

Polyp Segmentation in Colonoscopy Images with Convolutional Neural Networks

by

郭运博

Yunbo Guo

A thesis submitted in partial fulfilment for the requirements for the degree of

Doctor of Philosophy

at

the University of Central Lancashire

October 2019

Student Declaration

Concurrent registration for two or more academic awards

Either *I declare that while registered as a candidate for the research degree, I have not been a registered candidate or enrolled student for another award of the University or other academic or professional institution

or ~~*I declare that while registered for the research degree, I was with the University's specific permission, a *registered candidate/*enrolled student for the following award:~~

Material submitted for another award

Either *I declare that no material contained in the thesis has been used in any other submission for an academic award and is solely my own work

or ~~*I declare that the following material contained in the thesis formed part of a submission for the award of:~~

—
(state award and awarding body and list the material below):

Collaboration

Where a candidate's research programme is part of a collaborative project, the thesis must indicate in addition clearly the candidate's individual contribution and the extent of the collaboration. Please state below:

Signature of Candidate _____ Yunbo Guo _____
Type of Award _____ Doctor of Philosophy _____
School _____ School of Engineering _____

Abstract

The thesis looks at approaches to segmentation of polyps in colonoscopy images. The aim was to investigate and develop methods that are robust, accurate and computationally efficient and which can compete with the current state-of-the-art in polyp segmentation.

Colorectal cancer is one of the leading cause of cancer deaths worldwide. To decrease mortality, an assessment of polyp malignancy is performed during colonoscopy examination so polyps can be removed at an early stage. In current routine clinical practice, polyps are detected and delineated manually in colonoscopy images by highly trained clinicians. To automate these processes, machine learning and computer vision techniques have been utilised. They have been shown to improve polyp detectability and segmentation objectivity. However, polyp segmentation is a very challenging task due to inherent variability of polyp morphology and colonoscopy image appearance.

This research considers a range of approaches to polyp segmentation – seeking out those that offer a best compromise between accuracy and computational complexity. Based on analysis of existing machine learning and polyp image segmentation techniques, a novel hybrid deep learning segmentation method is proposed to alleviate the impact of the above stated challenges on polyp segmentation. The method consists of two fully convolutional networks. The first proposed network is based on a compact architecture with large receptive fields and multiple classification paths. The method performs well on most images, accurately segmenting polyps of diverse morphology and appearance. However, this network is prone to misdetection of very small polyps. To solve this problem, a second network is proposed, which primarily aims to improve sensitivity to small polyp details by emphasising low-level image features.

In order to fully utilise information contained in the available training dataset, comprehensive data augmentation techniques are adopted. To further improve the performance of the proposed segmentation methods, test-time data augmentation is also implemented.

A comprehensive multi-criterion analysis of the proposed methods is provided. The result demonstrates that the new methodology has better accuracy and robustness than the current state-of-the-art, as proven by the outstanding performance at the 2017 and 2018 GIANA polyp segmentation challenges.

Acknowledgements

I would like to express my sincere gratitude to my director of studies, Professor Bogdan Matuszewski. He has introduced me to the field of machine learning and computer vision and led me through the turns of my research. During my Ph.D. studies, he has provided great support and help me in making my research proceed smoothly. Besides, he has given me many opportunities to present my research, helping me to become more confident researcher.

I would like to express my gratitude to Dr. Pedro Henriquez Castellano, who has always kindly helped me solve problems in computer technologies.

I would like to express my gratitude to Professor Darren Ansell. Though we have not known each other for too long, he is always glad to know about my research and was very supportive during the thesis revisions.

I want to express my gratitude for the organizers of the GIANA challenge, who have given me a chance to show my abilities.

I would like to express my special gratitude to my parents, who have given me a chance to start again when I met with great difficulties and given me all the help and I needed. Whenever I was down, they would listen to me patiently and helped me to feel better. I think it would be hard for me to finish my study without their constant encouragement.

At last, I would like to express my gratitude to my colleague Liyang Wang. After tense research, we often relax together. When meeting with problems, we will encourage each other. In the past years, we have numerous memorable moments of our and I hope we will remain friends forever.

Contents

Abstract	ii
Acknowledgements	iii
Contents	iv
List of Figures	viii
List of Tables	xii
Chapter 1. Introduction	1
1.1 Aim and motivation	1
1.2 Colon cancer and colonoscopy	3
1.3 Image segmentation	6
1.4 Machine learning	9
1.5 Deep learning	11
1.6 Novel contributions	12
1.7 Thesis organization	13
Chapter 2. Polyp segmentation problem	14
2.1 Introduction	14
2.2 Polyp morphology	15
2.2 Review of polyp segmentation methods	18
2.2.1 Polyp segmentation based on shape	18
2.2.2 Polyp segmentation based on texture appearance	19
2.2.3 Polyp segmentation based on deep learning	20
2.4 Summary	24
Chapter 3. Artificial neural network and convolutional neural network	25
3.1 Introduction	25
3.2 Multilayer perceptron	26
3.2.1 Feedforward structure	26
3.2.2 Backpropagation	28
3.2.3 Activation function	31

3.2.4 Vanishing and exploding gradients	34
3.2.5 Multi-class and softmax	35
3.2.6 Loss function	36
3.3 Convolutional neural network	37
3.3.1 Convolutional layer	38
3.3.2 Sub-sampling	43
3.3.3 CNN structure	44
3.3.4 Dropout	45
3.3.5 Normalization	46
3.5 Typical CNN architecture	48
3.5.1 AlexNet & ZFnet	48
3.5.2 All convolution network	49
3.5.3 VGG16	49
3.5.4 Network in network	50
2.5.5 Inception	50
3.5.6 Deep residual network	52
3.5.7 Squeeze and excitation networks	53
3.6 Summary	54
Chapter 4. Review of image segmentation methods	55
4.1 Introduction	55
4.2 Traditional image segmentation methods	56
4.2.1 Thresholding	56
4.2.2 Clustering	57
4.2.3 Region growing	58
4.2.5 Machine learning and image segmentation	59
4.3 Image segmentation with fully convolutional networks	60
4.4 Unet	64
4.5 Deep segmentation networks architecture	65
4.5.1 DeepLab	65

4.5.2 SegNet	69
4.5.3 Global convolutional network	69
4.5.4 Pyramid scene parsing network	70
4.5.5 RefineNet.....	71
4.5.6 Deep contour-aware networks.....	71
4.5.7 Discriminative feature network.....	72
4.6 Summary.....	73
Chapter 5. Proposed polyp deep segmentation methods.....	74
5.1 Introduction	74
5.2 Polyp database	75
5.3 Image analysis.....	78
5.4 Pre-processing.....	83
5.5 Data augmentation.....	85
5.6 Segmentation methods	91
5.6.1 FCN8s.....	91
5.6.2 Proposed ResNet-FCN	91
5.6.3 Proposed Dilated-ResFCN	93
5.6.4 Proposed SE-Unet.....	97
5.6.5 Test time segmentation	100
5.7 Summary.....	101
Chapter 6. Experiment design and results	103
6.1 Introduction	103
6.2 Implementation details	104
6.3 Validation data	105
6.4 Metrics	106
6.5 Experiment Results.....	113
6.5.1 Validation results of FCN8s, ResFCN, Dilated ResFCN and SE-Unet.....	113
6.5.2 Validation results using background confidence map	123
6.5.3 Validation results using precision, recall, and Hausdorff distance metrics	128

6.5.4 Data augmentation ablation tests	130
6.5.5 Significance test	131
6.6 Results on testing dataset.....	134
6.6.1 Test data results.....	135
6.6.2 Ranking of submitted results in the second GIANA challenge	143
6.6.3 Ranking of submitted results for the third GIANA challenge	144
6.6.4 Comparison of segmentation methods.....	144
Chapter 7. Summary, contributions and future work	148
7.1 Summary.....	148
7.2 Future work.....	151
Appendix A: Histogram of polyp (SD) and background.....	155
Appendix B: ResNet-50 network.....	156
Appendix C: K-means clustering based HSV and lab colour space	158
Appendix D: Developed gradient descent algorithms	160
References	165

List of Figures

Figure 1. 1 Five-year survival rates of the colorectal cancer for each stage	3
Figure 1. 2 Explanation of colonoscopy procedure	4
Figure 1. 3 A typical colonoscopy image showing a polyp, with a brief explanation of the visible structures	5
Figure 1. 4 Polyp detection and localisation and segmentation.	6
Figure 1. 5 Images with low and high complexities.	8
Figure 1. 6 Examples of semantic segmentation and instance segmentation.	9
Figure 2. 1 The schematic representation of the supplemented Paris classification.	16
Figure 2. 2 Typical polyps in the GIANA SD training dataset.....	17
Figure 3. 1 The structure of a single neuron	26
Figure 3. 2 The structure of a multilayer perceptron.	27
Figure 3. 3 Shapes of the sigmoid and tanh activation function.....	32
Figure 3. 4 Shapes of the derivatives of the sigmoid and tanh activation functions.	33
Figure 3. 5 Different types of image processing via different convolutions.	39
Figure 3. 6 Representation of a convolutional layer.	40
Figure 3. 7 Illustration of how gradients are being transfer to weights.	41
Figure 3. 8 Illustration of transposed convolution.	41
Figure 3. 9 Transferring gradients to different layers.	42
Figure 3. 10 The different pooling methods in the feed forward step	43
Figure 3. 11 The different pooling methods in backward step.	44
Figure 3. 12 The structure and learnt features of LeNet-5.	45
Figure 3. 13 Dropout in a fully connected layer and a convolutional layer.....	46
Figure 3. 14 The results of gradient descent without normalization and with normalization.	47
Figure 3. 15 The structure of AlexNet.	49
Figure 3. 16 NIN structure.	50

Figure 3. 17 The structure of inception.	51
Figure 3. 18 The structure of a ResNet block.	52
Figure 3. 19 The structure of SE module.	53
Figure 4. 1 Histogram-based method image segmentation method.	57
Figure 4. 2 The structure of FCN8s, FCN16s and FCN32s.	63
Figure 4. 3 The structure of Unet.	64
Figure 4. 4 Regular convolution and atrous convolution.	66
Figure 4. 5 Representation of 2d convolution layer, with regular convolution and atrous convolution.	66
Figure 4. 6 The structure of ASPP module.	68
Figure 4. 7 The operation of global convolution.	70
Figure 4. 8 The structure of RefineNet.	71
Figure 5. 1 An example of typical SD and HD training images and their corresponding ground truth.	75
Figure 5. 2 A sample of images from the SD training dataset.	76
Figure 5. 3 A sample from the SD testing database.	77
Figure 5. 4 The clustering result of Image No. 251.	79
Figure 5. 5 Image No.6, 64 and 251 and their clustered results with three cluster centres in RGB colour space.	80
Figure 5. 6 Image No.6, 64 and 251 and their clustered results with four cluster centres in RGB colour space.	80
Figure 5. 7 The cumulative polyp size distribution of SD images for length and for width.	82
Figure 5. 8 Colonoscopy image with marked border. The values of different pixels are marked by different colour.	83
Figure 5. 9 Processing pipeline for border removal from SD and HD images.	84
Figure 5. 10 The re-scaling processing of HD images	86
Figure 5. 11 Selected examples of colour jittering experimental images.	89
Figure 5. 12 The image rotation and corresponding augmented colonoscopy image.	90

