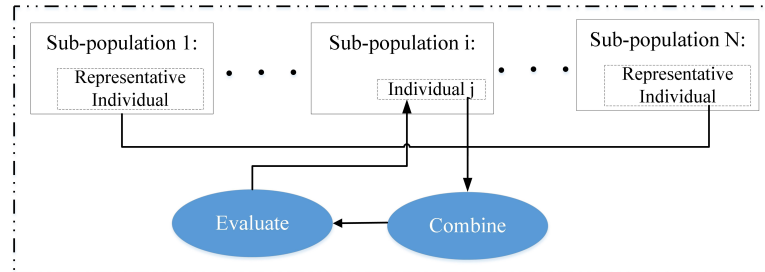


Figure 2.10: Fitness evaluation of an individual in cooperative coevolution.

### 2.3.2 Genetic Programming (GP)

GP can automatically create computer programs/algorithms to solve complex problems without pre-defined solution models [82]. Moreover, solution models for many complex problems may be unknown in practice. Compared with other evolutionary computation techniques, GP is more

flexible [52], as it can utilise complex and variable-length representations, such as trees. The flexibility of GP makes it possible to evolve better solutions than those designed by experts.

Figure 2.11 displays the flowchart of GP. Compared with Figure 2.8 (a general framework of EC techniques), the framework of GP does not include optional steps, e.g. local search for memetic algorithms.

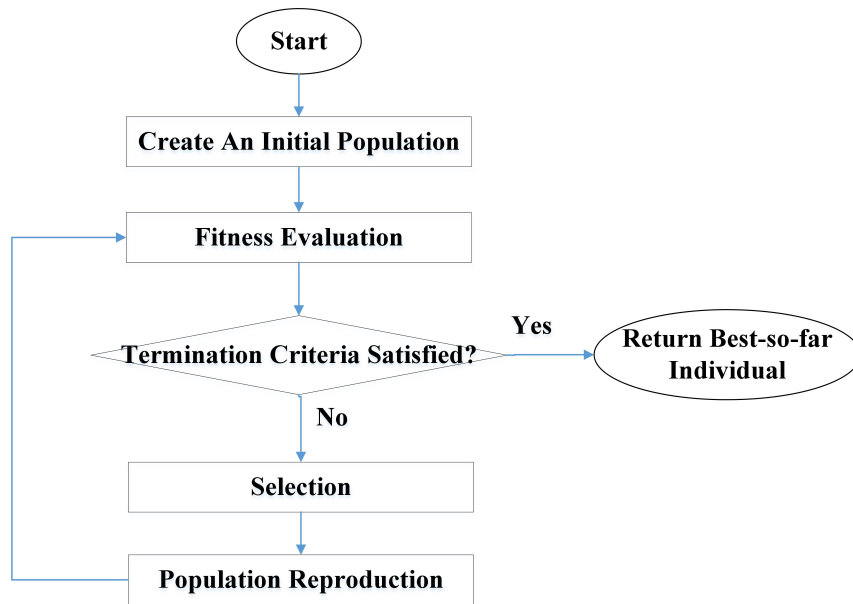


Figure 2.11: The framework of GP (adapted from Poli’s work [135]).

### 1. Initialise a GP population.

A variety of individuals are created randomly for subsequent evolution. There are three common structures used in GP: tree, linear and graph structures [135]. Among them, the tree structure is the most commonly used one, which will be described in Section 2.3.2. The initialisation methods in tree based GP include *full*, *grow* and *Ramped half-and-half* [135]. For the *Ramped half-and-half*, half of the initial population is constructed using the *full* method and half is constructed using the *grow* method. This method tends to bring more diversity to

## 2.3. EVOLUTIONARY COMPUTATION AND GENETIC PROGRAMMING<sup>47</sup>

the initial population than the *full* method or the *grow* method used in isolation.

### 2. Execute each program and evaluate its fitness.

Execute each individual (program) on the target problem at hand. A fitness function needs to be designed to evaluate how well each individual has learned to solve the target problem.

### 3. Select individual(s) from the population based on the fitness value to create usually better solutions through genetic operations.

There are various selection methods, e.g. *fitness-proportional selection*, *truncation selection* and *tournament selection* [16]. The *tournament selection* method is the most commonly used one. There are two steps in tournament selection. Firstly, a set number of individuals are chosen randomly from the population. Secondly, the selected individuals are compared with each other and the best one is chosen to be the parent. Therefore, individuals that have higher fitness values are more likely to be selected for reproduction. For the crossover operator, which needs two parents, two selection tournaments are conducted.

### 4. Create new individual program(s) by applying genetic operations.

The initial population is likely to contain low fitness solutions, so the evolution process is needed to transform the initial population by genetic operators to a new (plausibly better) population generation by generation. There are three principal genetic operators: *crossover*, *mutation* and *elitism*. How the operators are performed will be presented in the Section 2.3.2.

### 5. Stopping criteria.

Two common stopping criteria are that a maximal number of generation is reached or an acceptable solution is found.

### Individual Representation in Tree based GP

Each program in the tree structure is built from primitives in the function set and the terminal set. The terminal set contains the inputs (variables or constants) to GP programs and the function set consists of operators, functions and statements. Both the terminal set and the function set are problem-specific and should be provided by users. Figure 2.12 presents a tree-structured individual. The operators in the red circle are from the function set, which form the internal tree nodes, while the terminal nodes in the blue circle are from the terminal set.

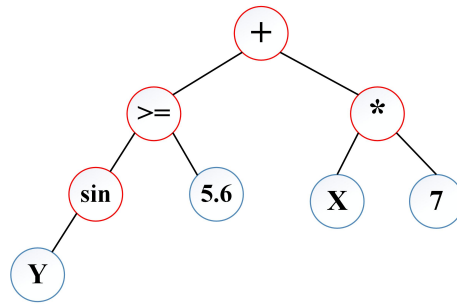


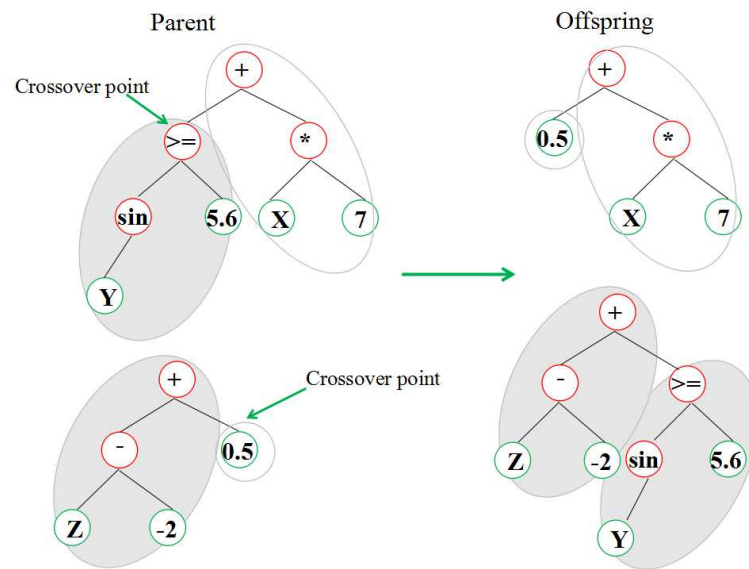
Figure 2.12: An individual in tree structure (*sin* means Sine function; *X*, *Y* are variables).

### Genetic Operators in Tree based GP

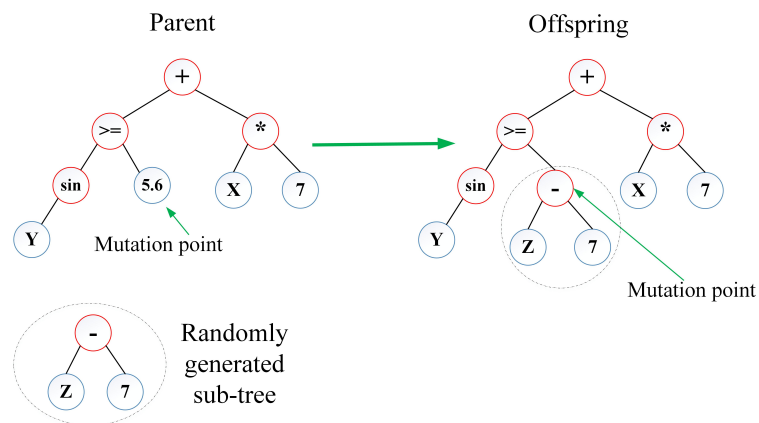
This part discusses the three principal genetic operators: crossover, mutation and elitism [134]. The crossover operation is shown in Figure 2.13a. As can be seen, two parents are needed. Firstly, a crossover point is selected randomly in each parent tree. Next, the offspring is created by replacing the subtrees rooted at the crossover point. The mutation operation is shown in Figure 2.13b. There are also two stages: select a mutation point and then replace the subtree rooted there with a subtree generated randomly. The elitism operator selects the best individuals based on the fitness and simply copies them to the next generation. Note that there has



been much work on optimising genetic operators in GP (e.g. three parents based crossover), but the well tested, understood and standard operators are used here as an initial study into GP for figure-ground segmentation.



(a) Crossover operation.



(b) Mutation operation.

Figure 2.13: Genetic operators (*sin* means Sine function;  $X, Y$  are variables).

### **Parsimony Pressure Techniques**

GP based methods tend to evolve large/complex solutions that are computationally expensive for evolution and later use, difficult to interpret, and are likely to have low generalisation performance [152]. Parsimony pressure is a simple and frequently used technique for bloat control in GP [111].

One parsimony pressure technique is to penalise the fitness of programs by decreasing the raw fitness by an amount dependent on the solution size (parsimony coefficient methods) [97], or by balancing the fitness and complexity of programs with a weight factor (weighted sum methods) [98]. Note that parsimony coefficient methods and weighted sum methods can also be employed for multi-objective optimisation problems by combining multiple objectives linearly to form a single objective. However, the selection of coefficients or weights is critical and difficult to achieve good results, as they depend on the target problem, the function set and terminal set, and parameters [135].

An alternative technique is lexicographic parsimony pressure that treats the fitness of evolved programs as the primary objective and the solution size as a secondary objective in a lexicographic order [111]. Specifically, it uses a modified tournament selection operator to control solution size, described as follows. To select an individual, two individuals are chosen randomly, and their fitnesses are compared: if one of the individuals has a superior fitness, it is selected; if both have the same fitness, the individual with a smaller size is selected; if both have the same fitness and same size, one individual is selected randomly.

## **2.4 GP for Figure-ground Segmentation**

In this part, the existing methods of GP based figure-ground segmentation, bloat control in GP, and GP based feature selection/construction will be discussed.

### 2.4.1 GP based Figure-ground Segmentation Methods

The existing GP based figure-ground segmentation works can be divided into two branches: 1) numerical operators<sup>3</sup> based methods; 2) image processing operators<sup>4</sup> based methods. In the numerical operators based methods, segmentation problems are transformed to window (subimages) classification based tasks. These methods use GP to evolve binary classifiers from image features (or pixel intensities) and numerical operators [47, 96, 100, 133, 160, 161], which can classify subimages as class *foreground* or *background*. In the image processing operators based methods, GP is employed to evolve segmentation algorithms directly from raw images and image processing operators [140, 157].

#### Numerical Operators based Methods

Poli [133] regards image segmentation as a filtering problem, and GP is employed to evolve image filters. The terminal set contains constants and average intensity values of square masks with different sizes, which can be calculated beforehand and stored to speed up GP's evolution process. The function set consists of four arithmetic functions (addition, subtract, multiply and division) and four mask shifting operators (left, right, up and down). The evolved algorithm is tested on two medical images, achieving much higher scores in sensitivity and specificity than an artificial neural network approach. However, two images are not sufficient for testing; moreover, the images are relatively simple with clean backgrounds.

Song et al. [160, 161] employ GP to segment texture images. The segmentation problems are formulated as cutout (window) classification based problems, and the cutouts are sampled from various textures on training images. Only raw pixel values of cutouts are used in the terminal set; the function set consists of arithmetic and logic operators, e.g. bigger

---

<sup>3</sup>Operators that process numeric values, e.g. add and subtract.

<sup>4</sup>Operators that process raw images, e.g. image filters.

than, smaller than and equal. Results on bitmap patterns and Bradatz textures show that the method can identify boundaries accurately, including irregular boundaries (such as curves). As the method used raw pixel values as input directly, it is a hundred times faster than traditional segmentation methods (with feature extraction) for testing. However, the training time for GP to evolve classifiers is quite long, e.g. from hours to days. Moreover, the cutout related parameters (e.g. cutout size) are required, which rely on expertise or trail-and-error.

Dong et al. [47] employs GP to evolve segmentors to segment the corpora lutea on 30 medical images. The terminal set contains features extracted by rotation invariant local binary patterns, and the function set includes arithmetic (e.g.  $+$ ,  $-$ ,  $*$ ,  $/$ ,  $\log$ ,  $\max$ ), logical (e.g.  $\text{and}$  &  $\text{\&}$ ) and relation (e.g.  $<$ ,  $>$ ) operators. Even though the segmentation accuracy is similar to a level set based method, the proposed method is fully automatic and five times faster. However, the proposed method is only tested on 30 medical images with clean backgrounds.

### **Image Processing Operators based Methods**

Singh et al. [157] use GP to evolve Matlab programs from images and primitive image operators for image segmentation, which produce consistently good results on cell images. The solution representation is a linear sequence of image operations in Matlab. The proposed method is used to segment cells on biological images (30 images for training and 100 images for testing). Results show that the proposed method achieves a higher segmentation accuracy and a higher cell count rate than a genetic algorithm based segmentation method. However, this work is only tested on relatively simple biological images that have clean backgrounds, and the linear sequence representation is less flexible than non-linear representations, e.g. trees.

Roberts [140] introduces GP to solve computer vision and image processing problems. To reduce the high computational cost, a caching mech-

anism is designed in the GP based system, which decreases the number of fitness evaluation by 66 % at most, leading to a 52 % decrease in evolution time. It suggests that the caching system can be used to solve real-world problems. This method performs well in the segmentation of pigmented skin lesions (moles) on images. As the segmentation performance is not the primary focus, extensive experiments are not conducted.

### Challenges

The works described above achieve promising results; however, there are several drawbacks/challenges listed as follows.

- The existing works only test a limited number of image domains, including texture images [160, 161], and medical images [133, 140, 157]. As the tested medical images all have clean backgrounds, they are considered as “simple images”. Therefore, whether GP-evolved segmentors can deal with a wide range of images, especially “complex images” with high variations, is still not clear.
- Compared with image processing operators based methods, numerical operators based methods are more efficient [140], as numerical operators require much less computational cost than image processing operators. Moreover, the numerical operators based approaches use a window classification based segmentation framework that is a common way to use GP in image-related tasks. Therefore, the window classification based framework will be employed as the basic approach to apply GP in figure-ground segmentation in this thesis. However, window related parameters (e.g. sizes and shifting steps of windows) in the framework are required. As the window related parameters rely on human expertise to be set, which are difficult for users, improvements on the window classification based framework for employing GP in figure-ground segmentation deserve investigation.

- Only intensities [140, 157, 160, 161], intensity statistics [133] and local binary patterns (LBP) [47] have been used as input features to evolve segmentors. As the input of GP has a great influence on its performance, it is hypothesised that if the terminal set contains certain kinds of features, such as histogram statistics and Gabor features, better performing segmentors will be evolved.
- Bloat problems are common for GP based methods that tend to evolve large/complex solutions [160, 161]. The evolved complex solutions are computationally expensive for evolution and later use, difficult to interpret, and are likely to have low generalisation performance. Therefore, new GP based methods, aiming to evolve segmentors balanced between the segmentation performance and the complexity in terms of solution size, deserve to be investigated.
- Even though promising results have been achieved by existing works, the selected image datasets are relatively simple. When segmenting images with high variations, effective feature sets are required to capture the distinguishing information between objects and backgrounds. Feature manipulation, e.g. feature selection and feature construction, can help produce more effective feature sets from primitive features extracted by existing image descriptors.

### 2.4.2 Bloat Control in GP

Bloat is a common problem in GP based methods, which is that the increase of the average tree size and depth does not lead to a corresponding increase in the solution fitness [112]. There are three commonly used techniques that have been introduced in GP to control bloat. The most common approach of bloat control is to place constraints on the depth of the evolved programs [21, 112]. For example, in Koza's work [83], children with a depth more than 17 are rejected and their parents are copied to the next generation. In addition, a popular approach is parsimony pressure,

which incorporates the solution size as an additional, but hidden, objective [163].

An alternative approach for bloat control is to employ multi-objective optimisation techniques in GP to produce trade-off solutions between the solution functionality and the complexity in terms of the solution size [21, 98, 147, 152, 168]. For multi-objective optimisation methods, each objective is treated separately and independently, aiming to evolve a Pareto front of trade-off solutions based on all the objectives. Compared with parsimony pressure techniques (parsimony coefficient and weighted sum methods) that aggregate the objectives into a scalar fitness function based on pre-defined weights or coefficients, multi-objective methods treat each objective separately and independently. Therefore, multi-objective methods search the space of two independent objectives, while the objective space of parsimony pressure based methods are restricted to only a part of that for multi-objective based methods, where the two objectives suit a linear relation determined by the weights or coefficients.

Wang et al. [168] propose a multi-objective GP approach (MOGPA) to identify implicit and explicit relationships in a given set of data. MOGPA has two objectives, the complexity of the expressions reflected by the maximum depth of solutions and the mean error. The performance of the evolved expressions are tested on four examples, i.e. Colebrook-White Formula, one circle and two elliptic curve equations. The proposed multi-objective GP method produces solutions that have smaller mean errors with lower sizes than methods from previous studies [40]. This reflects that GP has potential to solve problems with multiple objectives properly by incorporating with multi-objective techniques.

Shao et al. [152] developed a multi-objective GP method to evolve global feature descriptors for image classification tasks. The proposed MOGP method has two objectives, i.e. the classification error and the number of tree nodes. Better classification accuracies were achieved than other state-of-the-art hand-crafted feature extraction techniques, e.g. LBP

(local binary patterns) and Gabor filters (to be described in Section 2.4.2). However, the new MOGP method needs a long evolution time, e.g. 7.6 hours on Caltech-101 dataset.

Sarro et al. [147] exploited GP to estimate software development effort. They compared the single-objective GP (SOGP) with five different fitness functions, e.g. mean magnitude of relative error, and the MOGP that considers the five functions simultaneously. The result shows that the selection of the fitness function affects the estimation accuracy in single-objective GP significantly. Moreover, only SOGP with certain fitness functions can achieve similar results with those produced by MOGP.

Bleuler et al. [21] used a multi-objective technique in GP to reduce the effects of bloating by considering the program size as a separate objective. They combine GP and the multi-objective technique SPEA2 (strength Pareto evolutionary algorithm). The new method was compared with four other strategies (i.e. standard GP, Two Stage, constant parsimony, adaptive parsimony). The “Two Stage” method has two stages of fitness evaluation: optimise the functionality first and then the solution size. The proposed MOGP in paper [21] is claimed to outperform the four strategies to reduce bloat in regard with the convergence speed and the program size. However, this MOGP method was only tested on a simple parity problem.

In summary, placing constraints on the solution depth/size is simple, yet this technique makes the population tend to be filled up with large individuals on the edge of the constraints. Parsimony pressure techniques are effective, but the coefficients or weights of the penalty term in the fitness function are required. In addition, multi-objective techniques treat solution performance and complexity measures separately and independently. Compared with parsimony pressure techniques, multi-objective techniques do not require parameters (e.g. coefficients or weights) in the fitness evaluation. which do not require parameters in the fitness evaluation. Therefore, multi-objective techniques will be investigated in this thesis to solve multi-objective problems.



### 2.4.3 GP based Feature Selection/Construction Methods

This section analyses the existing works that specifically use GP to select/construct features, and presents the drawbacks and possible improvements. Figure 2.14 and Figure 2.15 show the general flowcharts of using GP for feature selection and construction respectively.

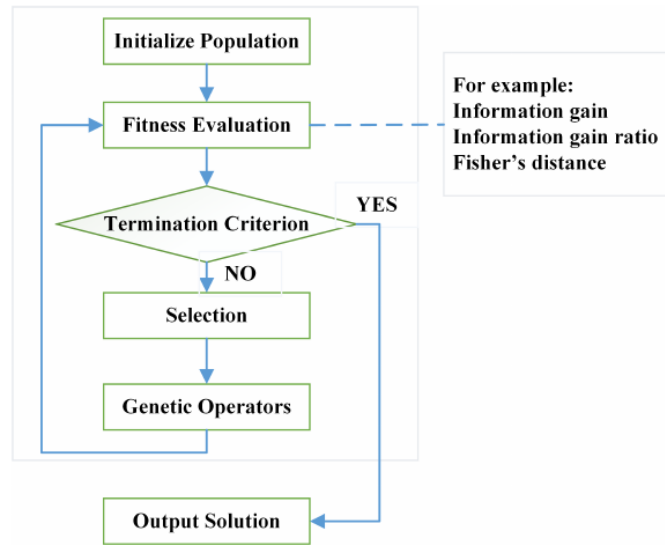


Figure 2.14: Filter Feature Selection/Construction based on GP.

#### GP based Feature Selection Methods

Recent works [3, 41, 124, 158] on GP based feature selection are described as follows, which shows GP's potential for feature selection (FS).

Nag et al. [124] apply embedded feature selection in designing classifiers using a multi-objective GP method. In this method, feature selection is conducted in the population initialisation, crossover and mutation operations of the GP's evolutionary process. To achieve this, a new initialisation method and new crossover and mutation operators are created. This method is tested on eight micro-array and eleven text datasets. Compared with seven FS methods (e.g. ReliefF [141] and Consist [39]), the proposed

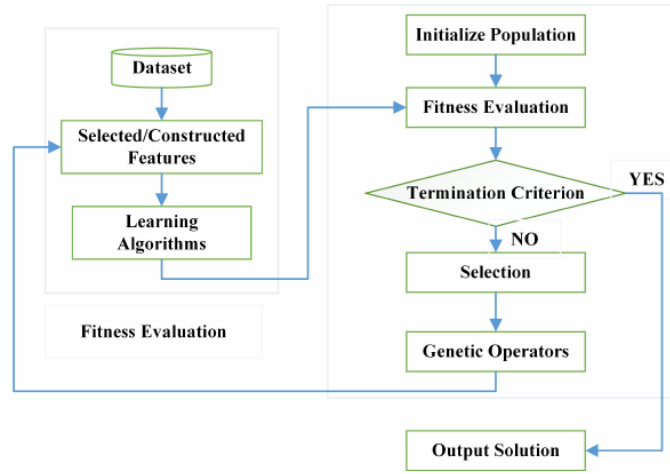


Figure 2.15: Wrapper Feature Selection/Construction based on GP.

method is better for 80% of the cases. However, this method may be inefficient for problems with a large number of samples or classes.

Smart et al. [158] apply GP to design a filter based method to select features for pattern classification problems on functional magnetic resonance imaging (fMRI) and intra-cranial electroencephalogram (iEEG) signals. The nearest neighbor classifier is employed for classification with the selected features. On fMRI dataset, over 60% median sensitivity and over 60% median selectivity are achieved, while on iEEG dataset, both median sensitivity and median selectivity are over 65%. One drawback is that this method requires parameter setting in the fitness function, and the parameter settings are not guaranteed optimal.

Davis et al. [41] present a two-stage genetic programming (GP) framework for feature selection. Firstly a subset of features is selected using GP based on its built-in feature selection process (embedded approach). Then GP is used again to evolve classifiers from the selected features. Due to the resultant smaller size of the feature subset, GP evolves smaller programs that are easier to interpret than using the complete feature set. In addition, this two-stage method obtains better classification results and faster con-

vergence rates than standard GP. However, the first stage just aims to find discriminant features and the evolved classifiers are discarded, then the second stage produces the final classifiers using the selected features. As the evolutionary process of GP is time-consuming, this two-stage process may be inefficient.

Ahmed et al. [3] take advantage of GP's implicit feature selection ability (or embedded feature selection, see Section 2.1.3). They use GP to evolve classifiers, and only certain features are used by the classifiers. Further processing is conducted. Specifically, the features selected are sorted, then as several top features perform better than all the selected features, less capable features are further reduced. Compared with other feature selection methods (e.g. linear Support Vector Machine Recursive Features Elimination, SVM-RFE) and original features, the results show that the proposed method achieves higher biomarker detection rates. This reflects that GP can implicitly select useful features, which will be investigated for figure-ground image segmentation in this thesis.

There are several issues in the existing works. Firstly, due to GP's characteristic that can implicitly select features in the evolutionary process, most existing works conduct embedded feature selection, while GP's ability for explicit feature selection, i.e. filter and wrapper feature selection, has not been sufficiently exploited. Secondly, feature selection is a multi-objective task, maximizing the effectiveness of the selected feature subset and minimizing the number of selected features. However, there are few multi-objective GP works for feature selection.

### **GP based Feature Construction Methods**

GP is a well-suited technique for feature construction (FC) [20]. As GP can utilise complex and variable-length representations, such as trees and graphs, it can combine input features to form linear or non-linear interactions. Meanwhile, GP searches the new feature space and helps generate high performing feature subsets. Moreover, it is possible to interpret and

understand the new features constructed by GP evolved transformations. Existing works [4, 20, 28, 64, 65, 129] are described as follows, which show GP's potential for feature construction.

Guo et al. [64, 65] develop feature construction methods using GP, whose fitness measure is based on Fisher criterion<sup>5</sup>. One of their works [65] presents a nonlinear feature construction method using GP for two-class and multi-class recognition problems. The new feature set is tested by k-nearest neighbour (KNN) and minimum distance classifier (MDC). In another work [64], a new approach is presented using GP to construct features for breast cancer diagnosis. The original features consist of constant values and 30 real-valued cancer attributes. A novel feature measure, modified Fisher linear discriminant analysis (MFLDA), is developed to measure the scattering of different classes. A neural network and a support vector machine are used to test the new features. As one GP run can produce one best solution, from which one feature can be constructed, these methods reduce the original feature set to only one dimensional feature set. However, it may not be sufficient to use only one feature for certain tasks, e.g. classification [127].

Neshatian et al. [129] employ GP to build a multiple feature construction system for classification problems, which construct one feature for each class, so the number of constructed features equals to the number of classes. The system is a filter method, which uses an entropy based fitness function. To construct multiple features, they propose a decomposable fitness function, which is based on a certain class that a feature is being constructed for. Therefore, each GP run can only build one feature for one class, and the construction of multiple features requires multiple GP runs, which is inefficient.

Bishop et al. [20] present a wrapper construction method using GP for image classification. As applications in image search, compression and

---

<sup>5</sup> Fisher criterion is a discriminant criterion function that is the ratio of the between-class scatter to the within-class scatter [95].

photography require the information of image aesthetic quality, it is necessary to construct features that can predict the aesthetic quality of an image. The fitness function is the classification accuracy. The constructed features outperform the original features in terms of the classification accuracy. Even though the features constructed by this GP-based wrapper approach are suitable for image aesthetics classification, it is not clear whether they can be generalised to other image domains.

Ahmed et al. [4] propose an embedded approach to construct multiple features using GP. Compared with common GP based FC methods, which only take the output of root node as the single constructed feature, they consider that the output of subtrees are also useful features, thus one GP tree solution can generate multiple features. The fitness function combines Fisher criterion<sup>5</sup> and p-values<sup>6</sup> to measure the discriminating information between different classes. The features constructed by the proposed method are more effective than original features on eight different mass spectrometry data sets in classification performance. The major drawback of this work is that the proposed method has not been compared with related feature construction methods.

Cano et al. [28] present a Pareto-based multi-objective genetic programming algorithm to extract feature and visualise data for both balanced and unbalanced high-dimensional data. It aims to optimise the classification performance and data visualisation simultaneously. The proposed algorithm is evaluated sufficiently by comparison with 11 well-acknowledged feature extraction methods (e.g. linear discriminant analysis, and random projection) on 22 balanced and 20 imbalanced data sets. The results show that accurate classification performance and good visualisation measures for both balanced and imbalanced data have been achieved.

Even though promising results have been achieved by the existing GP

---

<sup>6</sup>p-values measure the between-class and within-class separability, which are calculated based on the one way analysis of variance test [4].

based feature construction works, there are still several issues. Firstly, it is still challenging to design an effective fitness function to measure the goodness of the constructed features, especially for filter methods. Secondly, existing works construct features by mixing different types of image features homogeneously, which may lead to unnecessary interactions of non-related features, lowering the effectiveness of constructed features. Therefore, it is necessary to further investigate the potential of GP for feature construction.

## 2.5 Standard Image Datasets, Features, Segmentation Measures and GP Settings

### 2.5.1 Standard Image Datasets

Images from four databases, i.e. bitmap patterns [161], Brodatz texture database [67], Weizmann horse database [22, 24] and Pascal Visual Object Classes 2012 (Pascal VOC2012) dataset [53], are used to develop and test the novel methods in this work. These four datasets are selected as they cover diverse image domains and are increasingly difficult for the segmentation task. In practice, grayscale, large-sized and images with high variations in objects/backgrounds are more complex than binary, synthetic or small-sized images for segmentation tasks [133]. Note that once GP has been investigated on these datasets, other datasets (e.g. SED2 [8]) are used for further investigation (see Chapter 7).

Figure 2.16 shows a bitmap pattern image, a texture image and example images from Weizmann and Pascal datasets, along with their ground truth (foreground objects are in white colour and backgrounds in black colour). Specifically, there are two bitmap images, named as “Rectangular” and “Butterfly”, whose sizes are 256 by 256 pixels. They are synthesised from two bitmap patterns, P14 and P24. The two images are synthetic binary images, which are relatively simple. One example texture

image, D24vs34, is a grayscale image synthesised from two Brodatz textures that are D24 and D34. Its size is 320 by 160 pixels. In addition, images from the Weizmann and Pascal databases are object images that are relatively difficult segmentation tasks, as there exist high variations (e.g. varied object shapes and cluttered backgrounds) and some images have low quality.

The Weizmann horse dataset [25] consists of 328 horse images, among which certain images (e.g. horse219 and horse227 in Figure 2.16) have varying horse poses and cluttered backgrounds. Moreover, there are images (e.g. horse264) containing noise that lowers the image quality and makes segmentation tasks more challenging. Therefore, Weizmann images are regarded as difficult segmentation tasks. From the ground truth of Weizmann images (Figure 2.16), it can be seen that each image contains one target object (a horse). The average size of Weizmann images is around  $250 \times 200$  pixels.

The Pascal VOC challenges [53] were developed from 2005 to 2012. Pascal VOC2012 is the updated version, and the images of the aeroplane class are selected in this thesis. Note that only one class of images are selected as GP is mainly investigated to segment specific classes of objects in this thesis. As the GP methods are developed to be applicable to a wide range of images, after GP is fully studied on images with specific classes of objects, the proposed GP methods will also be tested on other classes in Pascal and other datasets with various classes of images (see Chapter 7).

Specifically, Pascal VOC2012 has 178 aeroplane images, which are highly varied in object sizes, object shapes (e.g. airliners and fighter aircrafts) and locations. Moreover, there are images with cluttered backgrounds, e.g. image 2011\_001880, and certain images have low quality, e.g. image 2007\_001761 (blurred by motion) and image 2010\_002939 (containing noise). Therefore, Pascal aeroplane images are considered as difficult segmentation tasks. According to the ground truth of Pascal images in Figure 2.16, except for object pixels (in white colour) and background pixels (in

black colour) there are also gray pixels, which are “void” pixels. To reduce the labelling time of the ground truth, this dataset treats the bordering region with a width of five pixels as “void” pixels that may contain both object and background pixels. In addition, the ambiguous or heavily occluded objects and small regions (too small to be labelled) are also labelled as “void” pixels. The “void” pixels are excluded for being selected as training samples and being used for the result evaluation. The average size of Pascal images is around  $500 \times 350$  pixels.

### 2.5.2 Image Features

Texture features, e.g. Gabor [178], local binary patterns (LBP) [132] and mean/median features, are widely-used for feature extraction in image analysis tasks. They will be described in this section.

The Gabor features are one of the most powerful edge descriptors, and are generated from the convolution response of Gabor filters to an image. A Gabor filter with specific parameters (scale and orientation) can capture information at the certain scale and orientation [178]. Filters generated from five scales ( $4, 4\sqrt{2}, 8, 8\sqrt{2}, 16$ ) and eight orientations ( $0, \frac{\pi}{8}, \frac{2\pi}{8}, \frac{3\pi}{8}, \frac{4\pi}{8}, \frac{5\pi}{8}, \frac{6\pi}{8}, \frac{7\pi}{8}$ ) are commonly used.

LBP is a simple and effective method for extracting texture features [130]. It can transform an image to an array of integer labels that can represent the small-scale appearance of the image [132]. In addition, the mean filter and the median filter are commonly used to remove noise in images, and are also employed to extract local statistical features. Specifically, for each pixel, the mean filter (or the median filter) studies its neighbors, which are captured by a window centered at this pixel, and then replaces the pixel value with the average value (or the median value) of the neighboring pixels.



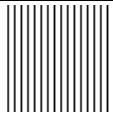

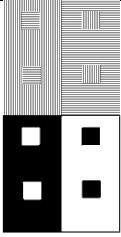
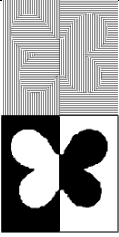
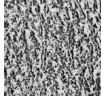
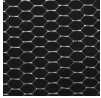
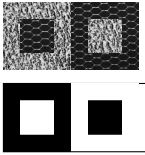




















Bitmap					
	P14	P24	Rectangular	Butterfly	
Brodatz					
	D24	D34	D24vs34		
Weizmann					
					
	horse219	horse224	horse225	horse227	horse264
Pascal					
					
	2007_001761	2009_000032	2010_002939	2010_003127	2011_001880

Figure 2.16: Example images.

### 2.5.3 Evaluation Methods

Three evaluation methods for evaluating segmentation result images will be described in this section. The first evaluation method is the segmentation accuracy (Equation 2.1). This measure is simple and commonly used, but it is insufficient to assess segmented results. For example, when objects take up a small proportion of an image, even though the whole image is

segmented as background, the segmentation accuracy would still be quite high. Therefore, another two evaluation methods,  $F_1$  measure (Equation 2.2) and NRM (negative rate metric, Equation 2.3), are also selected to compensate this measure.

$$\text{Segmentation Accuracy} = \frac{TP + TN}{\text{Total.Pixel.Number.of.All.Test.Images}} \quad (2.1)$$

$F_1$  score combines precision and recall together, which is a relatively unbiased measure. In addition, NRM takes mismatches between a prediction and the ground truth into account [12], and it is a single measure combining false negative rate and false positive rate.  $F_1$  reaches its worst at 0 and best value at 1, while NRM is worst at 1 and best at 0.

$$\begin{aligned} F_1 &= 2 \times \text{Precision} \times \text{Recall} / (\text{Precision} + \text{Recall}) \\ \text{Precision} &= TP / (TP + FP) \\ \text{Recall} &= TP / (TP + FN) \end{aligned} \quad (2.2)$$

$$\begin{aligned} \text{NRM} &= (FNR + FPR) / 2 \\ FNR &= FN / (TP + FN) \\ FPR &= FP / (FP + TN) \end{aligned} \quad (2.3)$$

where  $TP$ ,  $TN$ ,  $FP$  and  $FN$  stand for true positives, true negatives, false positives and false negatives respectively. To get the average segmentation accuracy,  $F_1$  score and NRM score on a test set,  $TP$ ,  $TN$ ,  $FP$  and  $FN$  are calculated based on all the test images. Specifically,  $TP$  means the number of object pixels that are correctly segmented as objects across all the test images;  $FP$  means the number of background pixels that are incorrectly segmented as objects across all the test images; and so forth for the  $TN$  and  $FN$ .

#### 2.5.4 Koza's GP Settings

John Koza is known for pioneering the use of genetic programming for optimisation of complex problems. Default parameter settings of GP in this

thesis follow the settings of Koza's works [83, 84, 85]. Table 2.1 presents the major GP parameters. Specifically, the number of generations is 51 and the population size is 1024. The initialisation method is Ramped half-and-half that builds initial individuals using the *grow* method based on the probability for *grow* (0.5), otherwise using the *full* method. In the *full* method, nodes are chosen randomly from the function set until this maximum tree depth is reached, and then only terminals can be chosen. In contrast, in the *grow* method, nodes are selected from both the function and terminal sets until the depth limit is reached, then only terminals may be chosen. The minimum depth of initial individuals is two and the maximum depth of initial individuals is six, while the maximum individual depth in the evolutionary process cannot exceed 17. In addition, the tournament selection method is used to select individuals for reproduction in the evolutionary process, which randomly selects individuals from the population to construct tournaments (tournament size: seven), and the winners of which are then selected. Moreover, the crossover and mutation operators are reproduction methods, whose rates are 0.90 and 0.10 respectively.

Table 2.1: Koza's GP Settings.

Parameter	Setting	Parameter	Setting
Generation	51	Population Size	1024
Initialiser	Ramped half-and-half	Probability for <i>grow</i>	0.5
Min-depth for Initialiser	2	Max-depth for Initialiser	6
Maximum Tree Depth	17		
Selection	Tournament	Tournament size	7
Crossover Rate	0.90	Reproduction (elitism) Rate	0.10

## 2.6 Chapter Summary

This chapter briefly described the basic concepts of figure-ground image segmentation, feature selection/construction, evolutionary computation

techniques and especially the GP algorithm. In addition, the existing works of GP based segmentation and GP based feature selection/construction were also reviewed. The limitations of the existing works were discussed, which leads to the motivations of this thesis. The overall motivation of applying GP for figure-ground segmentation is that GP has been successfully used to conduct figure-ground segmentation, and GP has the characteristics that make it suited for figure-ground segmentation, i.e. involving less human work, being flexible (e.g. variable-length representations), and having powerful search ability.

However, there are only a limited number of works investigating figure-ground segmentation using GP, and challenges/limitations still exist. Therefore, it is worth further investigating GP's potential for segmentation tasks. Specifically, the limitations of existing works and the motivations of this thesis are summarised as follows.

- Only intensity based features and LBP features have been employed as inputs of GP to evolve segmentors. Since the inputs have a great influence on the performance of evolved solutions [83], it is assumed that if the terminal set contains certain kinds of image information (e.g. texture) known to suit target image domains, better performing segmentors will be evolved. Therefore, it is worth investigating what image information is necessary for GP to derive effective segmentors.
- GP based methods often face bloat problems that cause large/complex solutions being evolved. The complex solutions are computationally expensive for evolution and later use, difficult to interpret, and likely to have low generalisation performance. Therefore, it is important to control bloat in GP when applying it to solve segmentation problems.
- GP has been applied to feature selection, and promising results have been achieved. However, there is no work that introduces GP based

feature selection for figure-ground segmentation tasks. Moreover, feature selection is a multi-objective task, maximizing the effectiveness of the selected feature subset and minimizing the number of selected features, but existing GP based feature selection works rarely consider the number of features.

- GP is well-suited for feature construction. Even though promising results have been achieved by existing GP based feature construction works, there are still several issues. Firstly, the constructed features of existing methods are mainly used for detection and classification problems, other tasks (e.g. segmentation problems) have not been extensively investigated. Secondly, it is still challenging to design an effective fitness function to measure the goodness of the constructed features, especially for filter methods. Thirdly, existing works construct features by mixing different types of image features homogeneously, which may lead to unnecessary interactions of non-related features, lowering the effectiveness of constructed features.
- GP's potential in different aspects for figure-ground image segmentation will be further investigated, e.g. by testing on datasets containing various types of foreground objects and by conducting transfer validation tasks. Moreover, comparison among the proposed GP methods and with non-EC methods deserves further investigations.

The following chapters will investigate those issues. Specifically, what kind of image information that GP determines necessary to evolve capable segmentors will be investigated in Chapter 3. A parsimony pressure and multi-objective techniques will be introduced in standard GP to design new segmentation methods, which are expected to balance the solution functionality and complexity in terms of the solution size, in Chapter 4. In addition, GP will be employed for the first time to develop feature selection methods for figure-ground segmentation tasks in Chapter 5, where feature selection will be treated as a multi-objective task: maximizing the

segmentation performance based on the selected features and minimizing the number of selected features. Moreover, GP will be applied for the first time to construct high-level features to improve the segmentation performance in Chapter 6. Specifically, a new entropy based fitness function will be developed for filter methods, and the cooperative coevolution technique will be introduced in GP to construct features from different types of features separately. The potential of proposed GP methods will be further investigated, and comparison among the proposed methods and with non-EC methods will be conducted in Chapter 7.

## Chapter 3

# GP for Evolving Segmentors from Diverse Image Features

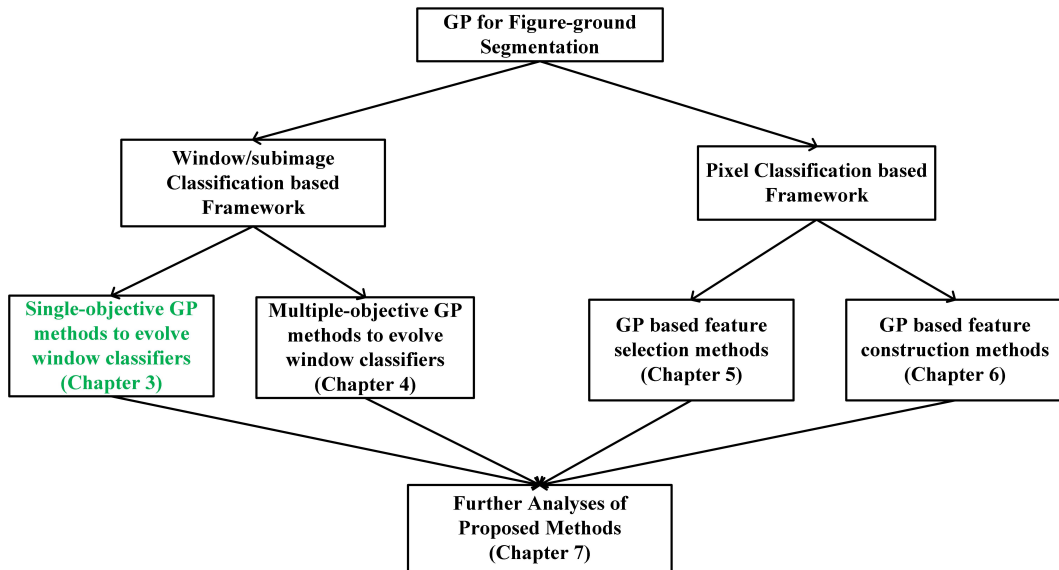


Figure 3.1: The overall structure of contributions.

Although GP has already been introduced in the field of image segmentation since the 1990s, there are only a limited number of related works. Therefore, GP has not been sufficiently investigated for image segmenta-

tion. As the inputs of GP have a great influence on the performance of evolved segmentors, it is necessary to investigate diverse types of image features, aiming to determine what image information is necessary for GP to evolve capable segmentation algorithms (segmentors).

## 3.1 Introduction

GP is developed to evolve segmentation algorithms from diverse image features in this chapter. This is because GP has only been applied by a limited number of segmentation works, i.e. numerical operators based methods [47, 133, 140, 160, 161], and image processing operators based methods [140, 157] (see Section 2.4.1), in a limited number of image domains. Moreover, even though the inputs of GP have a great impact on the performance of evolved solutions, the existing work had only investigated several image features as the inputs of GP. Therefore, it is worthwhile investigating what image information is necessary for GP to derive effective segmentors on a wide range of image domains.

### 3.1.1 Chapter Goals

The aim is to determine what kind of image information is necessary for GP to evolve capable segmentors. Diverse types of image features are investigated, which cover three general types of image properties (i.e. colour or brightness, texture and shape). Note that these types of features are tested individually rather than in combination as this chapter focuses on testing the effectiveness of each individual type of feature for GP based figure-ground segmentation. A combination of features may be better than a single feature alone, but this has not been tested in this chapter. In addition, images with different difficulty levels for segmentation are selected to test GP evolved segmentors, including simple bitmap, synthetic texture images and complex object images with high variations.



Specifically, this chapter will investigate the following objectives:

1. quantify the effectiveness of different image features that are used to evolve segmentors by GP,
2. investigate whether the GP evolved segmentors can have a consistent performance across a wide range of image domains,
3. investigate whether the proposed method can outperform widely-used conventional segmentation techniques.

### 3.1.2 Chapter Organisation

The rest of this chapter is organised as follows. The framework of the window classification based segmentation using GP, and seven feature descriptors selected for feature extraction are described in Section 3.2. Experiment preparations, e.g. image datasets and experiment design are introduced in Section 3.3. In addition, Section 3.4 and Section 3.5 discuss results on the selected image datasets, and conduct comparisons between the evolved segmentors and with conventional segmentation techniques respectively. Section 3.6 summarises the contributions of this chapter.

## 3.2 GP based Figure-ground Segmentation

The GP based figure-ground segmentation method in this chapter employs the window classification based segmentation framework, where GP is used to evolve subimage classifiers. The framework and settings of the GP method are described as follows.

### 3.2.1 Window Classification based Segmentation Framework

As displayed in Figure 3.2, there are three major steps in the window classification based segmentation framework.

Firstly, a classifier is evolved by GP. Specifically, an equal number of foreground and background cutouts (subimages) are captured by a pre-defined window (see Section 3.3.2 for settings) from training images. For example, as shown in Figure 3.2, five foreground cutouts and five background cutouts are captured on each training image. For each cutout, feature extraction is conducted to form a training sample, which is labelled as class  $A$  (foreground) or  $B$  (background) based on the ground truth, e.g.  $f_1, f_2, f_3, \dots A$ . Details of feature extraction methods are provided in Section 3.2.2. The training samples are input to GP so that it can evolve a classifier that can classify cutouts/subimages as class  $A$  or  $B$ .

Secondly, a sliding window (the same size as the window in the first step) is used to sweep across a test image. Specifically, the sliding window starts from the top-left of a test image, sweeping across to the bottom-right of the image with the moving step of  $dx$  and  $dy$  in horizontal and vertical directions respectively. Cutouts are captured by the sweeping window, and then feature extraction is conducted on the cutouts to form the test samples, which are then input to the evolved classifier in the first step. The evolved classifier categorises the cutouts as class  $A$  or  $B$ , based on which a class label is assigned to all pixels of each cutout. For example, if one cutout from the test image is categorised as class  $A$ , the pixels belonging to this cutout are all assigned a class  $A$ .

Thirdly, as the sweeping window has overlaps, most pixels in the test image have more than one assigned label, e.g.  $P(*, *)A; A; B; A\dots$ . The number of the assigned labels for each pixel is determined by the location of this pixel and window related parameters, i.e. the window size and the moving steps  $(dx, dy)$  (see Section 3.3.2 for settings). Therefore, a majority voting process is employed to obtain the final estimated label for each pixel in the test image (a random selection is applied to cases with an equal number of labels for each class, e.g.  $A; A; B; B$ ). Eventually, foreground pixels are assigned a gray-level value of 255 and background ones are given 0, based on which the segmentation result (a binary image) can

be generated.

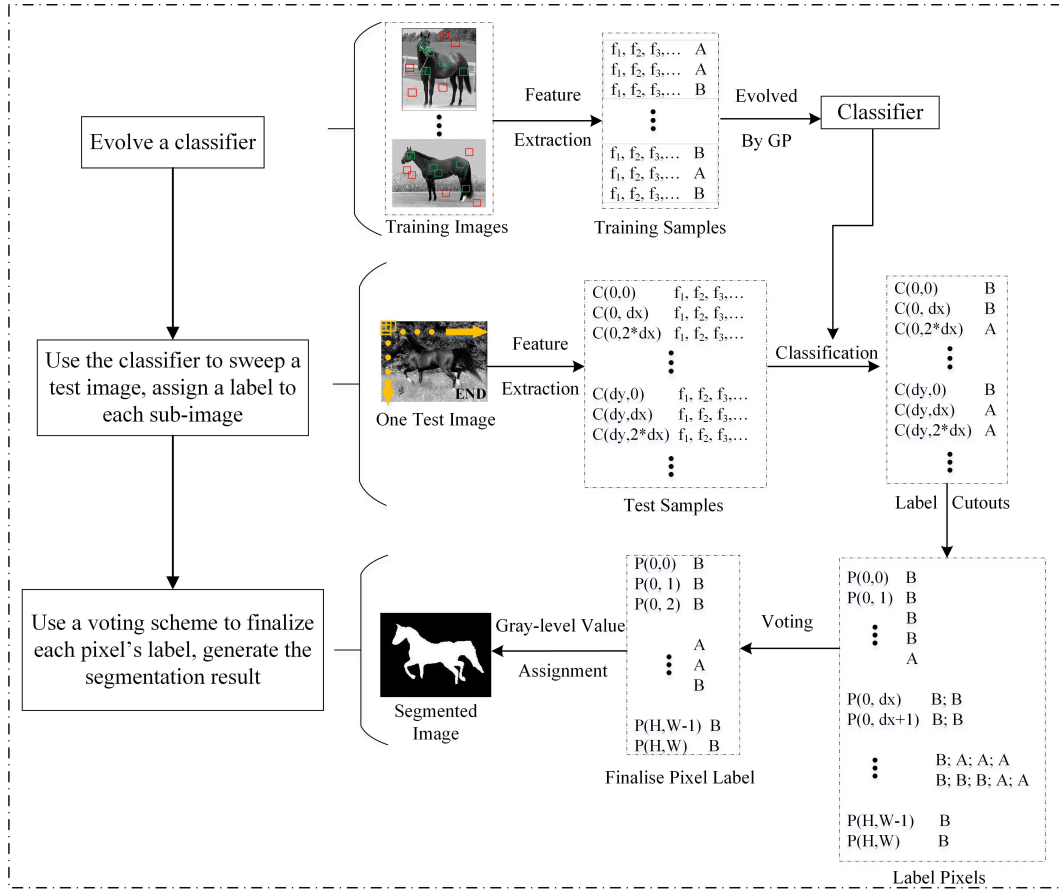


Figure 3.2: The window classification based segmentation using GP: sub-images captured by green and red windows are from class *foreground* (A) and *background* (B) respectively; windows in green, red and yellow hold the same size;  $f_1$  means an extracted feature with its index as 1, and so forth; C and P stand for cutout and pixel respectively;  $C(0,0)$  is the cutout with (0,0) as the coordinate of its top-left pixel, and so forth;  $P(0,0)$  is the pixel with (0,0) as its coordinate;  $dx$  and  $dy$  are the moving steps of the shifting window on test images in horizontal and vertical direction respectively; H and W are the height and width of a test image.

### 3.2.2 Terminal Set

Table 3.1 displays seven terminal sets (i.e. seven types of features) as GP inputs. As shown in Table 3.2, the features include raw pixel values as the brightness features; histogram statistics, GLCM (gray level co-occurrence matrices) statistics, LBP, Fourier power spectrum and Gabor features as texture features; “spatial moments + gradient statistics” as shape features. These features are selected as they are commonly-used and cover three general feature types, i.e. brightness, texture and shape.

Table 3.1: Terminal sets (GLCM refers to gray level co-occurrence matrices; LBP stands for local binary patterns).

Name	Description
Terminal Set 1	Raw Pixel Values
Terminal Set 2	Histogram Statistics
Terminal Set 3	GLCM Statistics
Terminal Set 4	LBP Features
Terminal Set 5	Fourier Power Spectrum
Terminal Set 6	Gabor Features
Terminal Set 7	Moments+Gradient Statistics

Specifically, intensities are the gray levels of pixels. The histogram statistics are first order measures [133] and six histogram statistics are used, i.e. mean, variance, skewness, kurtosis, energy and entropy. GLCM is a second-order statistical method that considers the spatial dependence of pixels [50]. Four types of statistics are derived from GLCM, i.e. contrast, correlation, energy and homogeneity. LBP descriptor transforms an image to an array of integer labels, which can represent the small-scale appearance of the image [132]. The Fourier power spectrum of an image or a region is the square of the magnitude of its Fourier transform [70]. Gabor filters [77] can extract image information in different scales and orientations, so they are one of the most powerful local appearance descriptors.

The first and second order spatial moments can describe the distribution of image regions [155]. Image gradients are directional changes in intensities or colour in an image, and are able to encode edges and local contrast [57]. Three gradient statistics, the maximum, minimum and average gradient of a region, are extracted.

Table 3.2: Description of features ( $M_1$  and  $M_2$  are the first order moment and the second order moment respectively; Min, Max and Ave mean the the maximum, minimum and average gradient respectively).

Category	Feature		Description or Parameter
Brightness	Raw Pixel Values		-
Texture	Statistical	Histogram Statistics	Mean, Variance, Skewness, Kurtosis, Energy, Entropy
		GLCM Statistics	Contrast, Correlation, Energy, Homogeneity
	Methods	LBP	P=8, R=1(P is the number of sampling points in a region with the radius R.)
		FourierPowerSpectrum	-
	Structural Methods	Gabor	$scale = 5, orientation = 8, d_1 = 4, d_2 = 4$
Shape	Moments + Gradient Statistics		$M_1 = \frac{\sum_{i=1}^n f(x_i, y_i) * d(x_i, y_i)}{n}$ $M_2 = \frac{\sum_{i=1}^n (f(x_i, y_i) * d(x_i, y_i) - M_1)^2}{n}$ Min, Max, Ave Gradient

Equations or parameters in Table 3.2 are based on the following definitions. For Gabor features,  $scale = 5$  means the number of filter scales is five;  $orientation = 8$  represents the number of filter orientations is eight;  $d_1 = 4, d_2 = 4$  means the down-sampling ratio in row and column respectively. For moments, given a square region with the size of  $n * n$ ,  $f(x_i, y_i)$  is the pixel value;  $d(x_i, y_i)$  is the distance of the pixel  $(x_i, y_i)$  to the top left corner of the region.

### 3.2.3 Function Set

Table 3.3 displays the function set, including four arithmetic and five conditional/relational operators. These operators are simple and efficient; moreover, they are sufficient to evolve both linear and non-linear classifiers.

Table 3.3: Function set of GP based figure-ground segmentation.

Function Name	Definition	Type
Add( $a_1, a_2$ )	$a_1 + a_2$	Arithmetic
Sub( $a_1, a_2$ )	$a_1 - a_2$	Arithmetic
Mul( $a_1, a_2$ )	$a_1 * a_2$	Arithmetic
Div( $a_1, a_2$ )	$\begin{cases} a_1/a_2 & \text{if } a_2 \neq 0 \\ 0 & \text{if } a_2 == 0 \end{cases}$	Arithmetic
IF( $a_1, a_2, a_3$ )	$\begin{cases} a_2 & \text{if } a_1 \text{ is true.} \\ a_3 & \text{if } a_1 \text{ is false.} \end{cases}$	Conditional
$\leq (a_1, a_2)$	$\begin{cases} true & \text{if } a_1 \leq a_2 \\ false & \text{if otherwise} \end{cases}$	Relational
$\geq (a_1, a_2)$	$\begin{cases} true & \text{if } a_1 \geq a_2 \\ false & \text{if otherwise} \end{cases}$	Relational
$= (a_1, a_2)$	$\begin{cases} true & \text{if } a_1 == a_2 \\ false & \text{if otherwise} \end{cases}$	Relational
Between( $a_1, a_2, a_3$ )	$\begin{cases} true & \text{if } a_2 \leq a_1 \leq a_3 \\ false & \text{if otherwise} \end{cases}$	Relational

### 3.2.4 Other GP Settings

Classification accuracy (Equation 3.1) is selected as the fitness function, because the segmentation problem has been transformed into a window classification based task in this chapter, and the classification accuracy is an efficient and commonly-used measure for binary classification tasks [5].

$$\text{Classification Accuracy} = \frac{\text{Number of correctly classified samples}}{\text{Number of total training samples}} * 100\%. \quad (3.1)$$

Most GP set-up parameters follow Koza's settings (e.g. the number of generation (51), the crossover rate (90%), the reproduction rate (10%) and the maximum tree depth (17)), which are commonly used in his works [83] (see Section 2.7.4 for details). In addition, considering the goal of this chap-

ter is to investigate the effectiveness of different features for GP rather than quantitative evaluation, a subset of images are selected from benchmark datasets. Moreover, larger population sizes can lead to higher computational cost and time cost. Therefore, the population size is set to 500 (Koza set it to 1024), which is considered sufficient for the given segmentation tasks.

### 3.3 Experiment Preparation

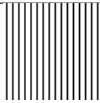
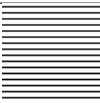
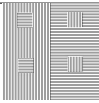
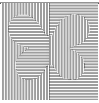
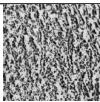
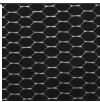
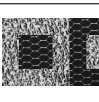





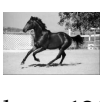

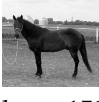










#### 3.3.1 Training and Test Images

Four image datasets are selected in this chapter, i.e. bitmap patterns [161], Brodatz texture images [67], Weizmann horse dataset [22, 24] and Pascal aeroplane dataset [53] (see Section 2.5.1 for details), as they cover a wide range of image domains and have different difficulty levels for segmentation. Since this chapter focuses on determining effective image features for GP to evolve capable segmentors rather than conducting the quantitative performance evaluation, a subset of images are randomly selected from the four datasets in this chapter, i.e. two bitmap images, one texture image, ten Weizmann images and eight Pascal images. The images are displayed in Table 3.4.

Note that images with larger sizes and/or with high variations are more complex than those with smaller sizes and/or with low variations, and grayscale images are more complex than binary ones for segmentation tasks [133]. In Table 3.4, the two bitmap images (named as “Rectangular” and “Butterfly”) are synthesised from two bitmap patterns (P14 and P24), which are relatively simple. The texture image (D24vs34) is a grayscale image and is synthesised from two Brodatz textures (D24 and D34). In addition, the Weizmann and Pascal images are complex object images and have high variations. Specifically, the Weizmann horse images (248 by 211 pixels) are varied in horse positions (e.g. standing, running and eating).

Each Weizmann image contains one object (horse), and all the horses are located in the center of images. The eight passenger air plane images from the Pascal dataset are the largest images (500 by 350 pixels). There are Pascal images that contain more than one object, and the object shapes and sizes on Pascal images vary greatly. Therefore, Pascal images are considered to be the most difficult for segmentation.

Table 3.4: Four types of images.

Database	Images				
Bitmap	 P14	 P24	 Rectangular	 Butterfly	
Brodatz	 D24	 D34	 D24vs34		
Weizmann	 horse006	 horse010	 horse027	 horse110	 horse119
	 horse121	 horse122	 horse159	 horse165	 horse317
Pascal	 2007_000033	 2007_000256	 2007_000738	 2007_001288	
	 2007_001761	 2007_002099	 2007_002266	 2007_002376	

### 3.3.2 Experiment Design

For bitmap (or texture) images, there are two different bitmap patterns (or textures) in each image, so one pattern (or texture) is chosen as class *foreground* and the other as *background* randomly. The training set has 2000 samples, equally from two patterns (or textures), with class label *back-*



*ground/foreground*. In addition, as only ten Weizmann images and eight Pascal images are used, the leave-one-out (LOO) cross validation [89] is employed on both datasets. The cross validation is on image level. Specifically, for a given set of images, one image is used as the training image and the others are used as test images. For Weizmann images, each training image provides 200 training samples, consisting of 100 foreground samples and 100 background samples. As Pascal images are larger than Weizmann images, each Pascal training image provides 500 samples, equally generated from foreground and background regions.

Window related parameters, i.e. the size of the window and its shifting steps, are set based on our initial experimental tests. Specifically, the window size is set to 4 for bitmap images and 16 for other images. As the window captures subimages, the window size must guarantee that each subimage contains sufficient information to distinguish itself from those belonging to other classes. Considering a big shifting step of the window may cause inaccurate results, yet a small step involves a high computation cost, both shifting steps ( $dx$  in the horizontal direction and  $dy$  in the vertical direction) are set to 2 for the four datasets.

Each experiment is run 30 times due to the stochastic nature of GP. In addition, three evaluation methods are employed to evaluate the segmentation results, i.e. the segmentation accuracy,  $F_1$  score and negative rate metric (NRM) (see Section 2.5.3 for details). For the segmentation accuracy and  $F_1$  scores, the higher the better; for NRM scores, the lower the better.





## 3.4 Results and Discussions

### 3.4.1 Results on Bitmap Images

As bitmap images are binary images, bitmap patterns can be distinguished by raw pixel values. Therefore, only intensities (raw pixel values) are used to evolve segmentation algorithms (segmentors) by GP for the bitmap images. In Table 3.5, result images show that regions of different patterns

are segmented accurately. Not only the linear boundaries of the “Rectangular” image but also the curve boundaries of the “Butterfly” image have been detected precisely. Based on the evaluation measures, high segmentation accuracies, high  $F_1$  scores and low NRM scores are obtained for both images. For example, the segmentation accuracy and  $F_1$  score reach 98.81% and 0.99 respectively on the “Rectangular” image, while the NRM reaches 0.12 on the “Butterfly” image.

Table 3.5: Test Results of segmentors evolved from intensities on bitmap images.

Ground Truth	Result Examples	Accuracy(%)	$F_1$	NRM
		$98.81 \pm 0.22$	$0.99 \pm 0.01$	$0.20 \pm 0.01$
		$95.96 \pm 1.27$	$0.96 \pm 0.01$	$0.12 \pm 0.00$









### 3.4.2 Results on Brodatz Texture Images

According to Table 3.6, the performance of segmentors evolved from the seven feature sets is generally good on the texture image, as most segmentation accuracies are over 90% and  $F_1$  scores are over 0.90. Specifically, intensities achieve the highest scores in the accuracy (94.26%) and  $F_1$  score (0.94), while the histogram features achieve the highest  $F_1$  score (0.94) and lowest NRM score (0.07). The results reflect that on the relatively simple texture images, pixel intensities are sufficient for GP to derive capable segmentation algorithms.

### 3.4.3 Results on Weizmann Horse Images

Example result images in Table 3.7 show that the segmentors evolved from intensities, histogram statistics and Gabor perform better than those
























































Table 3.6: Test Results of segmentors evolved from different features on a Brodatz texture image.

Feature	Result Examples	Accuracy(%)	$F_1$	NMR
Ground Truth		-	-	-
Intensity		$94.26 \pm 2.75$	$0.94 \pm 0.00$	$0.15 \pm 0.01$
Histogram Statistics		$93.98 \pm 2.30$	$0.94 \pm 0.00$	$0.07 \pm 0.00$
GLCM Statistics		$92.67 \pm 1.45$	$0.92 \pm 0.02$	$0.15 \pm 0.01$
LBP		$66.82 \pm 10.06$	$0.53 \pm 0.07$	$0.35 \pm 0.05$
Fourier Power Spectrum		$91.16 \pm 0.94$	$0.90 \pm 0.04$	$0.13 \pm 0.00$
Gabor		$90.91 \pm 0.72$	$0.90 \pm 0.01$	$0.15 \pm 0.02$
Moments + Gradient		$92.02 \pm 2.11$	$0.92 \pm 0.00$	$0.39 \pm 0.02$

from other features, as they locate horses accurately and generate clearer boundaries. The segmentors generated from LBP, Fourier power spectrum and “moments + gradient statistics” perform unsatisfactorily, since there are more false positive or negative pixels on the result images. The results based on the evaluation measures in Table 3.8 verify that histogram statistics and Gabor based segmentors obtain the best scores. They are followed by the histogram features. In contrast, the intensities that perform well on bitmap and texture images do not achieve satisfactory scores, with 3.88% lower, 0.04 lower and 0.05 higher than those of Gabor features. This reflects that for complex object images, employing only intensities is not sufficient. Instead, the inclusion of higher-level texture features (e.g. Gabor and histogram features) are necessary for GP to evolve effective segmen-

tors.

Table 3.7: Result examples on Weizmann images (GT: ground truth; Inten: intensity; Hist: histogram; FPS: Fourier power spectrum; MG: moments + gradient statistics).

Name (Prefix: horse)	006	010	027	110	119	121	122	159	165	317
GT										
Inten										
Hist										
GLCM										
LBP										
FPS										
Gabor										
M.G.										

### 3.4.4 Results on Pascal Aeroplane Images

Considering Table 3.9, the result examples based on intensities and GLCM features are visually the worst, since they contain more false positives and/or false negatives than results based on other features. In Table 3.10, Gabor and Fourier features perform best with similar segmentation scores,

Table 3.8: Test Results of segmentors evolved from different features on Weizmann images (the best result is in bold)





















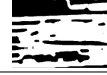




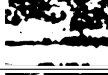
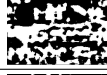















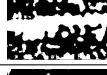















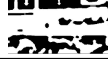





Feature	Accuracy (%)	$F_1$	NRM
Intensity	$74.41 \pm 8.37$	$0.62 \pm 0.03$	$0.47 \pm 0.01$
Histogram Statistics	$77.37 \pm 9.09$	<b><math>0.84 \pm 0.02</math></b>	$0.47 \pm 0.02$
GLCM Statistics	$76.74 \pm 3.92$	$0.68 \pm 0.01$	$0.47 \pm 0.01$
LBP	$66.19 \pm 10.95$	$0.52 \pm 0.04$	$0.48 \pm 0.03$
Fourier	$68.38 \pm 7.38$	$0.61 \pm 0.02$	$0.50 \pm 0.02$
Gabor	<b><math>78.29 \pm 5.40</math></b>	$0.66 \pm 0.02$	<b><math>0.42 \pm 0.01</math></b>
Moments + Gradient statistics	$65.04 \pm 10.39$	$0.58 \pm 0.06$	$0.50 \pm 0.04$

among which Gabor features are slightly better. They are followed by the histogram and “moments + gradient statistics” features. The remaining features, i.e. intensities, GLCM and LBP features, perform worst. For example, compared with that of Gabor features, the performance of GLCM features decreases by 8.21% and 0.13 in the segmentation accuracy and  $F_1$  respectively, while it also increases by 0.06 in the NRM. Therefore, even though the intensities perform well on relatively simple images, e.g. bitmap and texture images, they are not sufficient for GP to evolve effective segmentors for complex object images. Moreover, the shape features perform better, yet still not the best type of features. In contrast, certain texture features, e.g. Gabor and histogram features, are the best when used as inputs of GP to evolve segmentation solutions.

### 3.5 Analyses of Evolved Segmentors

To determine the influence of input features on the GP segmentation method, the performance of its evolved segmentors based on different features are compared. In addition, the well-performing evolved segmentors are also compared with four traditional segmentation techniques, i.e. thresholding

Table 3.9: Result examples on Pascal images (GT: ground truth; Inten: intensity; Hist: histogram; FPS: Fourier power spectrum; MG: moments + gradient statistics).

Name (Prefix: 2007_00)	0033	0256	0738	1288	1761	2099	2266	2376
GT								
Inten								
Hist								
GLCM								
LBP								
FPS								
Gabor								
MG								

[2] and ACM [1] from the Matlab R2014b document examples, a region growing method from Kroon [87] and a K-means method from Fonseca [56].

### 3.5.1 Comparison between Evolved Segmentors

Figure 3.3 shows the performance of the segmentors evolved from different features on three datasets. On the texture image (Figure 3.3a), most evolved segmentors achieve over 90% accuracy, over 0.9  $F_1$  score and around 0.1 NRM score, except for LBP and “moments + gradient statistics” based ones. On complex Weizmann images (Figure 3.3b), two kinds of segmen-

Table 3.10: Test Results of segmentors evolved from different features on Pascal images (the best result is in bold)

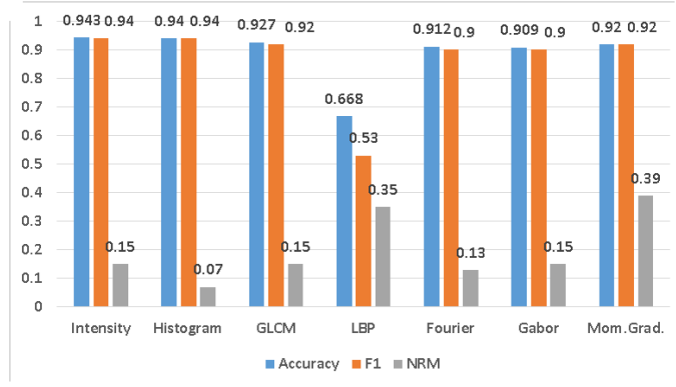
Feature	Accuracy (%)	$F_1$	NRM
Intensity	$71.39 \pm 10.63$	$0.49 \pm 0.05$	$0.50 \pm 0.05$
Histogram Statistics	$74.56 \pm 6.89$	$0.61 \pm 0.03$	$0.50 \pm 0.02$
GLCM Statistics	$67.39 \pm 9.60$	$0.49 \pm 0.04$	$0.52 \pm 0.03$
LBP	$63.75 \pm 14.07$	$0.54 \pm 0.07$	$0.50 \pm 0.05$
Fourier	$75.10 \pm 7.90$	$0.61 \pm 0.04$	<b><math>0.46 \pm 0.03</math></b>
Gabor	<b><math>75.60 \pm 8.10</math></b>	<b><math>0.62 \pm 0.04</math></b>	<b><math>0.46 \pm 0.03</math></b>
Moments + Gradient statistics	$74.53 \pm 7.83$	$0.59 \pm 0.05$	$0.48 \pm 0.03$

tors (using histogram statistics and Gabor features) are the best. In addition, on the most complex Pascal images, Figure 3.3c shows that four kinds of segmentors (histogram statistics, Fourier power spectrum, Gabor and “moments + gradient statistics” based ones) produce higher scores than others.

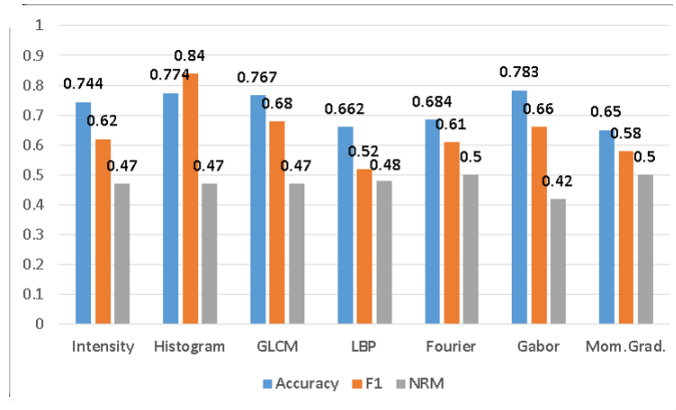
The analyses show that from simple texture images to complicated object images, certain texture features based segmentors (e.g. histogram statistics and Gabor features) produce consistently better results than others. In contrast, intensities based segmentors perform well on texture images, yet they fall far behind on the complex images, i.e. Weizmann and Pascal images. Moreover, the shape features cannot produce as good results as the histogram statistics and Gabor features generally.

### 3.5.2 Comparison with Traditional Methods

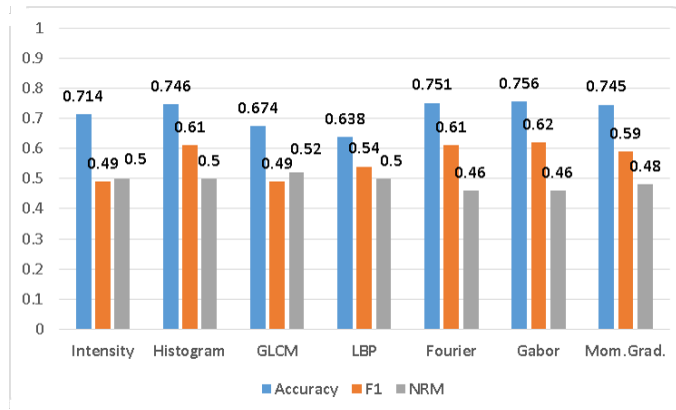
Four traditional segmentation techniques are selected for comparison, i.e. thresholding [2], active contour model (ACM) [1], a region growing method [87] and a K-means method [56]. As the segmentors evolved from the Gabor features achieve consistently good results on Weizmann and Pascal images, they are selected to compare with the four conventional methods.



(a) Results on the Brodatz images.



(b) Results on the Weizmann images.



(c) Results on the Pascal images.

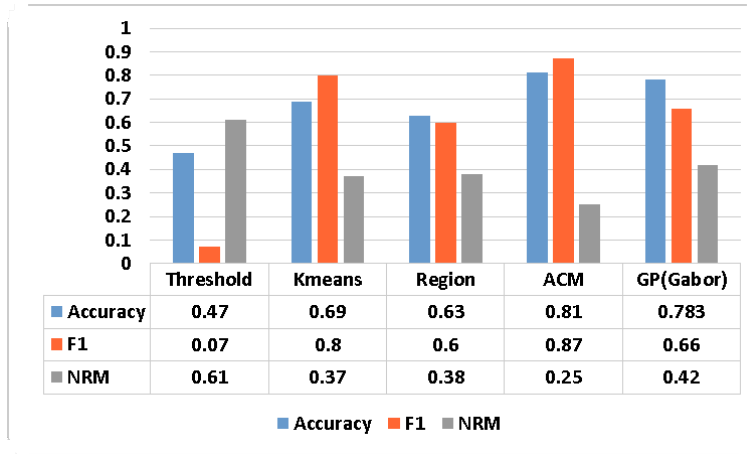
Figure 3.3: Comparison of GP evolved segmentors from different features (the higher the better for the accuracy and  $F_1$  score; the lower the better for NRM).



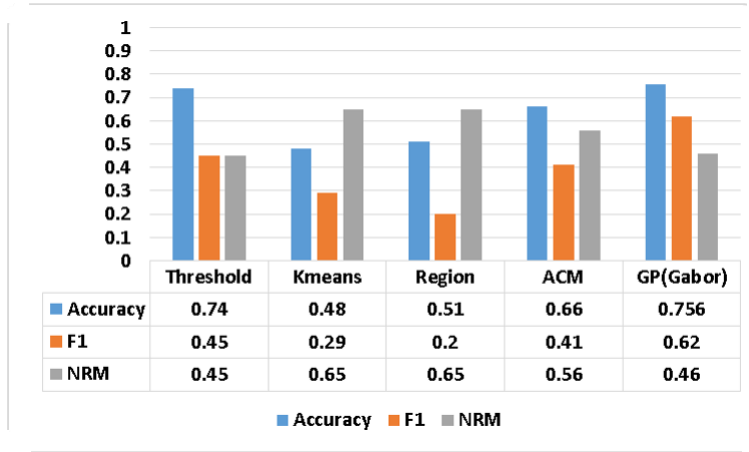
Figure 3.4 displays the performance of both the four traditional methods and the proposed GP method using Gabor features, termed as GP(Gabor). On Weizmann images (Figure 3.4a), ACM performs best with the highest accuracy, highest  $F_1$  score and lowest NRM score. It is well known that ACM relies heavily on the initial contour placement [106]. Only when initial contours are set near objects of interest, could ACM produce accurate results. GP(Gabor) and K-means clustering rank second with slightly worse results. According to Figure 3.4b on the more complex Pascal images, GP(Gabor) performs best, followed by thresholding, while K-means, region growing and ACM are much worse. Note that the thresholding method is only suitable for images with high contrast between objects and backgrounds. In contrast, the proposed GP based method can evolve segmentors matching different types of images, leading to satisfactory results consistently.

To reveal how GP affects the segmentation performance, Figure 3.5 displays the average training performance of the segmentors evolved in each generation by GP. Note that the initial segmentors (segmentors in generation one) are generated randomly, and the segmentors in later generations (generation two, three and so forth) are evolved by GP based on the initial ones. Figure 3.5 shows that the training accuracy increases with increasing generations on all the three datasets. In addition, Table 3.11 presents the average training performance of the derived segmentors at generation one and 51. Based on the Mann-Whitney U-Test [118] at the significance level 5%, the training accuracy on all the datasets rises significantly from generation one to generation 51. Meanwhile, the deviation of the accuracy drops. For example, on Brodatz dataset, the accuracy rises from 82.13% to 95.21% significantly, along with the decrease of the deviation from 2.46% to 0.97%. The analyses show that GP can improve the initial segmentors to produce ones with better segmentation performance.

Therefore, the GP based method can perform consistently well on the two complex datasets (the Weizmann and Pascal datasets), while some



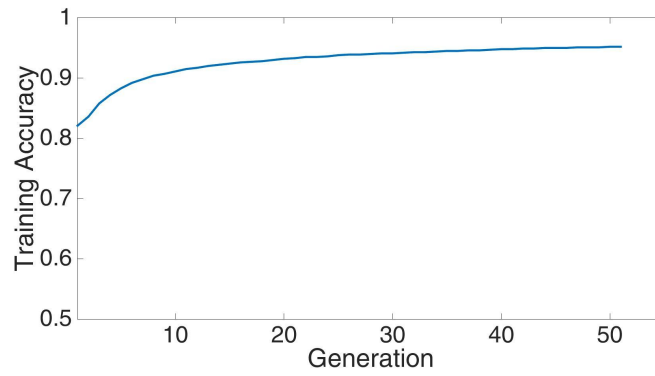
(a) Results on the Weizmann images.



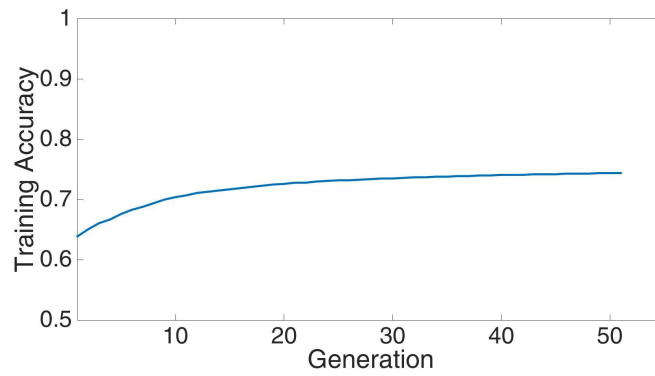
(b) Results on the Pascal images.

Figure 3.4: Comparisons of different methods on Weizmann and Pascal images (the higher the better for the accuracy and  $F_1$  score, and vice versa for NRM; GP(Gabor) means that the GP based segmentation method uses Gabor features).

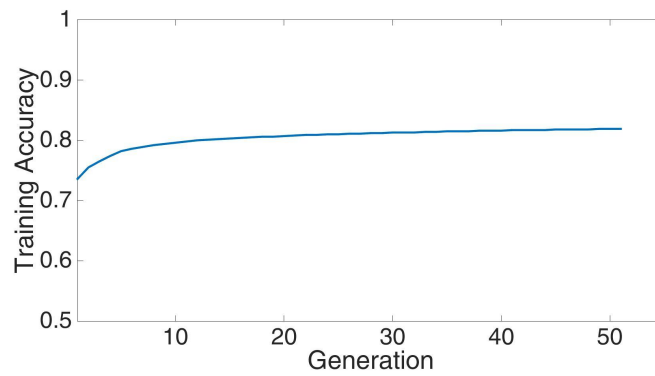
of the four traditional techniques (i.e. the thresholding and ACM methods) show unstable performance, and others (i.e. the K-means and region growing methods) are not effective to segment the real images. It can be claimed that the proposed method can outperform the conventional



(a) On Brodatz images.



(b) On Weizmann images.



(c) On Pascal dataset.

Figure 3.5: Average training performance of GP evolved segmentors from Gabor features.

Table 3.11: Average training performance of GP evolved segmentors from Gabor features (solutions in generation one are created randomly; generation 51 is the final generation).

Dataset	Generation	Training Accuracy (%)
Brodatz	1	$82.13 \pm 2.46$
	51	$95.21 \pm 0.97$
Weizmann	1	$63.89 \pm 1.29$
	51	$74.41 \pm 0.61$
Pascal	1	$73.65 \pm 1.85$
	51	$81.90 \pm 0.38$

commonly-used segmentation techniques on the given image datasets based on the Gabor features.

### 3.5.3 Example Segmentors

To understand why the evolved programs can be used as good segmentors, example programs are listed in Figure 3.6 and Figure 3.7. The solution in Figure 3.6 evolved for the texture image “D24vs34” (displayed in Table 3.3.1) is taken as an example.

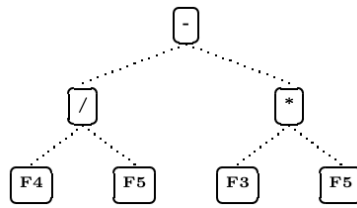


Figure 3.6: A GP tree solution evolved from histogram statistics for texture image “D24vs34” ( $F_0$  is a feature with 0 as its index, and so forth).

Figure 3.6 displays a solution derived from the histogram statistics, which consist of six input features, i.e. mean ( $F_0$ ), variance ( $F_1$ ), skewness

( $F_2$ ), kurtosis ( $F_3$ ), energy ( $F_4$ ) and entropy ( $F_5$ ). The functions include add (+), subtract (-), multiply (\*), divide (/), no bigger ( $\leq$ ), no smaller ( $\geq$ ), equal ( $=$ ), if (IF) and between (Bwt), which are described in Section 3.2.3. This tree solution can be equivalently expressed in Lisp S-expression format, shown in Equation 3.2 that can be simplified to Equation 3.3. In addition, Equation 3.4 describes the classification strategy of GP.

$$(-(/ F_4 F_5)(* F_3 F_5)) \quad (3.2)$$

$$GP\ Output = F_4/F_5 - F_3 * F_5 \quad (3.3)$$

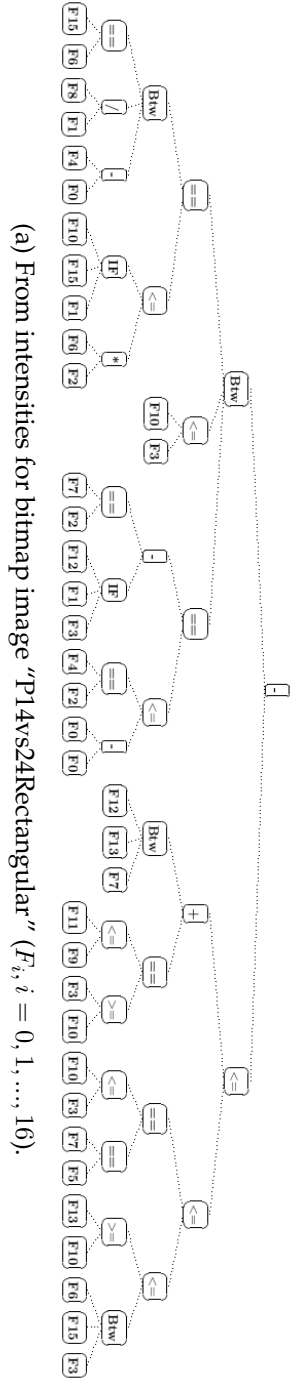
$$Class = \begin{cases} A & \text{if } GP\ Output < 0 \\ B & \text{if } GP\ Output \geq 0 \end{cases} \quad (3.4)$$

As the texture image “D24vs34” contains two patterns, “D34” and “D24”, which are set as class A and class B respectively. Equation 3.5 can be generated based on Equation 3.3 and Equation 3.4 to classify subimages in a test image. Specifically, Equation 3.5 shows that the evolved segmentor uses three features, i.e.  $F_3$  (kurtosis),  $F_4$  (energy) and  $F_5$  (entropy). Since texture “D34” should have lower energy, higher kurtosis and higher entropy than texture “D24” in reality, it is more likely for texture “D34” to obtain  $F_4 < F_3 * F_5^2$ , while texture “D24” to obtain  $F_4 \geq F_3 * F_5^2$ . Therefore, subimages with “D34” and “D24” textures can be correctly categorised as class A and B respectively.

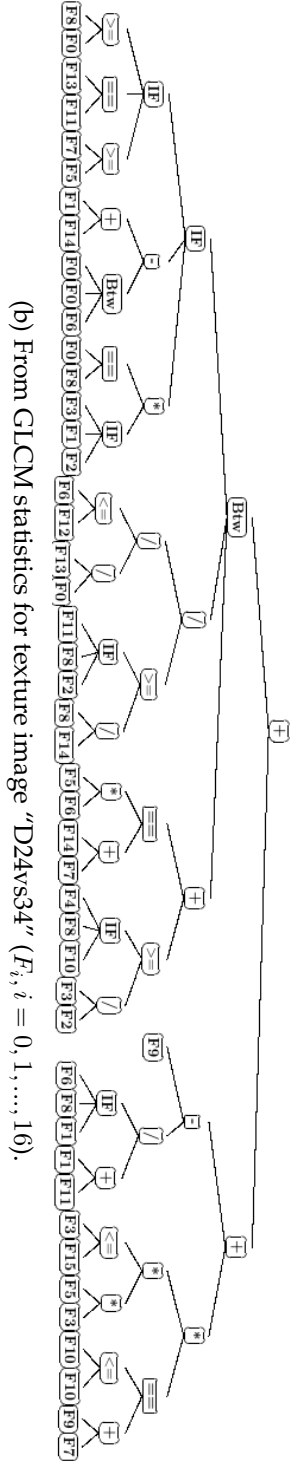
$$Class = \begin{cases} A(D34) & \text{if}(F_4 < F_3 * F_5^2) \\ B(D24) & \text{if}(F_4 \geq F_3 * F_5^2) \end{cases} \quad (3.5)$$

## 3.6 Chapter Summary

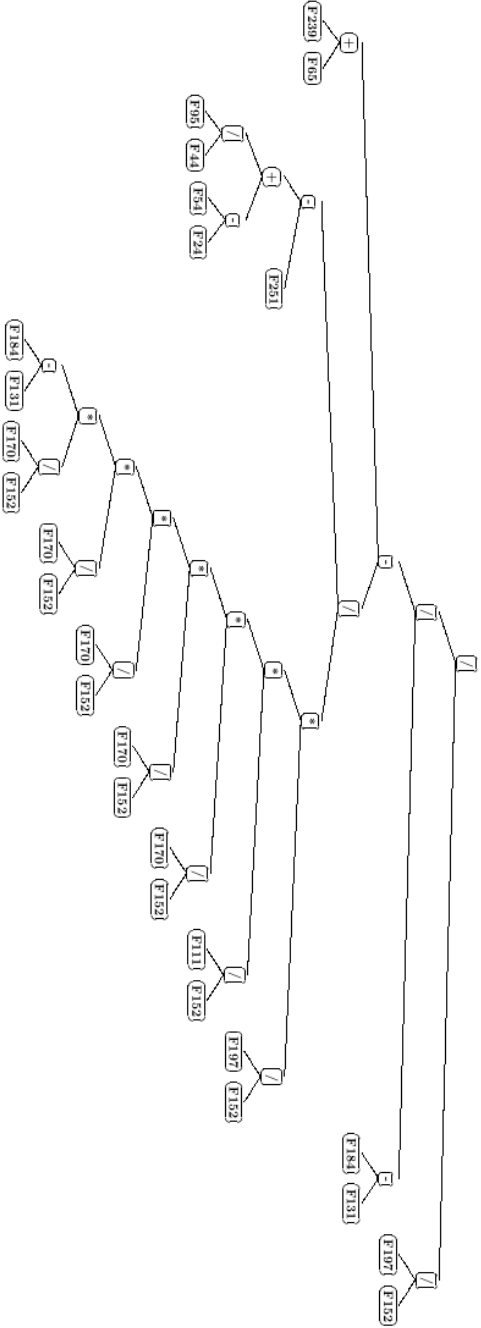
This chapter investigated a wide range of image features as inputs to GP, aiming to determine what image information is effective for GP to evolve capable segmentation algorithms. The contributions of this chapter are as follows.



(a) From intensities for bitmap image "P14vs24Rectangular" ( $F_i, i = 0, 1, \dots, 16$ ).



(b) From GLCM statistics for texture image "D24vs34" ( $F_i, i = 0, 1, \dots, 16$ ).



(c) From Fourier power spectrum for texture image "D24vs34" ( $F_i, i = 0, 1, 2, 3, 4$ ).

Figure 3.7: GP tree solutions ( $F_0$  is a feature with 0 as its index, and so forth).

Firstly, this chapter investigated seven types of image features as GP inputs (terminal sets) to evolve segmentors, covering three general feature categories (brightness, texture and shape). The effectiveness of different image features on the performance of evolved segmentors was studied. Results show that segmentors evolved from texture features (e.g. Gabor and histogram statistics) perform consistently well, better than the shape feature (i.e. “image moments + gradient statistics”) and the intensities, especially on complex object images. This shows that texture features are more effective than shape features and raw intensities for GP to derive capable segmentors for these datasets.

Secondly, this chapter tested the evolved segmentors on four types of images (e.g. bitmap, texture and two object datasets) that have increasing difficulty levels for segmentation. The segmentors evolved by the proposed GP method using texture features perform generally well on all the four datasets. This indicates that given proper inputs (e.g. texture features), the proposed methods can handle segmentation tasks on diverse image domains.

Thirdly, the best-performing segmentors derived based on Gabor features were compared with four conventional segmentation techniques, i.e. thresholding, K-means clustering, region growing and active contour model. The results show that the GP-evolved segmentors achieve consistently good performance, while traditional methods perform worse or cannot maintain good performance on the datasets. The results suggest that the proposed GP method with proper features, e.g. Gabor features, can outperform the conventional techniques.

Based on the achievements in this chapter, Gabor features will be employed as inputs to GP methods in the following chapters. Note that increasing the variety of available features worsens the problem that methods based on EC techniques, especially GP, tend to evolve complex solutions. They have large sizes, are computationally expensive and difficult to interpret. Therefore, the next chapter will consider reducing the solu-

tion complexity while retaining/improving the solution functionality for GP based segmentation methods.



## Chapter 4

# Multi-objective GP for Evolving Segmentors

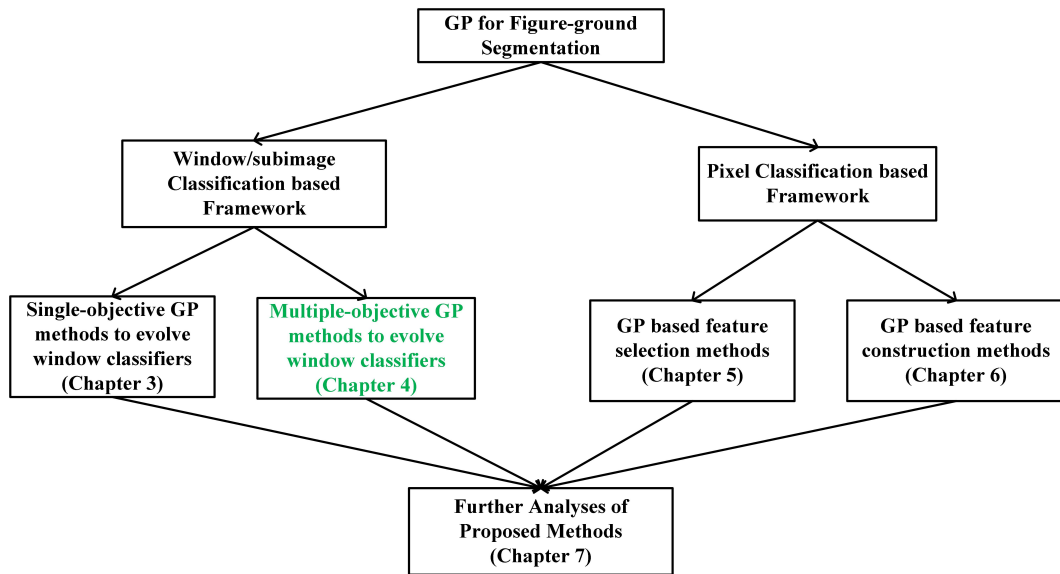


Figure 4.1: The overall structure of contributions.