Figure 5. 13 The structure of the proposed network using ResFCN.....	92
Figure 5. 14 The results of dilated Laplacian operator.	94
Figure 5. 15 Number of polyps fully covered by an increasing kernel size.....	95
Figure 5. 16 The results of a dilated Laplace operator for images of different resolutions.....	95
Figure 5. 17 The whole structure of Dilated-ResFCN	96
Figure 5. 18 Different dilated rates of the ASPP.....	98
Figure 5. 19 he whole structure of SE-Unet.....	99
Figure 5. 20 Polyp segmentation based rotated test image	101
Figure 6. 1 Image No. 181 and No. 182.	105
Figure 6. 2 The instance of Dice and Jaccard index.	107
Figure 6. 3 The example of Dice index and its components.	109
Figure 6. 4 The calculation of Hausdorff distance.....	112
Figure 6. 5 The Hausdorff Distance of different segmentation results.....	113
Figure 6.6 Typical results obtained for the SD images using FCN8s, ResFCN, Dilated ResFCN and SE-Unet networks..	115
Figure 6. 7 Visualization of Dice index from Table 6.5.....	116
Figure 6. 8 ResNet50, 101 and 152 based FCN.	118
Figure 6. 9 The SD polyp missed by Dilated ResFCN as well as by FCN8s and SE-Unet.....	119
Figure 6. 10 The SD polyp missed by Dilated ResFCN and segmented by FCN8s.....	121
Figure 6. 11 The polyp missed by Dilated ResFCN and segmented by SE-Unet.....	122
Figure 6. 12 The Dice index of each method during the training.....	125
Figure 6. 13 The difference between two neighbouring epochs.	126
Figure 6. 14 The mean rank of each method, with the blue segment indicating the best result.	134
Figure 6. 15 The Dice of each method in different value regions.	137
Figure 6. 16 Example of segmentation results obtained for SD images with the Dice index within the range of [0.9,1].....	138
Figure 6. 17 Example of results obtained for SD images with the Dice index within the range of [0.8, 0.9).	139

Figure 6. 18 Example of results obtained for SD images with the Dice index within the range of [0.6, 0.8). 140

Figure 6. 19 Example of results obtained for SD images with the Dice index within the range of [0.4, 0.6). 141

Figure 6. 20 Example of results obtained for SD images with the Dice index within the range of [0, 0.4). 142

List of Tables

Table 2. 1 Paris classification.....	15
Table 2. 2 The summary of polyp segmentation methods	22
Table 2. 3 Summary of polyps segmentation approaches advantages and disadvantages.	23
Table 5. 1 Number of augmented images using different augmentation method.....	91
Table 6. 1 Computer configurations.....	104
Table 6. 2 GPU divider, Cuda and Cudnn versions.....	104
Table 6. 3 The details of four subsets.	106
Table 6. 4 Definition of the confusion matrix.	108
Table 6. 5 The overall Dice index of four methods.....	114
Table 6. 6 The training details of four networks.....	117
Table 6. 7 Processing time for testing.....	117
Table 6. 8 Data missed by each method (chapters 5.5-5.8).....	118
Table 6. 9 The mean value of a re-segmented polyp generated by FCN8s and SE-Unet (chapters 5.5 and 5.8).....	119
Table 6. 10 Polyp missed by the Dilated ResFCN and successful segmented by the FCN8s. .	120
Table 6. 11 Polyp missed by the Dilated ResFCN and successful segmented by the FCN8s. .	120
Table 6. 12 Overall results after combining Dilated ResFCN with FCN8s or SE-Unet (chapters 5.5 and 5.8).....	123
Table 6. 13 Results generated by the background with each method (chapters 5.5-5.8).	123
Table 6. 14 Number of missing polyps using the background confidence map (chapters 5.5-5.7).	124
Table 6. 15 The results of networks without performance without the transfer learning.	128
Table 6. 16 The precision metric for each method.....	129
Table 6. 17 The recall metric for each method.	129
Table 6. 18 The Hausdorff Distance for each method.	130

Table 6. 19 Mean Dice index obtained on 4-fold validation data using Dilated ResFCN network	131
Table 6. 20 p-value: Dice.....	132
Table 6. 21 p-value: Precision.	132
Table 6. 22 p-value: Recall.	132
Table 6. 23 The mean value of ResFCN and SE-Unet (The summary of Table 6.13, 6.16, 6.17).	134
Table 6. 24 Results obtained on the test data using different architectures and networks outputs.....	135
Table 6. 25 The definition of different level of segmentation results (Dice index).	135
Table 6. 26 Ranking of the SD segmentation task.	143
Table 6. 27 Ranking of the HD segmentation task.	143
Table 6. 28 The comparison of existed polyp segmentation methods.....	146

Acronyms

ANN	Artificial neural network
ASPP	Atrous Spatial Pyramid Pooling
AUC	Area Under the Curve
CNN	Convolutional neural network
CWC	Colour wavelet covariance
DCAN	Deep contour-aware networks
DFN	Discriminative Feature Network
FCN	Fully convolutional network
GLCM	Grey-Level Co-occurrence Matrix
NIN	Network in Network
POCM	Polyp occurrence confidence map
PSPnet	Pyramid Scene Parsing
ResNet	Residual network
SE	Squeeze-and-excitation

Chapter 1. Introduction

1.1 Aim and motivation

The aim of this thesis is to investigate deep learning segmentation algorithms and design novel more accurate and computationally efficient polyp segmentation methods for colorectal endoscopy images. Typically, in the current routine clinical practice, polyps are detected and segmented manually. With an increase in the number of colorectal examination procedures, this approach has become ineffective and costly. To solve this problem, automated segmentation methods utilising machine learning and computer vision algorithms are being investigated.

However, polyp segmentation using computer-assisted methods is a very challenging task. The size, shape and appearance of a polyp are different at different stages. In an early stage, colorectal polyps are typically small, may not have a distinct appearance, and could be easily confused with other intestinal structures. In the later stages, the polyp morphology changes and the size begin to increase. Some polyps become so large that they take up most of the image space. Illumination in colon screening is also variable, producing local overexposure highlights and specular

reflections. Some polyps may look very differently from different camera positions, have no obvious boundary between the polyp and its surroundings tissue, be affected by intestinal content and luminal regions, inevitably leading to segmentation errors. So far, a number of polyp segmentation methods have been proposed mainly using active shape or texture-based algorithms. These methods can be used to segment polyps that have specific contours and/or similar appearance, but these conditions are often difficult to satisfy in practice. Since overall processing pipelines for these methods are composed of many processing stages, their structures and hyper-parameters often need to be re-set when the experimental settings are changed. Therefore, one of their main limitations is that they typically could perform well only when used to segment a specific polyp type in a predefined clinical setup.

The research reported in this thesis has been motivated by the limitations of previously proposed methods. It attempts to construct a new method that integrates feature descriptors and classification methods with parameters selection optimized via an integrated deep learning. The performance objectives for the new polyp segmentation methods are as follows:

Accuracy: The segmentation results of the proposed methods should be more accurate than those of the most current polyp segmentation approaches. To make this comparison more reliable, their performances should be evaluated by different complementary metrics. In addition, since it is not realistic to re-implement all previous methods, the most representative methods will be chosen for comparison purposes.

Robustness: Robustness indicates whether a method has sufficient generalization ability. The proposed segmentation methods should have a certain degree of robustness to cope with the variability in polyp images, including: varying polyp morphology, tissue deformations, displacement, specular reflections, size and varying polyp appearances.

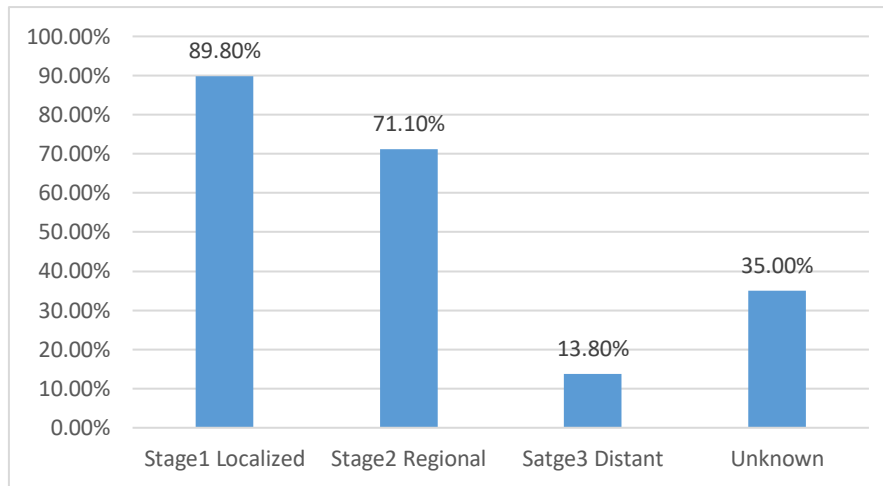


Figure 1. 1 Five-year survival rates of the colorectal cancer for each stage.¹: (a) Localized, i.e. confined to the primary site; (b)Regional, i.e. spread to regional lymph nodes; (c) Distant, i.e. cancer has metastasized; (c)Unknown, i.e. un-staged.

Computational efficiency: Deep learning methods usually require high performance GPU units with large internal memories, and therefore their implementation could be expensive. The developed methods should achieve a balance between performance and the size of deep model. Moreover, they should be computationally efficient, i.e. should enable a real-time processing.

1.2 Colon cancer and colonoscopy

Colorectal cancer is one of the major causes of cancer incidence and death worldwide. The latest survey shows that there were 1,096,601 new cases and 551,269 deaths in 2018 [1], and each account respectively for 6.1% and 5.8% of the total number of cancer related cases and deaths. Based on the current trend, it is estimated that the new cases and deaths will increase respectively to 2.2 million and 1.1 million by 2030. Colorectal cancer arises from benign polyps, however, with time some of them become malignant adenoma.

¹ <https://seer.cancer.gov/statfacts/html/colorect.html> [Accessed 20 Oct. 2019]

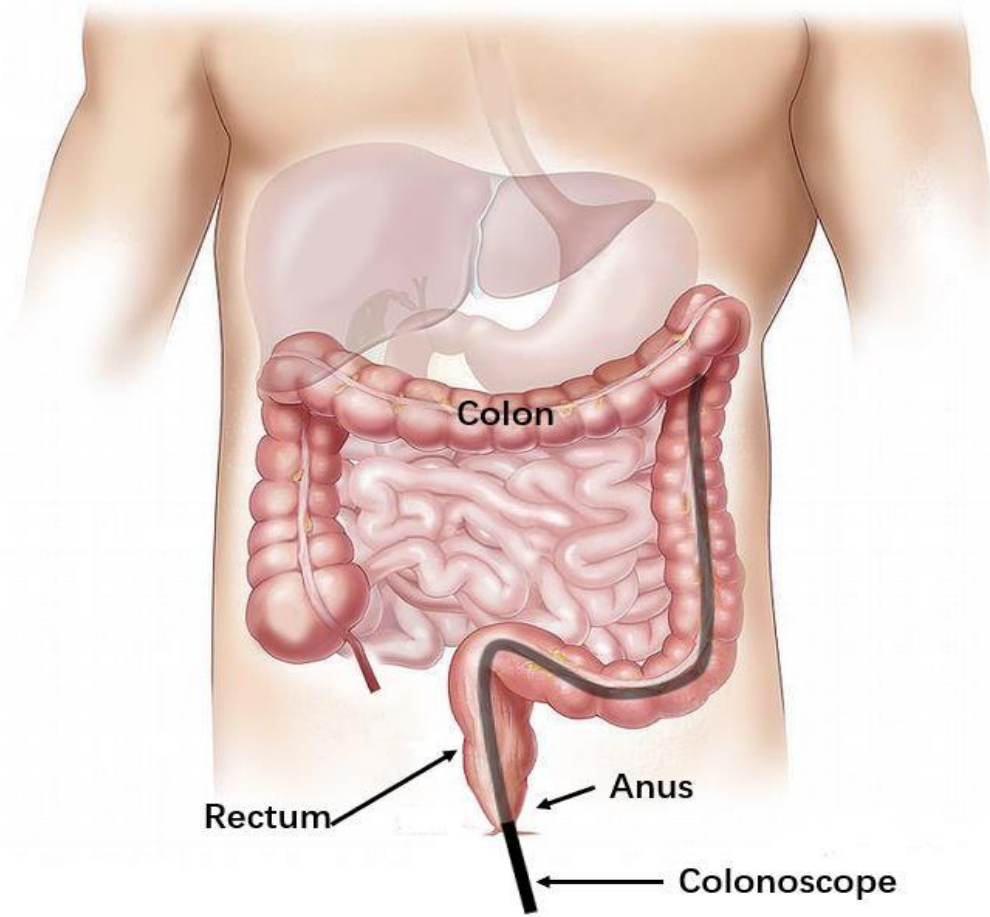


Figure 1. 2 Explanation of colonoscopy procedure².