GP is shown to be able to evolve capable segmentors for a wide range of image domains in Chapter 3. However, GP based methods tend to evolve large solutions, which is mainly caused by bloat. Therefore, new

GP based methods to control bloat will be investigated in this chapter.

## 4.1 Introduction

Based on the evolved segmentors in Chapter 3, their sizes can be more than 300 nodes for the segmentation task on a simple texture image, which is caused by bloat. Solutions with large sizes have several problems: being computationally expensive for evolution and later use, difficult to interpret, and likely to have low generalisation performance [135]. However, there has been no work that seeks to control bloat in GP for segmentation problems to date. Therefore, it is worth investigating GP based figure-ground segmentation methods that can evolve segmentors balancing between the segmentation performance and the complexity in terms of solution size.

### 4.1.1 Goals for Multi-objective GP Evolved Segmentors

The aim of this chapter is to control bloat for the GP based segmentation methods. Compared with the simple tree depth constraint method, parsimony pressure and multi-objective techniques are more effective for bloat control. Therefore, they are introduced in standard GP to design new segmentation methods. To investigate whether the new methods can perform well, they will be tested on a sequence of figure-ground segmentation problems with increasing difficulty. Moreover, the proposed methods will be compared with a reference method that is based on the standard GP and does not consider the solution size.

Specifically, the following objectives will be investigated:

1. whether the proposed methods can reduce the complexity in terms of solution size while maintaining or improving the segmentation performance,
2. whether the proposed GP methods can outperform the reference method,

3. whether the solutions produced can elicit useful operators and features for figure-ground segmentation.

### 4.1.2 Chapter Organisation

The remainder of this chapter is organised as follows. Section 4.2 introduces the proposed GP based methods, i.e. one parsimony pressure method and two multi-objective methods, which aim to balance the solution functionality and complexity. Experiment preparations are described in Section 4.3, including descriptions of the terminal set and the function set, the parameter settings and the experiment design. In addition, Section 4.4 discusses parameter settings based on experiments and the results of the proposed methods. The summary of this chapter is provided in Section 4.5.

## 4.2 Proposed Methods

Two multi-objective techniques (NSGA-II and SPEA2, see Section 2.3.1) are selected as basic multi-objective techniques to incorporate into GP to develop new methods with the aim to balance the solution functionality and complexity. NSGA-II and SPEA2 are selected as they are well understood and commonly-used techniques [42]. In addition, a parsimony pressure technique (weighted sum method, see Section 2.3.2) is selected for comparison, which is a popular bloat control technique in the GP community.

All the three techniques have to be applied in an efficient manner to be able to solve figure-ground segmentation tasks. As the window classification based framework used by the numerical operator based methods is more efficient than the whole image based framework used by the image processing operator based methods (see Section 2.4.1), the former framework is a popular way to apply GP to image segmentation tasks.

Therefore, the window classification based framework (see Section 3.2.1 for details) is employed in this chapter.

### 4.2.1 Weighted Sum GP Method

For weighted sum methods, the fitness function combines a performance measure with a penalty term (based on the solution size) linearly using a weight factor. Equation 4.1 shows the fitness function designed for the weighted sum GP method. As the window classification based segmentation approach transforms segmentation tasks into binary classification problems, the classification accuracy (Equation 4.2) is employed to measure the solution performance. The classification accuracy falls into the value range of  $[0, 1]$ .

The exponential function ( $\exp(-size)$ ) is employed as the base technique to measure the solution complexity. This function is selected as the contribution to the worth of the solution monotonically decreases with the increase in size and is in the range of  $[0, 1)$ , which makes it suitable to be linearly combined with the classification accuracy. Figure 4.2a depicts the curve of the exponential function, which approaches zero when the solution size grows above six. Note that the sizes (number of nodes) of the evolved solutions of standard GP based methods (see Chapter 3) can be tens or hundreds, such that this function would have no influence on complexity control. Therefore, a scaling factor  $\beta$  is introduced to the solution size contribution, thus forming the complexity measure ( $\exp(-\beta * size)$ ), where  $\beta$  is set to 0.01. Thus the complexity measure captures the differences in solution size in an appropriately large range (from tens to hundreds) effectively. The curve of the complexity measure is shown in Figure 4.2b. As  $\beta$  enables a large size range to be covered, the complexity measure is effective for different problem domains without tuning.

$$fitness = \alpha \times classification\ accuracy + (1 - \alpha) \times complexity\ measure. \quad (4.1)$$

$$\text{classification accuracy} = \frac{\text{Number of correctly classified samples}}{\text{Number of total training samples}}. \quad (4.2)$$

$$\text{complexity measure} = \exp(-\beta \times \text{size}). \quad (4.3)$$

where  $\alpha$  is a weight factor that reflects how much importance the fitness evaluation addresses the two terms, i.e. the solution performance (measured by the classification accuracy) and the solution complexity. A higher  $\alpha$  means more importance is attached to the solution performance, while less importance is attached to the solution complexity. As the segmentation performance is preferred over the solution complexity,  $\alpha$  is set to 0.95 based on initial experiments. In addition, *size* represents the solution size (the number of the solution tree nodes).

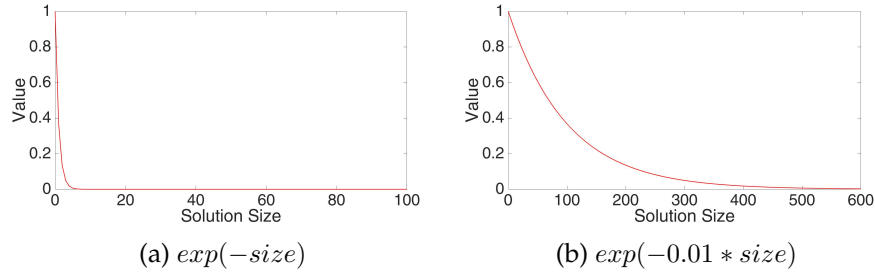


Figure 4.2: The complexity measure (*size* means solution size (the number of terminal and function nodes); *exp* stands for the exponential function; the higher the output of the complexity measure the better, which relates to lower solution sizes).

#### 4.2.2 Multi-objective GP Methods

NSGA-II and SPEA2 are selected as the basic multi-objective techniques to be incorporated with GP in order to develop the two new multi-objective

GP methods. Both methods have two objectives, one of which is the classification accuracy (Equation 4.2) and the other is the complexity measure (Equation 4.3), which are the same as the two terms in the fitness function of the proposed weighted sum method.

Between the two objectives, the classification accuracy is preferred, but different from the weighted sum method, a preference weighting is not assumed beforehand. This kind of preference information can be utilised to focus on searching an area on the Pareto front specified by a user's interest, thus helping produce a more fine-grained approximation of the most relevant areas on the Pareto front and/or reduce the computation time [26]. To incorporate the preference information, a new dominance mechanism (shown in Algorithm 1) is designed to guide the search to solutions with outstanding performance in the objective of the classification accuracy. Specifically, if solution A has a higher classification accuracy than solution B by a value defined by the *DominanceThreshold* parameter, A dominates B without comparing their scores in the other objective – the complexity measure.

The adapted NSGA-II and SPEA2 that use this new dominance mechanism are called INSGA-II (improved NSGA-II) and ISPEA2 (improved ISPEA) respectively. By employing the INSGA-II and ISPEA2 in GP respectively, two new multi-objective approaches are developed, which are INSGP (improved non-dominated sorting GP, shown in Algorithm 2) and ISPGP (improved strength Pareto GP, shown in Algorithm 3) respectively.

## 4.3 Experiment Preparation

### 4.3.1 Terminal Set

Gabor filters are widely-used to extract texture image features (see Section 2.5.2 for details), which are employed in this chapter. Twenty Gabor filters with popular scales and orientations (five scales: 4,  $4\sqrt{2}$ , 8,  $8\sqrt{2}$  and 16,

```

input : Solution A and B;
         DominanceThreshold.
output: Dominance result (i.e. A dominates B; B dominates A; A and
         B do not dominate each other).

1 Execute each solution and evaluate it based on two objectives (the
  accuracy (Equation 4.2) and the complexity measure (Equation
  4.3));
2 if A.accuracy – B.accuracy > DominanceThreshold (or B.accuracy
  – A.accuracy > DominanceThreshold) then
3   | A dominates B (or B dominates A)
4 else
5   | Apply the original dominance mechanism:
6   | if A is not inferior to B in all objectives and A is better than B in at
     | least one objective then
7   |   | A dominates B
8   | else if B is not inferior to A in all objectives and B is better than A in
     | at least one objective then
9   |   | B dominates A
10  | else
11  |   | A and B do not dominate each other
12  | end
13 end
14 Return the dominance result.

```

**Algorithm 1:** Pseudo-code of the new Pareto dominance mechanism.

and four orientations:  $0, \frac{\pi}{4}, \frac{2\pi}{4}$  and  $\frac{3\pi}{4}$ ) are selected.

Figure 4.3 shows the process of extracting Gabor features from one subimage (class object), which form one training/test sample. Specifically, for a given image, it is filtered by the 20 Gabor filters to produce 20 filtered images. The magnitudes of the filtered images are used as the Gabor

```

input : G: the maximum number of generations;
        N: the population size;
        The terminal set, function set and GP setting parameters (described in
        Section 4.3).
output: Pareto front.

1  Create an initial population at iteration zero ( $P_0$ ) of GP trees using the Ramped
   half-and-half method;
2   $g \leftarrow 0$ 
3  //  $Q$  is the child population of  $P$ ;  $C$  means the combined
   population of  $P$  and  $Q$ ;
4  while  $g < G - 1$  and the ideal solution is not found (the solution with accuracy == 1
   and solution size == 1) do
5      begin
6          // different procedures for generation 0 and
           generations onward //  $k$  records the index of the child
           generation to be created
7          if  $g == 0$  then
8              Sort  $P_0$  by non-dominated sorting based on the new Pareto dominance
               mechanism;
9               $k \leftarrow g$ ;
10         else
11              $C_g = P_g \cup Q_g$ ;
12             Sort  $C_g$  by non-dominated sorting based on the new Pareto dominance
               mechanism;
13              $P_{g+1} = C_g[0 : N - 1]$ ; // choose the first  $N$  elements;
14              $k \leftarrow g + 1$ ;
15         end
16         Create a child population  $Q_k$  from  $P_k$ :
17         begin
18             Set  $Q_k$  empty;
19             Assign each solution of  $P_k$  a fitness value equal to its non-dominance
               level;
20             for  $i = 0$  to  $N-1$  do
21                 Select parent(s) by tournament selection from  $P_k$ ;
22                 Generate children by mutation operation or crossover operation;
23                 Add the children to  $Q_k$ ;
24             end
25         end
26          $g \leftarrow g + 1$ ;
27 end
28 Return Pareto front (non-dominated solutions in the latest child population  $Q_g$ ).

```

**Algorithm 2:** Pseudo-code of INSGP (GP procedure is in red and NSGA-II procedure is in blue; detailed descriptions can be found in paper [44] for NSGA-II and paper [135] for GP).



```

input : G: the maximum number of generations;
        N: the population size;
         $\bar{N}$ : the archive size;
        The terminal set, function set and GP setting parameters (described in
        Section 4.3).
output: Pareto front.

1 Create an initial population at iteration zero ( $P_0$ ) of GP trees using the Ramped
  half-and-half method;
2 Create an empty archive ( $\bar{P}_0$ );
3  $g \leftarrow 1$ 
4 while  $g < G - 1$  and the ideal solution is not found (the solution with accuracy == 1
  and solution size == 1) do
5   Fitness assignment: sort  $P_g$  by non-dominated sorting based on the new
    Pareto dominance mechanism; assign each solution a fitness value equal to
    its non-dominance level;
6   Copy all non-dominated solutions in  $P_g$  and  $\bar{P}_g$  to  $P_{g+1}^-$ :
7   begin
8     if  $|P_{g+1}^-| > \bar{N}$  then
9       | Reduce  $P_{g+1}^-$  to  $\bar{N}$  by the truncation operator;
10    end
11    if  $|P_{g+1}^-| < \bar{N}$  then
12      | Fill  $P_{g+1}^-$  to  $\bar{N}$  with dominated solutions in  $P_g$  and  $\bar{P}_g$ ;
13    end
14  end
15  Create  $P_{g+1}$  from  $P_{g+1}^-$ :
16  begin
17    Set  $P_{g+1}$  empty;
18    for  $i = 0$  to  $N-1$  do
19      | Select one/two parent(s) by tournament selection from  $P_{g+1}^-$ ;
20      | Generate one child/two children by mutation operation or crossover
        operation;
21      | Add the child/children to  $P_{g+1}$ ;
22    end
23  end
24   $g \leftarrow g + 1$ ;
25 end
26 Return Pareto front (non-dominated solutions in the latest population  $P_g$ ).

```

**Algorithm 3:** Pseudo-code of ISPGP (GP procedure is in red and SPEA2 procedure is in blue; detailed descriptions can be found in paper [183] for SPEA2 and paper [135] for GP).

features. For a subimage ( $16 \times 16$  pixels) captured on the original image, a magnitude array ( $16 \times 16$  values) can be obtained on the identified location from each filtered image. Considering the high dimension of each magnitude array that leads to slow performance, down sampling is conducted on each magnitude array from  $16 \times 16$  to  $4 \times 4$  values, which are concatenated to a 16 dimension feature vector. By concatenating all the 20 feature vectors (from the 20 filtered images), a feature vector with 320 features ( $16 \times 20$ ) is obtained along with its class label (class *object* or *background*), which forms one Gabor feature sample. Therefore, the terminal set consists of 320 Gabor features.

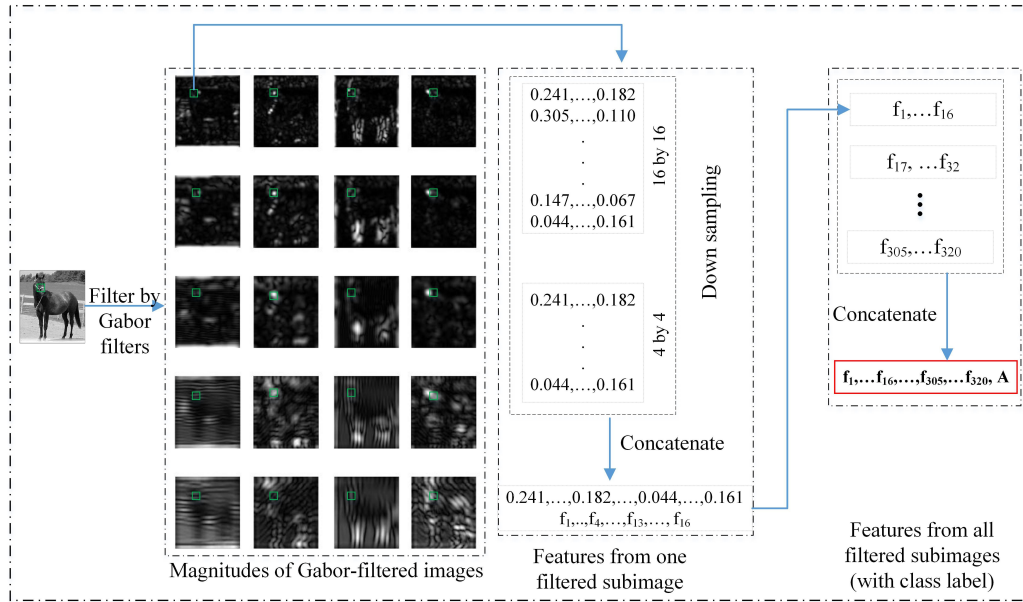


Figure 4.3: Gabor features extracted from one subimage ( $f_1$  means a feature value with 1 as its index, and so forth; A/B represents class *object/background* respectively).

### 4.3.2 Function Set

Table 4.1 is the function set, including four arithmetic operators ( $+$ ,  $-$ ,  $*$ ,  $/$ ), one conditional operator ( $IF$ ) and four relational operators ( $\leq$ ,  $\geq$ ,  $=$ ,  $\neq$ ).

, *Between*). These operators can be combined to form both linear and non-linear transformations, so they are effective to evolve binary classifiers.

Table 4.1: Function set ( $x, y, z$  stand for the variables required by the functions).

Function Name	Definition	Function Name	Definition
$+(x, y)$	$x + y$	$-(x, y)$	$x - y$
$*(x, y)$	$x * y$	$/(x, y)$	$\begin{cases} x/y & \text{if } y \neq 0 \\ 0 & \text{if } y = 0 \end{cases}$
$\text{IF}(x, y, z)$	$\begin{cases} y & \text{if } x \text{ is true} \\ z & \text{if } x \text{ is false} \end{cases}$	$\leq (x, y)$	$\begin{cases} 1 & \text{if } x \leq y \\ 0 & \text{if otherwise} \end{cases}$
$\geq (x, y)$	$\begin{cases} 1 & \text{if } x \geq y \\ 0 & \text{if otherwise} \end{cases}$	$== (x, y)$	$\begin{cases} 1 & \text{if } x == y \\ 0 & \text{if otherwise} \end{cases}$
$\text{Between}(x, y, z)$	$\begin{cases} 1 & \text{if } y \leq x \leq z \\ 0 & \text{if otherwise} \end{cases}$		

### 4.3.3 Parameters

#### GP Parameters

The population size is set to 512, and crossover and mutation are used as reproduction operators, where their rates are set to 90% and 10% respectively. Other GP parameters (e.g. the number of generations (51), the minimum (1) and maximum (17) tree depths) follow the settings used by Koza [83] (see Section 2.5.4 for details).

#### Other Parameters

The window related parameters (i.e. the size of the window and the shifting steps) in the window classification based segmentation framework are crucial in capturing sub-images. Specifically, the window size must guarantee that the captured subimages contain sufficient information to distinguish itself from those belonging to other classes. In addition, a big shift-

ing step of the window may cause inaccurate results, yet a small step involves a high computation cost. Based on initial experiments, the window size is set to 16 by 16 pixels, and both shifting steps ( $dx$  in the horizontal direction and  $dy$  in the vertical direction) are set to two.

#### 4.3.4 Experiment Design

A reference method with the same GP settings but without considering the solution size is used for comparison. In this chapter, the aggregated Pareto fronts over 30 runs are plotted for INSGP and ISPGP, while single best solutions from 30 runs (one best solution from each run) are plotted for the reference and weighted sum methods. Each experiment has 30 runs for statistical analyses due to the stochastic nature of GP.

For texture image segmentation, the training set consists of 1000 samples, including 500 samples extracted from each texture pattern (D23 or D34). The evolved segmentor is tested on one texture image, named D23vs34 (Section 2.5.1). On the Weizmann dataset, there are 328 horse images in total. Two thirds of all images (218 images) are used as training images, and the remainder (110 images) are used as test images. From each training image, 20 samples are extracted (10 from the foreground; 10 from backgrounds), so there are totally 4360 training samples. On the Pascal dataset, there are 178 aeroplane images. As suggested by Pascal VOC2012 [53], 88 images are for training and 90 images for test. There are 50 samples (25 from the foreground; 25 from backgrounds) extracted from each training image, as Pascal images are larger than the Weizmann images. In total, there are 4400 training samples.

Table 4.2 shows the number of foreground/background samples in the test sets, along with their rates. It can be seen that the Brodatz test set is balanced with an equal number of foreground and background samples. In contrast, the Weizmann and Pascal test sets are unbalanced with more background samples than foreground samples.

Table 4.2: Information on the test sets (the number of foreground/background samples are counted across the whole test set; Rate is calculated by dividing the number of foreground samples by that of background samples).

Dataset	Number of Foreground Samples	Number of Background Samples	Rate
Brodatz Test Set	5184	5184	1
Weizmann Test Set	321912	946797	0.34
Pascal Test Set	467339	3115593	0.15

#### 4.3.5 Performance Measures

Two measures are selected to assess the proposed methods, i.e.  $F_1$  score (see Section 2.5.3) [148] and the hypervolume indicator [182]. As in some test images the objects only occupy a small portion of the whole image,  $F_1$  score is employed. Compared with alternative measures, e.g. the segmentation accuracy,  $F_1$  score is relatively less biased for such cases with unbalanced classes. It reaches its worst at 0 and best at 1. In addition, the hypervolume indicator is popular for the performance assessment of the multi-objective optimisation algorithms [182]. It represents the relative size of the objective space correctly classified by the Pareto front solutions. The hypervolume measure is selected to evaluate INSGP and ISPGP. It reaches its worst at 0 and best at 1.

## 4.4 Results

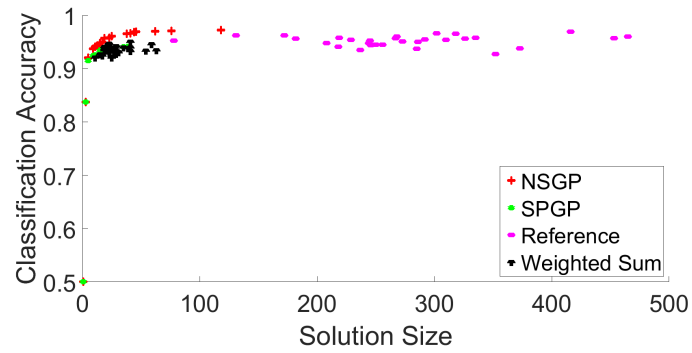
In this section, the performance of the proposed methods (the weighted sum method, INSGP and ISPGP) is compared with each other and also with the reference method. Moreover, the evolved solution trees are studied to determine how they function.

#### 4.4.1 Determination of Parameter *DominanceThreshold* in INSGP and ISPGP

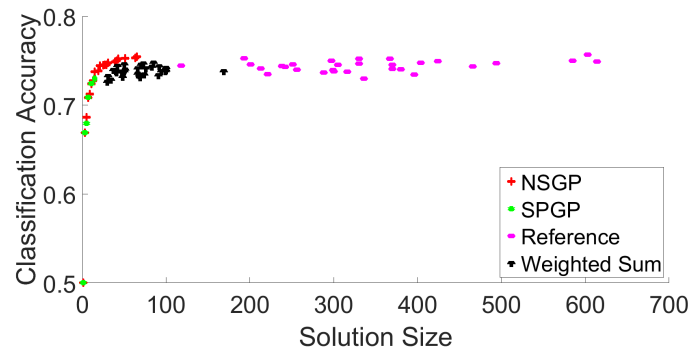
INSGP and ISPGP are created by introducing a new Pareto dominance mechanism in NSGA-II and SPEA2. This new mechanism is designed to assist in discovering solutions with extreme performance in the classification accuracy yet relatively large sizes and to enable them to survive in the evolutionary process. An important parameter, *DominanceThreshold*, is required by the new dominance mechanism (see Algorithm 1) to set the degree of dominance acceptable for increased accuracy.

The training performance of standard NSGA-II and SPEA2 based GP methods (termed as NSGP and SPGP) is shown in Figure 4.4 for comparison. It can be seen that solutions evolved by NSGP and SPGP have much smaller sizes than those of the reference method. For NSGP, its solutions can dominate those produced by both the reference and weighted sum methods in the majority of cases across the three datasets. However, this is not the case for SPGP. This figure shows that SPGP has a small search area, which does not include the solutions with relatively large sizes and high accuracies. For example, on the Weizmann dataset (Figure 4.4b), the biggest size of the non-dominated solutions is 19. For difficult segmentation tasks on the Weizmann dataset, solutions with small sizes may be too simple to produce accurate results. That is why solutions evolved by SPGP can rarely compete with those from the reference and weighted sum methods in terms of the classification accuracy.

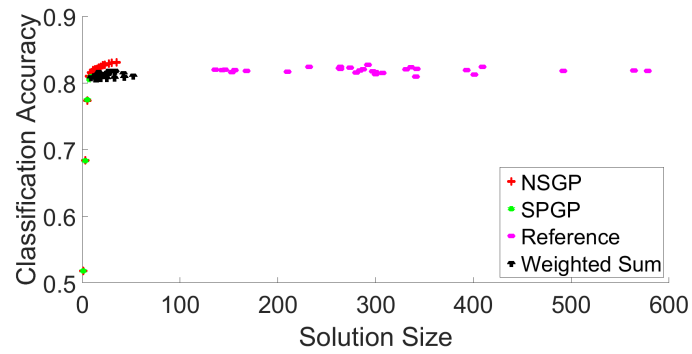
To tackle this problem, INSGP and ISPGP are created by introducing the new Pareto dominance mechanism (Algorithm 1). Different dominance thresholds using ISPGP are tested on the Pascal dataset, i.e. 0.1%, 0.5%, 1%, 3%, 5%, 7%, 10%, 20%. The smaller the threshold, the more the search process is biased towards the objective of classification accuracy. The bigger the threshold, the more equally the search process treats the two objectives. As shown in Figure 4.5, if the threshold is too small



(a) On Brodatz dataset



(b) On Weizmann dataset



(c) On Pascal dataset

Figure 4.4: Training performance of NSGP and SPGP (different scales are used on the vertical axis of the figures to show the variation on the Pareto fronts).

(e.g. 0.1%, 0.5%, 1% and 3%), there are no solutions with smaller sizes produced. If the threshold is too big (e.g. 10% and 20%), the search area almost corresponds to that of SPGP. Algorithms with the medium thresholds (5% and 7%), produce the most satisfactory training performance. They cover the search area of SPGP and extend to the area where solutions' sizes are relatively large, which suits the aim. Eventually, the threshold 5% is selected for the following experiments of INSGP and ISPGP.

#### 4.4.2 Proposed Methods vs Reference Method

Figure 4.6 and Figure 4.7 present the performance on the training sets and the test sets respectively. The non-dominated solutions of aggregated Pareto fronts over the 30 runs are plotted for INSGP and ISPGP, whereas single best solutions of 30 runs are plotted for the reference and weighted sum methods. In Table 4.3, the results are the average performance of the non-dominated solutions for INSGP and ISPGP, while results are the average performance of 30 single best solutions for the weighted sum and the reference methods.

Figure 4.6 presents the performance on each training set. As can be seen, the three proposed methods (the weighted sum, INSGP and ISPGP that consider the solution complexity), can generally evolve programs with smaller sizes than those produced by the reference method that does not address the complexity. Moreover, INSGP and ISPGP solutions can dominate those of the reference and weighted sum methods in the majority of cases. This means that with the same size, INSGP and ISPGP solutions have higher accuracies; or with the same accuracies, INSGP and ISPGP solutions have smaller sizes.

Figure 4.7 shows the performance on the test sets. For INSGP and ISPGP, each of the Pareto front solutions produced in the 30 runs is tested on the test set, and the average  $F_1$  value of each solution is calculated. Then the non-dominated solutions on the test sets are found and shown in Figure 4.7. Similar to the training performance, INSGP and ISPGP



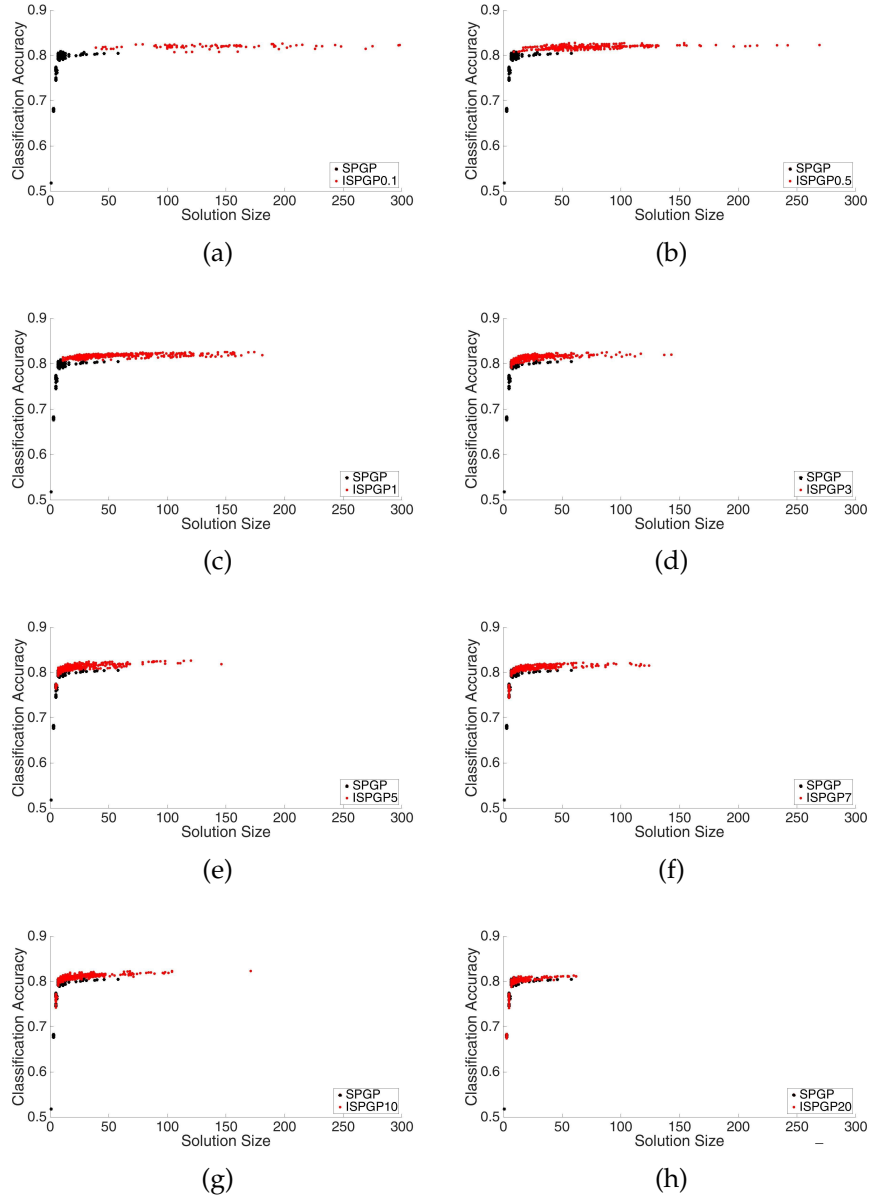
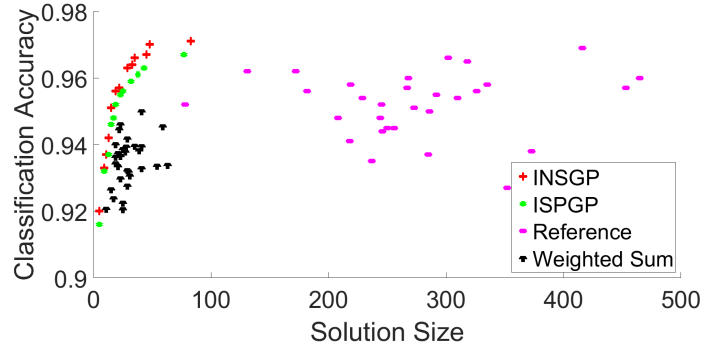
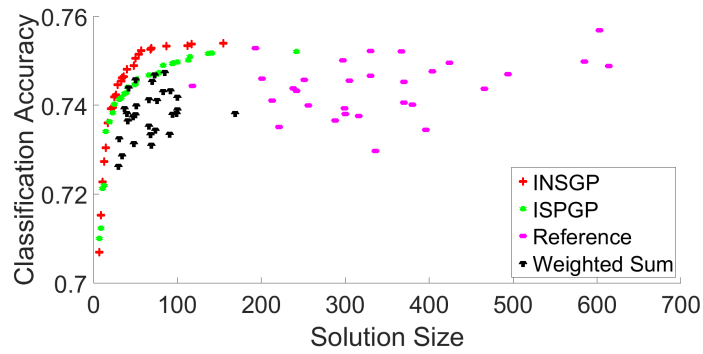


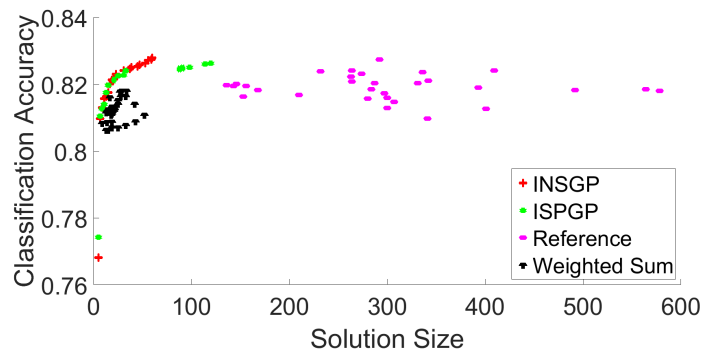
Figure 4.5: ISPGP training performance on Pascal dataset with different dominance thresholds (ISPGP0.1 means the threshold in ISPGP is 0.1% and so forth).



(a) On Brodatz dataset



(b) On Weizmann dataset



(c) On Pascal dataset

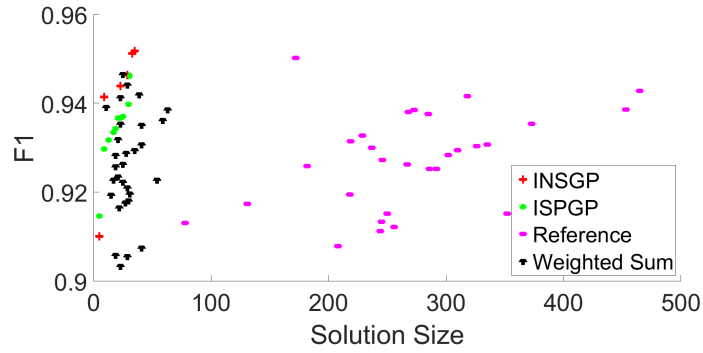
Figure 4.6: Training performance of proposed methods and the reference method.

solutions generally can dominate those of the other two methods on all three datasets. On the simple texture image (Figure 4.7a), INSGP solutions can dominate others, including the ISPGP solutions. On the two difficult datasets, the two multi-objective methods produce similar non-dominated sets in the objective space.

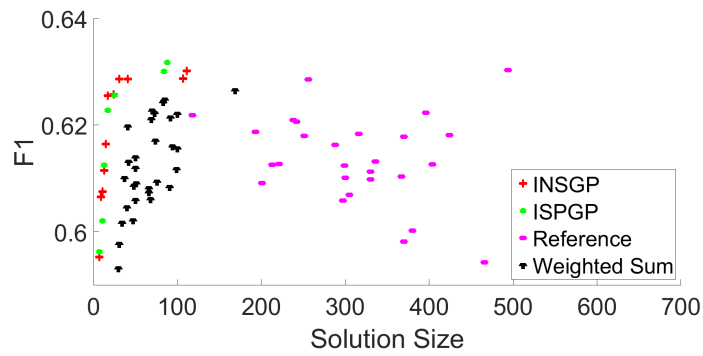
Table 4.3 provides the statistical results in terms of the training accuracy, the  $F_1$  score on the test set, the solution size, the training time and test time. For INSGP and ISPGP, the results are the average of the aggregated Pareto front solutions of 30 runs; while for the reference and weighted sum methods, the results are the average of best solutions of 30 runs. Based on the Mann-Whitney U-Test [118] at the significance level 5%, the three bloat control methods are compared with those of the reference method.

In regard to the training performance, INSGP and ISPGP show no significant difference from the reference method generally. In the test performance ( $F_1$  score), the three bloat control methods achieve generally similar performance to the reference method, with better results on Brodatz dataset for INSGP and on Pascal dataset for ISPGP and INSGP. This suggests that the three bloat control methods (weighted sum, INSGP and ISPGP) are able to produce capable solutions as the reference method. More importantly, compared with the reference method, which does not address the solution complexity, the three bloat control methods evolve solutions with much smaller sizes (around 10 to 20 times smaller). For example, on the Pascal dataset, the average size is 298 for the reference method, while 23, 13 and 14 for the three proposed methods respectively. Among the proposed methods, INSGP and ISPGP can further reduce solution sizes over the weighted sum method, which is especially obvious on the Weizmann dataset. With the simplification of evolved solutions, the training time and the test time decreased for the complexity control methods.

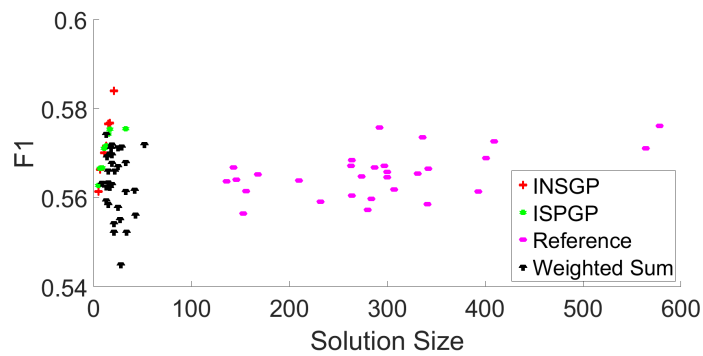
The analyses in this part show that the complexity control methods (INSGP, ISPGP and the weighted sum method) can evolve solutions much



(a) On Brodatz dataset



(b) On Weizmann dataset



(c) On Pascal dataset

Figure 4.7: Test performance of proposed methods and the reference method.

Table 4.3: Results (B., W. and P. refer to Brodatz, Weizmann and Pascal datasets; W.S. refers to the weighted sum GP method; s represents seconds).

	Method	Training Accuracy(%)	Test $F_1$	Size	Train Time Per Run(s)	Test Time Per Image(s)
B.	Reference	95.21 $\pm$ 0.98	0.93 $\pm$ 0.011	275 $\pm$ 90	150.239 $\pm$ 14.733	0.042 $\pm$ 0.012
	W.S.	93.48 $\pm$ 0.75	0.93 $\pm$ 0.011	29 $\pm$ 9	28.851 $\pm$ 6.120	0.010 $\pm$ 0.002
	INSGP	94.97 $\pm$ 1.71	0.94 $\pm$ 0.016	22 $\pm$ 8	26.779 $\pm$ 6.389	0.008 $\pm$ 0.000
	ISPGP	94.84 $\pm$ 1.45	0.93 $\pm$ 0.008	19 $\pm$ 5	24.582 $\pm$ 5.210	0.007 $\pm$ 0.001
W.	Reference	74.41 $\pm$ 0.62	0.61 $\pm$ 0.008	344 $\pm$ 98	3163.465 $\pm$ 100.002	0.055 $\pm$ 0.009
	W.S.	73.82 $\pm$ 0.53	0.61 $\pm$ 0.009	74 $\pm$ 25	1776.693 $\pm$ 88.252	0.025 $\pm$ 0.006
	INSGP	74.22 $\pm$ 1.27	0.62 $\pm$ 0.012	35 $\pm$ 12	1031.147 $\pm$ 58.793	0.013 $\pm$ 0.003
	ISPGP	74.13 $\pm$ 1.16	0.62 $\pm$ 0.014	35 $\pm$ 11	588.822 $\pm$ 21.773	0.010 $\pm$ 0.002
P.	Reference	81.81 $\pm$ 0.37	0.57 $\pm$ 0.005	298 $\pm$ 92	767.406 $\pm$ 27.338	0.110 $\pm$ 0.010
	W.S.	81.04 $\pm$ 0.30	0.56 $\pm$ 0.007	23 $\pm$ 6	249.424 $\pm$ 17.956	0.026 $\pm$ 0.005
	INSGP	81.92 $\pm$ 1.31	0.57 $\pm$ 0.007	13 $\pm$ 4	235.692 $\pm$ 19.254	0.029 $\pm$ 0.008
	ISPGP	81.50 $\pm$ 1.67	0.57 $\pm$ 0.005	14 $\pm$ 4	148.162 $\pm$ 13.064	0.023 $\pm$ 0.009

smaller than those of the reference method, leading to the training and test time being reduced significantly. Moreover, compared with the weighted sum method, the multi-objective methods (INSGP and ISPGP) can further reduce the solution size. Meanwhile, their performance on the test set is similar or even better than the reference method. Therefore, this reflects that multi-objective methods guided by the solution functionality and complexity measures can effectively explore the Pareto front to obtain a set of trade-off solutions between the two objectives.

To further show that the proposed methods with modified Pareto dominance outperform the methods with standard Pareto dominance, Figure 4.8 shows the test performance of ISPGP and SPGP on the Weizmann dataset. The non-dominated solutions of the aggregated Pareto fronts of 30 runs are depicted for ISPGP and SPGP. It shows that ISPGP solutions can generally dominate the SPGP solutions. Moreover, ISPGP provides more solutions with higher  $F_1$  scores yet larger solution sizes than SPGP, which is because the modified Pareto dominance leans towards solutions with outstanding segmentation performance.

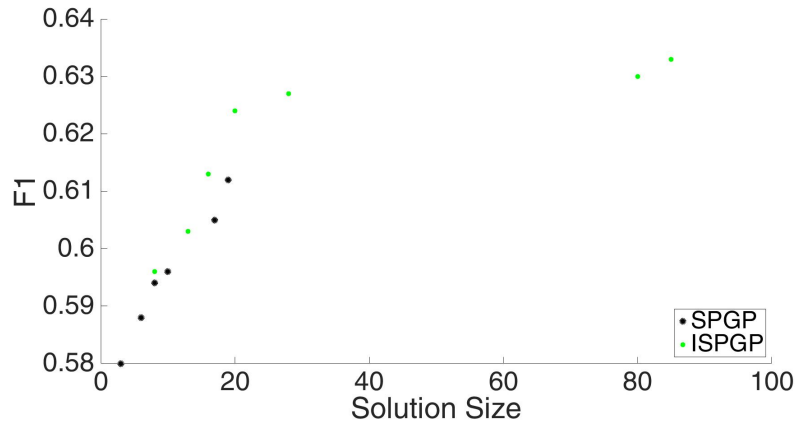


Figure 4.8: The comparison of ISPGP and SPGP on Weizmann dataset.

#### 4.4.3 INSGP vs ISPGP

Figure 4.9 presents selected image results. For Brodatz images, the results are visually accurate with objects correctly detected. For difficult images on the Weizmann and Pascal datasets, the results are relatively less accurate with some body parts of the objects incorrectly detected as class background and/or some background parts incorrectly detected as class object. This may be because the Gabor feature is not sufficiently precise to exactly capture object edges, and other types of features, e.g. colour, may be necessary to distinguish the objects from backgrounds. However, given the high variations in the test images, the results are still promising. From the result examples, no obvious difference can be seen in the performance of INSGP and ISPGP.

Table 4.4 compares the INSGP and ISPGP in terms of the hypervolume and the average solution size, which are based on the aggregated Pareto front solutions of 30 runs. For the hypervolume score, INSGP is a little higher on the training set (0.001 higher on Brodatz; 0.038 higher on Weizmann; 0.039 higher on Pascal) than ISPGP, while both methods produce generally similar scores on the test set (the same on Brodatz; 0.032 lower on Weizmann; 0.039 higher on Pascal). This reflects that solutions evolved





























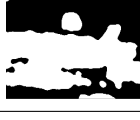




		Images				
B.						
		G.T.	INSGP	ISPGP		
W.	G.T.					
	Name	horse229	horse298	horse317	horse318	horse320
	INSGP					
	ISPGP					
P.	GT					
	Name	2007_001884	2007_002376	2007_002619	2010_003915	2010_005888
	INSGP					
	ISPGP					

Figure 4.9: Result examples (B., W. and P. refer to Brodatz, Weizmann and Pascal datasets;).

by INSGP and ISPGP can achieve similar segmentation performance. In addition, both methods evolve solutions with similar sizes, which reflects that they can reduce the solution complexity at a similar level.

The ability of preserving diversity can be shown by Figure 4.6, which depicts the non-dominated solutions produced by INSGP and ISPGP on the training set. Diversity refers to how well the solutions are distributed

along the Pareto Front. The solution diversity is important because a good distribution gives a better idea of the Pareto front shape that shows different options of trade-offs between objectives, and increases the chance to find a desired trade-off [107]. Figure 4.6 shows that INSGP solutions have a better spread along the whole frontier than the ISPGP solutions. The reason is that SPEA2 focuses more on solutions in the middle region, while NSGA-II tends to emphasise the edge points of the current non-dominated solutions and thus produce solutions along the whole frontier [44].

Based on the analyses, INSGP tends to evolve solutions that have similar performance as ISPGP solutions. Both methods can reduce solution sizes at a similar level. However, INSGP has a higher ability to preserve solution diversity than ISPGP for the given segmentation tasks.

Table 4.4: Comparison of ISPGP against INSGP (HV means hypervolume).

Dataset	Method	HV (Training Set)	HV (Test Set)	Solution Size (Test Set)
Brodatz	INSGP	0.967	0.973	$22 \pm 7$
	ISPGP	0.966	0.973	$19 \pm 6$
Weizmann	INSGP	0.791	0.741	$35 \pm 19$
	ISPGP	0.758	0.773	$35 \pm 13$
Pascal	INSGP	0.910	0.913	$13 \pm 5$
	ISPGP	0.866	0.874	$14 \pm 6$

#### 4.4.4 Further Analysis

In this part, the solutions on the Pareto front produced by both INSGP and ISPGP will be analysed to reveal the mechanism of how they function to distinguish objects from backgrounds. One test image (Figure 4.10a) is taken as an example. Three cutouts (A: object with a horizontal edge; B: object with an edge that has an angle around  $\frac{\pi}{4}$ ; C: background without edges) are highlighted on this image. Figure 4.10b shows the Gabor filters



used to extract features, along with the feature index. There are 20 Gabor filters, including filters with five scales (4,  $4\sqrt{2}$ , 8,  $8\sqrt{2}$  and 16) and four orientations ( $0$ ,  $\frac{\pi}{4}$ ,  $\frac{2\pi}{4}$  and  $\frac{3\pi}{4}$ ). Each filter contributes 16 feature values to the final 320-dimension feature vector of each cutout.