Figure 1.1 demonstrates five-year survival rates for the colon cancer. It can be seen that the earlier colorectal cancer is detected, the better the chances of surviving five years after being diagnosed. Therefore, the mortality due to colon cancer can be reduced through colon screening. Figure 1.2 illustrates the colonoscopy procedure: a colonoscope, a flexible instrument typically using an optical fibre or electronic camera, is inserted through the anus to visually examine the colon for abnormal tissue. Usually, patients need to take laxatives in advance of the procedure to remove waste material from the colon and sometimes carbon dioxide is injected to enable better access to different parts of the colon. During the examination, the colonoscope is controlled by a clinician (gastroenterologist) who navigates through the large intestine in search of abnormalities such as polyps. Figure 1.3 shows a typical colonoscopy image.

² <https://pixabay.com/en/offal-marking-medical-intestine-1463369/> [Accessed Oct. 2019]

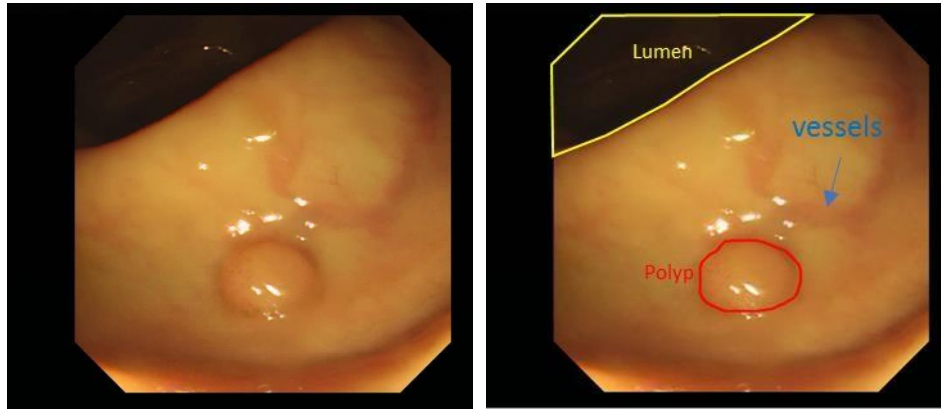


Figure 1.3 A typical colonoscopy image showing a polyp³, with a brief explanation of the visible structures.

There are also non-invasive approaches available for the colon screening, including computed tomography (CT colonoscopy) and colour Doppler ultrasound. The CT colonoscopy, also called virtual colonoscopy, produces reconstructed 2d or 3d images of the colon. Since these are non-invasive medical procedures, it is impossible to perform a resection or biopsy of the polyp during such examinations. If a lesion is found, a colonoscopy is required anyway to perform the resection or biopsy. Therefore, colonoscopy is the most common approach to colon screening as biopsy, and often the resection, are possible during the same screening procedure.

The Wireless Capsule Endoscopy (WCE), in the near future, replace the current colon scanning method. The capsule is swallowed by the patient and images of the esophagus, stomach, small intestine and colon are internally recorded or wirelessly transmitted to an external recording device. However, for this technique to be fully accepted in clinical practice an accurate and robust automatic lesion detection system has to be developed as the WCE generates a very large number of images making it very difficult for clinicians to use it in practice.

Usually, polyps need to be directly examined by clinicians during colonoscopy. However, with the increased number of colonoscopy procedures, such an approach becomes ineffective and costly. To address this problem, machine learning, and computer vision algorithms have become investigated with the view to automate analysis of colonoscopy images. Typical analysis of colonoscopy images includes: polyp

³ <https://giana.grand-challenge.org/> [Accessed 20 Oct. 2019]

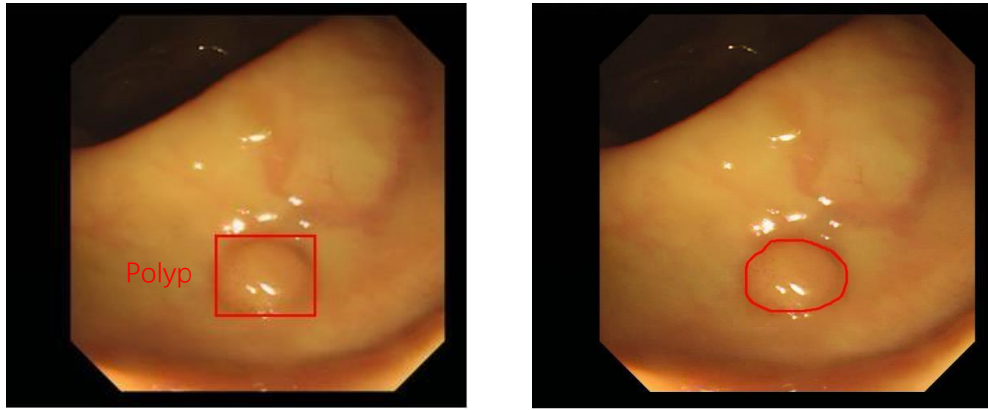


Figure 1. 4 Polyp detection and localisation (left) and segmentation (right).

classification, localisation, detection and segmentation.

There have been a number of different definitions used in literature for classification, detection, localisation, and segmentation. In this thesis the description of these tasks is aligned with the definitions used by the Gastrointestinal Image Analysis (GIANA) challenge⁴.

The objective of polyp detection is to identify the presence or absence of polyps in images and, in case of polyp(s) being identified as present finding, the location of polyp(s) in the image, typically using a bounding box to indicate relevant image regions, i.e. solving the localisation problem. Polyp segmentation is somewhat similar to detection, but the objective here is to find a contour accurately delineating the polyp rather than estimating position and size of a bounding box, see Figure 1.4. Polyp classification task is sometimes used to identify additional polyp characteristics, e.g. as hyperplastic, adenomas or deep submucosal invasive.

1.3 Image segmentation

Commonly, computer vision is tasked with detection, analysis and general processing of specific objects present in images. Frequently, to ensure the accuracy and stability of the processing, unnecessary contents need to be removed/ignored. A manual approach to this operation may guarantee an accurate result, but efficiency is low

⁴ <https://giana.grand-challenge.org/Tasks/> [Accessed 20 Oct. 2019]

when the number of images is even moderately large. Therefore, a technique that can automatically extract the objects of interest is required.

Image segmentation is an image processing methodology that could be used to meet such requirements. It is one of the most important constituents for many computer vision and digital image processing tools. Typically, the selected objects are called the foreground, and the other unimportant content is called the background. The foreground and background may have different appearances, with unique colour, texture or shape - with the foreground often represented by homogeneous regions. These properties are an important foundation for traditional image segmentation methods. Commonly used traditional methods include: thresholding, edge detection, region growing, clustering, and active contours. Sometimes, these properties are not easily defined and need to be further reinforced via specific pre-processing.

Despite the significant progress made, image segmentation remains a challenging problem. Traditional segmentation methods often are designed to work on a specific image type only, and their performances still are far from what is expected. This lack of performance is caused by the fact, that in such settings, algorithms cannot fully mimic humans' perception of images.

Traditional segmentation methods cannot fully "understand" the meaning of objects present in an image because the properties of the foreground and background are defined by a set of simple descriptors. Therefore, if the objects of interest are complex, with multiple nonhomogeneous sub-regions appearing in various configurations, the traditional segmentation methods do not work. This challenge is illustrated in Figure 1.5, where the objective is to segment sofas. For image (A), there is an obvious colour difference between the foreground (sofas) and background, therefore it is a manageable task for the traditional methods. However, for images (B) and (C), the sofa cannot be defined by colour, shape, size, or position alone. To overcome this problem, the task of image segmentation is further extended into so called semantic and instance segmentation.



Figure 1. 5 Images with low (A) and high complexities (B and C)⁵.

Semantic segmentation: This task can be understood as a combined object recognition and image segmentation performed as a single operation. The segmented foreground objects should correspond to their pre-set classes. Their differences are often reflected by the distinctive “meaning” of objects shown in an image and not just their local appearances. Therefore, semantic segmentation not only segments the foreground but also categorizes it to different classes. The key ingredients in the semantic segmentation are feature representation and machine learning. Moreover, with the development of deep learning, these two operations are merged into an end-to-end trained structure resulting in the superior performances when compared with other methods.

Instance segmentation:

Instance segmentation can be regarded as the extension of the semantic segmentation. This task requires that all segmented foreground objects representing the same object category are independently delineated. Figure 1.6 explains the difference between semantic and instance segmentations. The semantic segmentation cannot differentiate between different instances of the same object; therefore, all the “chairs” have assigned the same object label. However, the instance segmentation not only correctly recognise different object classes but also is able to differentiate between different object instances from the same class, i.e. all the “chairs” are individually delineated. The key technical difficulty is to deal with overlapping objects from the same class (e.g. chairs on the right side of the table).

⁵ <http://host.robots.ox.ac.uk/pascal/VOC/voc2012/> [Accessed 20 Oct. 2019]

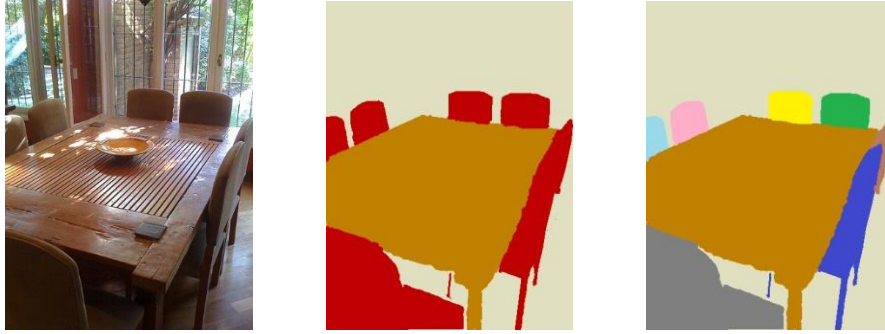


Figure 1.6 Examples of semantic segmentation and instance segmentation⁶. Left: original image. Middle: results of semantic segmentation (red: chair; brown: table). Right: results of instance segmentation.

1.4 Machine learning

Machine learning is a subject in computer science that aims to optimize the performance of tasks based on learning from observation or experience [2]. It aims to estimate properties of objects or events by analysing the data representative of these objects or events.

There is a large number of techniques developed to solve various problems in machine learning, most of which can be regarded as, linear or non-linear, mapping functions. The learning process can be explained as mechanism for modifying parameters and/or architectures of these mapping functions, which can describe patterns extracted from the available data. This can be used to replace humans in making decisions for often repetitive tasks.

In machine learning, objects (e.g., events, experiences, and observations) that need to be processed are called data samples. A descriptor explaining a sample is called a feature. Features are extracted from original data and can be used to describe the data more efficiently. Data sample can be described by a single feature or multiple features. The values of features can be continuous or discrete. Normally, a database used for learning purposes has large collections of samples, with all samples described by the same features.

⁶ <http://host.robots.ox.ac.uk/pascal/VOC/voc2012/> [Accessed 20 Oct. 2019]

Unsupervised and supervised learning

Based on the available data, machine learning methods can be divided into unsupervised and supervised learning methods. For supervised learning, the data samples used for learning have associated additional specific properties called labels and the objective of the learning process is to estimate stable mapping relations between the samples and the labels. The unsupervised learning aims to find the mapping patterns from unlabelled samples. In both of these cases, the final objective is to use the learned mapping for previously unseen data samples to obtain the predicted labels for these samples.

Classification and regression

The typical tasks in machine learning are classification and regression. The difference is that the output of the classification task is discrete (or categorical), and the output of a regression is continuous. For example, answering the question 'is it raining?' is an example of a classification problem, and response to question 'what is the temperature?' is an example of a regression problem. Many machine learning methods can be used for both classification and regression.

Feature extraction

Feature extraction is an important branch of machine learning and is tasked with efficient representation of the raw data.

Images are a fundamental source of information for this research. Image content can be encoded by high- and/or low- level features. High-level features can be understood as semantic objects in images, e.g. representing meaningful objects such as humans, cars or airplanes. Low-level features are the components that explain more fundamental image properties, such as colours, edges, corners, and gradients. Typically, low-level features are used to represent local image information and are commonly branded as low-level visual information. These features, when design based on human image interpretation, are often called handcrafted features. The machine learning

algorithms use image features for completion of recognition, segmentation and detection tasks. The performance of machine learning algorithms strongly depends on the descriptive properties of the features used. Selection of suitable features is one of the key ingredients in successful deployment of machine learning algorithms.

1.5 Deep learning

Deep learning is one of the branches of machine learning that was developed based on artificial neural networks (ANNs). In deep learning feature selection and feature classification is integrated through use of multiple non-linear hidden layers. This architecture aims to learn the patterns present in the training data, encoding them as features residing within the network (deep features) and, at the same time, perform pattern classification based on these deep features. Deep learning methods circumvent the selection of handcrafted features by discovering image dependencies which are hard to see by a human. In machine learning, this operation is called representation learning. The term 'deep' means that the number of hidden layers is far larger than those previously used in ANN methods (e.g. multilayer perceptron).

Convolutional neural networks (CNNs) are representative algorithms of deep learning in image processing. This method was inspired by research on the visual cortical cells [3]. The first successful prototype, called LeNet-5 [4], was used to perform hand-written digit recognition. However, the hardware at that time could not support processing of high-resolution images or construction of very deep networks.

In 2012, Krizhevsky et al. [5] developed CNN model which won first prize for the Large-Scale Visual Recognition Challenge (commonly this is considered as a birthday of the "deep learning revolution"). They reduced the influence of so-called vanishing gradients (previously preventing constructions of deep networks) and trained the network using a GPU (GPUs make the implementation of a large CNN computationally feasible). The error rate for that method was 15.3% lower than 26% error rate of the second-best method. After that, deep learning methods gradually became mainstream in computer vision and started to be a dominant approach for recognition,

segmentation and detection problems.

1.6 Novel contributions

This thesis presents a number of new approaches to complete polyp segmentation tasks in colonoscopy images based on the investigation of deep learning methods. The primary novel contributions of the research are two new segmentation convolutional neural network architectures. These are the Dilated ResFCN and the SE-Unet. The former performs well overall, while the latter is particularly effective at segmentation of small polyps, which could be missed by the former. The proposed optimal hybrid method combines these two CNNs to improve robustness, which allows for polyps of various types to be effectively segmented. In addition, these networks can be efficiently deployed on a standard desktop computer, allowing for real-time image segmentation.

The performances of the proposed and other reference polyp segmentation methods have been extensively evaluated. The performed comparison demonstrates that the proposed hybrid approach outperforms other methods. Therefore, the developed methods and the reported results can be used as a reference for the future research. Furthermore, a number of evaluation metrics have been used to validate the reported segmentation methods. Some of these metrics have not been used before in the context of polyp segmentation. The reliability of the performed evaluations has been validated using statistical significance tests. It has been demonstrated that the proposed method achieves statistically significantly better results than those of existing methods. The significance of different data augmentation methods has been evaluated using comparative ablation tests. It has been demonstrated that rotation, local deformation and colour jitter are the most important augmentation techniques.

1.7 Thesis organization

Chapter 2 introduces the polyp segmentation task and reviews previously proposed polyp segmentation methods. Chapter 3 provides an overview of current machine learning and deep learning methods, with the main focus on CNN's key building blocks. Chapter 4 summarizes the current image segmentation methods. It starts with descriptions of traditional methods, followed by descriptions of deep learning methods. The main focus is on the structure of semantic segmentation algorithms. Chapter 5 describes the polyp database and novel methods proposed in this thesis. Evaluation of the described methods is provided in chapter 6. Finally, Chapter 7 provides a summary of the work and hypothesizes about possible future work.

Chapter 2. Polyp segmentation problem

2.1 Introduction

Polyp segmentation in colonoscopy images is the central problem being investigated in this thesis. The key approach adopted in this work, to achieve a robust and accurate polyp segmentation, is based on the deep learning methodology and more specifically deep convolutional neural networks. Whereas Chapters three and four provide the necessary information about segmentation and the deep learning techniques, Chapter five describes the newly proposed methods, this chapter is focused on description of the problem itself and identifies the key challenges. Besides, the chapter provides a brief review of previously proposed polyp segmentation methods.

Table 2. 1 Paris classification of polyp morphology [6].

Types	Sub-groups
Type 0-I, polypoid	0-Ip, pedunculated
	0-Is, sessile
Type 0-II, non-polypoid and nonexcavated	0-IIa, slightly elevated
	0-IIb, completely flat
	0-IIc, slightly depressed without ulcer
Type 0-III, non-polypoid with a frank ulcer	-

2.2 Polyp morphology

Accurate segmentation of polyps in colonoscopy images is a challenging task. This is due to a number of factors such as illumination conditions, variable camera positions, different characteristics of the surrounding tissue or presence of intestinal content. However, more fundamentally the polyp segmentation is difficult because of the inherent variability of polyp morphology. Since the appearance of polyps and surrounding tissues is complex and variable, the distribution of pixel values cannot be quantified by homogeneous patterns.

A frequently used categorisation of different polyp morphology types is the so-called Paris classification of endoscopic polyps [6] introduced in Table 2.1. Paris classification was proposed in 2005 for the classification of superficial lesions in the esophagus, stomach, and colon. The method divides superficial lesions into three main classes, 0-I, 0-II and 0-III and corresponding sub-classes (see Table 2.1).

The Paris classification had been subsequently further supplemented by additional morphological subclasses graphically represented in Figure 2.1 [7]. This include two new polyp morphology types: 0-Isp and 0-IIa+s. Depending on the height of the polyposis, the lesion morphology can be divided into protruded, flat elevated and flat.

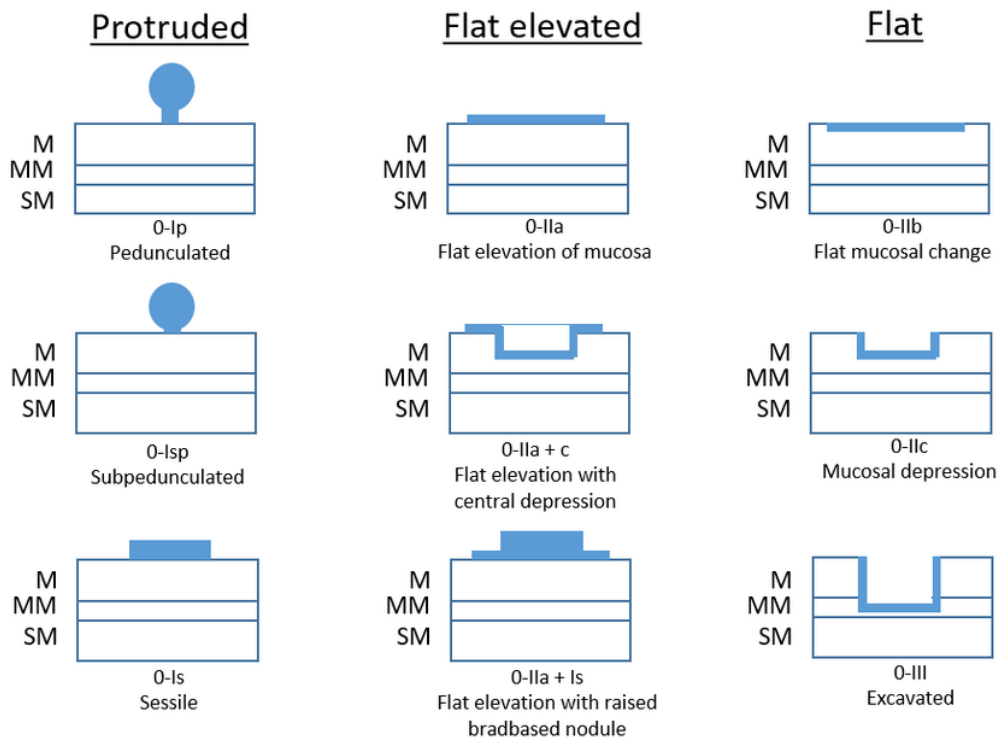


Figure 2. 1 The schematic representation of the supplemented Paris classification [7]. (M represents mucosa, MM Muscularis mucosa and SM represent sub- mucosa).

The main difference between flat elevated and flat polyps is whether lesion is raised from mucosa. It should be noted that since the 0-Is and 0-IIa types of lesions are very similar, the Paris classification selects a height of 2.5 mm as a threshold to differentiate between them.

The different types of polyps can significantly differ in terms of size, shape, colour, and texture. Even the same polyp may look significantly different depending on the colonoscopy camera position and/or used colon illumination, leading to changing pattern of shadows and highlights (Figure 2.1). Colon itself can deform, with the colon folds resembling polyps and polyps often “hiding” behind folds or indeed being masked within colon’s luminal region. The complexity of the problem can be judged based on image samples from the Figure 2.2 showing presence of specular highlights, overexposed regions, intestinal material, obscured (partially visible) polyps and so on. It should be also noted that the colonoscopy image is challenging as not only images have a low resolution but some of them are also of low quality, e.g. contain large areas of specular reflections as well as a motion blur or/and double exposure effect.

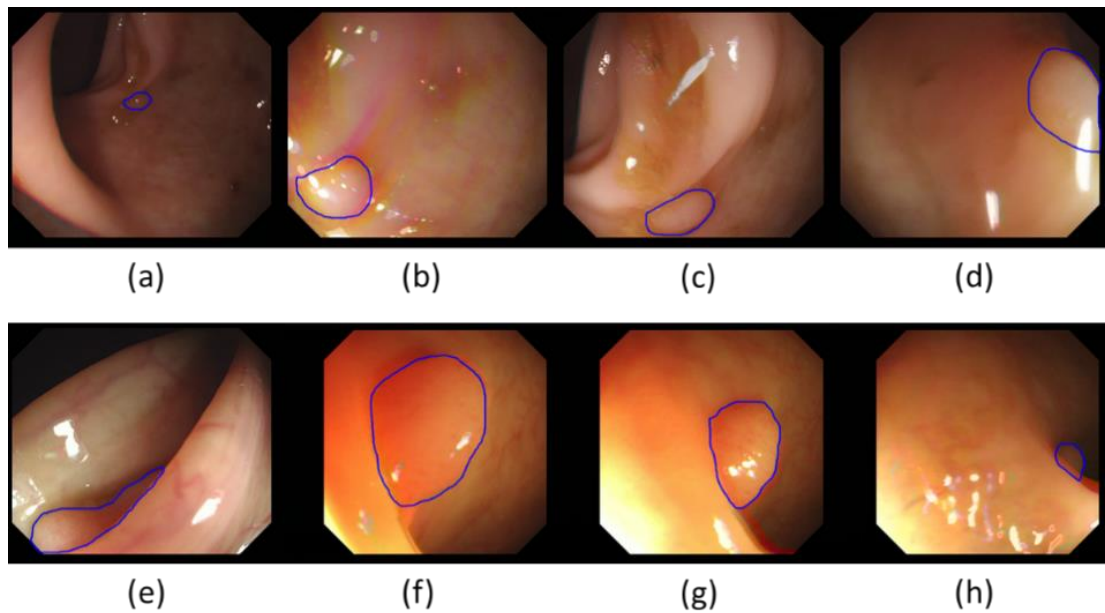


Figure 2. 2 Typical polyps in the GIANA SD training dataset [9], [10], [11] and [12]. (a) Small size; (b) Blur; (c) Intestinal content; (d) Specular highlights /defocused; (e) Occlusion; (f) Large size; (g) Overexposed areas; (h) Luminal region.

Another set of problems can be caused by poorly defined polyp edges as some of the flat and flat elevated types of polyps can be masked by mucosa, making it difficult to determine the boundary between polyps and other tissues, which leads to under segmentation or over-segmentation. This problem also affects the assessment of the methods, as human observers are faced with the same challenges when delineating polyps.

In order to solve above problems, the segmentation method may require multiple features to describe polyp properties. However, the multiple features increase the complexity of the input data. The uncertainty about feature types and method's hyperparameters can also make the method design more difficult, ultimately leading to increase in segmentation errors. For instance, if hyperparameters are too "tightly fitted" to the available data it may reduce the robustness of the developed segmentation method, resulting in the segmentation method detecting only specific types of polyps.

2.2 Review of polyp segmentation methods

This section provides a summary of the representative polyp segmentation methods that have been proposed in the recent years. The development of polyp segmentation algorithms is strongly affected by the progress in image processing, computer vision and machine learning. To date, most polyp segmentation methods can be characterised in terms of shape, texture, and applied machine learning (more recently deep machine learning) methodology.

2.2.1 Polyp segmentation based on shape

As most polyps have a well-defined edge, their shape becomes an important feature that can be used to distinguish between polyps and the background. Shape segmentation aims to reinforce polyp edges and detect the shape of a polyp with the corresponding enclosed area representing segmentation results.