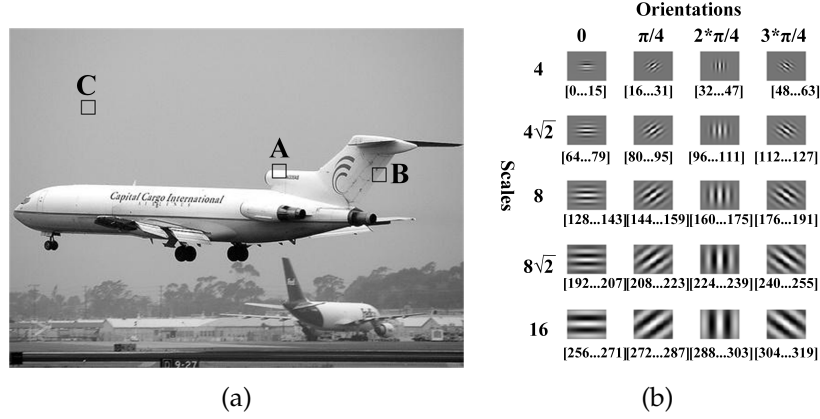


Figure 4.10: Tree analysis on an image (a. Image “2007\_002266” from Pascal dataset with cutout A, B and C; b. Gabor filters ([0...15] represents  $f_0, \dots, f_{15}$  extracted by the filter with the scale of 4 and the orientation of 0, and so forth)).

Figure 4.11 shows two solutions produced by INSGP and ISPGP respectively, which are non-dominated based on the segmentation performance ( $F_1$  score) on the test set. Both solutions have a size of five, which is regarded as a small size as the given segmentation tasks are complex. Based on INSGP and ISPGP, if the value of the root value is smaller than zero, the cutout is considered as class object; otherwise it is considered as class background. In Figure 4.11, the values of terminal nodes are features extracted by the filter being applied to the image, which are subsequently used to calculate the result at the root node. Both solutions have the same structure that contains a comparison operation and then a subtract operation (comparison-subtract structure) even though they were evolved completely separately.

An INSGP solution (Figure 4.11a) is taken as an example. It is known

that a Gabor filter produces high responses when it meets edges at the same orientation as itself. For cutout A (contains a horizontal edge), this solution (Figure 4.11a) considers the difference of edge information at three orientations ( $\frac{\pi}{4}$ ,  $\frac{2*\pi}{4}$  and  $\frac{3*\pi}{4}$ ). As the feature  $f_{170}$  ( $\frac{2*\pi}{4}$  orientation) is smaller than the feature  $f_{242}$  ( $\frac{3*\pi}{4}$  orientation), the comparison operation produces zero. In this case, as Gabor features used in this paper are all larger than zero, the root value will be negative after conducting the subtract operation, thus categorizing cutout A correctly as class object. For cutout B (contains an edge with an angle at  $\frac{\pi}{4}$ ), as feature  $f_{221}$  in the solution (Figure 4.11a) is oriented at  $\frac{\pi}{4}$ , feature  $f_{221}$  has a high value (larger than 1). In this circumstance, no matter the result of the comparison operation, the root value will be negative, classifying cutout B correctly as class object. As cutout C (background) does not contain edges, Gabor features may not be enough to describe it, yet the INSGP solution correctly detects it as class background by studying the pixel values. It reflects that the evolved tree structure and certain features used are effective for given segmentation tasks and not created by chance.

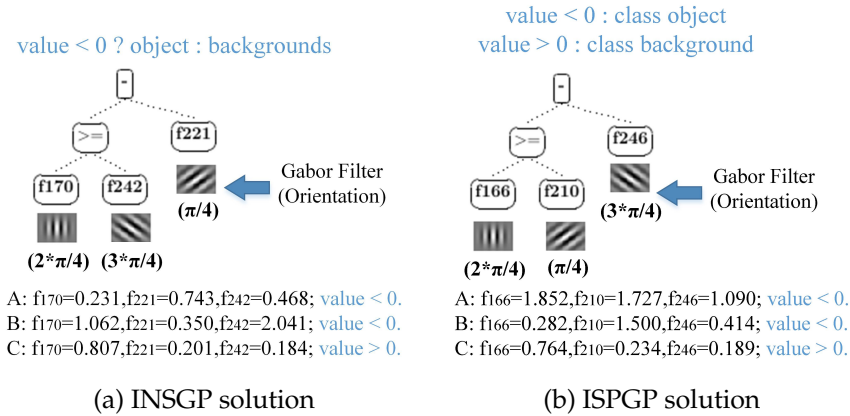
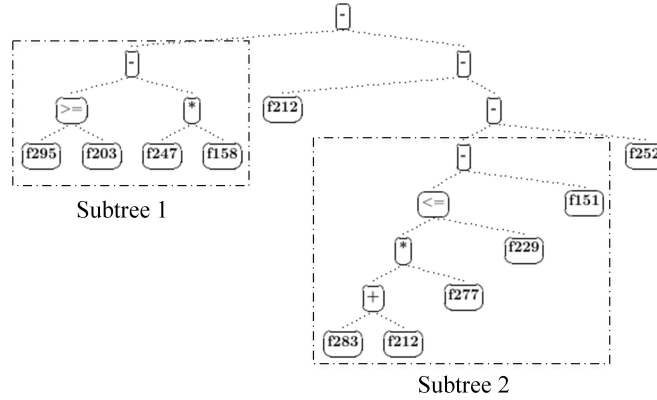


Figure 4.11: Analyses of small solutions non-dominated on Pascal test set (A, B and C represent the three cutouts in Figure 4.10a; value means the value of the tree root node).

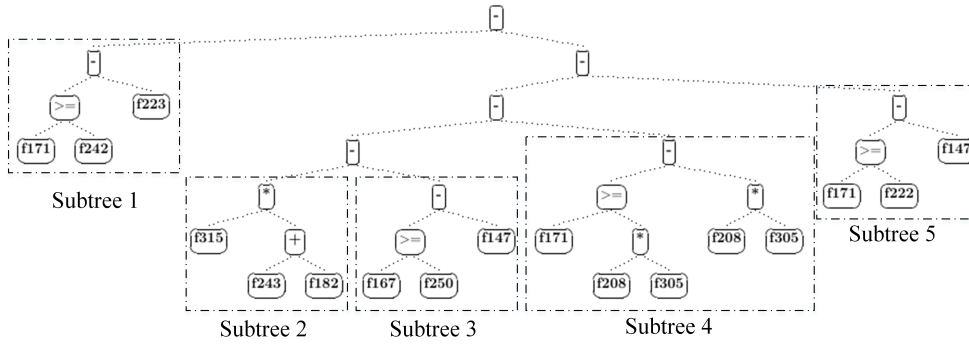
Figure 4.12 shows two relatively large solutions produced by INSGP

and ISPGP, compared with solutions in Figure 4.11. They are also non-dominated solutions based on the test performance. The same tree structure (comparison-subtract structure) that is used by small solutions can also be seen in the large trees. The solution in Figure 4.12a can be simplified as  $rootnode = (Subtree1 + Subtree2) - f_{212} - f_{252}$ . The subtree1 and subtree2 are of a similar comparison-subtract structure. The solution in Figure 4.12b can be simplified as  $rootnode = (Subtree1 + Subtree3 + Subtree4 + Subtree5) - Subtree2$ . The subtree1, subtree3, subtree4 and subtree5 hold the comparison-subtract structure. Compared to the small solutions in Figure 4.11, these large solutions add more operations (multiply and addition) and more features to construct new features. For example, the subtree2 in Figure 4.12b conducts multiply and addition operations with three input features  $(f_{182}, f_{243}, f_{315})$ , both of which are extracted by filters at orientation  $\frac{3*\pi}{4}$ . In this way, a new feature is produced, which provides more accurate information at this orientation. It is concluded that the larger solutions tend to conduct feature construction to produce more effective features to improve segmentation performance.

To further testify that the proposed methods (INSGP and ISPGP) can evolve solutions with certain patterns (e.g. to detect edges), the Pascal training images are rotated by 90 degrees. A new training set is extracted, from which INSGP and ISPGP evolve new solutions. Figure 4.13 shows four non-dominated solutions on the Pascal test set. For the small trees (Figure 4.13a and Figure 4.13b), they contain the same comparison-subtract structure as the small solutions (Figure 4.11a and Figure 4.11b) evolved from the original training set. The large trees (Figure 4.13c and Figure 4.13d) also use the comparison-subtract operation as their major structure, and similar with the large solutions (Figure 4.12) evolved from the original set, they employ more operations (e.g. multiply) and more features. This shows that the proposed methods search for solutions with certain patterns that are effective for the given segmentation tasks.



(a) INSGP solution



(b) ISPGP solution

Figure 4.12: Analyses of large solutions non-dominated on Pascal test set.

## 4.5 Chapter Summary

In this chapter, a parsimony pressure technique (weighted sum) and two multi-objective techniques (NSGA-II and SPEA2) were introduced in standard GP to design new segmentation methods, which are expected to reduce the solution complexity. Specifically, standard NSGA-II and standard SPEA2 were improved by including preference information to enable GP to focus on searching the area on the Pareto front of a user's interest, which are termed as INSGA-II (improved INSGA-II) and ISPEA2 (improved SPEA2). INSGA-II and ISPEA2 were incorporated with GP to de-

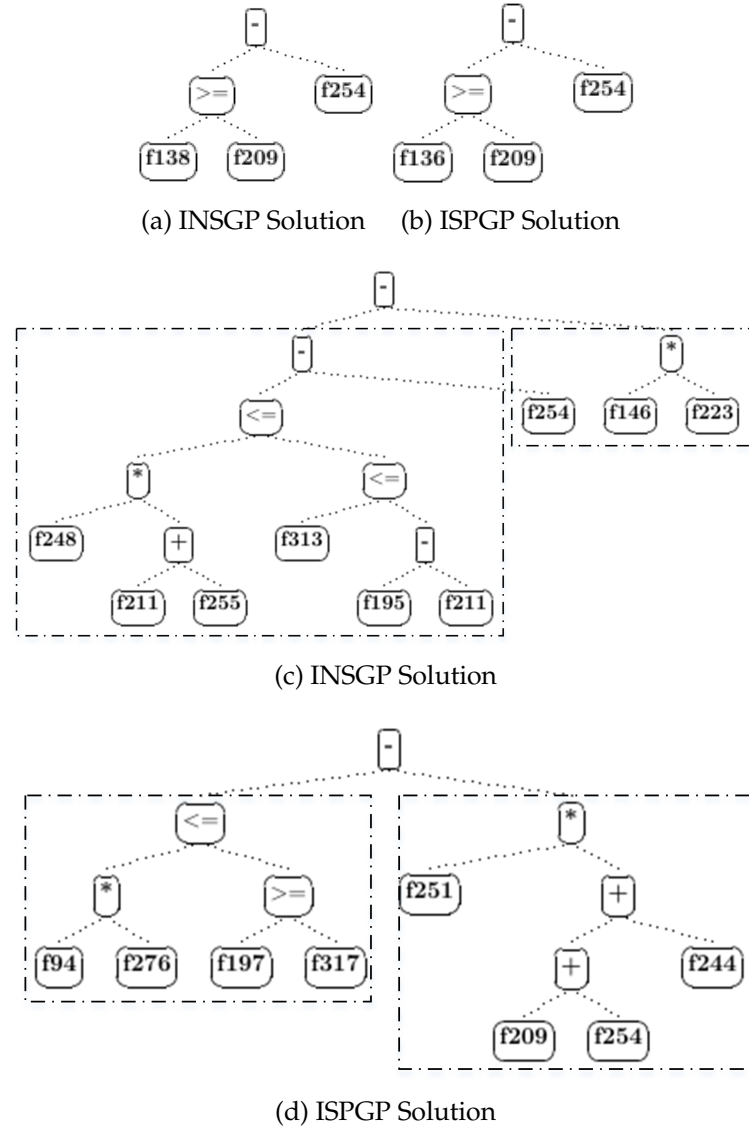


Figure 4.13: Solutions (evolved from a rotated Pascal training set) non-dominated on Pascal test set.

velop two new segmentation methods (INSGP and ISPGP). The weighted sum method considers both the segmentation performance and the solution complexity, but combines them into a single objective. In contrast, the

multi-objective methods treat two objectives (i.e. the segmentation performance and the solution complexity) independently, aiming to achieve good trade-offs between the two objectives. Subsequently, the proposed methods were tested on three benchmark image datasets, i.e. Brodatz texture, Weizmann and Pascal datasets, and were compared with a reference method (based on standard GP without complexity control).

The results show that the proposed methods (the weighted sum method, INSGP and ISPGP) can significantly reduce the solution sizes while achieving similar segmentation performance to the reference method. Moreover, INSGP and ISPGP can evolve solutions with further reduced sizes. This reflects that the proposed multi-objective methods (INSGP and ISPGP) can evolve effective trade-off solutions between the solution functionality and complexity. When comparing ISPGP with INSGP, solutions evolved by both methods have similar segmentation performance and similar size levels. However, INSGP solutions have higher diversity than ISPGP solutions for the segmentation tasks in this chapter. Further analysis on the evolved solutions reveals that certain patterns are captured to detect edges.

Even though promising results were achieved in this chapter, it is still challenging to achieve accurate segmentation on complex images with high variations (e.g. cluttered backgrounds and/or varying objects). Effective image features that can capture difference between object pixels and background ones are necessary. Feature selection can help remove noisy/redundant features. Thus introducing feature selection could improve the segmentation performance. Similarly, feature construction can produce new features that consider interactions of the primitive features. The constructed features are anticipated to be more distinctive than primitive features. Therefore, feature selection and construction will be investigated in the following chapters.

## Chapter 5

# GP for Feature Selection in Figure-ground Segmentation

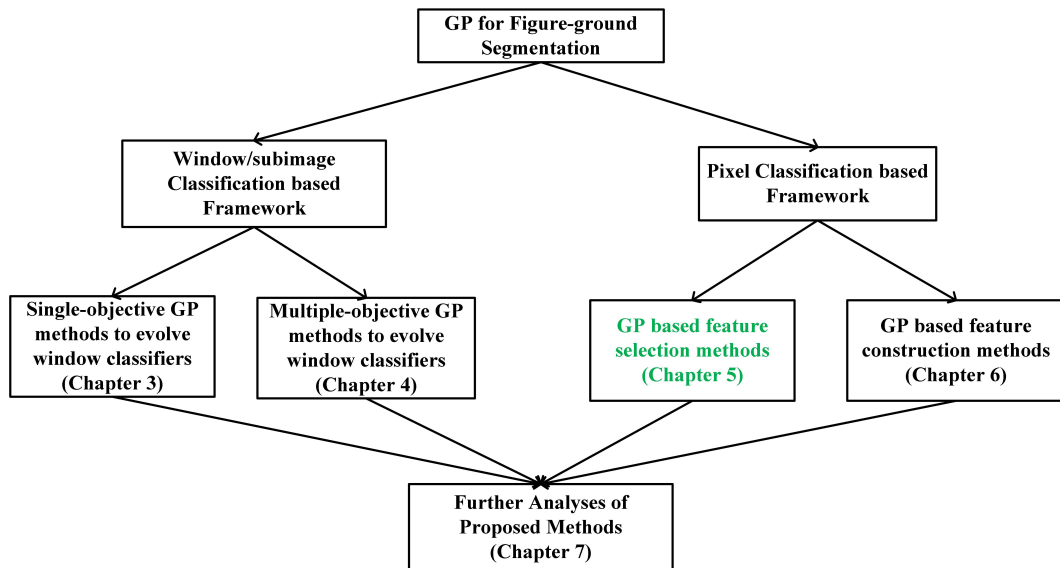


Figure 5.1: The overall structure of contributions.

Diverse types of image features with the aim to determine what image information is necessary for GP to evolve capable segmentors were investigated in Chapter 3. Results show that texture features based GP methods

perform better for figure-ground segmentation than those using raw pixel values and shape features. Thus, based on the Gabor features, the parsimony pressure and multi-objective techniques were introduced in GP to design new methods with the aim to balance the solution functionality and complexity in terms of solution size (see Chapter 4).

Even though well-performing solutions with smaller sizes are achieved, it is still challenging to segment images with high variations, which requires effective feature sets to capture the distinguishing information between objects and backgrounds. Therefore, feature selection and construction will be investigated to produce effective feature sets in the following chapters (see Chapter 5 and Chapter 6 respectively). This chapter focuses on feature selection first as removing features from primitive features is more straightforward than constructing new features.

## 5.1 Introduction

In figure-ground segmentation, one challenge is to segment images with high variations (e.g. containing a cluttered background) [96, 98], which requires effective feature sets to capture the distinctive information between objects and backgrounds. GP has been applied to feature selection [41, 121, 124, 158], and promising results have been achieved. However, there is no work that introduces GP based feature selection for figure-ground segmentation tasks. Moreover, feature selection is a multi-objective task, maximizing the effectiveness of selected features and minimizing the number of selected features, but existing GP based feature selection works rarely consider the number of features. Therefore, it is necessary to further investigate GP based feature selection for figure-ground segmentation.

### 5.1.1 Chapter Goals

The aim of this chapter is to design feature selection methods using GP for figure-ground segmentation tasks, which are expected to maximise the segmentation performance based on the selected features and minimise the number of selected features. To achieve this goal, the parsimony co-



efficient technique and two multi-objective techniques are introduced in GP to develop new feature selection methods. Note that the parsimony coefficient technique is popular for bloat control, which penalizes the fitness of candidate solutions by decreasing the raw fitness by an amount dependent on the tree size [97]. It is employed to handle multi-objective optimisation in feature selection by penalizing the fitness of selected feature subsets based on the number of features that they contain. On the other hand, the multi-objective techniques treat the two objectives independently. These techniques are selected to develop new methods since they are popularly used to handle multi-objective optimisation problems.

Specifically, the following objectives will be investigated:

1. explore whether the proposed methods can select effective feature subsets for complex image segmentation tasks,
2. investigate which one of the proposed methods performs better, e.g. in terms of segmentation performance, the number of selected features and the training/test time.

### 5.1.2 Chapter Organisation

The rest of the chapter is organised as follows. Section 5.2 introduces the overall structure of feature selection using GP for figure-ground segmentation tasks. Three new methods that consider the number of selected features, i.e. the parsimony pressure based method and two multi-objective GP methods, are described in Section 5.3. In addition, Section 5.4 and Section 5.5 provides the experimental preparation and results with discussions. In Section 5.6, the contributions of this chapter is summarised.

## 5.2 Overall Structure of GP based Feature Selection

Figure-ground segmentation tasks are formulated to window classification based problems in Chapter 3 and Chapter 4. However, there are two issues: the window related parameters are required, e.g. the window size and the number of the window shifting steps, which rely on users' expertise for setting; the window classification based methods tend to produce inaccurate object boundaries due to the overlapping of moving windows and the voting scheme. Therefore, in this chapter, a pixel classification based segmentation framework is developed. It does not need window related parameters and the voting scheme, which are required by the window classification based framework.

Note that the term "pixel classification" does not mean classification based on isolated pixels on an image. It is used to differentiate from "window classification" in two aspects. Firstly, for the pixel classification based framework, features are extracted based on the neighbourhood of each pixel to treat each pixel independently from others when conducting classification, and the predicted class label is assigned to this pixel. In contrast, for the window classification based framework, features are extracted based on an image patch and each image patch is treated as an individual sample when conducting classification, and the predicted class label is assigned to all pixels of this image patch. Secondly, for the pixel classification based framework, each pixel has one predicted class label; while for the window classification based framework, each pixel may have multiple predicted labels due to overlapping of the shifting windows (details are in Chapter 3).

In this chapter, based on the pixel classification framework, a feature selection process is introduced, aiming to select effective feature subsets for improving the segmentation performance.

### 5.2.1 Pixel Classification Based Figure-ground Segmentation

The framework of the pixel classification based segmentation with feature selection (shown in Figure 5.2) includes four major steps. Firstly, features are extracted from the training, validation and test image sets to form a training set, a validation set and a test set respectively. Secondly, the training set and the validation set (instances with extracted features) are used as inputs to a proposed GP based feature selection method, which selects a feature subset. Thirdly, the training and test sets are transformed based on the selected feature subset. A standard classifier (e.g. Naïve Bayes) is trained using the transformed training set, and then the trained classifier categorises the transformed test samples as class *object* or *background*. Eventually, based on the ground truth and certain measures (e.g. the segmentation accuracy), the segmentation performance based on the selected feature subset and the classifier can be generated.

### 5.2.2 Feature Selection using GP based Methods

Figure 5.3 illustrates how feature subsets are generated in the evolutionary process of GP. For each GP evolved program, its terminals are collected, and the unique features that occur in the terminals form the selected feature subset. For example, the left tree solution in Figure 5.3 contains four terminals  $[f_0, f_3, f_1, f_3]$ , in which there are three unique features  $[f_0, f_1, f_3]$ . Therefore, the feature subset selected by this solution is  $[f_0, f_1, f_3]$ .

### 5.2.3 Feature Extraction

For the pixel classification based figure-ground segmentation, the task of separating the object from background is to distinguish the object pixels from the background ones. Feature extraction is necessary to describe im-

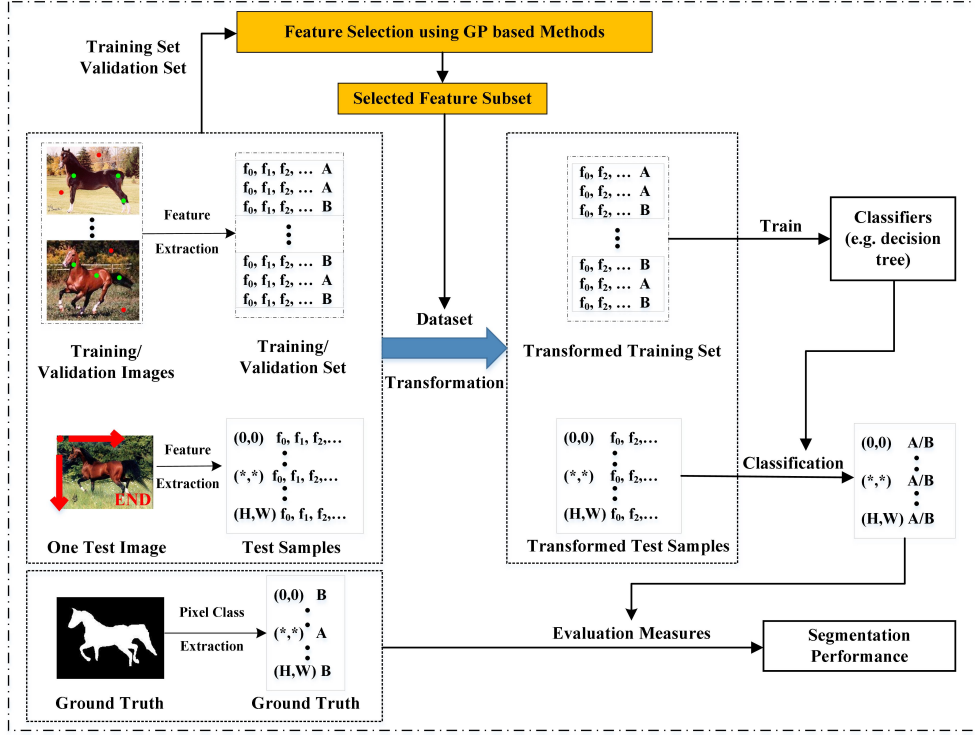


Figure 5.2: Framework of the pixel classification based figure-ground segmentation with feature selection ( $f_0$  is a feature with 0 as its index and so forth; A and B represent class *object* and *background* respectively;  $(0,0)$ ,  $(0,1)$  and  $(H,W)$  are the coordinates of pixels; H and W are the test image's height and width).

age pixels for accurate classification performance.

Given an image with the aeroplane as the region of interest (Figure 5.4), several pixels are highlighted, i.e. pixels  $a, b, c$  (class *object*) and pixels  $d, e, f$  (*background*) pixels. Different information is required to distinguish different pixels. For pixels  $c$  and  $e$ , as they have different colours, the colour information could be effective to separate them. For pixels  $b$  and  $f$ , they have a similar colour and a similar grayscale value, and both are edge pixels, so pure colour or intensity values would not work and higher-order information, e.g. grayscale statistics, is needed. For pixels  $a$  and  $d$ , as a is

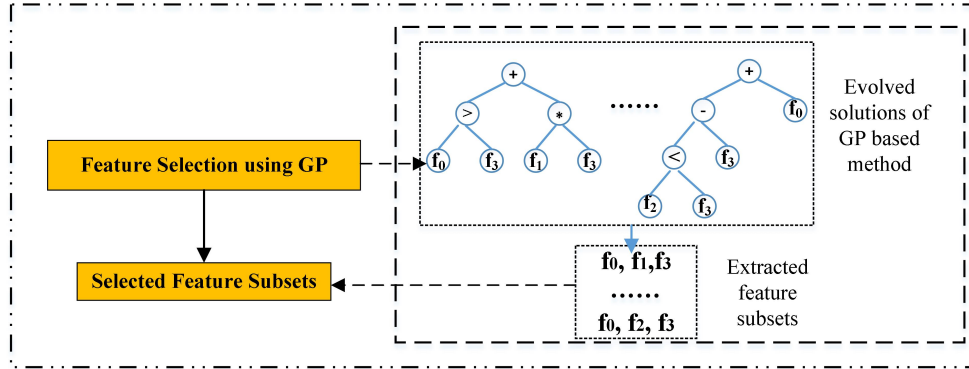


Figure 5.3: Feature subset generation ( $f_0$  is a feature with 0 as its index and so forth).

an edge pixel while  $d$  is not, edge features could be used to distinguish them easily. Therefore, features that contain edge information, colour and grayscale statistics information will be extracted from the neighborhood of an image pixel to describe it. Note that texture features (edge features and grayscale statistics) and colour features are used, while shape features are not included as they are suitable to describe image patches rather than individual pixels.

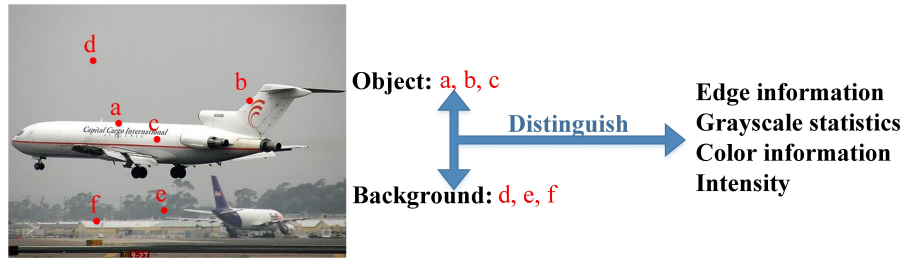


Figure 5.4: Distinguishable image information.

Figure 5.5 shows the feature descriptors employed in this chapter to extract the features mentioned above. The feature descriptors include Gabor filters, the mean and the median filters, LBP (local binary patterns) operators and pixel intensities. Specifically, Gabor filters are powerful edge descriptors [178], LBP operators are powerful texture descriptors that can

represent the small-scale image appearance [132], and the mean and the median filters can extract local statistical features. Details of feature descriptors can be found in Section 2.5.2.

As can be seen in Figure 5.5, there are in total 53 features to represent each pixel, and all the extracted features are normalized to  $[0, 1]$ . Forty Gabor filters are applied, generated from five common scales ( $4, 4\sqrt{2}, 8, 8\sqrt{2}, 16$ ) and eight common orientations ( $0, \frac{\pi}{8}, \frac{2\pi}{8}, \frac{3\pi}{8}, \frac{4\pi}{8}, \frac{5\pi}{8}, \frac{6\pi}{8}, \frac{7\pi}{8}$ ), hence there are a total of 40 Gabor features. The mean filter, median filter and LBP are employed to extract local statistical features. Three window sizes are employed, i.e.  $3 \times 3$  pixels,  $5 \times 5$  pixels and  $9 \times 9$  pixels, which can capture information at different scales.

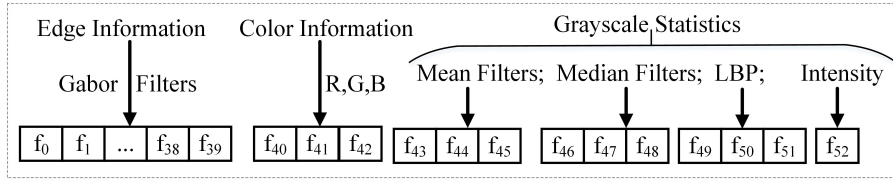


Figure 5.5: Extracted features and their dimension indexes ( $f_0$  is a feature with 0 as its index and so forth).

### 5.3 Proposed Feature Selection Methods

To balance the effectiveness and the number of features of the selected feature subsets, a parsimony pressure technique (i.e. the parsimony coefficient technique) is introduced in GP to develop the parsimony pressure GP method for feature selection, termed as PGP-FS. In addition, two multi-objective techniques, i.e. NSGA-II and SPEA2, are combined with GP to form two novel multi-objective GP methods for feature selection, termed as NSGP-FS (nondominated sorting GP feature selection) and SPGP-FS (strength Pareto GP feature selection).

### 5.3.1 The Parsimony Coefficient GP Method

The parsimony coefficient technique is used to combine with GP to develop the feature selection method (PGP-FS) in this section, aiming to balance the effectiveness and the number of selected features. In PGP-FS, the fitness function introduces a penalty term based on the number of selected features to the classification accuracy. Specifically, the classification accuracy can measure the effectiveness of selected feature subsets, while the penalty term can be designed to favour the feature subsets with smaller numbers. The classification accuracy is based on the Naïve Bayes algorithm as it is a robust and well understood classifier, although other classification methods could be substituted without changing the proposed method.

The fitness function is shown in Equation 5.1, where  $c$  means the parsimony coefficient that can reflect how much importance the fitness evaluation addresses the penalty measure (see Section 5.5.1 for the setting of  $c$ ), and  $penalty(number)$  refers to a penalty measure based on the number of selected features in a GP solution.

$$fitness = classification\ accuracy - c * penalty(number). \quad (5.1)$$

The sigmoid function ( $Sigmoid(x) = \frac{1}{1+exp(-x)}$ ) is selected as the basic penalty measure. This is because it rises monotonically with the increase of the input, which leads to higher penalties to solutions with larger numbers. Moreover, the output of this function is within the value range of (0, 1), which makes it suitable to be combined linearly with the classification accuracy (within [0, 1]). Note that other functions that suit the two criteria can also be employed.

Note that the fitness function (Equation 4.1) uses an exponential function (Equation 4.3) rather than the sigmoid function as the complexity measure of evolved solutions. That is because the function (Equation 4.1) is a weighted sum of the classification accuracy and the complexity measure based on the solution size. Therefore, to penalise the solutions with

large sizes, the complexity measure should decrease with the increase of the solution size. As the exponential function suits the above requirement, it is selected as the complexity measure in the fitness function (Equation 4.1).

The curve of the sigmoid function (Figure 5.6a) shows that it can transform the solution number into a value in range  $[0, 1)$ , and with the increase of the number, its output value rises, thus increasing the penalty. However, the output value calculated by the sigmoid function approaches 1 with a small solution number (around 6 from Figure 5.6a). Based on the experimental observation, without the penalty term, the solution number can be hundreds. Therefore, a scaling factor,  $\beta = 0.01$ , is applied to the solution number to make the penalty measure effective for solutions with numbers being hundreds (see Section 4.2.1 for the details of *beta* setting). The penalty measure is presented in Equation 5.2, with its curve in Figure 5.6b.

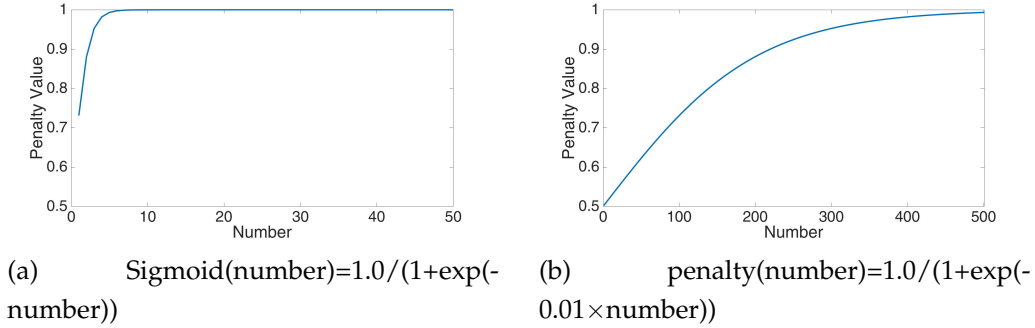


Figure 5.6: Penalty measure (number refers to the number of features selected in a GP solution; for tree based GP, it is the number of distinctive terminal nodes).

$$\text{penalty}(\text{number}) = \frac{1}{1 + e^{(-\beta * \text{number})}}, \quad (\beta = 0.01). \quad (5.2)$$



### 5.3.2 The Multi-objective GP Methods

Feature selection is considered to contain two objectives, i.e. minimizing the number of selected features and maximizing the segmentation performance. Therefore, feature selection can be formulated to be a multi-objective optimisation problem. Pareto dominance is a popular way to deal with multi-objective problems by treating two objectives independently, which can search the space of two independent objectives to produce trade-offs along all objectives rather than one best solution for parsimony pressure based methods. In contrast, parsimony pressure techniques aggregate the objectives into a scalar fitness function based on predefined weights or coefficients [134]. Thus, the objective space of parsimony pressure based methods is restricted to a part of that for Pareto dominance multi-objective based methods, i.e. where the two objectives suit a linear relation determined by the weights or coefficients.

There are many existing Pareto-based multi-objective optimisation techniques, i.e. nondominated sorting genetic algorithm (NSGA and NSGA-II [44]), strength Pareto evolutionary algorithm (SPEA and SPEA2 [183]) and Pareto archived evolutionary strategy (PAES) [37]. For similar reasons to selecting NSGA-II and SPEA2 as base multi-objective optimisation techniques in Chapter 4 (see Section 4.2), they are also chosen for feature selection in multi-objective GP.

NSGA-II and SPEA2 are combined with GP to develop the NSGP-FS (nondominated sorting GP feature selection) and the SPGP-FS (strength Pareto GP feature selection) respectively. NSGP-FS and SPGP-FS take the same two objectives: the classification accuracy based on Naïve Bayes (the same as the PGP-FS method) and the number of features (the number of terminal nodes) used in a GP solution. Both algorithms aim to maximise the classification accuracy of Naïve Bayes while minimizing the number of features, thus they are expected to provide solutions that can improve the segmentation performance while reducing the processing time. The Pseudo codes of NSGP-FS and SPGP-FS are provided in Algorithm 4 and

Algorithm 5 respectively.

## 5.4 Experiment Preparations

### 5.4.1 Benchmark Methods for Comparison

Since sequential forward selection (SFS) and sequential backward selection (SBS) are two standard and state-of-art feature selection methods that are widely used [30, 174, 153], they are selected to compare with the proposed methods. SFS and SBS methods are provided by the Weka package [66], in which the Naïve Bayes is used to evaluate the subsets in the process of feature selection as it is a robust and well understood classifier.

### 5.4.2 Image Datasets

Two benchmark image segmentation datasets (see Section 2.5.1 for details) are selected, i.e. the Weizmann horse dataset and the Pascal VOC2012 aeroplane dataset. The Weizmann dataset contains 328 horse images. Two thirds of all images are used as training/validation images (148 images for training and 70 for validation), and the remainder (110 images) are used for testing. From each training/validation image, 20 samples are extracted (10 from objects; 10 from background). The Pascal VOC2012 segmentation dataset has 178 aeroplane images. As suggested by Everingham et al. [53], 88 images are for training and 90 images for testing. The 88 training images are further divided into a new training set (60 images) and a validation set (28 images). Since Pascal images are larger than the Weizmann images, 50 samples (25 from objects; 25 from background) are extracted on each training/validation image.

```

input : G: the maximum number of generations;
        N: the population size;
        The terminal set, function set and GP setting parameters (Section 5.4.3).
output: Solutions (feature subsets).

1 Create an initial population at iteration zero ( $P_0$ ) of GP trees using the
  Ramped half-and-half method;
2  $g \leftarrow 0$ 
3 while  $g < G - 1$  and the ideal individual (the individual with
  classification accuracy == 1.0 and #selected features == 1) is not found do
4   begin
5     // different procedures for generation 0 and
     generations onward; k records the index of the
     child generation to be created
6     if  $g == 0$  then
7       Sort  $P_0$  by nondominated sorting based on the Pareto dominance
       mechanism;
8        $k \leftarrow g$ ;
9     else
10       $C_g = P_g \cup Q_g$ ; //  $C_g$  is the combined population of the
       $g$ th generation;
11      Sort  $C_g$  by nondominated sorting based on the Pareto dominance
      mechanism;
12       $P_{g+1} = C_g[0 : N - 1]$ ; // choose the first N elements;
13       $k \leftarrow g + 1$ ;
14    end
15    Create a child population  $Q_k$  from  $P_k$ :
16    begin
17      Set  $Q_k$  empty;
18      Assign each individual of  $P_k$  a fitness value equal to its
      nondominance level;
19      for  $i = 0$  to  $N-1$  do
20        Select parent(s) by tournament selection from  $P_k$ ;
21        Generate children by mutation operation or crossover operation;
22        Add the children to  $Q_k$ ;
23      end
24    end
25     $g \leftarrow g + 1$ ;
26 end
27 Get Pareto front individuals (nondominated individuals in the latest child
  population  $Q_g$ );
28 Return feature subsets (one feature subset consists of features that are
  terminals of one Pareto front individuals).

```

**Algorithm 4:** Pseudo-code of NSGP-FS (GP procedure is in red and NSGA-II procedure is in blue).

```

input : G: the maximum number of generations;
        N: the population size;
         $\bar{N}$ : the archive size;
        The terminal set, function set and GP setting parameters (Section 5.4.3).
output: Solutions (feature subsets).

1  Create an initial population at iteration zero ( $P_0$ ) of GP trees using the
    Ramped half-and-half method;
2  Create an empty archive ( $\bar{P}_0$ );
3   $g \leftarrow 1$ 
4  while  $g < G - 1$  and the ideal individual (the individual with
    classification accuracy == 1.0 and #selected features == 1) is not found do
5      Fitness assignment: sort  $P_g$  by nondominated sorting based on the Pareto
        dominance mechanism; assign each individual a fitness value equal to
        its nondominance level;
6      Copy all nondominated individuals in  $P_g$  and  $\bar{P}_g$  to  $P_{g+1}$ :
7      begin
8          if  $|P_{g+1}^-| > \bar{N}$  then
9              Reduce  $P_{g+1}^-$  to  $\bar{N}$  by the truncation operator;
10         end
11         if  $|P_{g+1}^-| < \bar{N}$  then
12             Fill  $P_{g+1}^-$  to  $\bar{N}$  with dominated individuals in  $P_g$  and  $\bar{P}_g$ ;
13         end
14     end
15     Create  $P_{g+1}$  from  $P_{g+1}^-$ :
16     begin
17         Set  $P_{g+1}$  empty;
18         for  $i = 0$  to  $N-1$  do
19             Select one/two parent(s) by tournament selection from  $P_{g+1}^-$ ;
20             Generate one child/two children by mutation operation or
                crossover operation;
21             Add the child/children to  $P_{g+1}$ ;
22         end
23     end
24      $g \leftarrow g + 1$ ;
25 end
26 Get Pareto front individuals (nondominated individuals in the latest
    population  $P_g$ );
27 Return feature subsets (one feature subset consists of features that are
    terminals of one Pareto front individuals).

```

**Algorithm 5:** Pseudo-code of SPGP-FS (GP procedure is in red and SPEA2 procedure is in blue).

### 5.4.3 GP Settings

The GP set-up parameters employ the default settings used in Koza's works [83] (see Section 2.5.4 for details) except for the population size. Compared with 1024 set by Koza, the population size is set to 500, which is sufficient to solve the related problems while also reducing the computational cost.

The function set, including four standard arithmetic operators and five conditional/relational operators, is displayed in Table 5.1. All the operators are simple and efficient to calculate. In addition, the terminal set consists of the input features to the GP based system. Based on the description in Section 5.2.3, there are 53 features in the terminal set, including 40 Gabor, three colour, three mean, three median, three LBP and one intensity features.

Table 5.1: Function set of GP based feature selection methods.

Function Name	Definition
$+(a_1, a_2)$	$a_1 + a_2$
$-(a_1, a_2)$	$a_1 - a_2$
$*(a_1, a_2)$	$a_1 * a_2$
$/(a_1, a_2)$	$\begin{cases} a_1/a_2 & \text{if } a_2 \neq 0 \\ 0 & \text{if } a_2 == 0 \end{cases}$
$\text{IF}(a_1, a_2, a_3)$	$\begin{cases} a_2 & \text{if } a_1 \text{ is true} \\ a_3 & \text{if } a_1 \text{ is false} \end{cases}$
$\leq (a_1, a_2)$	$\begin{cases} 1 & \text{if } a_1 \leq a_2 \\ 0 & \text{if otherwise} \end{cases}$
$\geq (a_1, a_2)$	$\begin{cases} 1 & \text{if } a_1 \geq a_2 \\ 0 & \text{if otherwise} \end{cases}$
$== (a_1, a_2)$	$\begin{cases} 1 & \text{if } a_1 == a_2 \\ 0 & \text{if otherwise} \end{cases}$
$\text{Between}(a_1, a_2, a_3)$	$\begin{cases} 1 & \text{if } a_2 \leq a_1 \leq a_3 \\ 0 & \text{if otherwise} \end{cases}$

### 5.4.4 Experiment Design

Three standard classifiers from the Weka package (i.e. decision tree (DT), Naïve Bayes (NB) and multilayer perceptron (MLP)) are employed for testing the selected subsets. This is because they are widely used and well understood classifiers. As the segmentation tasks are transformed to binary classification problems in this chapter and the segmentation accuracy and  $F_1$  score are popularly used to evaluate binary classifiers, they are applied to evaluate the segmentation results (see Section 2.5.3). Both measures reach the worst at 0 and the best at 1. The base code of GP and two multi-objective techniques (i.e. NSGA-II and SPEA2) are from the ECJ library [110]. All GP related experiments are run 30 times.

## 5.5 Results and Discussions

### 5.5.1 Determining Coefficient $c$ in Fitness Function of PGP-FS

The coefficient in fitness function is an important parameter, as it reflects how much control GP has on the penalty term. If it is too small, GP will mainly focus on the classification accuracy based on the selected features, so the number of selected features can hardly be controlled; if it is too large, GP will take the minimisation of the number of selected features as its main target and may ignore the solution functionality, thus converging towards extremely small but useless programs [136]. In addition, the coefficient is normally determined by trial-and-error [136].

Table 5.2 presents the training performance of PGP-FS with different coefficients ( $c$ ) in the fitness function (Equation 5.1) on the Pascal dataset. As the penalty value (calculated by the penalty measure, Equation 5.2) is not intuitive, the number of features is also provided. Normally, the larger the feature numbers, the more complex the evolved solutions. This

table shows that with the increase of the coefficient, the feature number decreases, yet the training accuracy also drops. Therefore, the choice of the coefficient is to select a balance between the performance and the size of selected feature subsets. Eventually,  $c$  is set to 0.1, as in this circumstance, the feature number is dramatically reduced compared with those under  $c = 0.001$  or 0.01. Moreover, the increase of  $c$  (0.3, 0.5, 0.7, 0.9) can hardly further reduce the feature number.

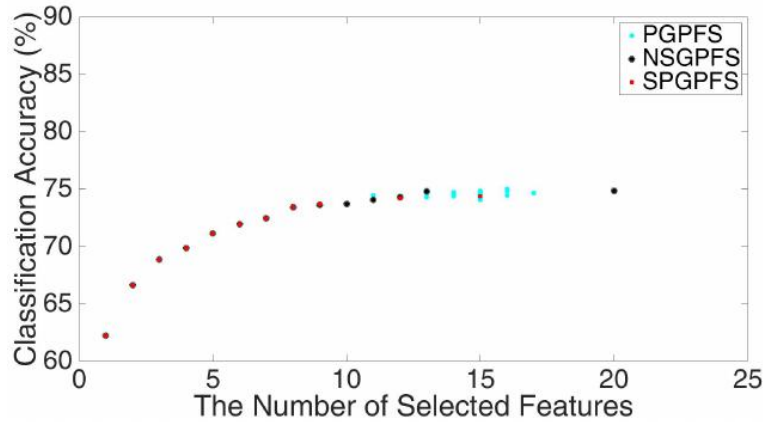
Table 5.2: Training performance of PGP-FS on the Pascal dataset with different coefficients in its fitness function.

Coefficient $c$	Training Accuracy(%)	Penalty Value	Number of Features
0.001	82.377	0.883	220
0.01	82.241	0.756	118
0.1	81.459	0.543	17
0.3	81.046	0.527	11
0.5	80.703	0.523	9
0.7	80.569	0.520	8
0.9	80.304	0.518	7

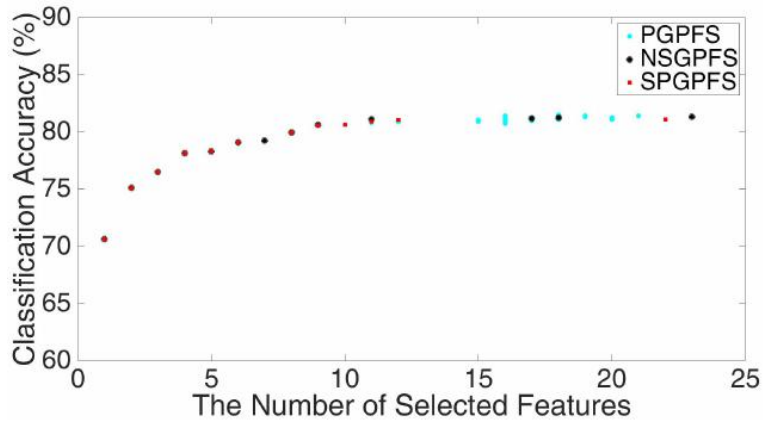
### 5.5.2 Training Results

This part analyses the results on the training stage (feature selection stage). Figure 5.7 displays the single best solutions over 30 runs for PGP-FS, and the aggregated Pareto fronts formed by nondominated solutions over 30 runs for NSGP-FS and SPGP-FS. From this figure, it can be seen that all solutions contain less than 25 features, which are much smaller than the whole set of 53 features. Moreover, most NSGP-FS and SPGP-FS solutions have fewer features than PGP-FS solutions.