Due to the fact that in many cases, polyps have well-defined shapes some of the early approaches attempted to fit predefined polyp shape models. Hwang et al. [12] divided the image into many small pieces using the watershed algorithm and subsequently fitted ellipses to all possible regions. Then, an optimal ellipse was selected based on curvature, edge distance and intensity values. Gross et al. [13] used the Canny edge detector with the image processed by using Non-Linear Diffusion Filtering (NLDF). The NLDF effectively removes the edges representing small blood vessels, leading to a better definition of the polyp edges. Subsequently, the detected edges are compared with the specific template, and only the most suitable edges are retained.

The above two methods are not suitable for polyps that do not have a well-defined shape. To solve this problem, Breier et al. [14], [15] investigated applications of active contour, active rays, and the Chan–Vese methods for polyp segmentation. For typical polyps, these methods are able to correctly fit polyp contours, however the Chan–Vese [16] method is easily affected by an uneven illumination, shadows and specular

reflections. In addition, these methods are not fully automatic as the initial contour position needs to be set manually before segmentation is performed.

The successful segmentation using active contour methods relies on the assumption that closed, complete and uniform polyp contours are visible in the image. To improve the robustness, further studies have focused on development of edge detectors. Ganz et al. [17] applied global Pb-Oriented Watershed Transform-Ultrametric Contour Map (gPbOWT-UCM) [18] in polyp segmentation and presented the Shape-UCM method. The idea is to use ellipses in the multi-level segmentation results of gPb-OWT-UCM selecting the shape that is closest to an ellipse. This method also removes the border of the image and inpaints specular reflections to improve polyp edge representation accuracy.

Bernal et al. [9] presented the 'depth of valley' approach to detect more general polyp shapes and segment the polyp by evaluating the relationship between pixels and the detected contour. To improve their segmentation results, the authors [10] decreased the influence of blood, highlights, and border and proposed a new method called Window Median Depth of Valleys Accumulation (WM-DOVA) maps to integrate "valley" information [11]. Tajbakhsh et al. [19] proposed a number of polyp segmentation methods based on edge classification. The initial method used a random forest to classify the features extracted by a Haar descriptor. In their follow-up work [20], [21], authors attempted to refine the background of an image and complete recognition via several sub-classifiers.

2.2.2 Polyp segmentation based on texture appearance

Given that some large polyps are frequently perfused with blood, their appearance is "redder" than the surrounding tissue and sometimes include bloodstains. This feature is typical and can be used to differentiate polyps from the other tissue in the colon. More generally, texture features can be used as an input for a machine learning algorithm to perform segmentation. For instance, Karkanis et al. [22] proposed colour wavelet covariance (CWC), which combines the Grey-Level Co-occurrence Matrix

(GLCM) with the 2-D discrete wavelet transform. The statistical measures of the GLCM [23] are obtained from a wavelet-transformed image. In the reported experiments, this method obtains the highest Area Under the Curve (AUC) value when compared to other texturebased methods. Using the same database and classifier, Lakovidis et al. [24] proposed a method that provided the best results in terms of the AUC metric. Furthermore, Alexandre et al. [25] and Ameling et al. [26] tested a colour-based method, the Local Binary Pattern and the original GLCMs; however, because of the use of different databases and design parameters they are difficult to compare directly.

2.2.3 Polyp segmentation based on deep learning

More recently, with the adoption of the deep learning methodology, the more traditional (based on the handcrafted features) segmentation methods are gradually being replaced by approaches based on convolutional neural networks (CNN). The deep polyp segmentation unifies the feature extraction and classification into one combined algorithm, significantly improving the accuracy of segmentation.

Deep learning methods can be divided into several categories. The first category uses patch based (sliding window) classification approach. In that case the CNN is only using a local image information. Park et al. [27] formulated a pyramid CNN to learn polyps' scale invariant features. The features are extracted from the same patch with three different scales through three CNN paths. To save computational load, the sliding window strides every 4 pixels, and then the classified pixels are up-sampled to the same size as the input image.

Ribeiro et al. [28] evaluated a CNN, comparing it against other state-of-the-art features for polyp classification. The authors found that the CNN has a superior performance when compared with methods based on handcrafted features (the CNN is used not only for classification but also for feature extraction). Zhang et al. [29] designed a transfer learning scheme, in which the low-level features are extracted from a pre-trained CNN and then classified using the support vector machine (SVM)

algorithm [30]. This transfer learning scheme illustrates that CNNs are able to robustly learn low-level image features.

The second category is developed based on the end-to-end CNN training model, and effectively has become the most popular (and successful) approach for generic image segmentation problems. Some studies [8], [31], [32] tested the performance of FCN8s [33] on different polyp databases. From their results, it can be concluded that end-to-end feature learning is in general a feasible approach for polyp segmentation, however several false positives appear in their segmentation results. To address this issue, Zhang et al. [31] added a random forest to recognize these wrongly detected structures.

Li et al. [34] chose the Unet as the segmentation method. The authors refine the structure of Unet [35] such that the smallest resolution of internal feature map is 1×1 . Zhou et al. [36] retain the original parameters setting of U-net, but redefine the loss using cross entropy combined with the Dice metric. This approach performs well for unbalanced data, with a small target (i.e. polyps).

Mohammed et al. [37] presented a two-path Unet architecture that obtains the deep features from two corresponding encoders. In that method, one encoder was trained on natural images and not re-trained on the polyp data, while another encoder was only trained on the polyp database with the decoder being trained on the combined outputs from the two encoders. It was observed that semantics are changed for different images, but some low-level features are similar. This observation can be used to reinforce the feature learning. Fan et al. [38] proposed an auto encoder to mitigate problems which could occur due to presence of specular reflection in colonoscopy images.

Table 2.2 is the summary of above polyp segmentation methods. Table 2.3 lists the advantage and disadvantages of shape-based, texture-based and deep learning-based polyp segmentation methods.

Table 2. 2 The summary of polyp segmentation methods.

Types	Methods	References
Shape-based methods	Watershed algorithm, Ellipse fitting	Hwang et al., [12] Gross et al., [13]
	Active contour, active rays, Chan–Vese	Breier et al., [14] Breier et al., [15]
	global Pb-Oriented Watershed Transform-Ultrametric Contour Map (gPb-OWT-UCM)	Ganz et al., [17]
	Depth of valley	Bernal et al., [9]
	Haar feature, random forest	Tajbakhsh et al., [19]
Texture appearance-based methods	Colour wavelet covariance (CWC)	Karkanis et al., [22] Lakovidis et al., [24]
	Colour-based, Local Binary Pattern (LBP), Grey-Level Co-occurrence Matrix (GLCM)	Alexandre et al., [25] Ameling et al., [26]
Deep learning	Classification CNN	Park et al., [27] Ribeiro et al., [28] Zhang et al., [29]
	End-to-End trained CNN	Vázquez et al., [8] Akbari et al., [32] Zhang et al., [31] Zhou et al., [36]

Table 2. 3 Summary of polyp's segmentation approaches advantages and disadvantages.

	Advantages	Disadvantages
Shape-based methods	<ul style="list-style-type: none"> • Do not require a large training dataset • Results and processing models could be easily interpreted in terms of image properties 	<ul style="list-style-type: none"> • Require complex data modelling process (e.g. PDE models in case is active contour) • Strongly depend on values of design parameters, which are difficult to select using analytical methods • The results are not robust and very strongly depending on possible image artefacts • May require manual initialization
Texture appearance-based methods	<ul style="list-style-type: none"> • Require only moderately sized dataset • Relatively simple to implement • Results and processing models could be easily interpreted in terms of image properties 	<ul style="list-style-type: none"> • It has low computational efficiency and typically not suitable for real-time segmentation applications. • The results are not very robust moderately depending on possible image artefacts
Deep learning methods	<ul style="list-style-type: none"> • Currently the best performing methods in terms of accuracy and robustness • Could implemented for real-time applications 	<ul style="list-style-type: none"> • For best computational performance, requires high-performance hardware • For best segmentation accuracy, requires very large training datasets • The results and models are difficult to interpret

2.4 Summary

As demonstrated in this chapter, polyp segmentation is a very challenging problem due to numerous reasons, including: variability of the polyp and colon morphology, acquisition conditions and methods, as well as different image representations. There have been variety of methods proposed in literature addressing these problems. All reported techniques could be approximately grouped into two main categories, the categories based on the handcrafted features and those based on the deep features. The handcrafted methods could be further subdivided into contour and texture based, whereas the deep methods in general could be subdivided into patch based and the end-to-end trained approaches. Although all these methods have their specific advantages and disadvantages, the deep methods currently provide the best compromised between segmentation accuracy and computational efficiency.

The novel polyp segmentation methods developed as part of the reported research belong to the deep convolutional neural network. They have been described in detail in Chapter 5.

Chapter 3. Artificial neural network and convolutional neural network

3.1 Introduction

In chapter 2, it can be seen the convolutional neural network (CNN) obtain the best results in polyp segmentation. This chapter introduces some important components of CNN. The whole chapter can be divided into four sub-sections: The first section explains the multilayer perceptron and backpropagation algorithm. The second section introduces the main components of convolutional neural network (CNN). A typical CNN architecture is also shown in this section. Next, some optimization methods that developed gradient descent algorithms are summarized. Finally, this chapter describes the representative CNNs architecture, some of them are used in this thesis.

3.2 Multilayer perceptron

Artificial neural network (ANN) is a kind of machine learning algorithm designed to simulate neurons in a biological brain. Some ANN methods have been available for a long time, but they have some shortcomings in performance and thus are not widely applied. In recent years, with the development of machine learning, this category of methods has been constantly improved and has gradually become an important topic of research in machine learning.

3.2.1 Feedforward structure

The structure of the multilayer perceptron can be divided into three sub-structures (Figure 3.2): the input layer, hidden layers, and output layer. The input layer receives data, and each unit of data represents one feature of the data samples. The hidden layer is used to learn these features, it can convert linearly non-separable data samples to linearly separable. A multilayer perceptron can have single or multiply hidden layers. Normally, linearly non-separable data sample requires more hidden layers. The output layer is used to receive the results generated by the last hidden layer, then transfer them to the prediction result.

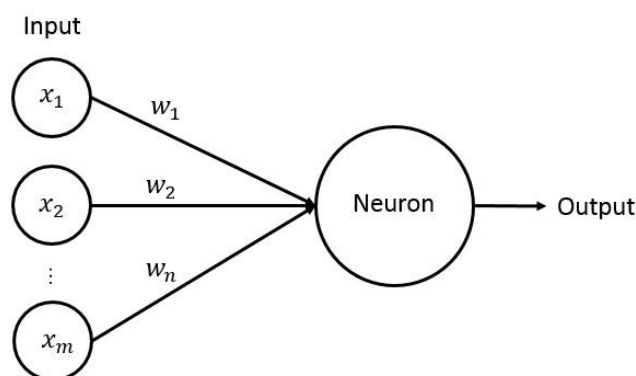


Figure 3. 1 The structure of a single neuron.

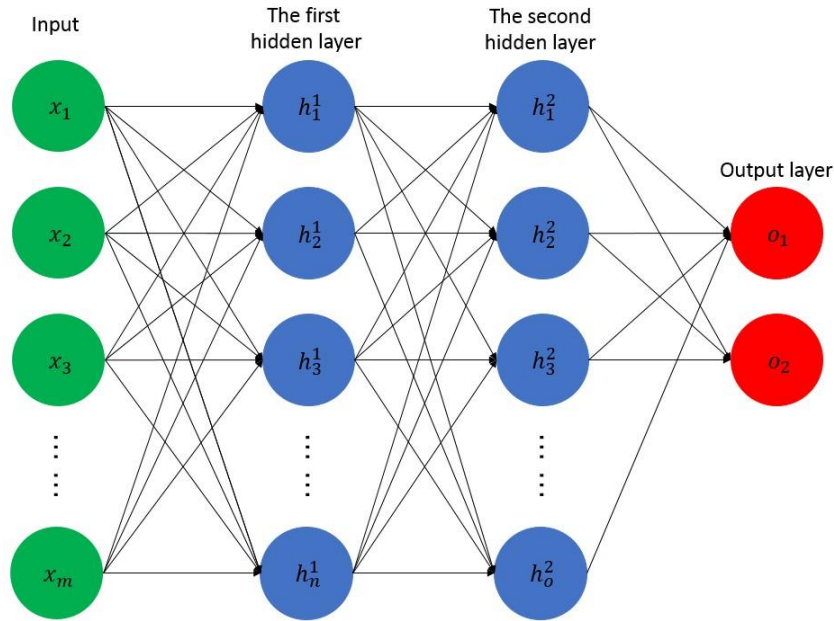


Figure 3. 2 The structure of a multilayer perceptron.