Table 5.3 provides the statistical results. For PGP-FS, the average of the best solutions in the feature number and the classification accuracy



(a) Weizmann set



(b) Pascal set

Figure 5.7: Training performance on the Weizmann and Pascal sets.

are shown. For the multi-objective methods, NSGP-FS and SPGP-FS, the nondominated solutions are trade-offs of the two objectives that are considered equivalently important, so it is not appropriate to calculate the average performance. To compare NSGP-FS and SPGP-FS against PGP-FS, whether NSGP-FS or SPGP-FS solutions can dominate the PGP-FS's assumed "average solution" is reported. For example, on the Weizmann dataset, it is considered that PGP-FS produces an "average solution" with 14 features and a classification accuracy of 74.57%.



According to Table 5.3, the nondominated solutions of NSGP-FS can dominate PGP-FS’s “average solution” on both datasets. Even though SPGP-FS solutions cannot dominate PGP-FS’s “average solution” on the Weizmann set, SPGP-FS and PGP-FS solutions achieve similar accuracies generally (Figure 5.7). Moreover, NSGP-FS and SPGP-FS (especially SPGP-FS) spend less training time than PGP-FS. For example, on the Weizmann dataset, NSGP-FS’s training time is less than a half, and SPGP-FS’s training time is only around one quarter of that of PGP-FS. This is because NSGP-FS and SPGP-FS (especially SPGP-FS) generate feature subsets with much fewer features than PGP-FS solutions, which speeds up training.

Table 5.3: Statistical results (#Feature means the number of features in the selected feature subsets; Acc. is classification accuracy; “Dominate” means whether NSGP-FS (or SPGP-FS) solutions can dominate the PGP-FS’s “average solution”).

Dataset	Method		#Feature	Training Acc.(%)	Time per run(s)
Weizmann	PGP-FS	Average	$14.27 \pm 1.32$	$74.57 \pm 0.23$	$303.81 \pm 48.60$
	NSGP-FS	Dominate		YES	$136.63 \pm 8.37$
	SPGP-FS	Dominate		NO	$80.33 \pm 6.23$
Pascal	PGP-FS	Average	$15.67 \pm 3.29$	$75.40 \pm 0.20$	$295.21 \pm 56.54$
	NSGP-FS	Dominate		YES	$138.22 \pm 17.56$
	SPGP-FS	Dominate		YES	$86.91 \pm 6.35$

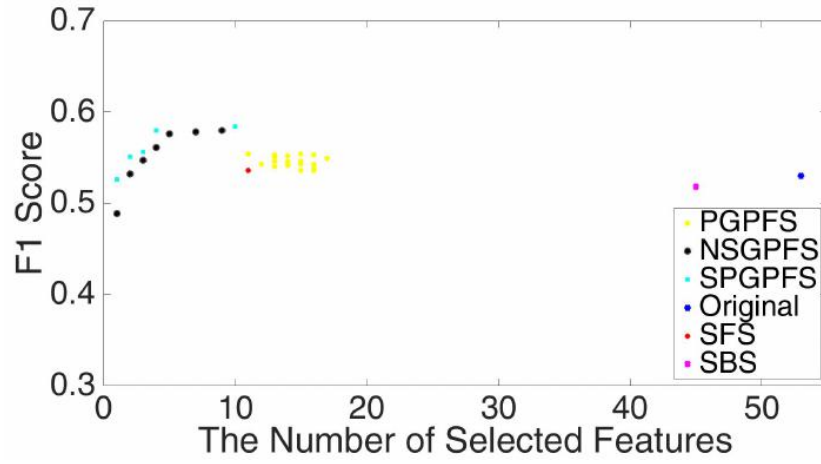
Based on the analyses, all the three proposed methods can generate feature subsets with much fewer features than the whole set (53 features), and the multi-objective methods (NSGP-FS and SPGP-FS, especially SPGP-FS) produce even smaller feature subsets, thus the training time is reduced accordingly. In addition, the training performance is similar for solutions with the same number of features produced by all three methods.

### 5.5.3 Test Results

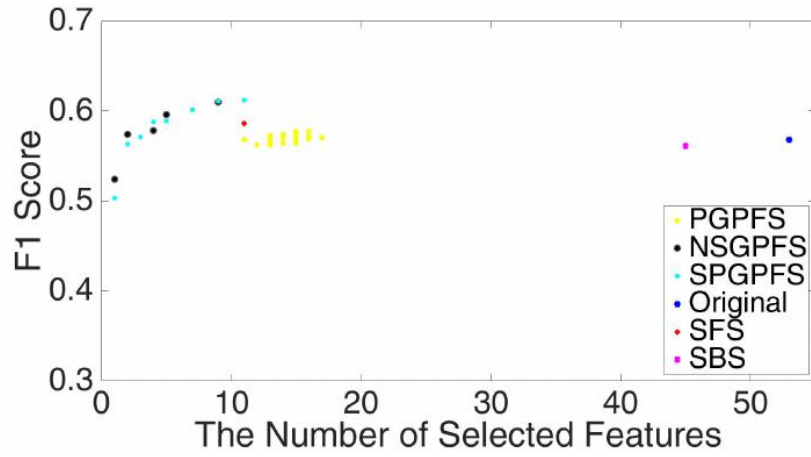
In this part, the feature subsets produced by the five feature selection methods (i.e. three proposed methods, SFS and SBS) and the whole feature set are tested by three classifiers (i.e. decision tree (DT), Naïve Bayes (NB) and multilayer perceptron (MLP)) for the segmentation tasks. The performance is measured by the segmentation accuracy (Equation 2.1) and the  $F_1$  score (Equation 2.2). For PGP-FS, the results of feature subsets that are from the single best solutions over the 30 training runs are displayed in the figures (Figure 5.8 and Figure 5.9). For NSGP-FS and SPGP-FS, all the solutions from the aggregated Pareto fronts over 30 training runs are tested; however, only the nondominated solutions are displayed. Table 5.4 and Table 5.5 show the statistical results on the test sets.

Figure 5.8 and Figure 5.9 show that all the five feature selection methods reduce the number of features compared with the original set, especially the proposed methods and SFS. Specifically, compared with the original set (53 features), NSGP-FS and SPGP-FS solutions are within 10 features on the Weizmann dataset and within 20 on the Pascal dataset; PGP-FS solutions are from around 10 to 20 features on both datasets; SFS solutions are around 10 on the Weizmann dataset and around 5 on the Pascal dataset. In terms of performance ( $F_1$  score), NSGP-FS and SPGP-FS solutions can dominate PGP-FS, SFS, SBS solutions plus the original set. It means that NSGP-FS and SPGP-FS solutions with the same number of features achieve better performance than that of other methods' solutions. In contrast, even though PGP-FS solutions are generally better than the SBS solutions and the original set with lower numbers and similar performance, they are worse than the SFS solutions for most cases, e.g. Naïve Bayes based segmentation on both datasets.

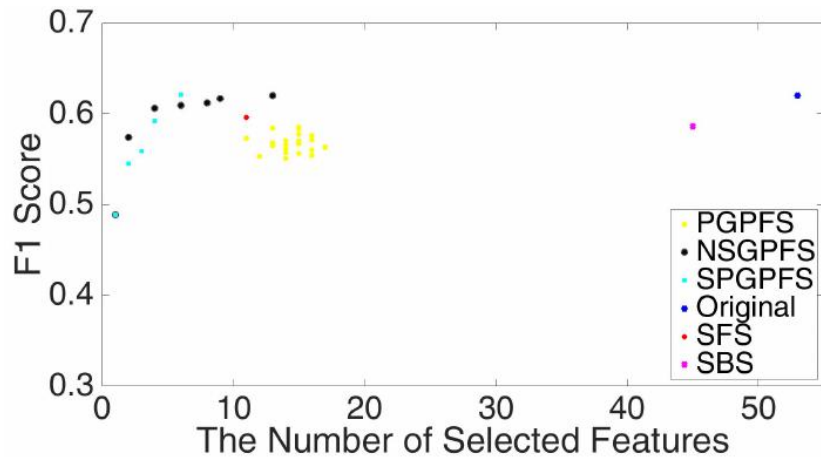
Table 5.4 and Table 5.5 display the statistical results. Compared with the whole feature set, the test time per image drops for all the three classifiers that use the feature subsets provided by the five feature selection methods, especially for the cases using NSGP-FS and SPGP-FS solutions.



(a) Based on decision tree.

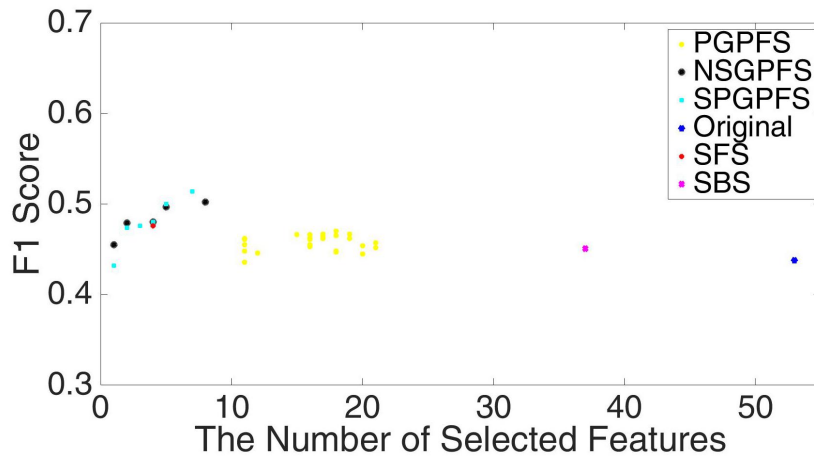


(b) Based on Naïve Bayes.

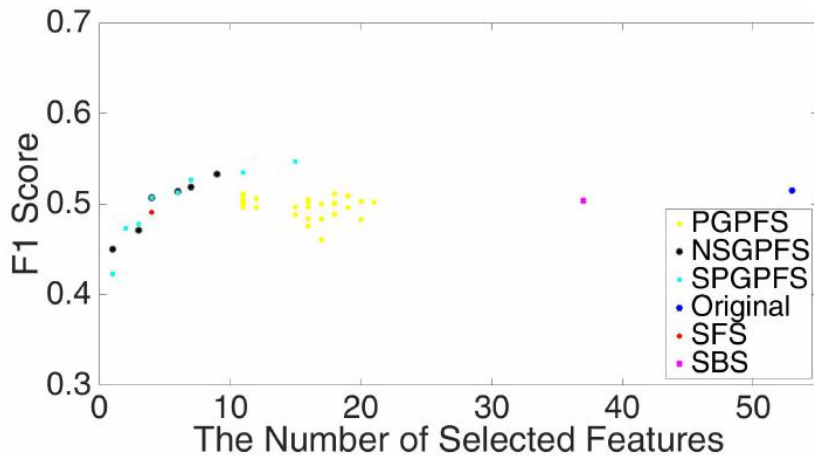


(c) Based on multi-layer perceptron.

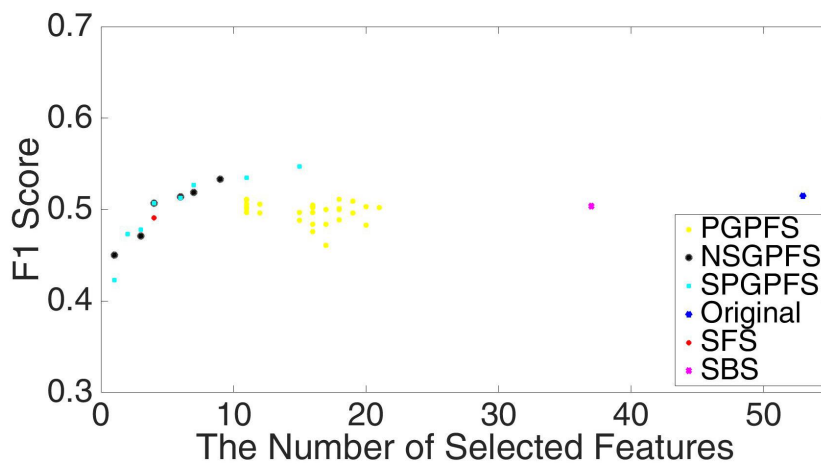
Figure 5.8: Performance ( $F_1$  score) of feature selection methods on the Weizmann test set.



(a) Based on decision tree.



(b) Based on Naïve Bayes.



(c) Based on multi-layer perceptron.

Figure 5.9: Performance ( $F_1$  score) of feature selection methods on the Pascal test set.

Table 5.4: Statistical results on Weizmann test set (s represents seconds; ↓, ↑ means the result is lower or higher than that of the original feature set; for NSGP-FS or SPGP-FS, YES (or NO) means its solutions dominate (do not dominate) the reference methods' solutions, i.e. the original feature set / SFS solution / SBS solution / the PGP-FS's "average solution").

Classifier	Feature Set	Test Accuracy(%)	Test $F_1$ Score	Feature Number	Test Time Per Image(s)
DT	Original	68.90	0.530	53	0.029
	SFS	68.25↓	0.536↑	11 ↓	0.016↓
	SBS	68.07↓	0.518↓	45↓	0.021↓
	PGP-FS	$69.89 \pm 1.93$ ↑	$0.545 \pm 0.005$ ↑	$14.27 \pm 1.34$ ↓	$0.017 \pm 0.002$ ↓
	NSGP-FS	YES/YES/YES/YES			$0.016 \pm 0.002$ ↓
	SPGP-FS	YES/YES/YES/YES			$0.016 \pm 0.002$ ↓
NB	Original	71.52	0.568	53	0.753
	SFS	73.01↑	0.586 ↑	11 ↓	0.197↓
	SBS	71.35↓	0.561↓	45 ↓	0.683↓
	PGP-FS	$71.00 \pm 0.48$ ↓	$0.570 \pm 0.004$ ↑	$14.27 \pm 1.34$ ↓	$0.217 \pm 0.019$ ↓
	NSGP-FS	YES/YES/YES/YES			$0.143 \pm 0.056$ ↓
	SPGP-FS	YES/YES/YES/YES			$0.133 \pm 0.046$ ↓
MLP	Original	76.61	0.620	53	0.650
	SFS	75.65↓	0.596↓	11↓	0.077 ↓
	SBS	73.92↓	0.586↓	45 ↓	0.457↓
	PGP-FS	$71.58 \pm 1.91$ ↓	$0.567 \pm 0.009$ ↓	$14.27 \pm 1.34$ ↓	$0.084 \pm 0.009$ ↓
	NSGP-FS	YES/YES/YES/YES			$0.061 \pm 0.019$ ↓
	SPGP-FS	YES/YES/YES/YES			$0.056 \pm 0.016$ ↓

This confirms that the multi-objective methods, NSGP-FS and SPGP-FS, can produce smaller solutions than other methods, which confirms the claim made from the training results in Section 5.5.2.

Based on the two tables, the SFS and SBS solutions cannot consistently improve the performance of standard classifier based segmentation. For example, on the Weizmann dataset, the SFS and SBS solutions reduce the multi-layer perceptron's performance compared with using the original feature set. This may be because the best first search method is a greedy

Table 5.5: Statistical results on Pascal test set (s represents seconds; ↓, ↑ means the result is lower or higher than that of the original feature set; for NSGP-FS or SPGP-FS, YES (or NO) means its solutions dominate (do not dominate) the reference methods' solutions, i.e. the original feature set / SFS solution / SBS solution / the PGP-FS's "average solution").

Classifier	Feature Set	Test Accuracy(%)	Test $F_1$ Score	Feature Number	Test Time Per Image(s)
DT	Original	77.10	0.438	53	0.072
	SFS	78.19↑	0.476↑	4↓	0.043↓
	SBS	76.54↓	0.451↑	37 ↓	0.067↓
	PGP-FS	79.74 ± 1.09 ↑	0.458 ± 0.008 ↑	15.67 ± 3.35 ↓	0.059 ± 0.006 ↓
	NSGP-FS	YES/YES/YES/YES			0.062 ± 0.014 ↓
	SPGP-FS	YES/YES/YES/YES			0.060 ± 0.013 ↓
NB	Original	83.28	0.466	53	2.288
	SFS	84.24↑	0.487 ↑	4↓	0.286↓
	SBS	84.89↑	0.511↑	37↓	2.040↓
	PGP-FS	84.58 ± 0.33 ↑	0.482 ± 0.004 ↑	15.67 ± 3.35 ↓	0.954 ± 0.175 ↓
	NSGP-FS	YES/YES/YES/YES			0.641 ± 0.254 ↓
	SPGP-FS	YES/YES/YES/YES			0.608 ± 0.212 ↓
MLP	Original	83.59	0.515	53	1.867
	SFS	81.88↓	0.491 ↓	4 ↓	0.151↓
	SBS	83.45↓	0.504↓	37 ↓	1.229↓
	PGP-FS	80.07 ± 1.64 ↓	0.498 ± 0.011 ↓	15.67 ± 3.35 ↓	0.362 ± 0.095 ↓
	NSGP-FS	YES/YES/YES/YES			0.236 ± 0.091 ↓
	SPGP-FS	YES/YES/YES/YES			0.220 ± 0.074 ↓

search method that does not examine all possible subsets, so it is not guaranteed to find the optimal feature subset. Similarly, PGP-FS solutions also have varying performance, e.g. performance improved for decision tree, while reduced for the multi-layer perceptron on the Pascal dataset. In contrast, NSGP-FS and SPGP-FS solutions can dominate not only the SFS, SBS solutions and the original feature set, but the PGP-FS's "average solution" on both datasets for all three classifiers based segmentation tasks.

The analyses indicate that all the three proposed methods can reduce

the number of features, especially the multi-objective methods (NSGP-FS and SPGP-FS). However, PGP-FS has limited ability to provide feature subsets that can improve the performance of the given figure-ground segmentation tasks, while the two proposed multi-objective methods are more capable in searching for effective feature subsets.

#### 5.5.4 NSGP-FS vs SPGP-FS

Figure 5.7, Figure 5.8 and Figure 5.9 are studied to compare NSGP-FS with SPGP-FS in two aspects, the dominance of each other's solutions and the ability of preserving diversity of front solutions. Diversity is an important factor for Pareto-based multi-objective optimisation techniques. It refers to how well the solutions are distributed on the Pareto front. A good diversity (evenly distributed along the Pareto front) provides users a better idea of the Pareto front shape that shows different options of trade-offs between objectives, which increases the chance of finding a desired solution [107]. Table 5.6 provides the number of front members on the training stage (derived from Figure 5.7) and the test stage (derived from Figure 5.8 and Figure 5.9).

According to the training performance in Figure 5.7, most NSGP-FS solutions achieve similar accuracies to SPGP-FS solutions that have the same number of features, as most of their solutions overlap (red and black points). Figure 5.8 and Figure 5.9 shows the test performance based on the  $F_1$  score. According to the two figures, no absolute claim that NSGP-FS solutions dominate SPGP-FS solutions can be made, as there are NSGP-FS solutions that dominate SPGP-FS solutions for certain cases and vice versa. It suggests that NSGP-FS and SPGP-FS are equally qualified in searching effective feature subsets for the segmentation problems in this work.

In terms of the diversity, Figure 5.7 based on the training performance shows that NSGP-FS produces a better distributed front and has more front members than SPGP-FS (e.g. 14 for NSGP-FS and 11 for SPGP-FS on the Weizmann set shown in Table 5.6). Moreover, Figure 5.8 and Fig-

ure 5.9 based on the test performance are more complicated and provide inconsistent results for both methods. For example, NSGP-FS solutions are more evenly distributed for DT on the Weizmann test set (Figure 5.8a), while SPGP-FS solutions have higher diversity for MLP on the Pascal test set (Figure 5.9c). However, further information in Table 5.6 shows that NSGP-FS has a higher average number of front solutions than SPGP-FS on the Weizmann test set. Therefore, it indicates that NSGP-FS is better at keeping diversity on the Pareto front than SPGP-FS. This may be because NSGA-II tends to emphasise the edge solutions and produce solutions along the whole frontier [44], while SPEA2 focuses more on solutions in the middle front region.

Table 5.6: Comparison of NSGP-FS against SPGP-FS in the number of front solutions (# means number).

Dataset	Method	#Front Solution (Training)	#Front Solution (Test)		
			Classifier		Average
Weizmann	NSGP-FS	14	DT	7	7
			NB	6	
			MLP	7	
	SPGP-FS	11	DT	5	6
			NB	8	
			MLP	6	
Pascal	NSGP-FS	13	DT	7	8
			NB	10	
			MLP	6	
	SPGP-FS	12	DT	6	8
			NB	9	
			MLP	8	



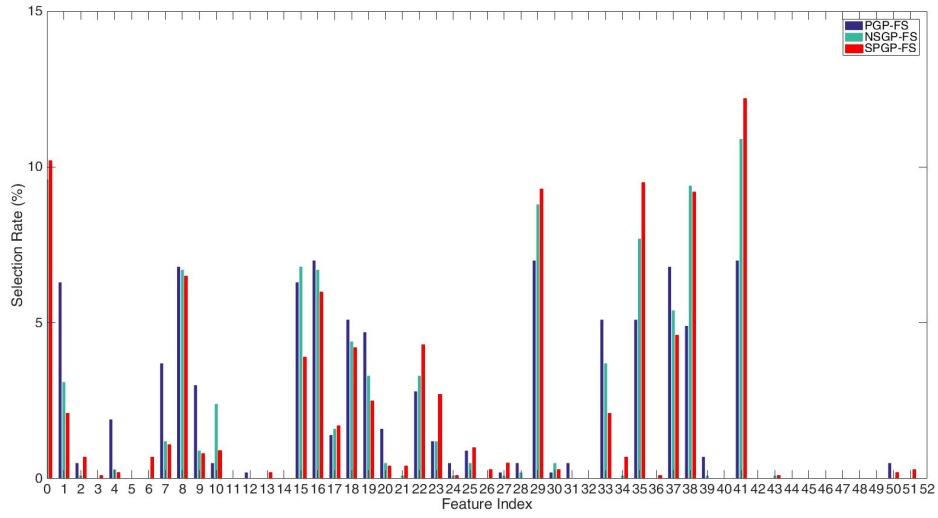
### 5.5.5 Analysis on Selected Features

Figure 5.10 shows the selection rate of each feature in the original feature set based on the proposed methods. The selection rate is calculated as follows. Firstly, for NSGP-FS and SPGP-FS, all the front solutions of 30 runs are collected; for PGP-FS, the best solutions of 30 runs are collected. Then, the number of each feature that are selected by the collected solutions is counted, based on which the occurrence rate of each feature is calculated for each method.

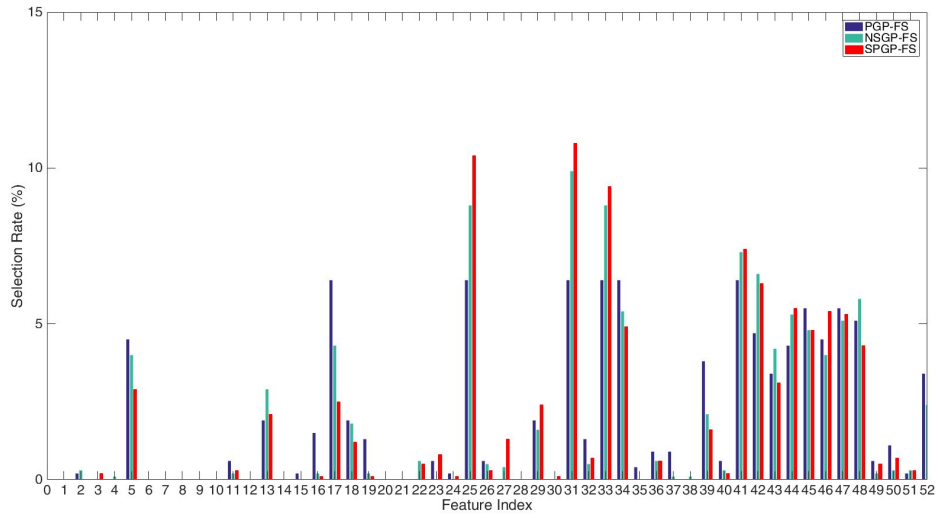
According to Figure 5.10, features ( $f_0, f_8, f_{15}, f_{16}, f_{29}, f_{35}, f_{37}, f_{38}, f_{41}$ ) have higher selection rates on the Weizmann dataset than others, while features ( $f_{25}, f_{31}, f_{33}, f_{34}, f_{41}, f_{42}, f_{44}, f_{45}, f_{46}, f_{47}, f_{48}$ ) are more often selected on the Pascal dataset. Based on Figure 5.5, features with indexes from 0 to 39 are Gabor features that can capture edge information;  $f_{40}, f_{41}$  and  $f_{42}$  are colour features; the remaining features (indexed from 43 to 52) are grayscale statistical features. This shows that for Weizmann images, edge and colour features are more useful than local statistical features, while for Pascal images, the frequently selected features cover all the three types of features. Considering that Pascal images are highly varied, all the edge, colour and local statistical features are necessary to achieve accurate segmentation. In contrast, on the Weizmann images, the objects of interest are horses (with distinctive curves as edges), and the backgrounds are mainly trees, grass and/or sky (with distinctive colour). Therefore, the edge and colour features may be more reliable than local statistical features for Weizmann images.

## 5.6 Chapter Summary

Three feature selection methods based on GP were proposed in this chapter, two of which (i.e. NSGP-FS and SPGP-FS) are multi-objective and one of which (i.e. PGP-FS) is single-objective. The contributions of this



(a) On Weizmann dataset.



(b) On Pascal dataset.

Figure 5.10: Selected features based on the proposed methods.

work are twofold. Firstly, this is the first time to investigate the ability of GP to select effective feature subsets for image segmentation tasks. Even

though GP has been studied for feature selection in existing works, these are mainly for general classification problems. GP has not been sufficiently investigated in the field of feature selection for figure-ground image segmentation. Secondly, feature selection is a multi-objective task, maximizing the effectiveness of the selected feature subset and minimizing the number of selected features. Compared with the existing GP based feature selection works that rarely consider the number of features, the proposed methods take both objectives into account. The feature subsets produced by the proposed methods were evaluated to reflect the methods' capacity in searching for effective feature subsets for complex segmentation tasks (images with high variations).

Two benchmark selection methods (sequential forward selection and sequential backward selection) were selected as the reference methods. Two benchmark datasets with high variations, the Weizmann and Pascal datasets, were employed to test the proposed and the reference methods. The results show that all the three proposed methods can generate feature subsets with a much lower number of features than the original set (53 features), and the multi-objective methods (NSGP-FS and SPGP-FS, especially SPGP-FS) can further reduce the number of features selected while improving the segmentation performance. However, PGP-FS has limited ability to provide effective feature subsets and produce unstable performance. Furthermore, the two multi-objective methods are more capable in searching effective feature subsets, whose solutions dominate both SFS and SBS solutions. When comparing NSGP-FS with SPGP-FS, they are equally qualified in searching effective feature subsets, yet NSGP-FS is better at keeping diversity among the nondominated solutions than SPGP-FS.

In addition to feature selection, feature construction is a process of transforming a given set of features to a new set of more powerful features. A new feature is usually created by a function of the original features and operators. As GP can utilise complex and variable-length representations, such as trees, in which various kinds of operators can be used to combine

input features into linear or non-linear forms, it is well-suited for feature construction. Therefore, the next chapter will investigate feature construction using GP.

## Chapter 6

# GP for Feature Construction in Figure-ground Segmentation

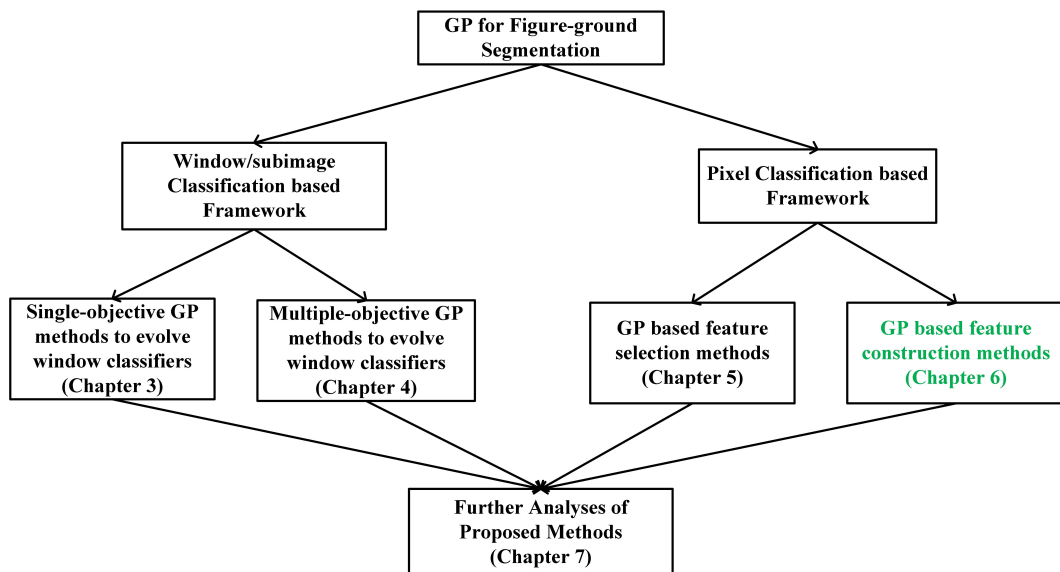


Figure 6.1: The overall structure of contributions.

New feature selection methods using GP were designed in Chapter 5. The results show that the selected feature subsets of the proposed methods achieve better segmentation performance with a lower number of features

than those of the sequential algorithms and the original feature set.

In addition to feature selection, feature construction is another type of feature manipulation, which is a process of transforming original features to a new set of more powerful features. Considering that GP based feature selection methods achieve promising results, and GP is a well-suited technique for feature construction, this chapter investigates feature construction using GP for figure-ground segmentation.

## 6.1 Introduction

GP is a well-suited technique for feature construction as it can use complex and variable-length representations and has high search ability, and GP's solutions can be interpreted [20]. Even though existing GP based feature construction works have achieved promising results, it is still challenging to design effective fitness functions and develop new methods to avoid constructing features by mixing different types of image features homogeneously in the existing works. Therefore, GP is investigated to construct features for figure-ground segmentation in this chapter.

Note that based on the evaluation of constructed features, there are three branches of feature construction methods, i.e. filter, wrapper and embedded [92]. This chapter only focuses on the filter and wrapper approaches, since the goal of feature construction in this thesis is to construct effective features explicitly, which can be used as inputs of segmentation algorithms to improve the segmentation performance. Since embedded approaches can only conduct implicit feature construction in the process of learning a model for a given problem, they are not considered in this chapter. In addition, based on whether a single feature or multiple features is/are constructed, there are single feature construction (SFC) [28] and multiple feature construction (MFC) methods [129]. Both the single and multiple constructed features consider the interactions of original features, which makes the constructed features more effective for target prob-

lems than the original features.

### 6.1.1 Chapter Goals

The goal of this chapter is to apply GP for the first time to construct high-level features from primitive features to improve the segmentation performance on complex images (e.g. images with high variations). Both single (SFC) and multiple feature construction (MFC) methods are designed. Specifically, a new entropy based fitness function is developed for filter feature construction methods. Techniques, i.e. subtrees (see Section 6.4.1) and cooperative coevolution techniques (see Section 6.4.2), are introduced in GP to design MFC methods. The SFC methods and the subtrees-based MFC method combine different types of primitive features homogeneously together, while the cooperative coevolution technique enables GP to construct features from different types of features separately, and conduct combination of the constructed features simultaneously. It is anticipated that the proposed methods can construct effective features that perform well for segmentation tasks on complex images.

Specific objectives are shown as follows to determine:

1. whether the constructed features can outperform the original features on the given complex segmentation tasks,
2. which of the proposed single or multiple feature construction methods can construct more effective features,
3. how to interpret the constructed features to reveal their effectiveness in distinguishing object and background pixels.

### 6.1.2 Chapter Organisation

The remainder of this chapter is organised as follows. Section 6.2 describes the framework of figure-ground segmentation with feature construction.

Section 6.3 and section 6.4 introduce the proposed single and multiple feature construction methods respectively. In addition, experiment preparations are described in section 6.5, including the function sets and terminal sets. Section 6.6 discusses the results, and section 6.7 analyses the constructed features. Eventually, the summary of this chapter is given in section 6.8.

## 6.2 Overall Structure of GP based Feature Construction

Figure 6.2 displays the framework of the figure-ground segmentation with a feature construction stage. In this chapter, segmentation problems are formulated to pixel classification problems. Compared with the window (or subimage) classification based methods in Chapter 3 and Chapter 4, the pixel based segmentation does not require window related parameters that rely on users' expertise to set up. The framework consists of three major steps. Firstly, a given number of pixels are captured randomly from training images whilst ensuring equal number of class *object* and *background*. Low /middle-level features are extracted for these pixels along with their class labels (class *object* or *background*) to form the training set. Accordingly, the same type of features are extracted for each pixel in test images. Secondly, the training set is used to evolve feature construction functions by GP. Based on the evolved construction functions, original features of each training/testing sample are converted to a new feature(s). Thirdly, the transformed training set is employed to train a standard classifier, e.g. a decision tree, which is then applied to classify the transformed test set. Based on certain evaluation measures and the ground truth, the segmentation results are generated.



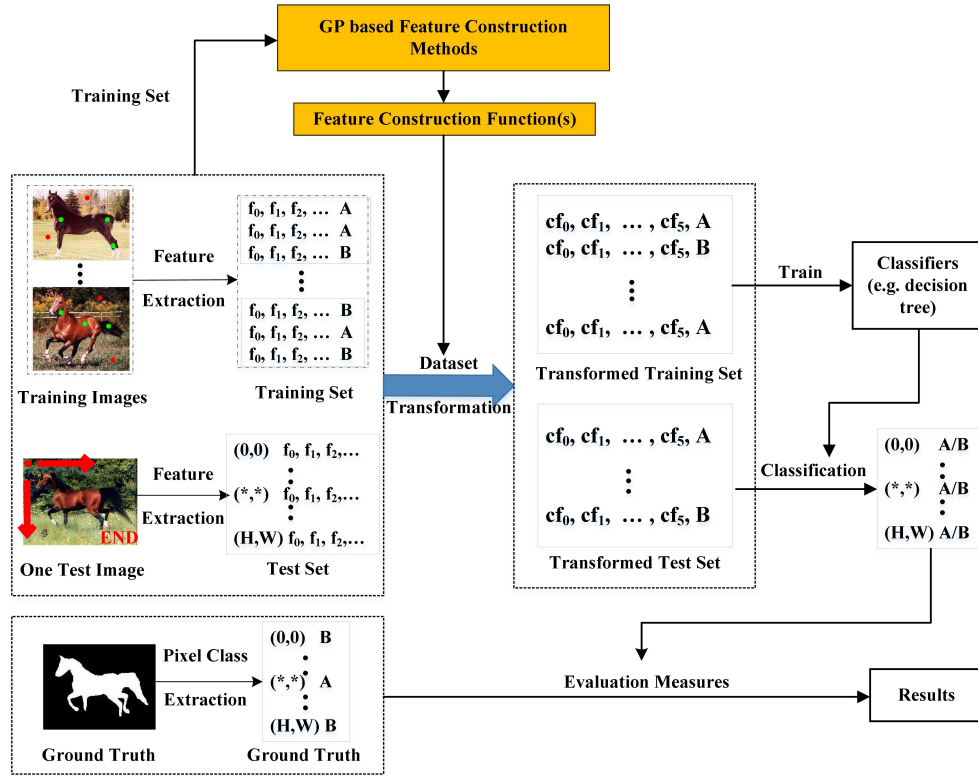


Figure 6.2: Framework of the pixel classification based figure-ground segmentation with feature construction ( $f_0$  is a feature with 0 as its index and so forth;  $(f_0, f_1, f_2, \dots)$  is a feature vector extracted by existing feature descriptors;  $f_c$  is the single constructed feature; A and B represent class *object* and *background* respectively;  $(0,0)$ ,  $(*,*)$  and  $(H,W)$  represent the pixel coordinates; H and W are the test image's height and width).

### 6.3 Proposed Single Feature Construction Methods

Two single feature construction methods are developed, i.e. a filter approach (termed as FSFC – Filter Single Feature Construction) and a wrapper approach (WSFC – Wrapper Single Feature Construction). An entropy based fitness function is designed for FSFC, since it can measure the im-

purity of two classes separated based on the constructed features, which can reflect the effectiveness of the constructed features. In addition, as the classification accuracy is a popular measure for binary classification problems, it is employed as the fitness function for WSFC. Figure 6.3 shows how to construct a single feature based on a derived solution of single feature construction methods. Specifically, the solution can be regarded as a feature construction function. By feeding the required input features to the tree solution, an output value can be generated from the root node, which is used as the single constructed feature of this solution.

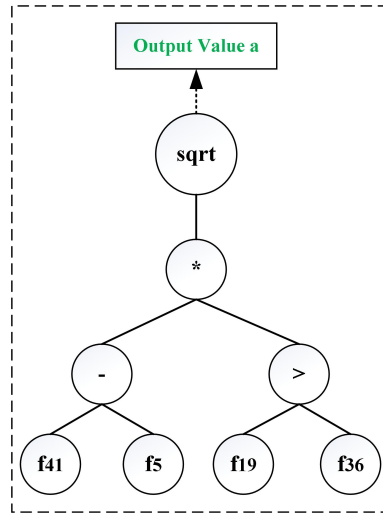
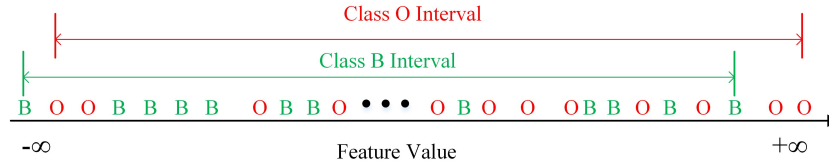


Figure 6.3: Feature construction using an evolved solution of single feature construction methods ( $f_0$  is an input feature, and so forth;  $+$ ,  $-$ ,  $*$ ,  $\%$ ,  $>$ ,  $<$ ,  $\sqrt{\phantom{x}}$  are operators).

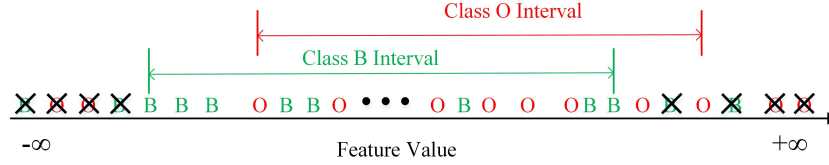
### 6.3.1 Filter Single Feature Construction Method

As segmentation tasks are transformed to binary pixel classification based problems, a “good” feature should be the one that can separate pixels of two classes, i.e. class *object* and *background*, by projecting them in the feature space. As shown in Figure 6.4a, pixels from two classes are commonly

overlapping. For example, the interval (the range of feature values) of class *background* contains object pixels. Since a less overlapping (or less impure) class interval means a better separation, the goodness of constructed features can be reflected by the impurity of the class intervals [127]. Therefore, a novel fitness function is proposed for FSFC that aims to minimise the impurity of the class intervals. Considering that there may exist outliers, which are noisy samples, 0.5% (determined empirically) of its total pixels on both minimum and maximal edges are removed from each class interval (illustrated in Figure 6.4b).



(a) Overlapping class intervals.



(b) Class intervals without outliers.

Figure 6.4: Pixel distribution in the feature space (O and B represent class *object* and *background* respectively;  $\times$  means removing the corresponding pixels).

As a popular entropy measure, Shannon entropy [151] is employed to measure the impurity of the class intervals. Shannon entropy is described as follows. Given discrete probability distributions  $\vec{p} = p_1, p_2, \dots, p_N$ , where the probabilities are all non-negative numbers and  $\sum_{i=1}^N p_i = 1$ , the Shannon entropy of  $\vec{p}$  is shown in Equation 6.1.

$$H(\vec{p}) = - \sum_{i=1}^N p_i * \ln p_i. \quad (6.1)$$

Based on the Shannon entropy, the impurity of one class interval can be calculated by Equation 6.2, where  $C$  is the set of class labels,  $Ic$  represents the interval in the feature space for a specific class  $c$  ( $c \in C$ ), and  $H(Ic)$  means the Shannon entropy of the interval  $Ic$ . Specifically, given a set of training samples (feature vectors with class labels), they are transformed based on the evolved feature construction function. Then in the constructed feature space, for a certain class  $c$ , the class interval  $Ic$  can be determined.  $NPIXS_{Ic}$  refers to the number of pixels that fall into the interval  $Ic$ , while  $NPIXS_{i,Ic}$  refers to the number of pixels whose class labels are class  $i$  ( $i \in C$ ) and who fall into the interval  $Ic$ .

$$H(Ic) = - \sum_{i \in C} p(NPIXS_{i,Ic} | NPIXS_{Ic}) * \ln p(NPIXS_{i,Ic} | NPIXS_{Ic}). \quad (6.2)$$

where  $C = \{c_1, c_2\}$ ,  $Ic = \{Ic_1, Ic_2\}$ . The fitness function (Equation 6.3) is defined as the mean of the two class intervals' entropies. A lower fitness value means a better separation of two classes, which suggests a better constructed feature.

$$fitness = \frac{H(Ic_1) + H(Ic_2)}{2}. \quad (6.3)$$

### 6.3.2 Wrapper Single Feature Construction Method

The fitness evaluation of an individual in the evolution process of the wrapper single feature construction method (WSFC) is shown in Figure 6.5. WSFC uses the classification accuracy as the fitness function. Since the training process is a balanced binary classification problem, the classification accuracy is an effective measure. Specifically, a training set and a validation set are required for measuring individuals' classification accuracy in the evolution process of WSFC (described in section 6.5.4). In Figure 6.5, the whole tree rooted on the root node is employed as a feature construction function, based on which the training and validation samples

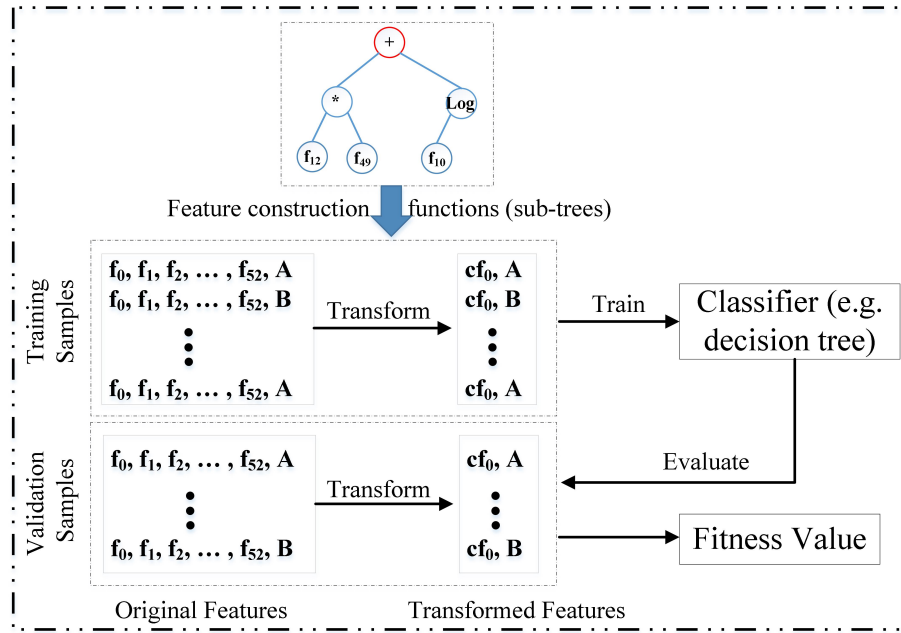


Figure 6.5: Fitness evaluation of an individual in the evolution process of WSFC.

are transformed to new samples. The new training and validation sets are then employed to train and evaluate a classifier (e.g. decision tree), and the classification accuracy is calculated as the fitness value of this individual.

## 6.4 Proposed Multiple Feature Construction Methods

Two multiple feature construction methods are proposed, including a sub-trees based method (SubtMFC) and a coevolutionary method (CoevoMFC). As Ahmed's work [4] discovered that the subtrees rooted on the non-terminal nodes of an evolved tree solution produce useful features for their classification tasks, SubtMFC is developed to construct multiple features from the subtrees of one solution. The cooperative coevolutionary technique is employed, since it allows GP to construct features from differ-

ent types of image features separately, and then combine the constructed features simultaneously. SubtMFC (as well as FSFC and WSFC) conducts homogeneously combination of varied types of primitive features, leading to the interactions of non-related features, which may lower the effectiveness of constructed features. In contrast, CoevoMFC does not allow the interactions across different features.

Both SubtMFC and CoevoMFC are wrapper approaches, and employ the classification accuracy as the fitness function (same as that of WSFC). Training and validation sets (described in section 6.5.4) are required for fitness evaluation. Note that the proposed filter method (FSFC) cannot use the feedback of a learning algorithm for fitness evaluation, which makes it more difficult to achieve an accurate evaluation of solutions.

### 6.4.1 Subtrees-based Multiple Feature Construction

Figure 6.6 shows how to construct features using a subtrees based multiple feature construction (SubtMFC) solution. It can be seen that by feeding the required primitive features to the solution, each subtree can produce an output. The outputs of all subtrees are collected, forming a new constructed feature set. For SubtMFC, constructed feature sets have varied numbers of features, as different solution trees can contain varied numbers of subtrees.

The fitness evaluation of an individual in SubtMFC is shown in Figure 6.7. The subtrees rooted on the non-terminal nodes are employed as feature construction functions, based on which the training and validation samples are transformed to new samples. The new training and validation sets are then employed to train and evaluate a classifier (e.g. decision tree), and the classification accuracy is calculated as the fitness value of this individual.

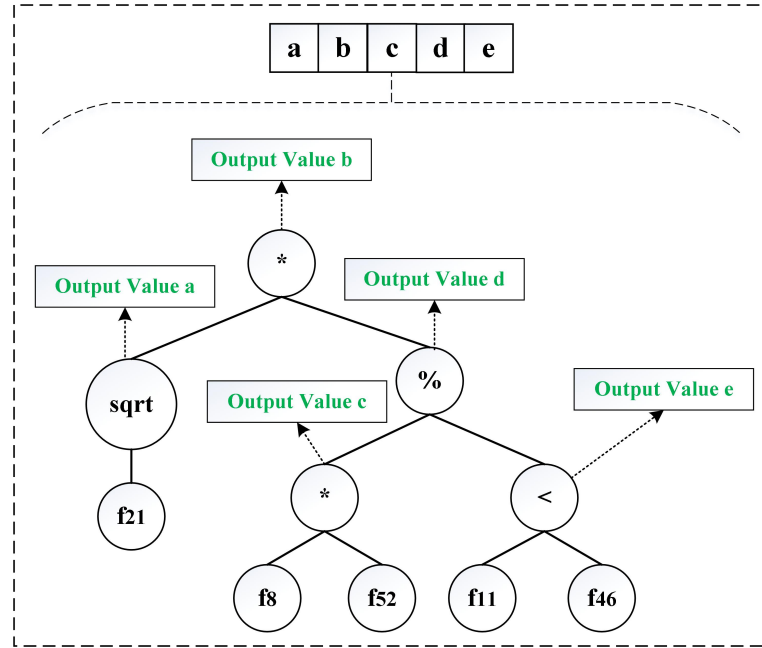


Figure 6.6: Feature construction using evolved solutions of SubtMFC ( $f_0$  is an input feature, and so forth;  $+$ ,  $-$ ,  $*$ ,  $\%$ ,  $>$ ,  $<$ ,  $\text{sqrt}$  are operators).

#### 6.4.2 Coevolution Multiple Feature Construction

The coevolution multiple feature construction (CoevoMFC) method is designed by introducing the cooperative coevolution technique in GP. Algorithm 6 shows the pseudo code of CoevoMFC. Compared with the standard GP, the major difference lies in the initialisation stage and the fitness evaluation stage (highlighted in blue colour). Specifically, a target problem is formulated into multiple subproblems. In the initialisation stage, multiple subpopulations are set up, each of which represents one subproblem. Then the coevolutionary search proceeds in each subpopulation independently, except for the fitness evaluation of each individual [86], which relies on other individuals.

Figure 6.8 shows an example of generating new features based on CoevoMFC solutions. Since there are three general types of image features,

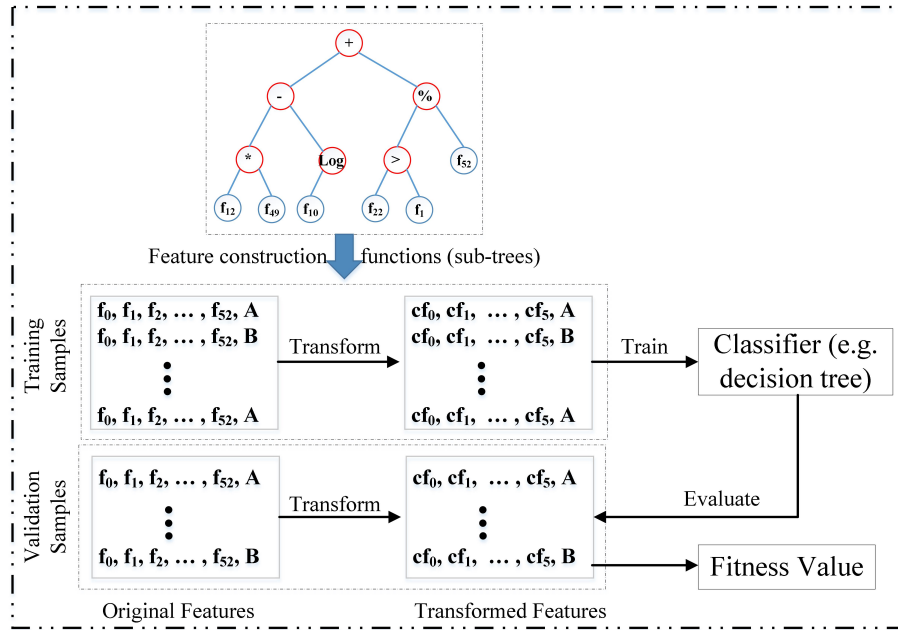


Figure 6.7: Fitness evaluation of one individual in the evolution process of SubtMFC.

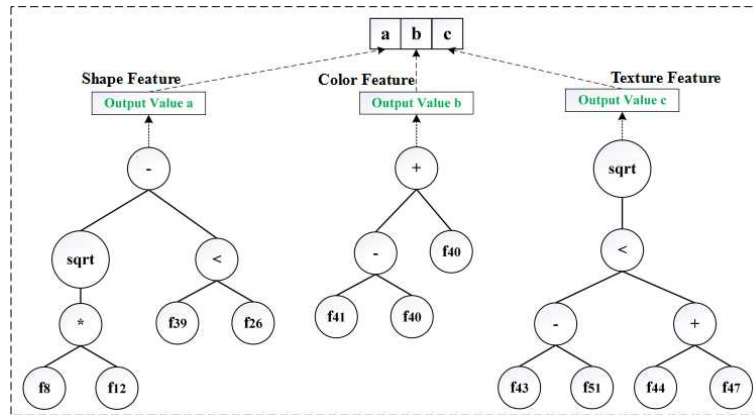


Figure 6.8: Feature construction using a group of evolved solutions of Co-evoMFC ( $f_0$  is an input feature, and so forth;  $+$ ,  $-$ ,  $*$ ,  $\%$ ,  $>$ ,  $<$ ,  $\sqrt{\phantom{x}}$  are operators).

i.e. edge, colour and grayscale statistics features, several feature descrip-



```

input : M: the number of subpopulations;
        G: the maximum number of generations;
        The terminal set, function set and GP parameters (section 6.5).
output: Solutions (feature construction functions).

1 for  $m = 0$  to  $M-1$  do
2   For subpopulation  $P_m$ , determine the subpopulation size  $N_m$ ;
3   Create an initial population of  $N_m$  GP trees at iteration zero ( $P_{m0}$ ) using the Ramped
   half-and-half method;
4 end
5  $g \leftarrow 1$ 
6 while  $g < G - 1$  and the ideal individual (the individual with fitness value == 1.0 ) is not found do
7   for  $m = 0$  to  $M-1$  do
8     Fitness assignment: Individuals from subpopulation  $P_m$  are tested by grouping them
     with the fittest individuals in the other subpopulations from the previous generation
     and assessing their joint fitness (Figure 6.9);
9     Create  $P_{m(g+1)}$  from  $P_{mg}$ :
10    begin
11      Set  $P_{m(g+1)}$  empty;
12      for  $i = 0$  to  $N_m - 1$  do
13        Conduct crossover, mutation or reproduction operations based on the
        individuals from  $P_{mg}$ ;
14        Add the child/children to  $P_{m(g+1)}$ ;
15      end
16    end
17  end
18   $g \leftarrow g + 1$ ;
19 end
20 Return the M best individuals from M subpopulations.

```

**Algorithm 6:** Pseudo-code of CoevoMFC (GP procedure is in red and coevolution procedure is in blue).

tors are selected to extract features that can cover all these three types of features (described in section 6.5.1). Therefore, CoevoMFC formulates a feature construction problem consisting of three subproblems, where each subproblem can handle one type of feature separately. Thus, a group of three feature construction functions are generated in each GP run, and then the output of the root node of each function is collected to form a new feature set with a fixed size of three features.

Figure 6.9 shows the fitness evaluation of an individual, which is individual  $j$  in the sub-population  $i$ . Firstly, one representative individual

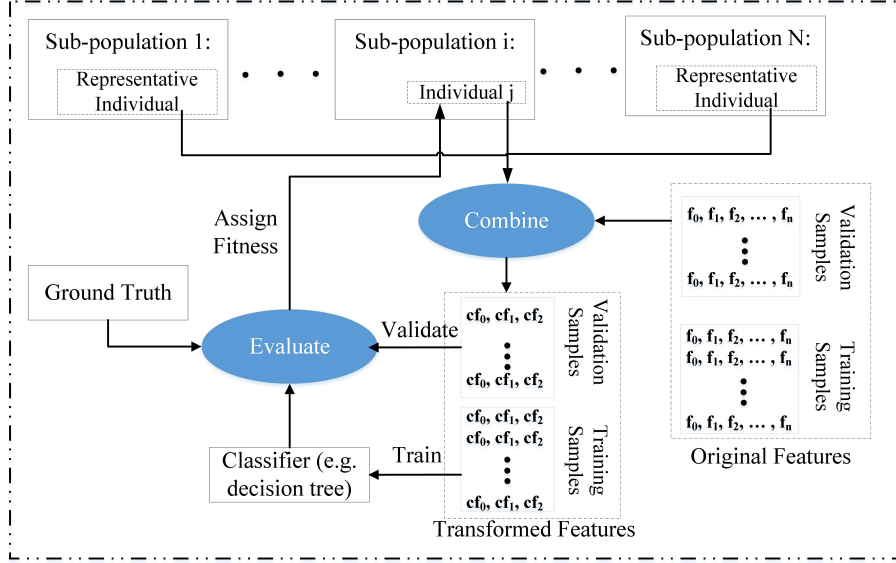


Figure 6.9: Fitness evaluation of an individual in the evolution process of CoevoMFC ( $N$  is the number of subpopulations;  $f_0$  and  $cf_0$  means a feature and a constructed feature with the index 0 respectively).

from each of the remaining subpopulations is selected to form the context of the target individual. The fittest individuals in other subpopulations from the previous generation are selected as the representative individuals. Secondly, the selected individuals are combined with the target individual, which are now three feature construction functions. With the input of primitive features, the training and validation datasets are transformed based on the feature construction functions (three high-level features are constructed for each training and validation sample). Thirdly, a classifier is trained and evaluated using the transformed training and validation sets to produce the classification accuracy, which is assigned to the target individual.

## 6.5 Experiment Preparations

### 6.5.1 Terminal Set

As there are three subpopulations in CoevoMFC, each subpopulation takes one type of image features as its terminal set (edge features for subpopu-

lation 0; colour features for subpopulation 1; grayscale statistics for subpopulation 2). In contrast, for SubtMFC and two SFC methods, as there is only one subpopulation, the terminal set is the same, which contains all the three types of image features. Several feature descriptors are selected to extract features (shown in Figure 6.10), which are the same as those in Chapter 5 (see Section 5.2.3 for details).

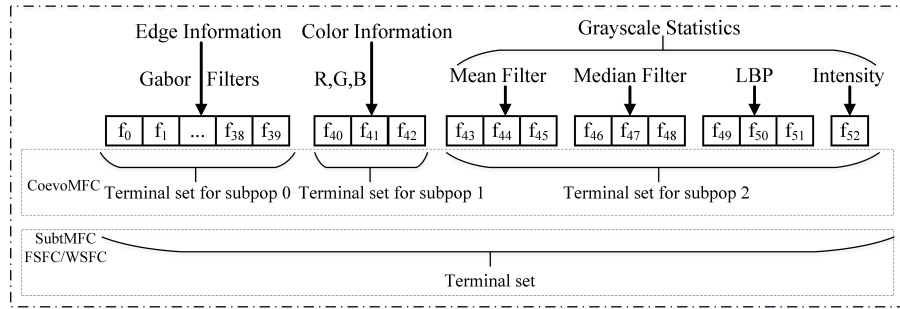


Figure 6.10: Terminal set ( $f_0$  is a feature with 0 as its index and so forth; subpop represents subpopulation).

### 6.5.2 Function Set

Table 6.1 displays the function set, which consists of seven functions: the first five are standard mathematical functions and the last two are relational functions. These functions are selected, since they are all computationally efficient, and can realise both linear and non-linear combinations of input features.

### 6.5.3 Other GP Settings

Major GP parameter settings are shown in Table 6.2, and the setting of the population size is as follows. Figure 6.10 shows that subpopulations 0, 1 and 2 of CoevoMFC take edge features (40 dimensions), colour features (3 dimensions) and grayscale statistics (9 dimensions) as inputs respectively. Since a larger number of input features can form a larger search

Table 6.1: Function set of GP based feature construction methods.

Function Name	Definition	Function Name	Definition
$+(a_1, a_2)$	$a_1 + a_2$	$-(a_1, a_2)$	$a_1 - a_2$
$*(a_1, a_2)$	$a_1 * a_2$	$\%(a_1, a_2)$	$\begin{cases} a_1/a_2 & \text{if } a_2 \neq 0 \\ 0 & \text{if } a_2 == 0 \end{cases}$
$\text{sqrt}(a_1)$	$\text{Sqrt} a_1 $		
$>(a_1, a_2)$	$\begin{cases} a_1 & \text{if } a_1 > a_2 \\ a_2 & \text{if otherwise} \end{cases}$	$<(a_1, a_2)$	$\begin{cases} a_1 & \text{if } a_1 < a_2 \\ a_2 & \text{if otherwise} \end{cases}$

space, a higher population size is often necessary to find optimal solutions. Therefore, for CoevoMFC, the sizes of three subpopulations are set to 512, 32 and 256 respectively based on the number of input features that each subpopulation handles. However, higher population sizes can lead to higher computation/time cost. Moreover, higher population sizes cannot guarantee better solutions. As long as the population size exceeds a critical size for GP methods, effective search can be ensured. Therefore, for SubtMFC and the two single construction methods, the size of the single subpopulation is set to 512 rather than 800 (the addition of subpopulation sizes for CoevoMFC). This is because the size of 512 is sufficient for the feature construction tasks based on initial experiments, in which significant improvements on the segmentation performance of constructed features were not achieved by setting the population size to 800.

Table 6.2: GP parameter settings of GP based feature construction methods.

Parameter	Setting	Parameter	Setting
Generation	51	Initialisation	Ramped half-and-half
Crossover Rate	0.80	Mutation Rate	0.19
Reproduction Rate	0.01	Maximum Depth	10

### 6.5.4 Experiment Design

Two standard image datasets, Weizmann horse database [22, 24] and Pascal aeroplane dataset [53] are used in this chapter (described in Section 2.5.1). The dataset processing (e.g. the splitting of a dataset to training, validation and test sets) is the same as that in Chapter 5 (see Section 5.4.2).

The outputs of the solutions (trees) derived by the proposed methods are normalised to the range  $[0, 1]$  using the sigmoid function, and the normalised values are regarded as the constructed features. This is because the original features are in  $[0, 1]$ , thus it is necessary to make the constructed features compatible with the original features. Specifically, the sigmoid function is employed, since it can transform input numeric values to the range  $[0, 1]$  monotonically. Four standard classifiers, i.e. J48 (a decision tree based classifier), NB (Naïve Bayes), PART (a rule based classifier) and SMO (Sequential Minimal Optimisation), from the Weka package [171] are selected to evaluate the constructed features, since they are popularly used for classification problems. All GP related experiments are run 30 independent times, and the results are the average of the 30 best solutions (one single best solution from each run). The  $F_1$  score (Equation 2.2) is utilised to evaluate segmentation results.

## 6.6 Results and Discussions

Table 6.3 and Table 6.4 displays the results of the proposed single and multiple construction methods respectively (the training performance in segmentation accuracy and the test performance in  $F_1$  score). The constructed features and the combined features (constructed features and the original features) are compared with the original features. The comparisons are based on a statistical significance test technique: the one sample median test [58] at the significance level 5%, which can test whether the median of a sample set of values is the same as a hypothesised value. Since the

proposed methods are indeterministic, thus they are run multiple times producing multiple independent feature sets, while there is only one original feature set.

### 6.6.1 Single Feature Construction Methods

Table 6.3: Results of FSFC and WSFC (#F means the number of features; Original means the original features; CF and Combined stands for the single constructed feature and the combined features (a constructed feature + the original features) respectively;  $\uparrow$ ,  $\downarrow$  or  $=$  mean significantly better, worse or similar, compared with original features).

	Method (#F)	Training Accuracy (%)				Test $F_1$ score			
		J48	NB	PART	SMO	J48	NB	PART	SMO
Weizmann	Original(53)	69.84	72.38	76.98	79.76	0.544	0.565	0.566	0.662
	FSFC	68.30 $=$	62.23 $\downarrow$	68.32 $\downarrow$	57.40 $\downarrow$	0.498 $\downarrow$	0.453 $\downarrow$	0.499 $\downarrow$	0.410 $\downarrow$
	CF(1)	$\pm 5.24$	$\pm 5.42$	$\pm 5.26$	$\pm 5.38$	$\pm 0.096$	$\pm 0.072$	$\pm 0.096$	$\pm 0.090$
	FSFC	98.02 $\uparrow$	72.46 $=$	84.60 $\uparrow$	80.28 $=$	0.539 $=$	0.569 $=$	0.554 $\downarrow$	0.651 $\downarrow$
	Combined(54)	$\pm 0.15$	$\pm 0.20$	$\pm 1.45$	$\pm 0.15$	$\pm 0.002$	$\pm 0.001$	$\pm 0.005$	$\pm 0.001$
	WSFC	76.51 $\uparrow$	74.72 $\uparrow$	76.09 $=$	75.82 $\downarrow$	0.599 $\pm$	0.590 $\pm$	0.596 $\pm$	0.601 $\pm$
	CF(1)	$\pm 1.25$	$\pm 1.95$	$\pm 1.51$	$\pm 1.81$	0.014 $\uparrow$	0.018 $\uparrow$	0.017 $\uparrow$	0.020 $\downarrow$
	WSFC	96.13 $\uparrow$	72.21 $=$	83.81 $\uparrow$	80.42 $=$	0.569 $\pm$	0.578 $\pm$	0.589 $\pm$	0.663 $\pm$
Pascal	Combined(54)	$\pm 1.14$	$\pm 0.49$	$\pm 1.46$	$\pm 0.30$	0.009 $\uparrow$	0.004 $\uparrow$	0.020 $\uparrow$	0.002 $=$
	Original(53)	75.44	83.56	79.60	83.01	0.442	0.475	0.475	0.517
	FSFC	72.85 $\downarrow$	67.60 $\downarrow$	72.86 $\downarrow$	64.91 $\downarrow$	0.351 $\downarrow$	0.332 $\downarrow$	0.382 $\downarrow$	0.270 $\downarrow$
	CF(1)	$\pm 6.29$	$\pm 6.13$	$\pm 6.27$	$\pm 7.15$	$\pm 0.072$	$\pm 0.061$	$\pm 0.071$	$\pm 0.079$
	FSFC	96.16 $\uparrow$	80.56 $\downarrow$	87.56 $\uparrow$	85.21 $\uparrow$	0.455 $\uparrow$	0.473 $=$	0.471 $=$	0.506 $\downarrow$
	Combined(54)	$\pm 0.82$	$\pm 0.10$	$\pm 2.45$	$\pm 0.09$	$\pm 0.006$	$\pm 0.002$	$\pm 0.018$	$\pm 0.000$
	WSFC	83.76 $\uparrow$	83.15 $=$	83.92 $\uparrow$	75.82 $\downarrow$	0.478 $\pm$	0.484 $\pm$	0.478 $\pm$	0.601 $\pm$
	CF(1)	$\pm 0.36$	$\pm 0.43$	$\pm 0.53$	$\pm 1.81$	0.009 $\uparrow$	0.010 $\uparrow$	0.010 $=$	0.020 $\uparrow$
	WSFC	94.97 $\uparrow$	79.83 $\downarrow$	86.76 $\uparrow$	84.45 $\uparrow$	0.447 $\pm$	0.479 $\pm$	0.470 $\pm$	0.514 $\pm$
	Combined(54)	$\pm 1.19$	$\pm 0.08$	$\pm 1.16$	$\pm 0.41$	0.009 $=$	0.001 $=$	0.027 $\downarrow$	0.006 $=$

The results of the proposed single feature construction methods, i.e. filter (FSFC) and wrapper (WSFC) methods, are shown in Table 6.3. In terms of the training performance, the single construction features of FSFC based on different classifiers are generally worse than the original features, while the combined features outperform the original features. As the single constructed feature is only one feature, it may not contain sufficient

distinguishing information for object/background pixels, which leads to worse performance than the original features. In addition, both the single and combined construction features have generally higher or similar training accuracies, compared with those of original features. Therefore, WSFC performs better than FSFC in training.

In terms of the test performance, both the single constructed features and combined features of WSFC generally perform significantly better than or similar to original features on both datasets. For example, compared with the original features on Weizmann dataset, single constructed features of WSFC increase the  $F_1$  score by 0.055, 0.025, 0.030 and -0.061 based on the four classifiers respectively, while the combined features increase performance by 0.025, 0.013, 0.023 and 0.001. In contrast, for FSFC, single constructed features achieve lower  $F_1$  scores on both datasets than original features, while its combined features have varying performance based on different classifiers on the two datasets (perform similar for four out of eight cases; perform worse for three out of eight cases; perform better for the remaining one case). Therefore, WSFC outperforms FSFC on test sets, which confirms the claim on the training sets. This is because WSFC is a wrapper approach that employs the feedback of a classifier to evaluate the constructed features for the classification based segmentation tasks, which tends to be more effective than FSFC that does not rely on any classifier to evaluate the constructed features.

### 6.6.2 Multiple Feature Construction Methods

Table 6.4 displays the results of the proposed multiple feature construction methods, i.e. SubtMFC and CoevoMFC. Compared with the original features using different classifiers, the combined features of the proposed methods have generally better or similar results in both training and test performance. For only the constructed features, they perform worse than original features in the training accuracy; however, they achieve better or similar testing performance in  $F_1$  score based on most classifiers. Note that

Table 6.4: Results of CoevoMFC and SubtMFC (CF and Combined refer to the multiple constructed features and the combined features respectively;  $\uparrow$ ,  $\downarrow$  or  $=$  mean significantly better, worse or similar, compared with original features).

	Method (Number of features)		Training Accuracy (%)				Test $F_1$ score			
			J48	NB	PART	SMO	J48	NB	PART	SMO
Weizmann	Original features(53)		69.84	72.38	76.98	79.76	0.544	0.565	0.566	0.662
	CoevoMFC	CF (3)	78.10 $\pm$ 1.00 $\uparrow$	75.70 $\pm$ 0.89 $\uparrow$	76.96 $\pm$ 0.87 $\downarrow$	76.07 $\pm$ 0.98 $\downarrow$	0.600 $\pm$ 0.014 $\uparrow$	0.595 $\pm$ 0.014 $\uparrow$	0.593 $\pm$ 0.013 $\uparrow$	0.601 $\pm$ 0.011 $\downarrow$
		Combined (56)	96.43 $\pm$ 1.13 $\uparrow$	72.61 $\pm$ 0.33 $\uparrow$	84.95 $\pm$ 1.33 $\uparrow$	80.59 $\pm$ 0.25 $=$	0.572 $\pm$ 0.007 $\uparrow$	0.580 $\pm$ 0.003 $\uparrow$	0.587 $\pm$ 0.013 $\uparrow$	0.664 $\pm$ 0.001 $\uparrow$
	SubtMFC	CF (30 $\pm$ 9.71)	82.61 $\pm$ 2.54 $\uparrow$	77.05 $\pm$ 1.23 $\uparrow$	78.95 $\pm$ 1.89 $=$	78.77 $\pm$ 1.25 $\downarrow$	0.592 $\pm$ 0.019 $\uparrow$	0.616 $\pm$ 0.018 $\uparrow$	0.604 $\pm$ 0.023 $\uparrow$	0.636 $\pm$ 0.020 $\downarrow$
		Combined (83 $\pm$ 9.71)	96.92 $\pm$ 1.06 $\uparrow$	76.06 $\pm$ 1.12 $\uparrow$	85.98 $\pm$ 1.85 $\uparrow$	81.20 $\pm$ 0.41 $\uparrow$	0.568 $\pm$ 0.013 $\uparrow$	0.620 $\pm$ 0.012 $\uparrow$	0.592 $\pm$ 0.016 $\uparrow$	0.668 $\pm$ 0.003 $\uparrow$
	Original features(53)		75.44	83.56	79.60	83.01	0.442	0.475	0.475	0.517
Pascal	CoevoMFC	CF (3)	85.23 $\pm$ 0.76 $\uparrow$	83.17 $\pm$ 0.40 $=$	84.38 $\pm$ 0.74 $\uparrow$	83.42 $\pm$ 0.47 $\uparrow$	0.488 $\pm$ 0.011 $\uparrow$	0.493 $\pm$ 0.009 $\uparrow$	0.480 $\pm$ 0.012 $\uparrow$	0.493 $\pm$ 0.012 $\downarrow$
		Combined (56)	95.74 $\pm$ 1.17 $\uparrow$	79.94 $\pm$ 0.08 $\downarrow$	88.01 $\pm$ 1.35 $\uparrow$	84.40 $\pm$ 0.31 $\uparrow$	0.446 $\pm$ 0.008 $\uparrow$	0.481 $\pm$ 0.001 $\uparrow$	0.470 $\pm$ 0.019 $=$	0.522 $\pm$ 0.007 $\uparrow$
	SubtMFC	CF (29 $\pm$ 9.03)	86.65 $\pm$ 1.09 $\uparrow$	83.28 $\pm$ 0.54 $=$	84.44 $\pm$ 0.52 $\uparrow$	83.86 $\pm$ 0.27 $\uparrow$	0.461 $\pm$ 0.012 $\uparrow$	0.506 $\pm$ 0.007 $\uparrow$	0.471 $\pm$ 0.016 $=$	0.502 $\pm$ 0.007 $\downarrow$
		Combined (82 $\pm$ 9.03)	95.91 $\pm$ 1.22 $\uparrow$	82.40 $\pm$ 0.45 $=$	87.98 $\pm$ 1.57 $\uparrow$	84.53 $\pm$ 0.27 $\uparrow$	0.440 $\pm$ 0.010 $=$	0.509 $\pm$ 0.005 $\uparrow$	0.469 $\pm$ 0.024 $=$	0.524 $\pm$ 0.005 $\uparrow$
	Original features(53)		75.44	83.56	79.60	83.01	0.442	0.475	0.475	0.517
	Original features(53)		75.44	83.56	79.60	83.01	0.442	0.475	0.475	0.517

$F_1$  score is more reliable than the segmentation accuracy due to the unbalanced objects/backgrounds on the test images. Moreover, the constructed features (3 for CoevoMFC; 30 on Weizmann dataset or 29 on Pascal dataset for SubtMFC) are less than the original features (53) in size. The analyses indicate that the constructed features of both MFC methods are more effective and robust than the original features.

When comparing CoevoMFC with SubtMFC in the test performance, the constructed features and combined features achieve similar segmentation performance. For example, on the Pascal dataset, the combined features of CoevoMFC achieve 0.446, 0.481, 0.470 and 0.522 in  $F_1$  for J48, NB, PART and SMO respectively, while those of SubtMFC achieve 0.440, 0.509, 0.469 and 0.524 respectively. However, considering that CoevoMFC



only produces three features, which is much less than those of SubtMFC (around 30 on Weizmann dataset and around 29 on Pascal dataset), CoevoMFC features are more effective.

### 6.6.3 Single Methods vs Multiple Methods

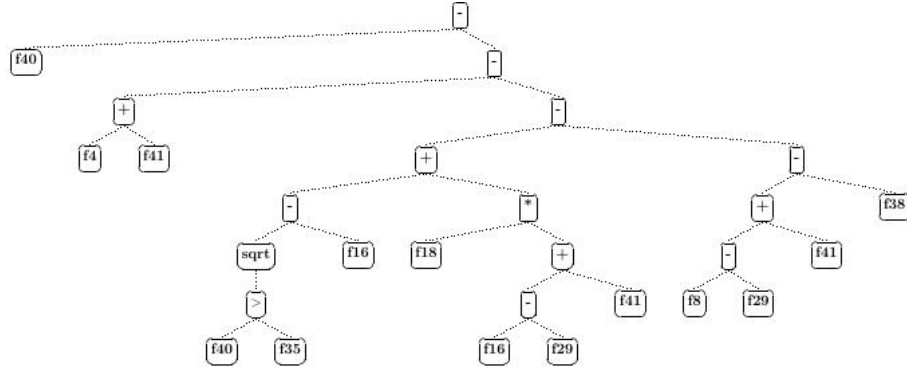
Compared with those of single construction methods (FSFC and WSFC), both the constructed features and the combined features of multiple construction methods (SubtMFC and CoevoMFC) generally achieve better or similar training and testing performance. As WSFC performs better than FSFC based on the analyses in section 6.6.1, WSFC is taken for comparison with the MFC methods.

According to Table 6.3 and Table 6.4, the constructed features of multiple construction methods are relatively more effective than those of the single construction methods. For example, in terms of the training accuracy on the Weizmann dataset, the constructed features of WSFC achieved 76.51% (J48), 74.72% (NB), 76.09% (PART) and 75.82% (SMO), while those of SubtMFC rose by around 6%, 3%, 2% and 3% and those of CoevoMFC increased by approximately 2%, 1%, 1% and 0.2% accordingly. In the test performance on the Weizmann dataset, the constructed features of WSFC achieves 0.599 (J48), 0.590 (NB), 0.596 (PART) and 0.601 (SMO), while those of SubtMFC rose by about -0.031, 0.026, 0.008 and 0.035 and those of CoevoMFC increased by around 0.001, 0.005, -0.003 and 0 respectively. This indicates that the proposed multiple feature construction methods (i.e. SubtMFC and CoevoMFC) are better than the single feature construction methods (FSFC and WSFC) in constructing effective features for the given segmentation problems.

## 6.7 Further Analyses

To reveal the effectiveness of the constructed features, this section analyses the evolved solutions (feature construction functions) of WSFC and CoevoMFC, since WSFC and CoevoMFC perform best among the proposed single and multiple construction methods respectively.

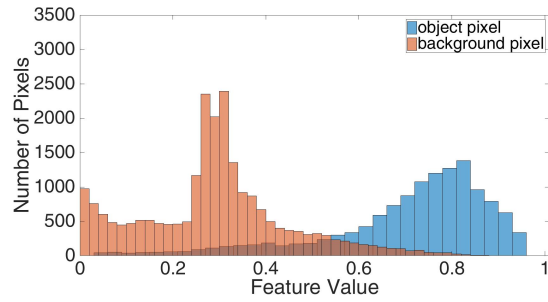
### 6.7.1 Analyses of A Single Feature Construction Solution



(a) One WSFC solution.



(b) A test image.



(c) The constructed feature space based on the solution in Figure 6.11a.

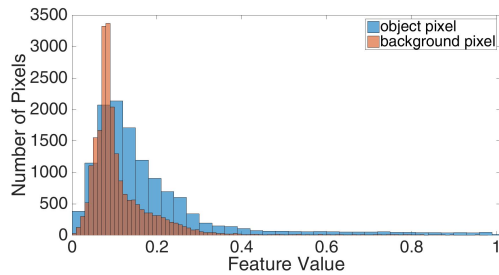
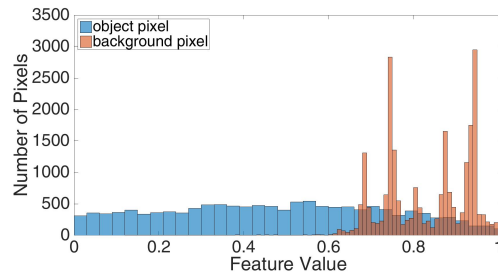
(d) One original feature  $f_{38}$ .(e) One original feature  $f_{40}$ .

Figure 6.11: Distribution of class *object/background* pixels in different feature spaces.

Figure 6.11a shows an example feature construction function derived

by WSFC. Based on the function, original features of a Weizmann test image (Figure 6.11b) are transformed into a constructed feature. Figure 6.11c shows the histogram of object/background regions in the constructed feature space. It can be seen that the majority of object pixels and background pixels can be separated. Specifically, most background pixels fall into the value range of above 0.5 and the object pixels are all within  $[0, 0.5]$ , while only a minority of pixels are misclassified. This reflects that the constructed feature is effective to distinguish object/background pixels.

Figure 6.11d and Figure 6.11e display the object/background histogram in the spaces of two original features, e.g.  $f_{38}$  and  $f_{41}$ , which are used in the feature construction solution (Figure 6.11a). Figure 6.11d and Figure 6.11e show that the majority of background and object pixels are overlapping, so are not as separable as in the constructed feature spaces. This reflects that the single constructed feature is more distinctive than the original ones, which indicates that the proposed WSFC can evolve useful functions to build effective high-level features.

### 6.7.2 Analyses of Multiple Feature Construction Solutions

Figure 6.12 displays one best group of feature construction functions evolved by CoevoMFC. Specifically, Figure 6.12a, Figure 6.12b and Figure 6.12c are the solutions derived based on edge features, colour features and grayscale statistics respectively. Figure 6.12b shows that the feature construction function is to select the feature  $f_{41}$ , and no further operation is implemented. Based on these functions, original features of a Pascal test image (Figure 6.13a) are transformed to constructed features.

Figure 6.13b, Figure 6.13c and Figure 6.13d show the object/background histogram in the constructed feature spaces, in which the majority of object pixels and background pixels can be separated. Specifically, Figure 6.13c shows that feature  $f_{41}$  itself is distinctive, which explains that the solution in Figure 6.12b only selects  $f_{41}$  without additional operations. In contrast, Figure 6.13e and Figure 6.13f display the object/background histogram in

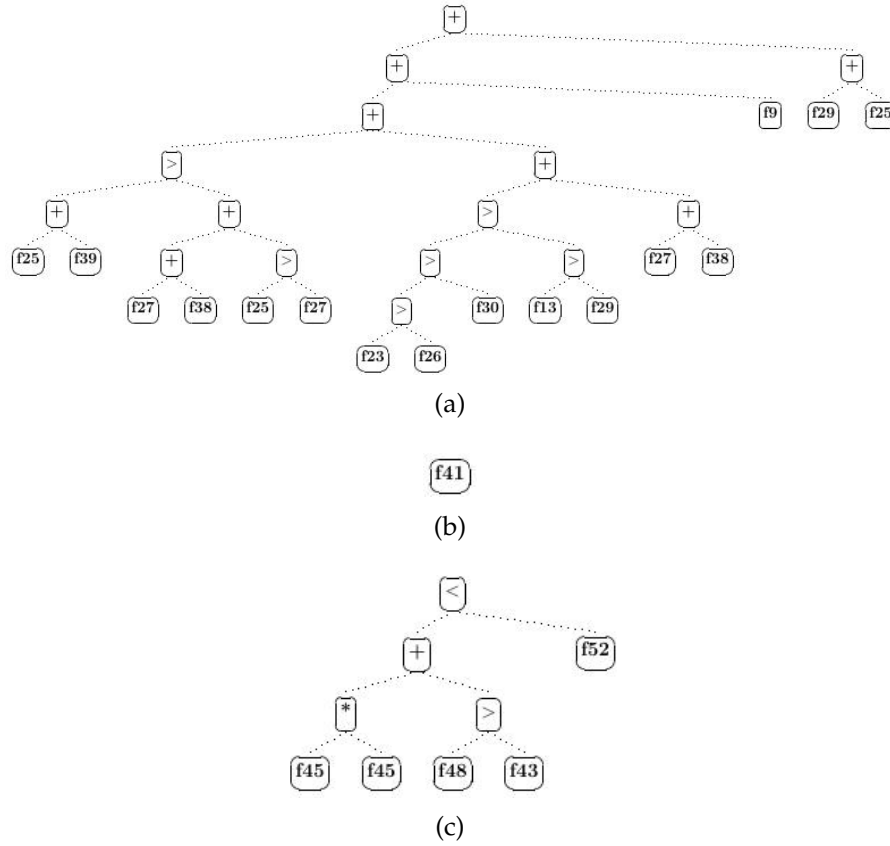


Figure 6.12: Example solutions (feature construction functions) evolved by CoevoMFC ((a), (b) and (c) are the best solutions of subpopulations 0, 1 and 2 respectively).

two original feature spaces, e.g.  $f_{13}$  and  $f_{52}$ , which shows that the majority of background and object pixels are overlapping. This suggests that the constructed features contain more distinguishing information than the original ones for segmentation on the given complex images.

## 6.8 Chapter Summary

This chapter introduced GP for the first time to construct high-level features from primitive image features, aiming to improve the performance

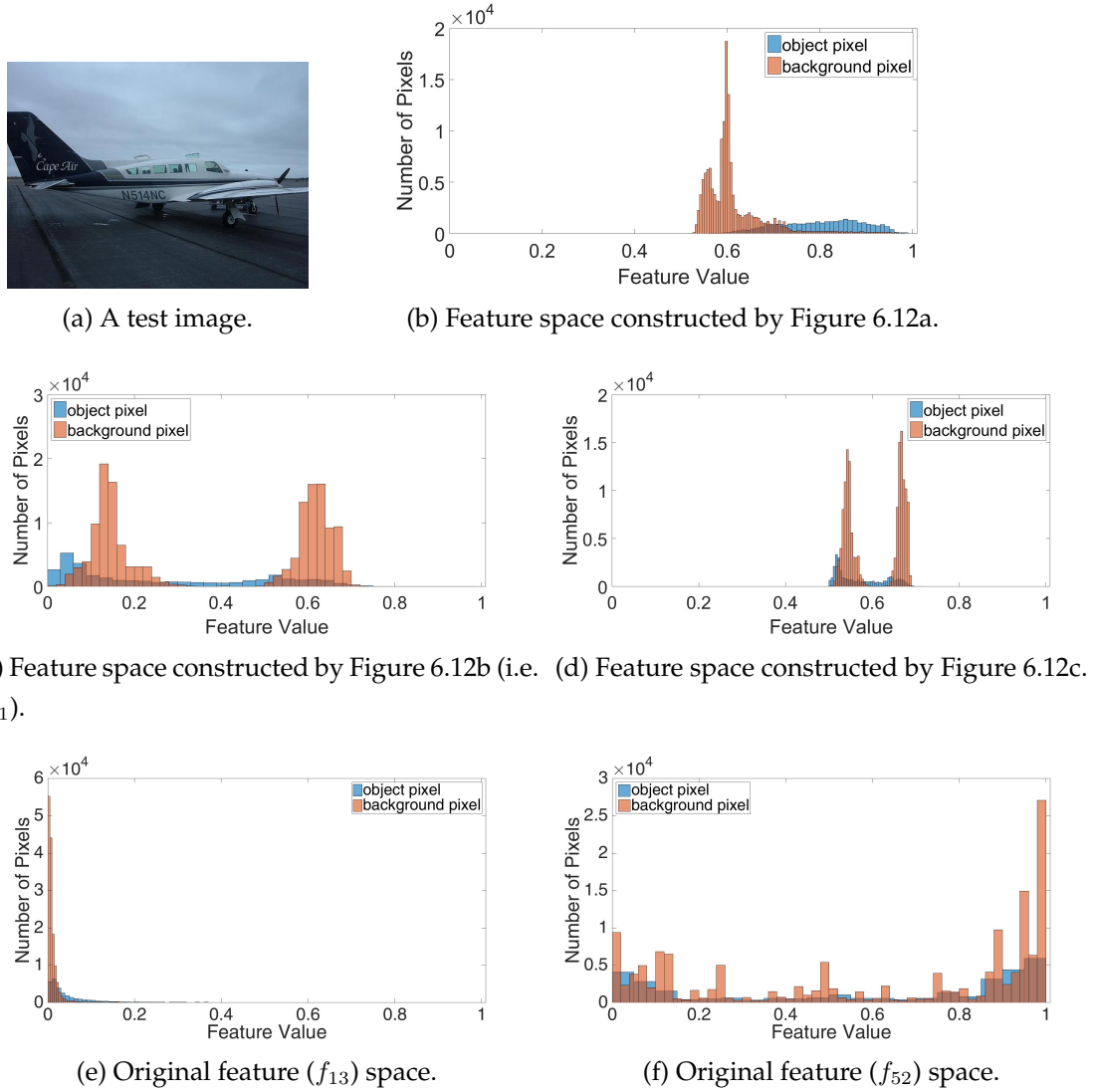


Figure 6.13: Distribution of class *object/background* pixels in different feature spaces.

of subsequent classifiers based segmentation tasks, especially on complex images. Both single and multiple feature construction methods were designed using GP, termed as SFC and MFC. Specifically, the proposed single construction methods include a filter (FSFC) method and a wrapper (WSFC) method, where a novel entropy based fitness function was de-

signed for the FSFC. The subtrees and cooperative coevolution techniques were introduced in GP to develop new multiple feature construction methods. The subtrees technique was selected, as the outputs of subtrees contain useful information. In contrast to the single construction methods and the subtrees based method that conduct homogeneous combination of different types of features, the cooperative coevolution technique was employed, which enables GP to construct features from different types of features separately, and conduct combination of the constructed features simultaneously.

Among the single feature construction methods (FSFC and WSFC), both the single constructed features and combined features of WSFC perform better than the original features for testing; for FSFC, the single constructed features perform worse and the combined features hold varied performance. It reflects that WSFC is better than FSFS for constructing more effective features. Similarly, multiple feature construction methods (SubtMFC and CoevoMFC) constructed features, producing better or similar testing performance in  $F_1$  score based on most classifiers. Moreover, the number of constructed features (3 for CoevoMFC; 30 on Weizmann dataset and 29 on Pascal dataset for SubtMFC) is less than that of the original features (53). This indicates that the constructed features of both MFC methods are more effective and robust than the original features. When comparing between SubtMFC and CoevoMFC, they achieve similar testing performance; however, CoevoMFC only has three constructed features, a much smaller number than those of SubtMFC (30 on Weizmann and 29 on Pascal). Therefore, CoevoMFC is more capable than SubtMFC in generating effective features based on the performance for given segmentation tasks.

GP's potential in different aspects for figure-ground image segmentation tasks has been investigated (Chapters 3-6), including investigating the effectiveness of image features for GP to evolve capable segmentors, introducing multi-objective techniques to GP methods to balance solution func-

tionality and complexity, developing feature selection/construction methods using GP to generate effective features. It would be meaningful to further investigate the proposed GP methods in the field of figure-ground segmentation, e.g. by comparing the proposed methods or by comparing with non-EC segmentation methods.





# Chapter 7

## Further Analyses of Proposed Methods

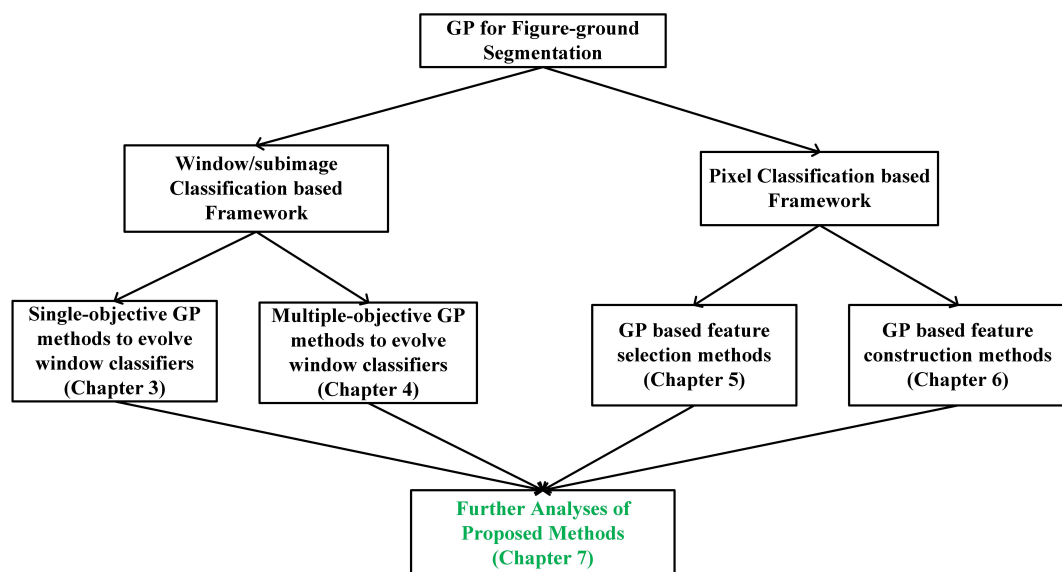


Figure 7.1: The overall structure of contributions.

## 7.1 Introduction

GP's potential in different aspects for figure-ground image segmentation tasks was investigated (Chapters 3-6), including the effectiveness of image features for GP to evolve capable segmentors (Chapter 3), multi-objective GP methods to balance solution functionality and complexity (Chapter 4), GP based feature selection (Chapter 5) and feature construction (Chapter 6). In this chapter, the proposed GP methods will be further investigated in the field of figure-ground segmentation.