The hidden and output layers are mainly composed of units called neurons. Figure 3.1 shows the structure of a single neuron. Equation 3.1 shows the processing inside the neuron is weight summation. The variable w_i and b represents the weights and bias, n denotes the number of input data samples. Next, the output of Equation 3.1 is inputted into the activation function to do non-linear mapping (Equation 3.2).

$$z = \sum_{i=1}^n w_i x_i + b \quad 3.1$$

$$y = \varphi(z) \quad 3.2$$

The output of a multilayer perceptron is obtained by fusing the output of different neurons. The computing is same to the Equation 3.1. Each neuron needs to receive all the outputs from the previous layer and generates an output value. For a hidden or output layer, Equation 3.1 can be extended into:

$$z_i^l = \sum_{j=1}^{n^{l-1}} w_{ij}^l y_j^{l-1} + b_i^l \quad 3.3$$

Where w_{ij}^l denotes the j^{th} weight of i^{th} neurons in l^{th} hidden layer (or output layer). The variable b_i^l is the bias of i^{th} neurons in l^{th} hidden layer. The input vectors y_j^{l-1} represents the output from last layer and activated by Equation 3.2. If Equation 3.3 is the first hidden layer, y_j^{l-1} represents the input data samples.

In the training stage, a loss function is used to evaluate the error (residual). The error can be regarded as the similarity between the predication and observation (label). Here, mean square error loss function (MSE) is used as an example to explain this processing. Supposing Equation 3.3 represents the output layer, if a single data sample is inputted into a multilayer perceptron, the error can be expressed:

$$MSE = \zeta = \frac{1}{2} \sum_{i=1}^{n^l} (y_i' - y_i^l)^2 \quad 3.4$$

Where y_i' is the observation, n^l denotes the number of input data samples. In the training stage, the weights and biases of multilayer perceptron are optimized by minimizing the error between the observation and predication.

3.2.2 Backpropagation

Backpropagation [39], [40] is used to correct the trainable parameters of multilayer perceptron. This technique is developed based on gradient descent and chain rule. As for the reverse step, errors are given to the output of the network in back-to-front sequence in order to update parameters using the gradient method.

Gradient descent

Gradient descent algorithm is an iterative optimization method. Because the direction of the gradient is the direction along which the function increases at the highest rate at a given point, the opposite direction is similarly the direction in which the function decreases the fastest. On this basis, a minimum of the function can be determined. It can be defined as follows:

$$\theta(t + 1) \leftarrow \theta(t) - \eta \frac{\partial J(\theta)}{\partial \theta} \Big|_{\theta=\theta(t)} \quad 3.5$$

where $J(\theta)$ represents a loss function, θ denotes a trainable parameter in the model, and t represents a particular iteration, while the $\theta(t + 1)$ represents the corrected θ . In addition, η is a hyper-parameter called the step size or learning rate; it controls how much gradients that used to update the parameters.

Backpropagation in multilayer perceptron

The structure of a multi-layer perception can be regarded as a function composition. For the variable in a function composition, the gradient (or derivative) is computed by chain rule (Equation 3.6). Supposing $y = f(u)$, $u = g(x)$, the derivative of $\frac{\partial y}{\partial x}$ is:

$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial u} \frac{\partial u}{\partial x} \quad 3.6$$

Supposing Equation 3.3 is the output layer, the gradient of $w_i^l(t)$ can be written:

$$\frac{\partial \zeta(t)}{\partial w_i^l(t)} = \frac{\partial \zeta(t)}{\partial y_i^l(t)} \frac{y_i^l(t)}{\partial z_i^l(t)} \frac{\partial z_i^l(t)}{\partial w_i^l(t)} \quad 3.7$$

Where

$$\frac{\partial \zeta(t)}{\partial y_i^l(t)} = -(y_i'(t) - y_i^l(t)) \quad 3.8$$

$$\frac{y_i^l(t)}{\partial z_i^l(t)} = \varphi'(z_i^l(t)) \quad 3.9$$

$$\frac{\partial z_i^l(t)}{\partial w_i^l(t)} = y_j^{l-1}(t) \quad 3.10$$

φ' represents the derivate of φ . Hence, the use of Equation 3.8, 3.9 and 3.10 in 3.7 yields

$$\frac{\partial \zeta(t)}{\partial w_i^l(t)} = -(y_i'(t) - y_i^l(t))\varphi'(z_i^l(t))y_j^{l-1}(t) \quad 3.11$$

The gradient of $w_i^l(t)$ is obtained, then put Equation 3.11 into 3.5 to update the $w_i^l(t)$.

$$w_i^l(t+1) = w_i^l(t) - \eta \frac{\partial \zeta(t)}{\partial w_i^l(t)} \quad 3.12$$

It can be seen the most important thing is to obtain the $\frac{\partial \zeta(t)}{\partial z_i^l(t)}$. Normally, this component called the local gradient, it is defined by:

$$\delta_i^l(t) = \frac{\partial \zeta(t)}{\partial y_i^l(t)} \frac{y_i^l(t)}{\partial z_i^l(t)} \quad 3.13$$

In order to update $w_i^{l-1}(t)$, the local gradient of l^{th} layer needs to be propagated to the $l - 1^{th}$ layer. Based on the chain rules, the processing is:

$$\frac{\partial \zeta(t)}{\partial w_j^{l-1}(t)} = \frac{\partial \zeta(t)}{\partial y_i^l(t)} \frac{\partial y_i^l(t)}{\partial z_i^l(t)} \frac{\partial z_i^l(t)}{\partial y_j^{l-1}(t)} \frac{\partial y_j^{l-1}(t)}{\partial z_j^{l-1}(t)} \frac{\partial z_j^{l-1}(t)}{\partial w_j^{l-1}(t)} \quad 3.14$$

$$\frac{\partial \zeta(t)}{\partial w_i(t)} = \delta_i^l(t) \quad 3.15$$

$$\frac{\partial z_i^l(t)}{\partial y_j^{l-1}(t)} = w_i^l(t) \quad 3.16$$

$$\frac{\partial y_j^{l-1}(t)}{\partial z_j^{l-1}(t)} = \varphi' (z_j^{l-1}(t)) \quad 3.17$$

$$\frac{\partial z_j^{l-1}(t)}{\partial w_j^{l-1}(t)} = y_k^{l-2}(t) \quad 3.18$$

Hence, $\frac{\partial \zeta(t)}{\partial w_j^{l-1}(t)}$ is equal to the product of equation 3.15-2.18. Then put the product into 3.5 to update the weights in $l - 1^{th}$ layer. For the updating of $l - 2^{th}$ or other layers, repeat the above computing method.

3.2.3 Activation function

In multilayer perceptron, an activation function aims to improve the performance in order to learn more complex data. This technique is inspired by the activation and inhibition of biological neurons. When the intensity of the received signal is greater than a certain threshold, the neuron is activated and produces an output. In contrast,

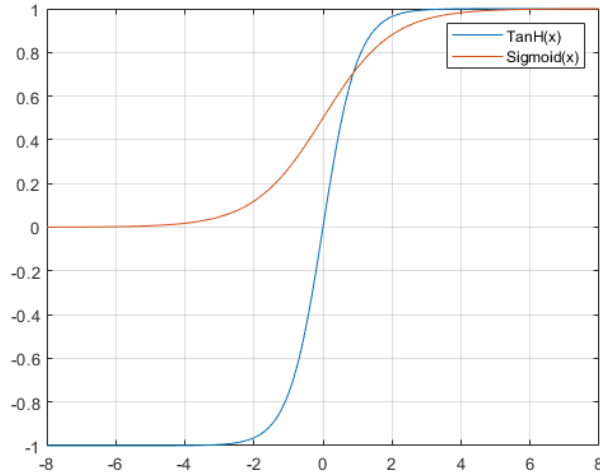


Figure 3. 3 Shapes of the sigmoid and tanh activation function.

when the signal intensity is relatively small, the neuron is inhibited. The biological neuron can be modelled as a threshold function that outputs a step signal. The activation function must a non-linear function. In addition, because the operation of an ANN is very complex, the computing of activation function should be simple, whether in the feed forward or backward. At present, the most commonly used activation functions are the sigmoid, tanh, and ReLU functions.

Sigmoid & Tanh

These two activation functions are widely used in multilayer perceptron. The sigmoid function is an 'S'-shaped function (Figure 3.3) in the range of [0,1]. The tanh function is also an 'S'-shaped function but with a range of [-1,1]; it is defined as

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \quad 3.19$$

$$Tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad 3.20$$

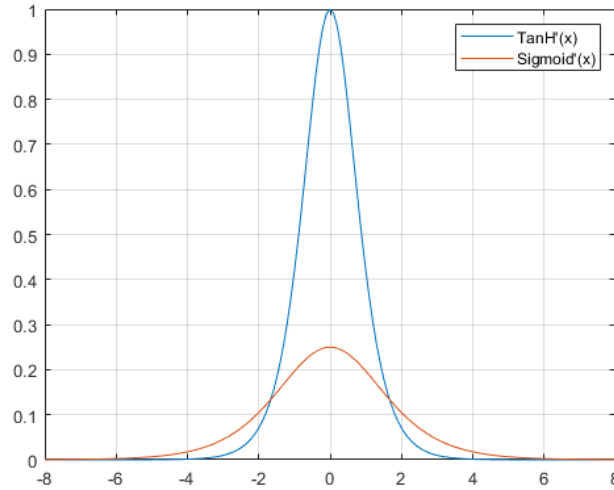


Figure 3. 4 Shapes of the derivatives of the sigmoid and tanh activation functions.

Their derivatives are as follow, the corresponding curves are shown in Figure 3.4.

$$\frac{\partial \text{sigmoid}(x)}{\partial x} = \text{sigmoid}(x) * (1 - \text{sigmoid}(x)) \quad 3.21$$

$$\frac{\partial \tanh(x)}{\partial x} = 1 - \tanh^2(x) \quad 3.22$$

The ReLU function [41], [42] is a piecewise function that outputs 0 when the input is less than or equal to 0 and outputs the original value otherwise. It is defined as follows:

$$\text{Relu}(a) = \max(x, 0) \quad 3.23$$

In the backward, the processing of Relu is as follows:

$$\text{Relu}(x)' = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases} \quad 3.24$$

ReLU is very simple to calculate in both the forward and reverse directions, which helps to improve the training speed for deep networks. Leaky ReLU [43] and ELU [44]

are developed based on ReLU. These two activation functions aim to obtain more information when the gradient less than zero.

3.2.4 Vanishing and exploding gradients

In theory, the backpropagation algorithm can train a neural network with an arbitrary number of hidden layers. However, in practice, too many hidden layers are likely to cause vanishing and/or exploding [45] gradient problems, which makes the network unable to learn. This is an inherent flaw of the backpropagation algorithm.

Vanishing gradient means that during backpropagation weights' update, at some network layers, gradients' values get very close to zero, which means that the corresponding weights are unable to be effectively updated. Taking Equation 3.14 as an example, the backpropagation of the gradient involves a chain of multiplications. If the magnitude of the derivative for each component is smaller than one, the multiplication value of all items will become very small. It can be seen from this, that when the number of hidden layers is too large, the deeper layers cannot obtain sufficient update of their weights. Similarly, exploding gradients means that the derivative of each term is larger than one, and the final gradient value could become very large. Both of these effects will cause the parameters to be updated incorrectly.

Currently, ReLU [41], [42] is one of the popular approaches to decrease the impact of vanishing gradients. Equation 3.24 shows that the derivative of ReLU, for positive input values, is always equal one, and therefore it does not affect the propagation of the gradient so strongly.

Weights regularization [46] is used to prevent the exploding gradients from happening. For example, the L2 regularization $\|w\|_2^2$ is included in loss function, preventing weights become too large.