Figure-ground segmentation problems are transformed into window classification based tasks in Chapters 3 and 4, in which the effectiveness of image features employed as inputs of GP and the bloat control were investigated respectively. The window classification based framework has two drawbacks, i.e. setting the window related parameters and producing inaccurate object boundaries. Therefore, the pixel classification based framework was proposed in Chapter 5, which avoided these drawbacks. Based on the pixel classification framework, the feature selection and construction methods were investigated in Chapters 5 and 6 respectively, which generated effective features for figure-ground segmentation tasks. Therefore, it is necessary and meaningful to conduct the following comparisons: the window classification based versus the pixel classification based methods, the single-objective methods versus multiple-objective methods, and the selected features versus the constructed features for figure-ground segmentation tasks. Note that the comparison between the single-objective methods and the multiple-objective methods was already conducted in Chapters 4 and 5, so it is not included in this chapter.

The benchmark datasets selected in this thesis (see Section 2.5.1) contains one class of foreground objects. Therefore, whether the proposed methods can perform well on datasets containing various classes of foreground objects (a wide range of objects, e.g. plants, animals and vehicles) deserves to be investigated in this chapter. In addition, it is hypothesised

that the evolved segmentors based on one type of foreground objects can generalise well on other types of objects, so it would be interesting to conduct transfer validation tests. Moreover, compared with other segmentation techniques, the strengths and weaknesses of GP in this field deserve investigation.

## 7.2 Performance on Datasets with Various Foreground Objects

Various foreground objects refer to multiple foreground objects that can cover a wide range of objects, e.g. plant, animals and vehicles. The benchmark datasets used in this thesis, i.e. Weizmann horse dataset and Pascal aeroplane dataset (see Section 2.5.1), are partitioned to contain one class of foreground objects. For example, on Weizmann horse dataset, the horse images are used to train the proposed GP methods to evolve segmentors that are then used to segment horses on test images. Whether GP can evolve useful segmentors that can handle various types of objects simultaneously was not determined. In this section, the proposed GP based segmentation methods will be tested on datasets containing various types of foreground objects.

### 7.2.1 Design of Experiments

Two new datasets (i.e. the SED2 dataset [8] and the ImgSal dataset [94], which will be described in Section 7.2.2), are selected. This is because they are well explored benchmark object segmentation and detection datasets; the images in both datasets contains diverse classes of objects; and the ground truth on pixel level of both datasets is provided. Note that other image datasets that meet the above three criteria can also be used.

Two segmentation frameworks using GP were developed in this thesis, i.e. the window classification based framework (see Section 3.2.1) and


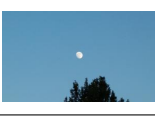




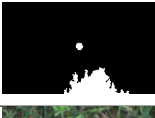


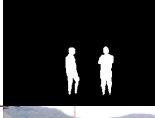










the pixel classification based framework (see Section 5.2.1), both of which are employed in this section. Three GP based segmentation methods (the weighted sum method, INSGP and ISPGP) were developed to evolve segmentation solutions in Chapter 4. As INSGP and ISPGP perform better than the weighted sum method, while INSGP and ISPGP produce solutions with similar segmentation performance, ISPGP (see Section 4.2.2) is selected in this section. In addition, the GP algorithm is used as a reference method. Therefore, both the window and pixel classification frameworks based GP and ISPGP methods will be employed on the two new datasets.

### 7.2.2 Datasets

Table 7.1 shows selected images from SED2 [8] and ImgSal (image saliency) [94] datasets. Specifically, SED2 consists of 100 images with two salient objects in each image. The SED2 images contain a wide range of foreground regions, e.g. birds, planes, buildings, boats, flowers and people. The sizes of SED2 images are around  $300 \times 225$  (or  $225 \times 300$ ) pixels.

ImgSal is a benchmark dataset for saliency detection [94]. There are 235 colour images in the ImgSal dataset, which are varied in certain aspects, e.g. salient region size and cluttered backgrounds. Specifically, ImgSal images can be divided into six categories: images with large, intermediate and small salient regions; images with cluttered backgrounds; images with repeating distractors; images with both large and small salient regions. In addition, ImgSal images contain diverse salient regions, e.g. animals (e.g. horses, cows, butterflies and pandas), people in different activities (e.g. working in the field and biking) and plants. The sizes of ImgSal images are all  $480 \times 640$  pixels. Note that the ground truth is derived from human segmentors, so it could include bias and ambiguity, e.g. in Table 7.1 whether the viaduct in the fourth ImgSal image is salient or not.

Table 7.1: Two new datasets (G.T. means ground truth; Ori. stands for original images).

SED2	Ori.					
	GT					
ImgSal	Ori.					
	GT					

### 7.2.3 Experiment Preparation

Two thirds of images in ImgSal and SED2 are used for training, and the remaining images are for testing, where appropriate fifty samples are extracted from each ImgSal training image, while twenty samples are extracted on each SED2 training image, since ImgSal images are larger than SED2 ones.

The required parameters for the window classification based methods are set based on the previous settings, i.e. the window size ( $16 \times 16$  pixels) and the shifting steps of the window (two in both vertical and horizontal directions) (see Section 4.3.3 for details).

### 7.2.4 GP Settings

The settings of GP (i.e. the function set, the terminal set, the fitness function and GP parameters) are similar to the settings in previous chapters, which are shown as follows.

Table 7.2 shows the function set used in this section, which removes the redundant operators included in the function set of Chapter 4 (see Section

4.3.2). The two objectives of ISPGP are the classification accuracy (Equation 4.2) and the complexity measure (Equation 4.3), and the fitness function of GP is the classification accuracy (Equation 4.2). In addition, major GP parameters are as follows: the rates of crossover, mutation and elitism operators are 80%, 19% and 1% respectively; the population size is 512, and the minimum and the maximum tree depth is 1 and 10 respectively. Other GP parameters follow the default settings used in Koza's works [83] (see Section 2.5.4 for details).

Several feature descriptors (i.e. Gabor filters, RGB colour features and mean/median filters (see Section 2.5.2)) are employed. Specifically, 12 Gabor filters are selected, which are generated from three common scales (4, 8, 16) and four common orientations ( $0, \frac{\pi}{4}, \frac{2\pi}{4}, \frac{3\pi}{4}$ ) to extract Gabor features. The pixel values on red, green and blue channels are used to capture colour information. In addition, the mean filter and the median filter with three block sizes ( $3 \times 3$ ,  $5 \times 5$  and  $9 \times 9$  pixels) are also employed to extract local grayscale statistics. Therefore, there are 21 features (12 Gabor features, 3 RGB colour features, 3 mean features and 3 median features) in each feature vector for the pixel classification based methods. Note that features are extracted from subimages for the window classification based methods. As the window is  $16 \times 16$  pixels (set up in Section 7.2.3), there will be 5376 features ( $16 \times 16 \times 21$ ) in each feature vector for the window classification based methods. It is considered that this feature dimension is too high, as it is computationally expensive and may cause overfitting. Therefore, the extracted features are down-sampled to 336 dimensions ( $4 \times 4 \times 21$ ) for the window classification based methods. The details of feature extraction for the window and pixel classification based methods can be seen in Section 4.3.1 and Section 6.5.1 respectively.

## 7.2.5 Results

Table 7.3 displays several image results. On SED2, the results are visually accurate with objects segmented, but edges are blurred and inner pixels

Table 7.2: Function set of GP based figure-ground segmentation methods.


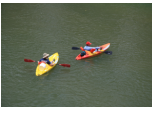


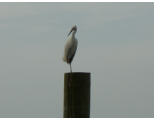








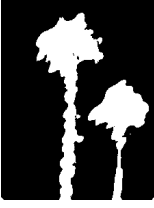
















Function Name	Definition	Function Name	Definition
$+(a_1, a_2)$	$a_1 + a_2$	$-(a_1, a_2)$	$a_1 - a_2$
$*(a_1, a_2)$	$a_1 * a_2$	$\%(a_1, a_2)$	$\begin{cases} a_1/a_2 & \text{if } a_2 \neq 0 \\ 0 & \text{if } a_2 == 0 \end{cases}$
$>(a_1, a_2)$	$\begin{cases} a_1 & \text{if } a_1 > a_2 \\ a_2 & \text{if otherwise} \end{cases}$	$<(a_1, a_2)$	$\begin{cases} a_1 & \text{if } a_1 < a_2 \\ a_2 & \text{if otherwise} \end{cases}$
$\text{IF}(a_1, a_2, a_3)$	$\begin{cases} a_2 & \text{if } a_1 \text{ is true} \\ a_3 & \text{if } a_1 \text{ is false} \end{cases}$	$==(a_1, a_2)$	$\begin{cases} 1 & \text{if } a_1 == a_2 \\ 0 & \text{if otherwise} \end{cases}$

of objects are mis-labelled. On ImgSal, the results are too specific with background parts captured as foreground (the first and fourth images). Interestingly the train carriages and two additional bathers are captured more clearly than GT as well as separating the two dolphins (the second, fifth and third images respectively).

Compared with SED2 images, ImgSal images are more difficult segmentation tasks, as ImgSal contains images with small salient regions, cluttered backgrounds and/or repeated distractors. This is supported by statistical results in Table 7.4), where the test  $F_1$  scores on ImgSal are significantly lower than those on SED2 based on the Mann-Whitney U-Test [118] at the significance level 5%. Even though the proposed GP based segmentation methods cannot perform close to optimum on ImgSal due to its high variations, the promising results on SED2 reflect that it is possible for GP to evolve segmentors with distinctive patterns for various types of objects (e.g. straight lines from buildings and curves from flowers).

When comparing ISPGP with GP, both achieve similar test  $F_1$  scores on each dataset based on the Mann-Whitney U-Test. However, both the training and test time are reduced for ISPGP, which are around half of those for GP. This is because ISPGP uses multi-objective techniques to balance the solution performance and complexity (Section 4.2.2), thus the evolved solutions of ISPGP are smaller/simpler than those of GP, which helps reduce the time cost in both training and test stages. This reflects that the pro-

Table 7.3: Result examples using pixel classification based ISPGP (G.T. means ground truth; Ori. stands for original images).

SED2	Ori.					
	GT					
	Result					
ImgSal	Ori.					
	GT					
	Result					

posed ISPGP performs better than GP for the figure-ground segmentation tasks on the given datasets.

When comparing the window classification based and pixel classification based methods, their  $F_1$  scores on the test sets are compared based on the Mann-Whitney U-Test. Specifically, on SED2, the window based and



Table 7.4: Results (Window and Pixel means the window classification based and the pixel classification based GP methods respectively; s represents seconds).

	Method		Training Accuracy(%)	Test $F_1$	Train Time Per Run(s)	Test Time Per Image(s)
SED2	Window	GP	$79.96 \pm 0.73$	$0.544 \pm 0.019$	$14.616 \pm 2.802$	$0.025 \pm 0.005$
		ISPGP	$81.13 \pm 3.00$	$0.549 \pm 0.020$	$10.865 \pm 1.607$	$0.019 \pm 0.004$
	Pixel	GP	$80.49 \pm 0.44$	$0.540 \pm 0.017$	$9.047 \pm 2.004$	$0.055 \pm 0.009$
		ISPGP	$81.67 \pm 2.36$	$0.546 \pm 0.020$	$6.153 \pm 0.734$	$0.025 \pm 0.007$
ImgSal	Window	GP	$72.97 \pm 0.56$	$0.261 \pm 0.010$	$195.960 \pm 21.594$	$0.120 \pm 0.025$
		ISPGP	$78.16 \pm 7.28$	$0.259 \pm 0.013$	$90.569 \pm 9.370$	$0.108 \pm 0.024$
	Pixel	GP	$72.63 \pm 2.44$	$0.250 \pm 0.014$	$58.470 \pm 12.561$	$0.190 \pm 0.040$
		ISPGP	$79.44 \pm 6.49$	$0.248 \pm 0.010$	$30.817 \pm 5.366$	$0.086 \pm 0.028$

pixel based GP methods achieve similar  $F_1$  scores, as well as the window based and pixel based ISPGP methods. However, on ImgSal, the window based methods are significantly better in  $F_1$  score than the pixel based methods. Considering the ImgSal images are more complex with high variations, the results suggest that the window classification based segmentation framework perform better than the pixel based framework on complex image datasets. This may be because the window based framework uses a shifting window that generates multiple predicted labels on most image pixels and a voting scheme is employed to produce the final predicted label, which can help lower the influence of inaccurate predictions.

Two ISPGP solutions for the SED2 dataset are shown in Figure 7.2. Take Figure 7.2b as an example. Based on the feature extraction in Section 7.2.4, feature subsets,  $f_0, \dots, f_{191}$ ,  $f_{192}, \dots, f_{239}$ ,  $f_{240}, \dots, f_{287}$  and  $f_{288}, \dots, f_{335}$ , are Gabor features, colour features, mean features and median features respectively. From Figure 7.2b, eight features are used in this solution, among which six features are Gabor features, one feature is a colour feature and the other one is a mean feature. This reflects that edge features are popular for datasets with multiple types of objects. This may be because the variations of different types of foreground objects and backgrounds

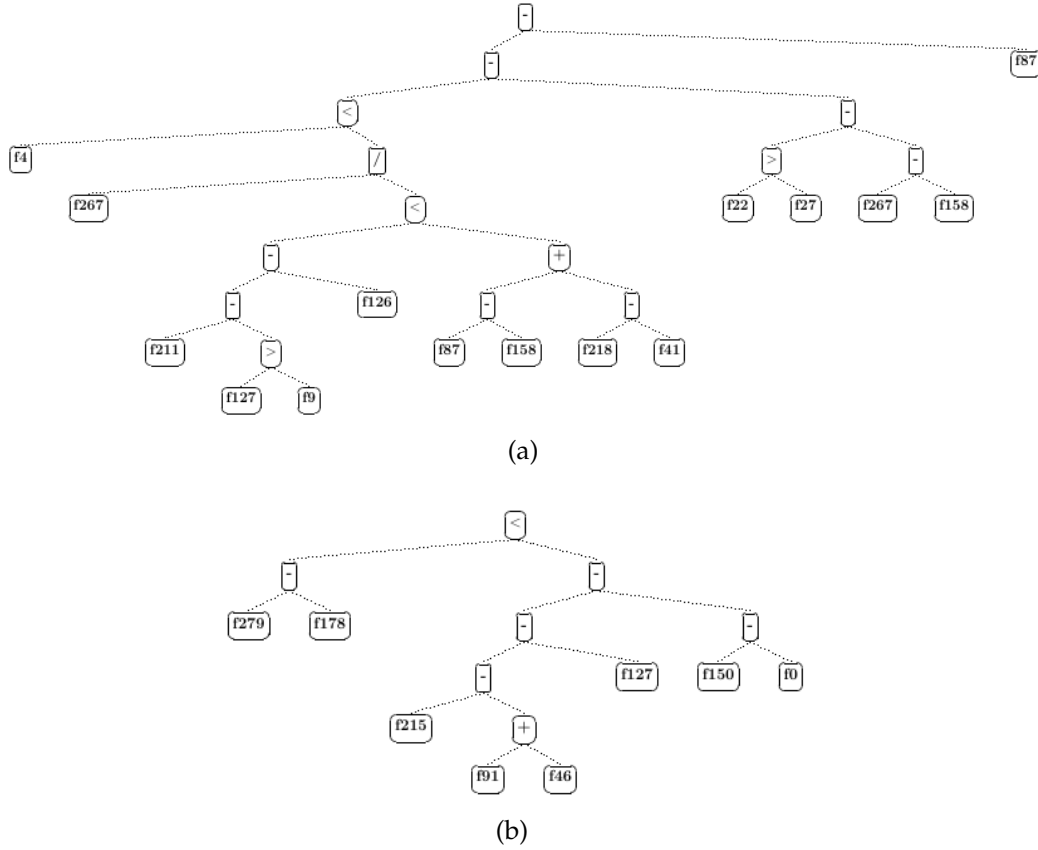


Figure 7.2: Two example segmentors evolved by the window classification based ISPGP on the SED2 dataset.

that lie in color and texture are more difficult to capture than edges.

### 7.3 Transfer Validation

This section aims to investigate how well the segmentors evolved from images with one type of objects (e.g. horses) can generalise on the same images (images with the same type of foreground objects, e.g. horses, from different datasets to remove confounding variables, e.g. lighting effects), similar images (images with biologically similar types of foreground objects, e.g. cows) and different images (images with biologically different

types of foreground objects, e.g. planes).

### 7.3.1 Design of Experiments

To achieve the goal of this section, the training set of the Weizmann horse dataset will be used to train the proposed methods to evolve segmentors, and then the evolved segmentors will be tested on the test set of the Weizmann horse dataset, the validation sets of Pascal horse images, Pascal cow images and Pascal aeroplane images respectively. The cows are considered as similar objects to horses, as they share common characteristics in both shape and backgrounds, while the planes are regarded as different objects from horses, as they are different in these aspects. Furthermore, an experiment of using Pascal aeroplane images as the training set and the Weizmann horse images as the test set is conducted for cross-comparison.

The description of Weizmann and Pascal datasets are provided in Section 2.5.1. There are 218 images in the Weizmann training set, and 20 samples are extracted from each training image. In addition, there are 56 horse images, 56 cow images and 88 plane images in the Pascal dataset for validation. The settings of the GP methods (i.e. the terminal set, the function set, the fitness function and the parameter settings) are the same with those in Section 7.2.4.

### 7.3.2 Results

Table 7.5 presents the training results based on two training sets, i.e. Weizmann horse images and Pascal plane images. It shows that similar classification accuracies are achieved for the window and pixel based methods on both training sets. The evolved segmentors from the two datasets are then tested on several test sets, and the results are presented as follows.

Table 7.6 shows the performance of the segmentors (evolved from the Weizmann horse images and Pascal plane images) on different test sets. Considering the four sets of experiments based on the segmentors evolved

Table 7.5: Training results (Window and Pixel means the window classification based and the pixel classification based GP methods respectively; s represents seconds).

Training Set	Method	Training Accuracy(%)	Train Time Per Run(s)
Weizmann horse	Window	$76.59 \pm 1.59$	$81.6 \pm 18.4$
	Pixel	$77.69 \pm 0.92$	$30.3 \pm 5.9$
Pascal plane	Window	$80.44 \pm 0.82$	$98.8 \pm 19.5$
	Pixel	$80.79 \pm 0.80$	$28.1 \pm 5.3$

Table 7.6: Test results (Window and Pixel means the window classification based and the pixel classification based GP methods respectively; s represents seconds; Weizmann horse VS Weizmann horse means that the segmentors evolved from Weizmann horse images are tested on unseen Weizmann horse images, and so forth).

	Method	Test $F_1$ Score	Test Time Per Image(s)
Weizmann horse VS Weizmann horse	Window	$0.490 \pm 0.013$	$0.062 \pm 0.011$
	Pixel	$0.597 \pm 0.015$	$0.042 \pm 0.009$
Weizmann horse VS Pascal horse	Window	$0.492 \pm 0.032$	$0.063 \pm 0.016$
	Pixel	$0.476 \pm 0.019$	$0.114 \pm 0.028$
Weizmann horse VS Pascal cow	Window	$0.562 \pm 0.038$	$0.066 \pm 0.023$
	Pixel	$0.550 \pm 0.027$	$0.112 \pm 0.028$
Weizmann horse VS Pascal plane	Window	$0.359 \pm 0.047$	$0.058 \pm 0.014$
	Pixel	$0.376 \pm 0.037$	$0.118 \pm 0.028$
Pascal plane VS Weizmann horse	Window	$0.413 \pm 0.014$	$0.034 \pm 0.007$
	Pixel	$0.393 \pm 0.014$	$0.040 \pm 0.006$

from Weizmann horse images, the test  $F_1$  scores are significantly worse on Pascal plane images than those on horse/cow images based on the Mann-Whitney U-Test [118] at the significance level 5%. This suggests

that segmentors cannot generalise well on images with different objects. Specifically, the training images are horse images that contain different patterns from plane images. For example, the shape of horses mainly consists of curves, while planes are formed by straight lines. In addition, the backgrounds of horse images tend to be grass and/or trees, while those of planes tend to be sky and/or buildings.

In contrast, when applying Weizmann horse segmentors on the Pascal horse images and Pascal cow images, promising results are achieved. Specifically, compared with the  $F_1$  scores on Weizmann test images, they are similar on Pascal horse images and are even better on Pascal cow images. This reflects that Weizmann horse segmentors can generalise well on Pascal horse images and Pascal cow images. The reason is that the GP methods have captured common characteristics that horses and cows share, e.g. object shape and backgrounds. The ability to capture common characteristics across a class of foreground objects in a human interpretable manner is a major strength of GP based image segmentation. Note that on the Pascal cow images, better test performance is achieved than that on the Pascal horse images (having the same type of object as the training images). By checking the Pascal cow images, they tend to be easier segmentation tasks than the Pascal horse images, since they tend to contain relatively clean backgrounds and objects with bigger object sizes in relation to background.

Table 7.3 shows several result examples for the transfer validation test. It can be seen that based on the Weizmann training set (horse images), the Weizmann test horse, Pascal test horse and Pascal test cow images are segmented visually accurately. Even though there exist false positives on Pascal test horse and cow images, the results are still promising due to the cluttered backgrounds. This is because the training and test images are the same type or contain similar types of objects, which share common properties (e.g. in object shape and object size). In addition, when training on the horse images and testing on the plane images or vice versa, the results
















	W.horse VS W.horse	W.horse VS P.horse	W.horse VS P.cows	W.horse VS P.plane	P.plane VS W.horse
Original					
GT					
Result					

Figure 7.3: Result examples of the window classification based GP for transfer validation test (W.horse VS P.horse refers to the segmentor trained on the Weizmann horse training set is used to segment Pascal horse test set, and so forth; GT means ground truth).

are relatively worse. For example, the “P.plane VS W.horse” result image shows that the boundaries of the horse cannot be accurately detected, which may be because the GP method cannot learn specific patterns to detect curves on a horse from plane images with straight lines.

## 7.4 Selected Features Versus Constructed Features

This section will compare the selected feature subsets with the constructed features of the proposed methods. Specifically, three feature selection methods were proposed in Chapter 5, i.e. PGP-FS, NSGP-FS and SPGP-FS. As NSGP-FS and SPGP-FS perform better than PGP-FS, while NSGP-FS and SPGP-FS produce feature subsets with similar segmentation performance, SPGP-FS is selected in this section. In addition, four feature construction methods were developed in Chapter 6, i.e. FSFC, WSFC, SubtMFC and CoevoMFC, among which CoevoMFC is more capable than others in generating effective features for given segmentation tasks. Therefore, SPGP-

FS and CoevoMFC are the best-performing feature selection/construction methods proposed in this thesis. Note that the previous results of SPGP-FS and CoevoMFC from Chapters 5 and 6 respectively cannot be compared directly, as they use slightly different settings. The two methods are rerun using the same settings (e.g. the terminal/function set, the fitness function and GP parameters) for the same segmentation tasks. Features selected by SPGP-FS and features constructed by CoevoMFC will be compared in this section.

Note that feature selection and construction are not combined together for figure-ground segmentation in this thesis. This is because to understand GPs performance and how GP functions on feature selection or feature construction, it is necessary to conduct research/investigation on each of them individually. Moreover, even though combining feature selection and construction might be a method to further manipulate the selected/constructed features to improve the segmentation performance, there are many other possible methods, which are beyond the scope of this thesis.

### 7.4.1 Design of Experiments

Two standard image datasets, Weizmann horse database [22, 24] and Pascal aeroplane dataset [53] are used in this section (see Section 2.5.1 for details). The dataset processing (e.g. the splitting of datasets to training/test sets) follows previous settings (Section 6.5.4), as well as the terminal set (Section 6.5.1), the fitness function (i.e. the classification accuracy for GP; the classification accuracy and the number of selected features for SPGP-FS (Section 5.3.2)) and the GP parameter settings (Section 6.5.3).

The function set is shown in Table 6.1, which combines the function sets used for both feature selection (Chapter 5) and feature construction (Chapter 6). This makes the function set more likely to be sufficient for both SPGP-FS and CoevoMFC. Specifically, it consists of eight functions: five standard mathematical functions ( $+$ ,  $-$ ,  $*$ ,  $\%$ ,  $\text{sqrt}$ ), two relational functions

( $>$ ,  $<$ ) and one conditional function ( $IF$ ). These functions are selected, as they are all computationally efficient, and can realise both linear and non-linear combinations of input features.

Table 7.7: Function set of GP based feature selection/construction methods.

Function Name	Definition	Function Name	Definition
$+(a_1, a_2)$	$a_1 + a_2$	$-(a_1, a_2)$	$a_1 - a_2$
$*(a_1, a_2)$	$a_1 * a_2$	$\%(a_1, a_2)$	$\begin{cases} a_1/a_2 & \text{if } a_2 \neq 0 \\ 0 & \text{if } a_2 == 0 \end{cases}$
$\text{sqrt}(a_1)$	$\text{Sqrt} a_1 $		
$>(a_1, a_2)$	$\begin{cases} a_1 & \text{if } a_1 > a_2 \\ a_2 & \text{if otherwise} \end{cases}$	$<(a_1, a_2)$	$\begin{cases} a_1 & \text{if } a_1 < a_2 \\ a_2 & \text{if otherwise} \end{cases}$
$IF(a_1, a_2, a_3)$	$\begin{cases} a_2 & \text{if } a_1 \text{ is true} \\ a_3 & \text{if } a_1 \text{ is false} \end{cases}$	$==(a_1, a_2)$	$\begin{cases} 1 & \text{if } a_1 == a_2 \\ 0 & \text{if otherwise} \end{cases}$

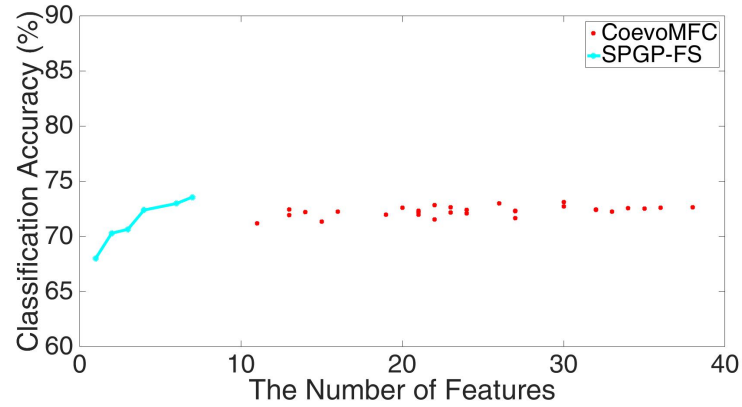
Naïve Bayes (NB) from the Weka package [171] is employed to evaluate the selected/constructed features in the fitness function in the training/test stage. This is because Naïve Bayes (NB) is a robust and well understood classifier. The  $F_1$  score (Equation 1.2 in Section 1.5.2) is utilised to evaluate segmentation results. All experiments are run 30 times independently.

## 7.4.2 Results

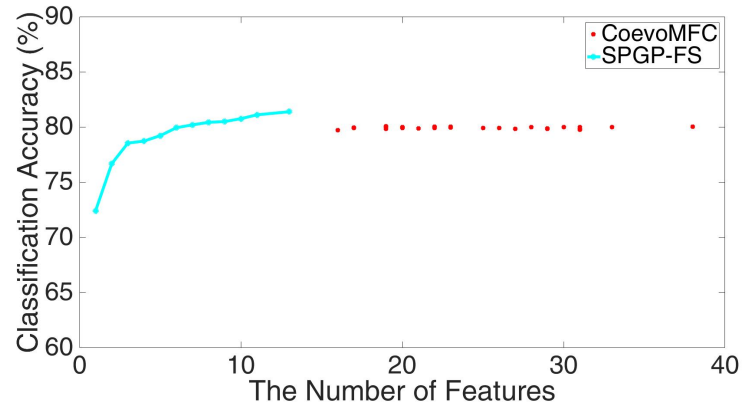
Figure 7.4 and Figure 7.5 compare the training and test performance of SPGP-FS and CoevoMFC respectively. For SPGP-FS, the aggregated Pareto fronts of selected feature sets from 30 runs are displayed, while for SPGP-FS, 30 constructed feature sets (each from one GP run) are displayed. Note that the number of features in the figures refer to the number of original features selected in the feature subsets by SPGP-FS or used to construct the new features by CoevoMFC. These figures show that based on both



the training accuracy and the test  $F_1$  score, the selected feature subsets of SPGP-FS can dominate most constructed feature sets of CoevoMFC. In other words, for a selected feature set and a constructed feature set that have the same training accuracy (or test  $F_1$  score), the selected feature set has smaller number of features than those used to construct the new feature set.



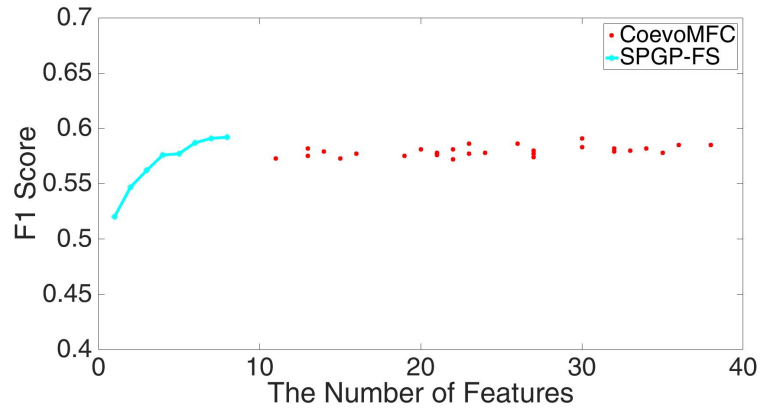
(a) On Weizmann dataset.



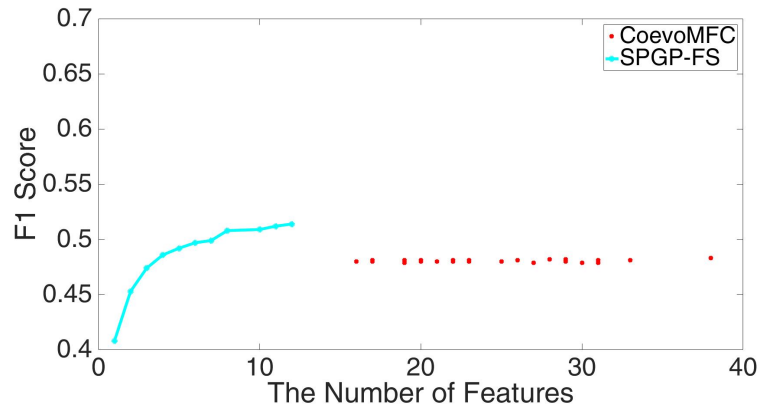
(b) On Pascal dataset.

Figure 7.4: Training performance (solutions with high classification accuracies and low number of features are desired).

In Table 7.8, the training and test performance of CoevoMFC constructed



(a) On Weizmann dataset.



(b) On Pascal dataset.

Figure 7.5: Test performance (solutions with high classification accuracies and low number of features are desired).

feature sets are the average based on 30 runs. For SPGP-FS, a Pareto front of solutions (trade-offs based on the two objectives, i.e. the classification accuracy and the number of selected features) are generated in each run. To compare with the constructed feature sets of CoevoMFC, the Pareto front solution with the highest classification accuracy of SPGP-FS is selected in each run, and the average values of the 30 front solutions (selected feature sets) are displayed in Table 7.8.

Table 7.8 shows that similar training accuracies are achieved for CoevoMFC and SPGP-FS based on the Mann-Whitney U-Test [118] at the significance level 5%. In terms of the test performance, similar  $F_1$  scores are achieved on the Weizmann dataset. However, on the Pascal dataset, the selected features perform significantly better than the constructed features. Therefore, for the two given figure-ground segmentation tasks, SPGP-FS can produce feature sets that are more effective than those of CoevoMFC.

According to Table 7.8, there are more features in the selected feature sets than the new features in the constructed feature sets. Specifically, there are around 9 and 11 selected features based on SPGP-FS for the two datasets respectively, while the number of constructed features based on CoevoMFC is fixed (i.e. three) due to the characteristics of CoevoMFC that one new feature is constructed from one type of primitive features and three general types of primitive features are used (see Section 6.4). However, considering Figure 7.4 and Figure 7.5, more primitive features are used to construct the three new features by CoevoMFC than the features selected by SPGP-FS.

In addition, CoevoMFC is more time-consuming than SPGP-FS. One major reason lies in the fitness evaluation, where primitive features need to be transformed to new features based on each CoevoMFC individual (regarded as a feature construction function). Compared with SPGP-FS, the processing of feature transformation leads to extra calculation for CoevoMFC.

Table 7.8: Results (s represents seconds).

Dataset	Method	Training Accuracy(%)	Test $F_1$ Score	Num. of Features	Train Time Per Run(s)
Weizmann	SPGP-FS	$71.95 \pm 0.82$	$0.578 \pm 0.007$	$9.00 \pm 1.74$	$33.4 \pm 2.3$
	CoevoMFC	$72.29 \pm 0.44$	$0.579 \pm 0.004$	3	$88.5 \pm 6.5$
Pascal	SPGP-FS	$80.36 \pm 0.56$	$0.499 \pm 0.009$	$11.10 \pm 2.30$	$40.5 \pm 4.3$
	CoevoMFC	$79.93 \pm 0.09$	$0.480 \pm 0.001$	3	$87.0 \pm 2.3$

## 7.5 Further Discussion

Image segmentation is an active area with many algorithms and techniques proposed. According to Khan's work [80], traditional segmentation techniques can generally fall into the following six categories:

- Threshold based methods [2];
- Region based methods (region growing, region splitting, and region merging) [76];
- Edge based methods [6];
- Clustering based methods [56];
- Neural network based methods [61, 108];
- Partial Differential Equation (PDE) based methods (active contour model (ACM), level sets, Mumford-Shah model and so on) [29, 73].

Four segmentation methods are selected for comparison with the proposed GP methods in Chapter 3, i.e. thresholding [2] and active contour model (ACM) [1], a region growing method [87] and a K-means method [56]. The four methods are selected as they are from four different branches, and each of them is a standard or popular algorithm in its branch. The results (see Section 3.5.2) show that the GP-evolved segmentors achieve consistently good performance, while traditional methods perform worse or cannot maintain good performance on all the given datasets. Specifically, automatic or manual initial contour placement is required by ACM, and it is well known that ACM relies heavily on the placement of the initial contour [106]. Only when initial contours are set near objects of interest, could ACM produce accurate results. In addition, the thresholding method is only suitable for images with high contrast between objects and backgrounds. For K-means clustering and region growing methods,

the number of clusters and regions is needed to be set beforehand respectively. Compared with the selected traditional segmentation methods, the advantage of GP is that it is easily adapted to diverse image domains without major parameter tuning or human intervention.

In addition to the above four techniques, the remaining two types of segmentation techniques, i.e. edge based methods and neural network based methods, were not tested experimentally with the following reasons. Considering that results of figure-ground segmentation are commonly pixelwise separation of objects and backgrounds, edge based methods were not selected as they produce object boundaries as segmentation results and extra processing is needed to generate pixelwise segmentation for fair comparison with the proposed methods. In addition, among neural network based methods, deep learning techniques, e.g. convolutional neural networks [108], are popular and are gaining success in image segmentation. Deep neural networks were not selected, because they require large annotated image datasets and processing with high computation power, which are beyond the scope of this work. Even though the edge based and deep neural networks based methods are not experimentally compared with the proposed methods, their strengths and weaknesses are discussed as follows.

Edge based methods detect and link edge pixels of different regions based on certain criteria (e.g. discontinuity or a similarity) to form contours, thus separating regions in an image [6]. However, edges of a region can often be difficult to be detected due to noise or occlusions. In addition, the key advantage of the deep learning techniques lies in the ability to learn appropriate feature representations for given problems [61]. One major challenge of deep learning methods is to construct appropriate datasets as they should have large numbers of examples that represent the given problem, which requires domain expertise, infrastructure resources and human labour. Moreover, expertise is required to design the architecture of deep networks for given problems. Compared with deep learning

techniques, the strengths of GP are that it does not require pre-defined problem models, and GP does not need large-scale datasets for training.

## 7.6 Chapter Summary

This chapter investigated the following four issues. Firstly, whether the proposed methods can perform well on datasets that contain various foreground objects. Secondly, whether the segmentors evolved from images with one type of foreground objects can generalise well on images with similar/different types of foreground objects. Thirdly, the selected features were compared with the constructed features of the proposed feature selection/construction methods. Finally, the strengths and weaknesses of GP were discussed in comparison with alternative segmentation techniques.

This chapter shows that the proposed GP based segmentation methods can evolve segmentors with useful class characteristic patterns to segment various types of objects simultaneously, although it is still challenging to handle highly varied datasets with various foreground objects. In addition, the window classification based segmentation framework is shown to be better than the pixel based framework on the complex image datasets. This may be because the window based framework has a voting scheme to generate the final class label of image pixels, which helps reduce the influence of inaccurate predictions.

The proposed GP based segmentation methods are argued to be able to evolve segmentors that can generalise well on images with the same/similar foreground objects. In contrast, when tested on images with different types of objects, the evolved segmentors cannot perform well (e.g. segmentors trained on horse images and tested on plane images).

In addition, SPGP-FS can select feature subsets more effective than those constructed by CoevoMFC, and SPGP-FS is more efficient than CoevoMFC as extra feature transformation is required by CoevoMFC in eval-

uating new feature sets. In addition, more primitive features are used to construct new features by CoevoMFC than those selected by SPGP-FS.

This chapter considers that compared with deep learning techniques qualitatively, the strengths of GP are that it does not require pre-defined problem models, and does not need large-scale datasets for training. In addition, compared with other traditional segmentation methods, e.g. region growing and K-means clustering, the advantage of GP is to be easily adapted to diverse image domains without major parameter tuning or human intervention.





## Chapter 8

# Conclusions and Future Work

This thesis focuses on figure-ground image segmentation using GP. The overall goal is to investigate and improve the capability of GP for figure-ground segmentation. This goal has been achieved by developing new GP based methods to evolve segmentation algorithms that can perform well on diverse image domains, and to conduct feature selection/construction that can produce effective features for supplementary segmentation approaches. This thesis contributes to both computer vision and genetic programming fields. Based on the proposed methods (e.g. segmentation methods and feature selection/construction methods), similar/better segmentation performance was achieved with less manual and/or computational cost.

This thesis finds that GP is suited to figure-ground image segmentation, especially for the following three issues. Firstly, given diverse image domains, as GP based segmentation methods do not require pre-defined solution models, much parameter tuning and human intervention, they can be easily adapted to a wide range of image domains, e.g. bitmap, texture and object images (details in Chapters 3, 4 and 7). Secondly, given complex images with high variations (e.g. in object shapes, object sizes and cluttered backgrounds), as GP based feature selection/construction methods can generate effective features, they can help improve the seg-

mentation performance of the subsequent segmentation based on standard classifiers (details in Chapters 5 and 6). Thirdly, if discovered feature relations are required to be interpretable and reusable, the GP based feature construction methods should be considered, as it is possible to interpret constructed features by analysing the evolved tree solutions of GP (details in Chapter 6).

The proposed segmentation methods were tested and compared with existing GP based methods and traditional segmentation methods on standard image datasets with varying difficulty levels. The results show that GP is able to derive well-performing segmentation algorithms that can adapt to a wide range of image domains, e.g. binary images, texture images and images with different types of objects. Specifically, the weighted sum method, INSGP, and ISPGP were developed to evolve segmentation algorithms (see Chapter 4). INSGP and ISPGP can evolve segmentors with similar or better segmentation performance, yet smaller sizes than those of the weighted sum method. Moreover, INSGP and ISPGP have similar performance in evolving effective segmentors.

GP can be effectively used for feature selection/construction with the selected/constructed features achieving better segmentation performance and containing a smaller number of features than the original feature set. Specifically, three methods were designed in Chapter 5 for feature selection, i.e. PGP-FS, NSGP-FS and SPGP-FS. NSGP-FS and SPGP-FS can select feature subsets with similar segmentation performance, yet smaller numbers of features than PGP-FS. Moreover, NSGP-FS and SPGP-FS have similar performance in selecting feature subsets for the given problems. In addition, four feature construction methods were developed in Chapter 6, i.e. FSFC, WSFC, SubtMFC based and CoevoMFC. Among them, CoevoMFC is more capable to construct effective features than the other methods.

Conclusions for each of the research objectives of this thesis are provided as follows, along with the potential research areas that this work

enables.

## 8.1 Achieved Objectives

This work has achieved the following research objectives:

1. Diverse image features were investigated to determine what image information is necessary for GP to evolve capable segmentors for diverse image domains. Seven terminal sets that include the three general types of image features, i.e. brightness (intensities), texture and shape features, were used as inputs of GP to evolve segmentation algorithms. Results show that texture features are more effective than intensities and shape features. Moreover, the evolved segmentors using texture features (e.g. Gabor features) can perform consistently well on a wide range of image domains, e.g. binary images, texture images and images with different types of objects.
2. Multi-objective GP based methods were proposed for figure-ground segmentation to balance the evolved solutions' functionality and complexity in terms of solution size. The GP based methods often have bloat problems that cause the complex segmentors being evolved. As complex segmentors normally have large sizes, are computationally expensive and difficult to interpret, multi-objective GP based methods were designed to optimise two objectives, i.e. maximising the segmentation performance of solutions and minimising their complexity. The proposed methods can evolve a set of non-dominated segmentors that are trade-offs between the two objectives. Since the evolved segmentors have smaller sizes, the fitness evaluation of individuals (segmentors) in the evolutionary process is less time-consuming, leading to the training and test time being reduced.
3. New feature selection methods using GP were developed for the first time to address figure-ground segmentation problems. The GP

based feature selection is not extensively investigated for figure-ground segmentation. Moreover, feature selection is a multi-objective task: maximizing the segmentation performance and minimizing the number of features. Different from existing works that rarely consider the number of selected features, this thesis introduced the parsimony coefficient technique and multi-objective techniques in GP to balance the two objectives. Results show that compared with the original features, the proposed GP based methods can select feature subsets that can improve the segmentation performance while reducing the number of features.

4. Novel feature construction methods using GP were developed for the first time to construct discriminative features for figure-ground segmentation. GP based feature construction has not been sufficiently investigated for figure-ground segmentation, and the existing works mix different types of features simultaneously. In this thesis, both single feature construction and multiple feature construction (MFC) methods were developed. Compared with the original features, the constructed features are more discriminative, which helps improve the segmentation performance and reduce the number of features. Moreover, the coevolution technique is introduced in GP to develop CoevoGP, that allows new features to be constructed separately from different types of features simultaneously. The results show that CoevoGP performs best in generating effective features for the given segmentation problems.
5. The strengths and weaknesses of the proposed GP methods were further investigated in the field of figure-ground image segmentation. Specifically, the following four issues are addressed. Firstly, the proposed GP segmentation methods were tested on image datasets with various classes of foreground objects. The proposed methods can evolve segmentors with useful class characteristic patterns to

segment various types of objects simultaneously. Secondly, transfer validation tests were conducted by employing the evolved segmentors from images with one type of foreground objects on images with other types of objects. The results show that the evolved segmentors can generalise well on images with similar foreground objects. Thirdly, the best-performing proposed feature selection method and feature construction method are compared, which shows that the feature selection method produces more effective features, uses fewer primitive features and consumes less time for the given segmentation tasks. Finally, the proposed GP methods were compared with other segmentation techniques, e.g. thresholding, K-means clustering and deep learning techniques. Compared with deep learning techniques, the strengths of GP are that it does not require pre-defined problem models, and does not need large-scale datasets for training. Compared with other traditional segmentation methods, GP can be easily adapted to diverse image domains without major parameter tuning or human intervention.

## 8.2 Main Conclusions

This section discusses the major conclusions for the five research objectives corresponding to the five contribution chapters (Chapter 3 to Chapter 7).

### 8.2.1 GP for Evolving Segmentors from Diverse Image Features

The first work on determining effective image features for GP to evolve capable segmentors (segmentation algorithms) were conducted in Chapter 3. Specifically, seven terminal sets were investigated, that cover three general feature types, i.e. brightness, texture and shape features. It is found that intensities are not sufficient for complex images, as they are

not discriminative when foreground and background patches have similar brightness levels. Shape features are better than intensities, but using only shape information is not sufficient to distinguish the foreground and background patches with similar pixel distributions. In contrast, texture features are the most effective since they are discriminative for different materials that foreground/background regions normally belong to (e.g. metal or wood). In addition, this thesis finds that GP can use texture features (e.g. Gabor features or histogram statistics) to derive segmentors that perform consistently well on diverse image domains.

### 8.2.2 Multi-objective GP for Evolving Segmentors

New GP methods were proposed to evolve figure-ground segmentors that are balanced between the segmentation performance and the solution complexity in terms of solution size in Chapter 4. Specifically, a parsimony pressure technique (i.e. the weighted sum technique) and two multi-objective techniques (INSGA-II and ISPEA2) were incorporated with GP to optimise the two objectives at the same time. The weighted sum GP method considers both the functionality and the complexity, but combines them into a single objective. The multi-objective GP methods treat two objectives independently, aiming to achieve good trade-offs between solution functionality and complexity. In addition, a GP based method without considering solution complexity was developed as a reference method. It is found that GP can be successfully combined with parsimony pressure techniques or multi-objective techniques to evolve balanced segmentors that reduce significantly in size while achieving similar segmentation performance to that of segmentors produced by the reference method.

#### Weighted Sum based Method

This thesis finds that the weighted sum based method can produce solutions that are better than those of the reference method, yet worse than

those of the multi-objective methods. Considering the weighted sum technique combines objectives into a scalar fitness function based on a pre-defined coefficient, its search space is restricted to a certain part of that based on two independent objectives for the multi-objective methods. Therefore, the weighted sum based method is not as flexible as the multi-objective methods in searching for trade-offs between objectives.

### **Multi-objective Methods**

This thesis finds that the two multi-objective GP methods (INSGP, ISPGP) outperform the parsimony pressure method and the reference method in evolving balanced solutions. Moreover, INSGP solutions have found to have a better spread along the whole frontier than the ISPGP solutions. This is mainly due to the difference in the two multi-objective techniques: SPEA2 focuses more on solutions in the middle region, while NSGA-II tends to emphasise the edge points of the current nondominated solutions and produce solutions along the whole frontier [44].

### **8.2.3 GP for Feature Selection in Figure-ground Segmentation**

New feature selection methods using GP were developed for figure-ground image segmentation for the first time (see Chapter 5). The proposed methods handle feature selection as a multi-objective task (maximizing the segmentation performance based on the selected features and minimizing the number of selected features) by using the parsimony coefficient technique and two multi-objective techniques. This thesis finds that parsimony pressure techniques and multi-objective techniques can be successfully combined with GP to select effective features for figure-ground segmentation.

### **Parsimony Coefficient based GP Method**

It is found that the new fitness function designed for the parsimony coefficient method can effectively guide it to search for feature subsets based on the two objectives. Specifically, the fitness function consists of two terms that are linearly combined, i.e. the classification accuracy and a penalty term. As in Chapter 5, the segmentation tasks were transformed to pixel classification based problems, the classification accuracy was selected to measure the classification performance based on the selected features. The parsimony coefficient technique introduced a penalty (based on the number of selected features) into the classification accuracy. The penalty measure is based on the sigmoid function, as it can monotonically transform solution sizes into the range value  $[0, 1)$ , making the penalty value suitable to be linearly combined with the classification accuracy.

Compared with the original features, the feature subsets produced by the proposed parsimony pressure based method can improve segmentation performance while reducing the number of features. However, compared with two traditional feature selection methods (sequential forward selection (SFS) and sequential backward selection (SBS)), this method cannot produce consistently better performance on the given datasets.

### **Multi-objective GP Methods**

It is found that Pareto dominance based multi-objective techniques (e.g. NSGA-II and SPEA2) enable GP to select effective features based on the two objectives. Specifically, the proposed multi-objective methods produced trade-off solutions along the two objectives, which can dominate the original feature set, SFS/SBS solutions and the solutions of the parsimony pressure method. This means that compared with the parsimony pressure based method, the multi-objective methods can further reduce the number of features selected while improving the segmentation performance. The reason is that, unlike the parsimony pressure based method (aggregates the objectives into a scalar fitness function), the multi-objective methods treat two objectives independently, making them more flexible in



searching for trade-offs between objectives.

#### 8.2.4 GP for Feature Construction in Figure-ground Segmentation

GP was applied for the first time to construct high-level features from primitive features to improve the segmentation performance on complex images (e.g. images with high variations) in Chapter 6. Based on whether a single feature or multiple features is/are constructed, there are single feature construction (SFC) [28] and multiple feature construction (MFC) methods [129]. Both single (SFC) and multiple feature construction (MFC) methods were designed in this work. This work finds that by considering interactions of the original features, the constructed features of the proposed methods achieve better segmentation performance than the original features yet are fewer in number.

##### Single Feature Construction

Two single feature construction methods were developed, i.e. a filter approach (termed as FSFC) and a wrapper approach (WSFC). Specifically, an entropy based fitness function was designed for FSFC and the classification accuracy was used as the fitness function for WSFC. It is found that WSFC is better than FSFC in constructing more effective features for the given figure-ground segmentation tasks. Note that WSFC employs the feedback of a classifier to evaluate the constructed features, while FSFC depends on the general characteristics of the training data. FSFC tends to be less effective than WSFC in feature evaluation. This is because the avoidance of inductive algorithms in filter approaches makes the constructed features not optimised to be used with a learning algorithm for specific tasks.

### Multiple Feature Construction

Two multiple feature construction methods were proposed, including a subtree based method (SubtMFC) and a coevolutionary method (CoevoMFC). It is found that compared with the constructed features of single construction methods (FSFC and WSFC), the features constructed by the multiple construction methods (SubtMFC and CoevoMFC) generally achieve better or similar training and test performance. As the single constructed feature is only one combination of original features, it may not contain sufficient distinguishing information for foreground/background pixels, which leads to worse performance than multiple constructed features.

Moreover, it is found that CoevoMFC is more capable than SubtMFC in generating effective features for the given segmentation tasks. This is because CoevoMFC is developed by incorporating the cooperative coevolutionary technique with GP, which allows GP to construct features from different types of image features separately, and then combines the constructed features simultaneously. Therefore, CoevoMFC does not allow the interactions across different types of features. In contrast, SubtMFC (as well as FSFC and WSFC) conducts homogeneously combination of varied types of primitive features, leading to the interactions of non-related features, which may lower the effectiveness of constructed features.

#### 8.2.5 Holistic Considerations

Further investigations of the proposed GP methods in the field of figure-ground image segmentation were conducted in Chapter 7. Specifically, four issues were addressed. Firstly, the proposed GP segmentation methods were hypothesised to generalise well on image datasets with multiple classes of foreground objects. Secondly, it was shown that the segmentors evolved from images with one type of foreground objects can generalise well on images with other types of objects. Thirdly, the selected features and constructed features of the proposed GP methods were compared. Fi-

nally, the proposed GP segmentation methods were compared with other segmentation techniques beyond the scope of EC techniques.

### **Performance on Datasets with Various Foreground Objects**

Note that the major benchmark datasets used in this thesis, i.e. Weizmann horse dataset and Pascal aeroplane dataset, contain one type of foreground objects. In practice, datasets with various types of foreground objects are more common than those containing one type of foreground objects. Therefore, after GP is well investigated on datasets with one object class, it would be meaningful to investigate whether the proposed GP methods can handle datasets with various object classes.

Two datasets (i.e. the SED2 dataset and the ImgSal dataset) were selected, because they are well explored benchmark object segmentation and detection datasets, and the images in both datasets contains diverse classes of objects. This thesis finds that the proposed GP based segmentation methods can evolve segmentors with useful class characteristic patterns to segment various types of objects simultaneously, but it is still challenging to handle highly varied datasets with various types of foreground objects.

### **Transfer Validation Tests**

It is found that the proposed GP based segmentation methods are able to evolve segmentors that can generalise well on images with the similar foreground objects. The reason is that the GP methods have captured common characteristics (e.g. the object shape and backgrounds) that similar images share (e.g. horse images and cow images). The ability to capture common characteristics across a class of similar foreground objects in a human interpretable manner is a major strength of GP based image segmentation.

### **Selected Features versus Constructed Features**

The best-performing feature selection method and feature construction method proposed in this thesis, i.e. SPGP-FS and CoevoMFC, were selected for comparison. This thesis finds that SPGP-FS can select feature subsets that are more effective than those constructed by CoevoMFC, and SPGP-FS uses less primitive features than those used to construct new features by CoevoMFC. In addition, SPGP-FS is more efficient than CoevoMFC as CoevoMFC requires feature transformation in fitness evaluation, which leads to high computation/time cost.

### **Comparison with Other Image Segmentation Techniques**

The main scope of this thesis is GP, but other non-EC segmentation techniques exist. Compared with deep neural networks, the major strengths of GP are that it does not require pre-defined solution models and large-scale datasets for training. Specifically, architectures of deep networks for given problems should be designed beforehand, while problem models are not required by GP. In addition, deep learning methods require appropriate datasets that should have large number of images and represent the given problems, which requires domain expertise, infrastructure resources and human labour. Moreover, compared with other traditional segmentation techniques, i.e. thresholding, K-means clustering, region growing and active contour model, the advantage of GP is that it can be easily adapted to diverse image domains without much parameter tuning or human intervention.

## **8.3 Future Work**

This section highlights key areas of future work that are enabled by the achieved contributions.

### 8.3.1 A Challenge for The Proposed GP Methods

This work shows that the segmentation results (binary images with foreground/background labelled as colour white/black) contain “holes” inside foreground objects, which means that inner parts of objects are detected incorrectly as background (see Chapter 3 and Chapter 4). However, the edges of objects are generally detected accurately. The reason may be that the information is not sufficient to recognise certain body parts of foreground objects. In contrast, the edge parts of objects contain more information, e.g. the edge information, which helps to detect them.

A post-processing stage can be employed to solve the above problem in the proposed methods by employing morphological operators, e.g. dilate, erode, open and close. Specifically, image opening and erosion are the dual of closing and dilation respectively. Image opening and erosion can remove foreground pixels from the edges of foreground regions and enlarge background holes in foreground regions, while closing and dilation can enlarge the boundaries of foreground regions and shrink background holes in foreground regions [68].

To avoid introducing an extra post-processing stage, investigations can be made to improve the proposed GP methods to generate segmentation results without “holes”. As the common framework of applying GP to figure-ground segmentation is to transform segmentation tasks to subimage/pixel classification problems, GP is used to evolve classifiers to classify each subimage (or pixel) as class foreground/background. Therefore, the spatial relations of subimages/pixels, e.g. pixel connectivity and region continuity, are lost. Considering that edge regions of foreground objects can generally be accurately separated, it is plausible that spatial relations of neighboring subimages/pixels can help remove “holes” on foregrounds/backgrounds. Therefore, how to use the spatial relations of subimages/pixels in the evolutionary process of GP deserves further investigations.

### 8.3.2 Filter GP Methods

As filter methods are not optimised to be used with a learning algorithm for specific tasks in fitness evaluation, so filter methods must discover features that a priori suit an algorithm's learning style. Therefore, it is worth developing effective fitness functions for filter methods.

As GP is commonly applied to figure-ground segmentation by transforming segmentation problems to binary classification tasks (classifying subimage/pixel as class foreground or background), two promising avenues for filter measures can be investigated. As a discriminant criterion function, Fisher criterion is the ratio of the between-class scatter to the within-class scatter. It may be possible to design fitness functions that can minimise within-class scatter and maximise the between-class scatter [95]. In addition, information theory (e.g. mutual information and entropy) [151] is used to measure the information of the random variables. It is plausible that this can also be used to develop fitness functions that can measure the relevance of image features with classes (background or foreground).

### 8.3.3 Datasets with Various Types of Objects

GP was well investigated on datasets containing one type of foreground objects in this work. As it is often the case that a dataset contain images with various types of foreground objects, the proposed GP based methods are tested on two benchmark datasets (SED2 and ImgSal) with diverse types of objects in Chapter 7. Compared with SED2 images, ImgSal images represent more difficult segmentation tasks, as ImgSal contains images with small salient regions, cluttered backgrounds and/or repeated distractors. Even though promising results are achieved on SED2, the proposed GP methods cannot perform close to optimum on ImgSal due to its high variation. This suggests that GP cannot evolve segmentors with distinctive patterns for highly varied classes of objects. Therefore, how to

improve the proposed GP methods to handle datasets containing varied classes of objects deserves further investigations.

One possible method is to improve the learning ability of the proposed GP methods by using ensemble learning techniques<sup>1</sup>, e.g. boosting. Specifically, boosting algorithms iteratively learn weak “rules” by re-weighting the training data to address certain examples (e.g. misclassified examples for a classification task), and then combine weak “rules” to form a strong “rule” [59]. Therefore, the proposed GP methods can be used to learn weak segmentors in boosting algorithms that can develop a strong segmentor eventually.

## 8.4 Chapter Summary

This chapter summarised the achieved objectives of this thesis in GP evolved segmentors, feature selection and construction, and further investigation of GP’s effectiveness in figure-ground segmentation. In addition, the major conclusions corresponding to the achieved objectives were also described. Moreover, key areas of future work were discussed in this chapter, i.e. avenues to improve the proposed GP methods, to build effective fitness functions for filter GP methods, and to enable GP to handle datasets with various object classes. Overall, this thesis shows that GP is suited to figure-ground image segmentation on complex images with high variations, and it is worth investigating the remaining issues in this field.

---

<sup>1</sup>Ensemble learning is to combine the predictions of multiple learning algorithms to obtain better predictive performance than that based on any single constituent learning algorithm [45].





# Bibliography

- [1] Active contour based segmentation. Available at <http://au.mathworks.com/help/images/ref/activecontour.html#btuep4x-7>.
- [2] Thresholding segmentation. Available at <http://au.mathworks.com/help/images/examples/correcting-nonuniform-illumination.html>.
- [3] AHMED, S., ZHANG, M., AND PENG, L. Enhanced feature selection for biomarker discovery in LC-MS data using GP. In *IEEE Congress on Evolutionary Computation (CEC)* (2013), IEEE, pp. 584–591.
- [4] AHMED, S., ZHANG, M., PENG, L., AND XUE, B. Multiple feature construction for effective biomarker identification and classification using genetic programming. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation* (2014), ACM, pp. 249–256.
- [5] AL-SAHAF, H., SONG, A., NESHATIAN, K., AND ZHANG, M. Extracting image features for classification by two-tier genetic programming. In *IEEE Congress on Evolutionary Computation (CEC)* (2012), IEEE, pp. 1–8.
- [6] ALBREGTSEN, F. Region & edge based segmentation. Available at <http://www.uio.no/studier/emner/>

matnat/ifi/INF4300/h11/undervisningsmateriale/  
INF4300-2011-f04-segmentation.pdf.

- [7] ALPAYDIN, E. *Introduction to machine learning*. MIT press, 2014.
- [8] ALPERT, S., GALUN, M., BRANDT, A., AND BASRI, R. Image segmentation by probabilistic bottom-up aggregation and cue integration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, 2 (2012), 315–327.
- [9] ANZAI, Y. *Pattern recognition and machine learning*. Elsevier, 2012.
- [10] ARBELAEZ, P., MAIRE, M., FOWLKES, C., AND MALIK, J. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 5 (2011), 898–916.
- [11] ARLOT, S., CELISSE, A., ET AL. A survey of cross-validation procedures for model selection. *Statistics surveys* 4 (2010), 40–79.
- [12] ASHBURNER, J., AND FRISTON, K. J. Unified segmentation. *Neuroimage* 26, 3 (2005), 839–851.
- [13] BÄCK, T., FOGEL, D. B., AND MICHALEWICZ, Z. *Evolutionary computation 1: Basic algorithms and operators*, vol. 1. CRC press, 2000.
- [14] BANIA, R. Survey on feature selection for data reduction. *International Journal of Computer Applications* 94, 18 (2014).
- [15] BANIMELHEM, O., AND YAHYA, Y. A. Multi-thresholding image segmentation using genetic algorithm. *Jordan University of Science and Technology,, Irbid, Jordan* (2011), 1–6.
- [16] BANZHAF, W., NORDIN, P., KELLER, R. E., AND FRANCONI, F. D. *Genetic Programming: An Introduction*. New York: Morgan Kaufmann, 1998.

- [17] BECKER, M., AND MAGNENAT-THALMANN, N. Deformable models in medical image segmentation. In *3D multiscale physiological human*. Springer, 2014, pp. 81–106.
- [18] BHANDARKAR, S. M., AND ZHANG, H. Image segmentation using evolutionary computation. *IEEE Transactions on Evolutionary Computation* 3, 1 (1999), 1–21.
- [19] BILOTTA, E., CERASA, A., PANTANO, P., QUATTRONE, A., STAINO, A., AND STRAMANDINOLI, F. A cnn based algorithm for the automated segmentation of multiple sclerosis lesions. *Applications of Evolutionary Computation* (2010), 211–220.
- [20] BISHOP, A., CIESIELSKI, V., AND TRIST, K. Feature construction using genetic programming for classification of images by aesthetic value. In *International Conference on Evolutionary and Biologically Inspired Music and Art* (2014), Springer, pp. 62–73.
- [21] BLEULER, S., BRACK, M., THIELE, L., AND ZITZLER, E. Multiobjective genetic programming: Reducing bloat using SPEA2. In *Proceedings of the Congress on Evolutionary Computation* (2001), vol. 1, IEEE, pp. 536–543.
- [22] BORENSTEIN, E. Weizmann horse database. Available at <http://www.msri.org/people/members/eranb/>.
- [23] BORENSTEIN, E., AND ULLMAN, S. Class-specific, top-down segmentation. In *Proceedings of the 7th European Conference on Computer Vision, ECCV*. Springer, 2002, pp. 109–122.
- [24] BORENSTEIN, E., AND ULLMAN, S. Learning to segment. In *European Conference on Computer Vision* (2004), Springer, pp. 315–328.
- [25] BORENSTEIN, E., AND ULLMAN, S. Combined top-down/bottom-up segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, 12 (2008), 2109–2125.

- [26] BRANKE, J. Consideration of partial user preferences in evolutionary multiobjective optimization. In *Multiobjective Optimization*. Springer, 2008, pp. 157–178.
- [27] BUI, L. T., ESSAM, D., ABBASS, H. A., AND GREEN, D. Performance analysis of evolutionary multi-objective optimization methods in noisy environments. In *Proceedings of the 8th Asia Pacific symposium on intelligent and evolutionary systems* (2004), Citeseer, pp. 29–39.
- [28] CANO, A., VENTURA, S., AND CIO, K. J. Multi-objective genetic programming for feature extraction and data visualization. *Soft Computing* (2015), 1–21.
- [29] CHAN, T. F., SHEN, J., AND VESE, L. Variational pde models in image processing. *Notices AMS* 50, 1 (2003), 14–26.
- [30] CHANDRASHEKAR, G., AND SAHIN, F. A survey on feature selection methods. *Computers & Electrical Engineering* 40, 1 (2014), 16–28.
- [31] CHEN, B., CHEN, L., AND CHEN, Y. Efficient ant colony optimization for image feature selection. *Signal processing* 93, 6 (2013), 1566–1576.
- [32] CHEN, L., CHEN, B., AND CHEN, Y. Image feature selection based on ant colony optimization. In *Australasian Joint Conference on Artificial Intelligence* (2011), Springer, pp. 580–589.
- [33] CHEN, Y., PENG, B., AND LIU, D. Fully automated ultrasound common carotid artery segmentation using active shape model. *International Journal of Signal Processing, Image Processing and Pattern Recognition* 7, 2 (2014), 99–106.
- [34] COELLO, C. C. Evolutionary multi-objective optimization: a historical view of the field. *IEEE Computational Intelligence Magazine* 1, 1 (2006), 28–36.

- [35] CRISTIANINI, N. Supervised and unsupervised learning. In *Dictionary of Bioinformatics and Computational Biology*, Wiley Online Library.
- [36] CUEVAS, E., OSUNA, V., AND OLIVA, D. *Evolutionary Computation Techniques: A Comparative Perspective*, vol. 686. Springer, 2017.
- [37] DAS, S., AND PANIGRAHI, B. K. Multi-objective evolutionary algorithms. *Encyclopedia of Artificial Intelligence* 3 (2009), 1145–1151.
- [38] DASH, M., AND LIU, H. Feature selection for classification. *Intelligent Data Analysis* 1, 1-4 (1997), 131–156.
- [39] DASH, M., LIU, H., AND MOTODA, H. Consistency based feature selection. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (2000), Springer, pp. 98–109.
- [40] DAVIDSON, J., SAVIC, D., AND WALTERS, G. Method for the identification of explicit polynomial formulae for the friction in turbulent pipe flow. *Journal of Hydroinformatics* 1 (1999), 115–126.
- [41] DAVIS, R. A., CHARLTON, A. J., OEHLISCHLAGER, S., AND WILSON, J. C. Novel feature selection method for genetic programming using metabolomic <sup>1</sup>H NMR data. *Chemometrics and Intelligent Laboratory Systems* 81, 1 (2006), 50–59.
- [42] DEB, K. Multi-objective evolutionary algorithms. In *Springer Handbook of Computational Intelligence*. Springer, 2015, pp. 995–1015.
- [43] DEB, K., AND JAIN, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Trans. Evolutionary Computation* 18, 4 (2014), 577–601.

- [44] DEB, K., PRATAP, A., AGARWAL, S., AND MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 182–197.
- [45] DIETTERICH, T. G., ET AL. Ensemble methods in machine learning. *Multiple Classifier Systems* 1857 (2000), 1–15.
- [46] DONALEK, C. Supervised and unsupervised learning. Available at [http://www.astro.caltech.edu/~george/aybil99/Donalek\\_Classif.pdf](http://www.astro.caltech.edu/~george/aybil99/Donalek_Classif.pdf).
- [47] DONG, M., ERAMIAN, M. G., LUDWIG, S. A., AND PIERSON, R. A. Automatic detection and segmentation of bovine corpora lutea in ultrasonographic ovarian images using genetic programming and rotation invariant local binary patterns. *Medical & Biological Engineering & Computing* 51, 4 (2013), 405–416.
- [48] EDELKAMP, S., AND SCHRODL, S. *Heuristic Search: Theory and Applications*. Elsevier, 2012.
- [49] EIBEN, A. E., AND SMITH, J. E. *Introduction to evolutionary computing*, vol. 53. Springer, 2003.
- [50] ELEYAN, A., AND DEMIREL, H. Co-occurrence matrix and its statistical features as a new approach for face recognition. *Turk J Elec Eng & Comp Sci* 19, 1 (2011), 97–107.
- [51] EMAMBAKHSI, M., EBRAHIMNEZHAD, H., AND SEDAAGHI, M. H. A hybrid top-down/bottom-up approach for image segmentation incorporating colour and texture with prior shape knowledge. In *18th Iranian Conference on Electrical Engineering (ICEE)* (2010), IEEE, pp. 270–275.
- [52] ESPEJO, P. G., VENTURA, S., AND HERRERA, F. A survey on the application of genetic programming to classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C* 40, 2 (2010), 121–144.

- [53] EVERINGHAM, M., ESLAMI, S. A., VAN GOOL, L., WILLIAMS, C. K., WINN, J., AND ZISSERMAN, A. The Pascal visual object classes challenge: a retrospective. *International Journal of Computer Vision* 111, 1 (2015), 98–136.
- [54] FLEET, D., AND WEISS, Y. Optical flow estimation. *Handbook of mathematical models in computer vision* (2006), 237–257.
- [55] FONSECA, C. M., FLEMING, P. J., ET AL. Genetic algorithms for multiobjective optimization: Formulation discussion and generalization. In *Icga* (1993), vol. 93, pp. 416–423.
- [56] FONSECA, P. K-means image segmentation. Available at <http://www.mathworks.com/matlabcentral/fileexchange/authors/129300>.
- [57] FREEMAN, B., AND DURAND, F. Gradient image processing. Available at [http://groups.csail.mit.edu/graphics/classes/CompPhoto06/html/lecturenotes/10\\_Gradient.pdf](http://groups.csail.mit.edu/graphics/classes/CompPhoto06/html/lecturenotes/10_Gradient.pdf).
- [58] FREIDLIN, B., AND GASTWIRTH, J. L. Should the median test be retired from general use? *The American Statistician* 54 (2000), 161–164.
- [59] FREUND, Y., SCHAPIRE, R., AND ABE, N. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence* 14, 771–780 (1999), 1612.
- [60] FU, H., XIAO, Z., DELLANDRÉA, E., DOU, W., AND CHEN, L. Image categorization using ESFS: a new embedded feature selection method based on SFS. In *International Conference on Advanced Concepts for Intelligent Vision Systems* (2009), Springer, pp. 288–299.

- [61] GARCIA-GARCIA, A., ORTS-ESCOLANO, S., OPREA, S., VILLENA-MARTINEZ, V., AND GARCIA-RODRIGUEZ, J. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857* (2017).
- [62] GILL, G., TOEWS, M., AND BEICHEL, R. R. Robust initialization of active shape models for lung segmentation in CT scans: A feature-based atlas approach. *Journal of Biomedical Imaging 2014* (2014), 13.
- [63] GOLOVANOV, N. *Geometric Modeling: The mathematics of shapes*. Create Space Independent Publishing Platform, 2014.
- [64] GUO, H., JACK, L. B., AND NANDI, A. K. Feature generation using genetic programming with application to fault classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 35, 1 (2005), 89–99.
- [65] GUO, H., AND NANDI, A. K. Breast cancer diagnosis using genetic programming generated feature. *Pattern Recognition* 39, 5 (2006), 980–987.
- [66] HALL, M., FRANK, E., HOLMES, G., PFAHRINGER, B., REUTEMANN, P., AND WITTEN, I. H. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter* 11, 1 (2009), 10–18.
- [67] HE, D., AND SAFIA, A. Brodatz texture database. Available at [http://multibandtexture.recherche.usherbrooke.ca/original\\_brodatz.html](http://multibandtexture.recherche.usherbrooke.ca/original_brodatz.html).
- [68] HEIJMANS, H. J. Connected morphological operators for binary images. *Computer Vision and Image Understanding* 73, 1 (1999), 99–120.
- [69] HORN, J., NAFPLIOTIS, N., AND GOLDBERG, D. E. A niched pareto genetic algorithm for multiobjective optimization. In *IEEE World Congress on Computational Intelligence* (1994), Ieee, pp. 82–87.



- [70] ILONEN, J., EEROLA, T., MUTIKAINEN, H., LENSU, L., KÄYHKÖ, J., AND KÄLVIÄINEN, H. Estimation of bubble size distribution based on power spectrum. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Springer, 2014, pp. 38–45.
- [71] JAIN, A. *Scene and Video Understanding*. PhD thesis, University of Maryland, College Park, 2014.
- [72] JAIN, H., AND DEB, K. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach. *IEEE Trans. Evolutionary Computation* 18, 4 (2014), 602–622.
- [73] JIANG, X., ZHANG, R., AND NIE, S. Image segmentation based on pdes model: A survey. In *3rd International Conference on Bioinformatics and Biomedical Engineering, ICBBE 2009* (2009), IEEE, pp. 1–4.
- [74] JIAO, L. Evolutionary-based image segmentation methods. In *Image segmentation*. InTech, 2011.
- [75] JOLLIFFE, I. T. Principal component analysis and factor analysis. *Principal component analysis* (2002), 150–166.
- [76] KAGANAMI, H. G., AND BEIJI, Z. Region-based segmentation versus edge detection. In *Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP'09* (2009), IEEE, pp. 1217–1221.
- [77] KAMARAINEN, J. Gabor features in image analysis. In *3rd International Conference on Image Processing Theory, Tools and Applications (IPTA)* (2012), IEEE, pp. 13–14.
- [78] KANUNGO, P., NANDA, P., AND SAMAL, U. Image segmentation using thresholding and genetic algorithm.

- [79] KASS, M., WITKIN, A., AND TERZOPOULOS, D. Snakes: Active contour models. *International Journal of Computer Vision* 1, 4 (1988), 321–331.
- [80] KHAN, W. Image segmentation techniques: A survey. *Journal of Image and Graphics* 1, 4 (2013), 166–170.
- [81] KLETTE, R. *Concise computer vision: an introduction into theory and algorithms*. Springer Science & Business Media, 2014.
- [82] KOZA, J. R. What is genetic programming (GP)? Available at <http://www.genetic-programming.com/>.
- [83] KOZA, J. R. *Genetic programming: on the programming of computers by means of natural selection*, vol. 1. MIT press, 1992.
- [84] KOZA, J. R. *Genetic programming III: Darwinian invention and problem solving*, vol. 3. Morgan Kaufmann, 1999.
- [85] KOZA, J. R., KEANE, M. A., STREETER, M. J., MYDLOWEC, W., YU, J., AND LANZA, G. *Genetic programming IV: Routine human-competitive machine intelligence*, vol. 5. Springer Science & Business Media, 2006.
- [86] KRAWIEC, K., AND BHANU, B. Coevolution and linear genetic programming for visual learning. In *Genetic and Evolutionary Computation Conference* (2003), Springer, pp. 332–343.
- [87] KROON, D. Region growing. Available at <http://www.mathworks.com/matlabcentral/fileexchange/19084-region-growing>.
- [88] KUETTEL, D., AND FERRARI, V. Figure-ground segmentation by transferring window masks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2012), IEEE, pp. 558–565.

- [89] KUHN, M. Futility analysis in the cross-validation of machine learning models. *Computing Research Repository* 1405.6974 (2014).
- [90] KUMAR, D. S., AND CHAUHAN, E. A. A survey on image feature selection techniques. *International Journal of Computer Science and Information Technologies (IJCSIT)* 5, 5 (2014), 6449–6452.
- [91] KUMAR, M., SAXENA, R., ET AL. Algorithm and technique on various edge detection: A survey. *Signal & Image Processing* 4, 3 (2013), 65.
- [92] KUMAR, V., AND MINZ, S. Feature selection. *SmartCR* 4, 3 (2014), 211–229.
- [93] LANE, M. C., XUE, B., LIU, I., AND ZHANG, M. Particle swarm optimisation and statistical clustering for feature selection. In *Australasian Conference on Artificial Intelligence* (2013), Springer, pp. 214–220.
- [94] LEVINE, M. D., AN, X., AND HE, H. Saliency detection based on frequency and spatial domain analysis. *Neuroscience* 8, 8 (2005), 975–977.
- [95] LI, STAN Z. AND JAIN, A. Fisher criterion. In *Encyclopedia of Biometrics*. Springer, 2009, pp. 549–549.
- [96] LIANG, Y., ZHANG, M., AND BROWNE, W. N. A supervised figure-ground segmentation method using genetic programming. In *European Conference on the Applications of Evolutionary Computation* (2015), Springer, pp. 491–503.
- [97] LIANG, Y., ZHANG, M., AND BROWNE, W. N. Figure-ground image segmentation using genetic programming and feature selection. In *IEEE Congress on Evolutionary Computation (CEC)* (2016), IEEE, pp. 3839–3846.

- [98] LIANG, Y., ZHANG, M., AND BROWNE, W. N. Multi-objective genetic programming for figure-ground image segmentation. In *Australasian Conference on Artificial Life and Computational Intelligence* (2016), Springer, pp. 134–146.
- [99] LIANG, Y., ZHANG, M., AND BROWNE, W. N. Feature construction using genetic programming for figure-ground image segmentation. In *Proceedings of Intelligent and Evolutionary Systems: The 20th Asia Pacific Symposium, IES 2016* (2017), Springer, pp. 237–250.
- [100] LIANG, Y., ZHANG, M., AND BROWNE, W. N. Genetic programming for evolving figure-ground segmentors from multiple features. *Applied Soft Computing* 51 (2017), 83–95.
- [101] LIM, S. H., WANG, L.-L., AND DEJONG, G. Explanation-based feature construction. In *IJCAI* (2007), vol. 7, pp. 931–936.
- [102] LIN, C.-H., CHEN, H.-Y., AND WU, Y.-S. Study of image retrieval and classification based on adaptive features using genetic algorithm feature selection. *Expert Systems with Applications* 41, 15 (2014), 6611–6621.
- [103] LIU, C.-Y., IGLESIAS, J. E., TU, Z., INITIATIVE, A. D. N., ET AL. Deformable templates guided discriminative models for robust 3D brain MRI segmentation. *Neuroinformatics* 11, 4 (2013), 447–468.
- [104] LIU, H., AND YU, L. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering* 17, 4 (2005), 491–502.
- [105] LIU, J., AND WANG, J. Application of snake model in medical image segmentation. *J. Convergence Inf. Technol* 9, 1 (2014), 105–109.
- [106] LIU, T., XU, H., JIN, W., LIU, Z., ZHAO, Y., AND TIAN, W. Medical image segmentation based on a hybrid region-based active con-

- tour model. *Computational and mathematical methods in medicine* 2014 (2014).
- [107] LIZÁRRAGA, G. L., AND RIONDA, S. B. On the diversity of non-dominated sets. Available at <http://www.micai.org/2009/proceedings/complementary/cd/ws-imso/191/diversity.pdf>.
- [108] LONG, J., SHELHAMER, E., AND DARRELL, T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 3431–3440.
- [109] LU, X., TANG, K., SENDHOFF, B., AND YAO, X. A review of concurrent optimisation methods. *International Journal of Bio-Inspired Computation* 6, 1 (2014), 22–31.
- [110] LUKE, S. The ecj owners manual. *San Francisco, California, A user manual for the ECJ Evolutionary Computation Library* (2010), 1–206.
- [111] LUKE, S., AND PANAIT, L. Lexicographic parsimony pressure. In *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation* (2002), Morgan Kaufmann Publishers Inc., pp. 829–836.
- [112] LUKE, S., AND PANAIT, L. A comparison of bloat control methods for genetic programming. *Evolutionary Computation* 14, 3 (2006), 309–344.
- [113] MALDONADO, S., AND WEBER, R. Embedded feature selection for support vector machines: state-of-the-art and future challenges. In *Iberoamerican Congress on Pattern Recognition* (2011), Springer, pp. 304–311.
- [114] MANIKANDAN, P., AND VENKATESWARAN, C. J. Feature selection algorithms: Literature review. *International Journal* 5, 3 (2015).

- [115] MAO-GUO, G., LI-CHENG, J., DONG-DONG, Y., AND WEN-PING, M. Evolutionary multi-objective optimization algorithms. *Journal Software* 20, 2 (2009), 271–289.
- [116] MAYFIELD, E., AND PENSTEIN-ROSÉ, C. Using feature construction to avoid large feature spaces in text classification. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation* (2010), ACM, pp. 1299–1306.
- [117] MCCLUNEY, W. R. *Introduction to radiometry and photometry*. Artech House, 2014.
- [118] MCKNIGHT, P. E., AND NAJAB, J. Mann-Whitney U test. *Corsini Encyclopedia of Psychology* (2010), 1.
- [119] MICHALEWICZ, Z., AND MICHALEWICZ, M. Evolutionary computation techniques and their applications. In *IEEE International Conference on Intelligent Processing Systems, ICIPS'97* (1997), vol. 1, IEEE, pp. 14–25.
- [120] MICHALSKI, R. S., CARBONELL, J. G., AND MITCHELL, T. M. *Machine learning: An artificial intelligence approach*. Springer Science & Business Media, 2013.
- [121] MUNI, D. P., PAL, N. R., AND DAS, J. Genetic programming for simultaneous feature selection and classifier design. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 36, 1 (2006), 106–117.
- [122] MURPHY, K. P. *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [123] NAG, K., AND PAL, N. R. A multiobjective genetic programming-based ensemble for simultaneous feature selection and classification. *IEEE Transactions on Cybernetics* 46, 2 (2015), 499–510.

- [124] NAG, K., AND PAL, N. R. A multiobjective genetic programming-based ensemble for simultaneous feature selection and classification. *IEEE transactions on cybernetics* 46, 2 (2016), 499–510.
- [125] NANDI, G. An enhanced approach to las vegas filter (LVF) feature selection algorithm. In *2011 2nd National Conference on Emerging Trends and Applications in Computer Science (NCETACS)* (2011), IEEE, pp. 1–3.
- [126] NASERIPARSA, M., BIDGOLI, A.-M., AND VARAEE, T. A hybrid feature selection method to improve performance of a group of classification algorithms. *arXiv preprint arXiv:1403.2372* (2014).
- [127] NESHTATIAN, K. Feature manipulation with genetic programming. *Doctoral Dissertation, Victoria University of Wellington* (2010).
- [128] NESHTATIAN, K., AND ZHANG, M. Using genetic programming for context-sensitive feature scoring in classification problems. *Connection Science* 23, 3 (2011), 183–207.
- [129] NESHTATIAN, K., ZHANG, M., AND ANDREAEE, P. A filter approach to multiple feature construction for symbolic learning classifiers using genetic programming. *IEEE Transactions on Evolutionary Computation* 16, 5 (2012), 645–661.
- [130] OJALA, T., PIETIKAINEN, M., AND MAENPAA, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 7 (2002), 971–987.
- [131] PAREKH, H. S., THAKORE, D. G., AND JALIYA, U. K. A survey on object detection and tracking methods. *International Journal of Innovative Research in Computer and Communication Engineering* 2, 2 (2014), 2970–2979.

- [132] PIETIKÄINEN, M., HADID, A., ZHAO, G., AND AHONEN, T. Local binary patterns for still images. In *Computer Vision Using Local Binary Patterns*. Springer, 2011, pp. 13–47.
- [133] POLI, R. Genetic programming for image analysis. In *Proceedings of the 1st Annual Conference on Genetic Programming* (1996), MIT Press, pp. 363–368.
- [134] POLI, R., LANGDON, W. B., MCPHEE, N. F., AND KOZA, J. R. *A field guide to genetic programming*. Lulu. com, 2008.
- [135] POLI, R., LANGDON, W. B., AND N.F.M. *A Field Guide to Genetic Programming*. Published via <http://lulu.com> and freely available at <http://www.gp-field-guide.org.uk>, UK, 2008.
- [136] POLI, R., AND MCPHEE, N. F. Covariant parsimony pressure in genetic programming. Tech. rep., Technical Report CES-480, Department of Computing and Electronic Systems, University of Essex, 2008.
- [137] REMONDINO, F., SPERA, M. G., NOCERINO, E., MENNA, F., AND NEX, F. State of the art in high density image matching. *The Photogrammetric Record* 29, 146 (2014), 144–166.
- [138] REN, X., AND GU, C. Figure-ground segmentation improves handled object recognition in egocentric video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2010), vol. 2, IEEE, pp. 3137–3144.
- [139] REN, Z., GAO, S., CHIA, L.-T., AND TSANG, I. W.-H. Region-based saliency detection and its application in object recognition. *IEEE Transactions on Circuits and Systems for Video Technology* 24, 5 (2014), 769–779.



- [140] ROBERTS, M. E. The effectiveness of cost based subtree caching mechanisms in typed genetic programming for image segmentation. In *Applications of Evolutionary Computing*. Springer, 2003, pp. 444–454.
- [141] ROBNIK-ŠIKONJA, M., AND KONONENKO, I. Theoretical and empirical analysis of relieff and rrelieff. *Machine learning* 53, 1-2 (2003), 23–69.
- [142] ROTH, V., AND LANGE, T. Adaptive feature selection in image segmentation. In *Joint Pattern Recognition Symposium* (2004), Springer, pp. 9–17.
- [143] RUSSELL, S. J., AND NORVIG, P. *Artificial intelligence: a modern approach (International Edition)*. Pearson US Imports & PHIPES, 2002.
- [144] RUSU, A.-N., ET AL. A survey on edge detection in images. *Journal of Information Systems & Operations Management* (2016), 1.
- [145] SAHA, I., MAULIK, U., AND BANDYOPADHYAY, S. An improved multi-objective technique for fuzzy clustering with application to iris image segmentation. *Applications of Evolutionary Computing* (2009), 426–431.
- [146] SÁNCHEZ-MAROÑO, N., ALONSO-BETANZOS, A., AND TOMBILLA-SANROMÁN, M. Filter methods for feature selection—a comparative study. In *International Conference on Intelligent Data Engineering and Automated Learning* (2007), Springer, pp. 178–187.
- [147] SARRO, F., FERRUCCI, F., AND GRAVINO, C. Single and multi objective genetic programming for software development effort estimation. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing* (2012), ACM, pp. 1221–1226.
- [148] SASAKI, Y., ET AL. The truth of the f-measure. *Teach Tutor Mater* 1, 5 (2007).

- [149] SEN-CHING, S. C., AND KAMATH, C. Robust techniques for background subtraction in urban traffic video. In *Electronic Imaging 2004* (2004), International Society for Optics and Photonics, pp. 881–892.
- [150] SENTHILKUMARAN, N., AND RAJESH, R. Edge detection techniques for image segmentation—a survey of soft computing approaches. In *International Journal of Recent Trends in Engineering* 1, 2 (2009).
- [151] SHANNON, C. E. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review* 5, 1 (2001), 3–55.
- [152] SHAO, L., LIU, L., AND LI, X. Feature learning for image classification via multiobjective genetic programming. *IEEE Transactions on Neural Networks and Learning Systems* 25, 7 (2014), 1359–1371.
- [153] SHAW, D., STONE, K., HO, K., KELLER, J. M., LUKE, R. H., AND BURNS, B. P. Sequential feature selection for detecting buried objects using forward looking ground penetrating radar. In *SPIE Defense+ Security* (2016), International Society for Optics and Photonics, pp. 98231L–98231L.
- [154] SHIRAKAWA, S., AND NAGAO, T. Evolutionary image segmentation based on multiobjective clustering. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on* (2009), IEEE, pp. 2466–2473.
- [155] SHU, H., LUO, L., AND COATRIEUX, J.-L. Moment-based approaches in imaging. part 1, basic features. *IEEE Engineering in Medicine and Biology Magazine* 26, 5 (2007), 70.
- [156] SIA, F., AND ALFRED, R. Evolutionary-based feature construction with substitution for data summarization using dara. In *Data Mining and Optimization (DMO), 2012 4th Conference on* (2012), IEEE, pp. 53–58.

- [157] SINGH, T., KHARMA, N., DAOUD, M., AND WARD, R. Genetic programming based image segmentation with applications to biomedical object detection. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation* (2009), ACM, pp. 1123–1130.
- [158] SMART, O., AND BURRELL, L. Genetic programming and frequent itemset mining to identify feature selection patterns of iEEG and fMRI epilepsy data. *Engineering Applications of Artificial Intelligence* 39 (2015), 198–214.
- [159] SONDHI, P. Feature construction methods: a survey. *sifaka. cs. uiuc. edu* 69 (2009), 70–71.
- [160] SONG, A., AND CIESIELSKI, V. Fast texture segmentation using genetic programming. In *The Congress on Evolutionary Computation* (2003), vol. 3, IEEE, pp. 2126–2133.
- [161] SONG, A., AND CIESIELSKI, V. Texture segmentation by genetic programming. *Evolutionary Computation* 16, 4 (2008), 461–481.
- [162] SONKA, M., HLAVAC, V., AND BOYLE, R. *Image processing, analysis, and machine vision*. Cengage Learning, 2014.
- [163] SOULE, T., AND FOSTER, J. A. Effects of code growth and parsimony pressure on populations in genetic programming. *Evolutionary Computation* 6, 4 (1998), 293–309.
- [164] SUTTON, R. S., AND BARTO, A. G. *Reinforcement learning: An introduction*, vol. 1. MIT press Cambridge, 1998.
- [165] TALAVERA, L. An evaluation of filter and wrapper methods for feature selection in categorical clustering. In *International Symposium on Intelligent Data Analysis* (2005), Springer, pp. 440–451.
- [166] VANNESCHI, L., CASTELLI, M., AND SILVA, S. Measuring bloat, overfitting and functional complexity in genetic programming. In

- Proceedings of the 12th annual conference on Genetic and evolutionary computation* (2010), ACM, pp. 877–884.
- [167] VENKATADRI, M., SASTRY, H. G., AND REDDY, L. C. A survey on genetic programming in data mining tasks. *International Journal of Advanced Research in Computer Science* 3, 2 (2012).
- [168] WANG, B., SINGH, H. K., AND RAY, T. A multi-objective genetic programming approach to uncover explicit and implicit equations from data. In *IEEE Congress on Evolutionary Computation (CEC)* (2015), IEEE, pp. 1129–1136.
- [169] WANG, L., AND PAN, C. Robust level set image segmentation via a local correntropy-based k-means clustering. *Pattern Recognition* 47, 5 (2014), 1917–1925.
- [170] WANG, L., YU, Z., AND PAN, C. Medical image segmentation based on novel local order energy. In *Asian Conference on Computer Vision* (2010), Springer, pp. 148–159.
- [171] WITTEN, I. H., FRANK, E., HALL, M. A., AND PAL, C. J. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.
- [172] WU, Y. An introduction to computer vision. Available at <http://users.eecs.northwestern.edu/~yingwu/teaching/EECS432/Notes/intro.pdf>.
- [173] XUE, B., ZHANG, M., DAI, Y., AND BROWNE, W. N. Pso for feature construction and binary classification. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation* (2013), ACM, pp. 137–144.
- [174] YAN, K., MA, L., DAI, Y., SHEN, W., AND JI, Z. Cost-sensitive and sequential feature selection for chiller fault detection and diagnosis. *International Journal of Refrigeration* (2017).

- [175] YEOH, T., ZAPOTECAS-MARTÍNEZ, S., AKIMOTO, Y., AGUIRRE, H., AND TANAKA, K. Genetic algorithm assisted by a svm for feature selection in gait classification. In *Intelligent Signal Processing and Communication Systems (ISPACS), 2014 International Symposium on* (2014), IEEE, pp. 191–195.
- [176] YU, S., AND SHI, J. Object-specific figure-ground segregation. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2003), vol. 2, IEEE, pp. II–39.
- [177] ZABINSKY, Z. B. Random search algorithms. *Wiley Encyclopedia of Operations Research and Management Science* (2009).
- [178] ZHANG, D., WONG, A., INDRAWAN, M., AND LU, G. Content-based image retrieval using gabor texture features. *IEEE Transactions PAMI* (2000), 13–15.
- [179] ZHANG, J., MA, X., CHEN, Y., FANG, L., AND WANG, J. Mr image segmentation based on local information entropy geodesic active contour model. *International Journal of Multimedia and Ubiquitous Engineering* 9, 4 (2014), 127–136.
- [180] ZHANG, J., ZHAN, Z.-H., LIN, Y., CHEN, N., GONG, Y.-J., ZHONG, J.-H., CHUNG, H. S., LI, Y., AND SHI, Y.-H. Evolutionary computation meets machine learning: A survey. *IEEE Computational Intelligence Magazine* 6, 4 (2011), 68–75.
- [181] ZHANG, M., DING, C. H., ZHANG, Y., AND NIE, F. Feature selection at the discrete limit. In *AAAI* (2014), pp. 1355–1361.
- [182] ZITZLER, E., BROCKHOFF, D., AND THIELE, L. The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In *International Conference on Evolutionary Multi-Criterion Optimization* (2007), Springer, pp. 862–876.

- [183] ZITZLER, E., LAUMANN, M., AND THIELE, L. Spea2: Improving the strength pareto evolutionary algorithm. In *Proc. Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems* (2001), pp. 95–100.
- [184] ZOU, W., BAI, C., KPALMA, K., AND RONSIN, J. Online global transfer for automatic figure-ground segmentation. *IEEE Transactions on Image Processing* 23, 5 (2014), 2109–2121.