3.2.5 Multi-class and softmax

Since sigmoid function only maps the output to the range of 0 to 1, it cannot be applied in multi-class classification tasks. To solve this problem, the sample labels are represented in the form of sequences of 0 and 1. Consider an example with three classes:

The first class: 1, 0, 0

The second class: 0, 1, 0

The third class: 0, 0, 1

The position of the 1 in each sequence corresponds to the label number. Then, the number of neurons in the output layer is set equal to the sequence length. For example, if the number of classes is 3, then there are three neurons in the output layer. Each neuron in the output layer is connected to the outputs of the last hidden layer.

It should be noted that since the sigmoid function renders the output of multiple neurons close to 1, the neurons in the output layer have no activation function, which means that their range is \mathbb{R} . Then, these outputs are normalized such that their range is from 0 to 1 and their sum is 1. That is, the output of each neuron is the probability that the current sample belongs to the corresponding class. The normalization function used is the softmax function.

Supposing Equation 3.3 is the output layer, then softmax is defined as follows:

$$\text{Softmax}(z_i^l) = \frac{e^{z_i^l}}{\sum_{i=1}^{n^l} e^{z_i^l}} \quad i = 1, 2, 3, \dots, n^l \quad 3.25$$

Where n^l is equal to the number of classes, so i represents as a specific class. In this way, multi-class samples can be classified. When the probability output by a neuron

is greater than other neurons, the sample is considered to belong to the class corresponding to that neuron.

3.2.6 Loss function

The choice of loss function is depending on the type of task and data samples. Normally, there are two popular loss functions, they are mean square error (MSE) and cross entropy. MSE can be used in both regression and classification, but it performs better at regression tasks. Cross entropy only can used to classification, and it is more popular than MSE. Section 3.2.1 has been introduced MSE. Here, if there are multi data samples, Equation 3.4 can be rewritten:

$$MSE = \frac{1}{2M} \sum_{m=1}^M \sum_{i=1}^{n^l} (y_{mi}' - y_{mi}^l)^2 \quad 3.26$$

Where M represents the number of samples.

For a binary classification task, cross entropy can be written:

$$cross\ entropy = - \sum_{i=1}^{n^l} y_i' \log(y_i^l) + (1 - y_i') \log(1 - y_i^l) \quad 3.27$$

For multi input data samples, the error is obtained by Equation 3.29.

$$cross\ entropy = - \frac{1}{M} \sum_{m=1}^M \sum_{i=1}^{n^l} y_{mi}' \log(y_{mi}^l) + (1 - y_{mi}') \log(1 - y_{mi}^l) \quad 3.28$$

The negative sign is used to take the minimal value, because logarithm is incremental. Then, both weights and bias are corrected by solving for the minimal value. Based on cross entropy, the parameter w_i^l in Equation 3.3 can be updated by

$$w_i^{l+1} = w_i^l - \eta \frac{1}{M} \sum_{m=1}^M \sum_i^{n^l} (y_{mi}^l - y_{mi}') z_i^l \quad 3.29$$

For a multi-class classification task, the combination of softmax (Equation 3.25) and cross entropy can be modified to:

$$\textit{Softmax with cross entropy} = -\frac{1}{M} \sum_{m=1}^M \sum_{l=1}^{n^l} y_{mi}' \log(\textit{softmax}(z_{mi}^l)) \quad 3.30$$

If M equal to 1, y_i' represents a class in the form of a sequence. Since only one element in the sequence is 1 and the others are 0, the above equation can be further simplified as follows:

$$\textit{Softmax with cross entropy} = -\frac{1}{M} \sum_{m=1}^M y_m' \log(\textit{softmax}(z_m^l)) \quad 3.31$$

3.3 Convolutional neural network

Convolutional neural networks (CNN) are a representative deep learning method used in image processing. The main feature of a convolutional neural network is that it uses convolutions, instead of the operators used in traditional feature extraction, and the whole network is built by stacking multiple convolutional layers. During training, the weights of the convolutional layers are adjusted through backpropagation, enabling these layers to search for valuable image information for classification. In addition, the shallow hidden layers of a convolutional neural network produce low-level features, but after processing through several hidden layers, these low-level features can be fused into high-level features, giving the CNN strong robustness.

3.3.1 Convolutional layer

Convolution is a fundamental operation in digital signal and image processing. It is used to modify a function by taking its weighted sum with another function. The mathematical definition for a 1-D function is as follows:

$$\int f(u)g(x - u)du \quad 3.32$$

In discrete form,

$$(f \times g)[n] = \sum_{m=-\infty}^{\infty} f[m][n - m] \quad 3.33$$

An image is a 2-D discrete signal; in this case, the 2-D convolution operation can be defined as follows:

$$s[i, j] = (I \times K)[i, j] = \sum_n \sum_m I[m, n]K[i - m, j - n] \quad 3.34$$

The operator function for convolution is called the kernel; it is actually a square matrix built by weights. By adopting different sets of weights, convolution can be applied for image smoothing, edge detection, sharpening and other purposes (Figure 3.5).

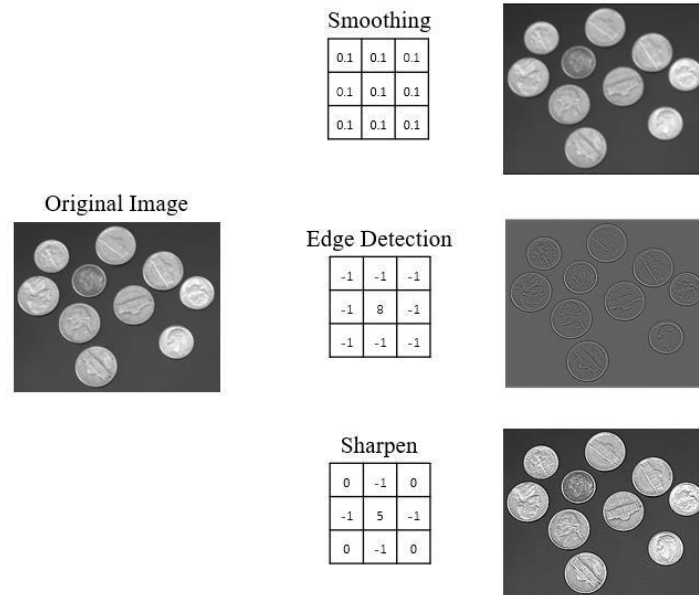


Figure 3. 5 Different types of image processing⁷ via different convolutions.

In a CNN, a hidden layer is created by several kernels, which are used to learn different features from an image. The output of convolutional layer called feature maps. Convolution can be regarded as a filter, such that useful information can be reinforced, and useless information can be suppressed. The difference between kernels and neurons is: In an MLP, two adjacent layers are fully connected; each neuron must assign weights for all of the input data (or the outputs from last layer). In a CNN, the connections between two layers are local, and the number of connections depends on certain manual settings, such as the size of the kernel. These manual settings mean that a kernel uses the same weights to process the whole image (Figure 3.6). The benefit of convolution is that the local connection approach reduces the number of necessary calculations and saves resources.

The weights of a CNN are also corrected by backpropagation. Since each kernel is locally connected to the input, the weights must be updated by means of the corresponding gradients. Here, the 2-D kernel is reshaped into a 1-D kernel to make it easier to illustrate the relationships and the full and local connections. First, for the feed forward stage, the convolution processing can be represented as shown below:

⁷ <https://uk.mathworks.com/help/images/> [Assessed 21 Oct. 2019]

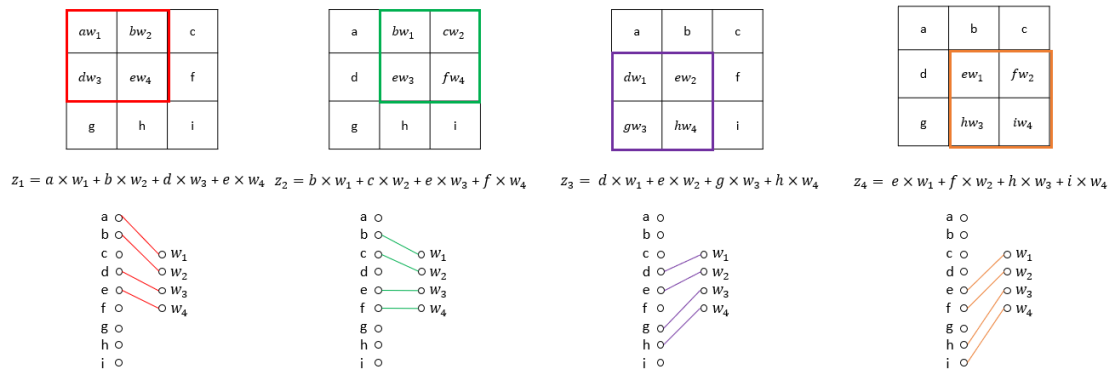


Figure 3. 6 Representation of a convolutional layer.

The w , letter and coloured squares represent weights, input and position, respectively. The different coloured lines represent the connections between the weights and pixels in different position. For the whole image, only pixels a , b , d and e are ever connected to weight w_1 in all operations; thus, the gradient for w_1 generated by the product of these pixels. Figure 3.7 shown the operation that used to assign the gradient to each weight, this operation is the extension of equation 3.11.

D represents the gradient generated by the pixels in different positions, and the coloured lines represent the corresponding connections of the weights in the forward step. The black square represents the convolution steps, and each step corresponds to a set of coloured connections. For simplicity, these connections can be transferred to the convolution operation [47].

After correcting the current weights, the CNN still needs to propagate the local gradient to the previous layer. First, it needs to check the correction between the input and output and then assign the correct weighted gradient to each input. Figure 3.9 shows the whole operation; this operation is the extension of equation 3.15.

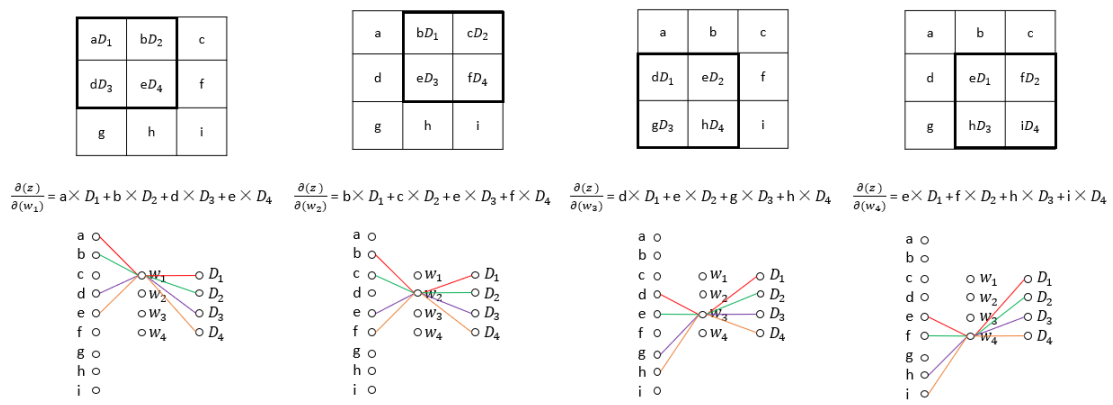


Figure 3. 7 Illustration of how gradients are being transfer to weights.

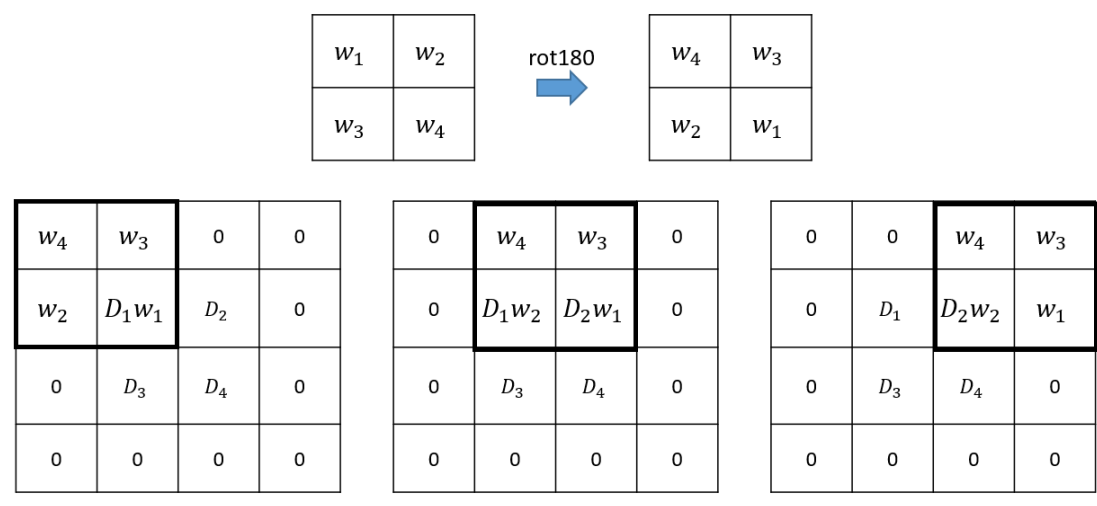


Figure 3. 8 Illustration of transposed convolution.

For pixel e, since this pixel is used in all calculations, it corresponds to D_1 , D_2 , D_3 and D_4 . It should be noted that the pixel with the most corrections does not necessarily make the greatest contribution. Figure 3.8 shows how the gradients are transferred in the normal implementation. First, the kernel is rotated, and then the rotated kernel is used to perform the convolutional operation on a padded gradient map. The padding rate is half the kernel size (if the kernel size is odd, the rate is $2N-1$). It can be seen that the result of each convolution step is the same as the connections shown in Figure 3.9.

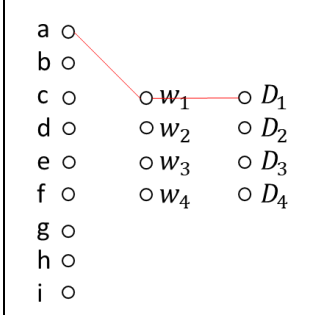
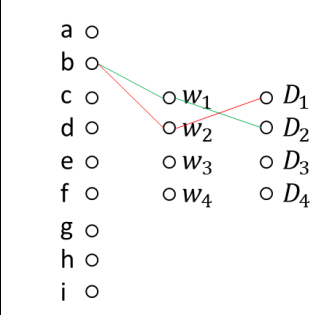
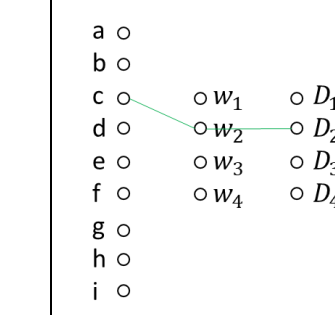
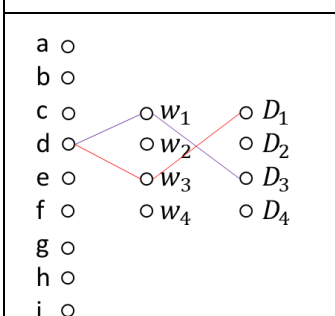
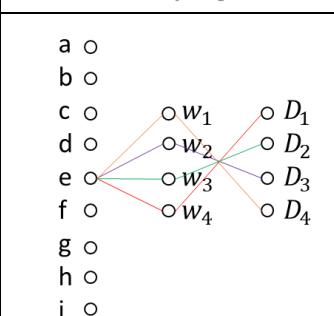
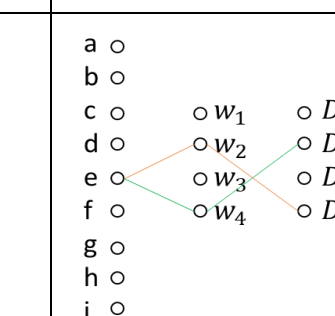
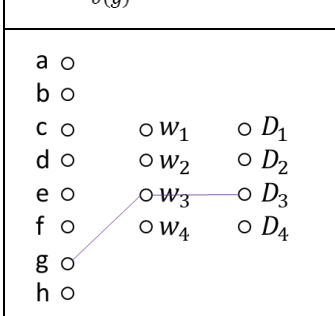
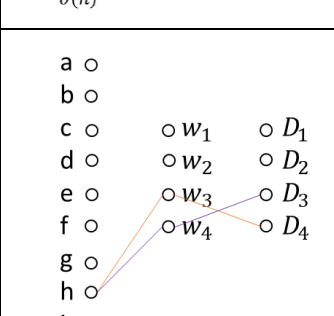
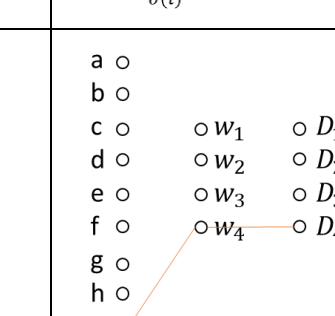
$\frac{\partial(z)}{\partial(a)} = w_1 \times D_1$ 	$\frac{\partial(z)}{\partial(b)} = w_1 \times D_2 + w_2 \times D_1$ 	$\frac{\partial(z)}{\partial(c)} = w_2 \times D_2$ 
$\frac{\partial(z)}{\partial(d)} = w_1 \times D_3 + w_3 \times D_1$ 	$\frac{\partial(z)}{\partial(e)} = w_1 \times D_4 + w_2 \times D_3 + w_3 \times D_2 + w_4 \times D_1$ 	$\frac{\partial(z)}{\partial(f)} = w_2 \times D_4 + w_4 \times D_2$ 
$\frac{\partial(z)}{\partial(g)} = w_3 \times D_3$ 	$\frac{\partial(z)}{\partial(h)} = w_3 \times D_4 + w_4 \times D_3$ 	$\frac{\partial(z)}{\partial(i)} = w_4 \times D_4$ 

Figure 3. 9 Transferring gradients to different layers.

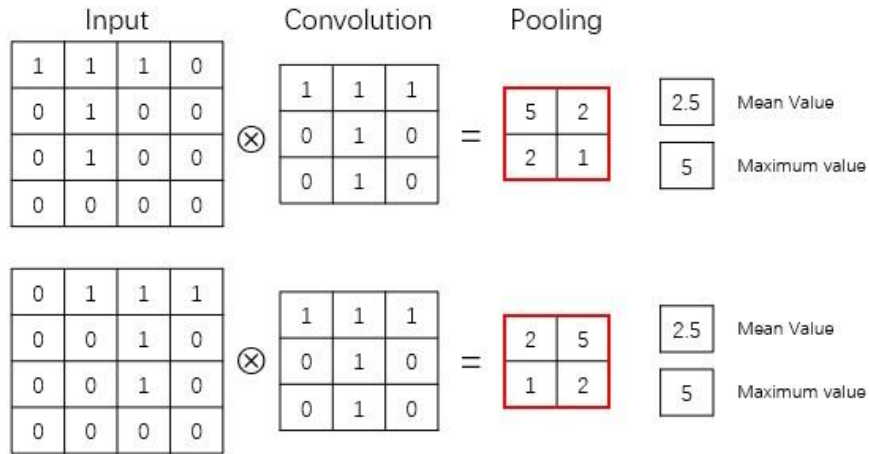


Figure 3. 10 The different pooling methods in the feed forward step.

3.3.2 Sub-sampling

A sub-sampling layer is used to further extract features after a convolutional operation. The operation performed in a sub-sampling layer is pooling; the slight displacement of each target in the image has little effect on the pooling operation, which helps to improve the resistance of a CNN to image deformation. In the pooling operation, the feature map is divided into overlapping or nonoverlapping blocks, and the maximum or mean value is then obtained from each block. Pooling is similar to convolution. It also uses a kernel to scan the image and obtain the results, but there are no weights in the kernel.

Normally, a sub-sampling layer is defined by four hyperparameters. The first three parameters are the kernel size, stride and padding, which are the same as in a convolutional layer; they control how the pooling kernel scans the input data (or feature maps). The last parameter is the type of pooling performed, i.e., max pooling or average pooling. Generally, max pooling is more popular than average pooling because average pooling degrades the purity of the learned features and reduces the difference between high and low responses.

Figure 3.10 shows a typical example of pooling. The convolutional kernel generates a feature map. The highest response is equal to 5, but less important responses are

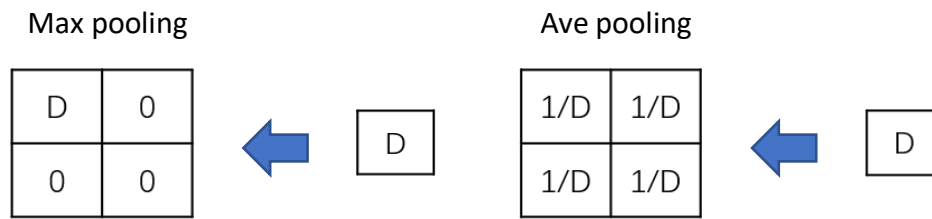


Figure 3. 11 The different pooling methods in backward step.

also generated. Either max or average pooling further extracts a higher value from the feature map. A comparison of these two operations reveals that the result of max pooling is more reasonable than average pooling, because the latter reduces the high response. In addition, these two pooling methods are both translationally invariant. Even if the positions of the values in 'T' are changed, the extracted feature value remains the same.

Because there are no trainable weights in a pooling layer, it is only necessary to propagate the gradient during the backward step. For max pooling, the gradient is assigned to the position of the feature that remains. The other values are set equal to zero, meaning that their corresponding inputs do not contribute. For average pooling, the gradients are evenly assigned to all elements of each block. Figure 3.11 illustrates the whole operation.

3.3.3 CNN structure

LeNet-5 is a famous classification network in the early stage of CNN developments [4]. This concept was validated on the MNIST dataset of handwritten numerals. The number of convolutional, sub-sampling and fully connected layers is 2 (Figure 3.12). The first convolutional layer has 6 kernels and the second convolutional layer has 12 kernels. The number of neurons in the first fully connected layer is 84.

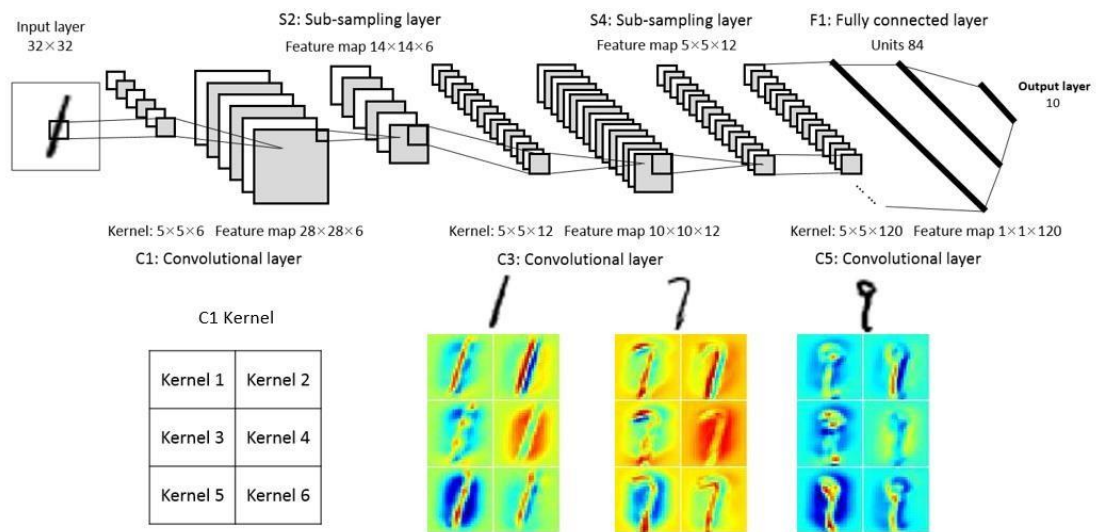


Figure 3. 12 The structure and learnt features of LeNet-5.

The left side represents 6 kernels in LeNet-5, while the right side shows the corresponding feature maps for different input data. Red in the feature maps represents the high value response, while the blue part represents low value response. It is shown that different kernels are found for different features, and network learnt them without any handcrafted features.

3.3.4 Dropout

With dropout [48], in each epoch, a certain number of randomly selected nodes are ignored in the BP update to address the over-fitting problem. This operation is implemented by means of the element-wise multiplication of the features by a mask of the same size. The elements of the mask take values of 0 and 1, and the quantities of these two values are controlled by the dropout rate. This rate represents the proportion of dropped neurons. Dropout can also be added to convolutional layers. In Figure 3.13, the black blocks represent zeroes, and the white blocks represent ones [49]. Through the element-wise multiplication of the features and the dropout mask, some features are replaced with zeroes; these features correspond to dropped nodes.

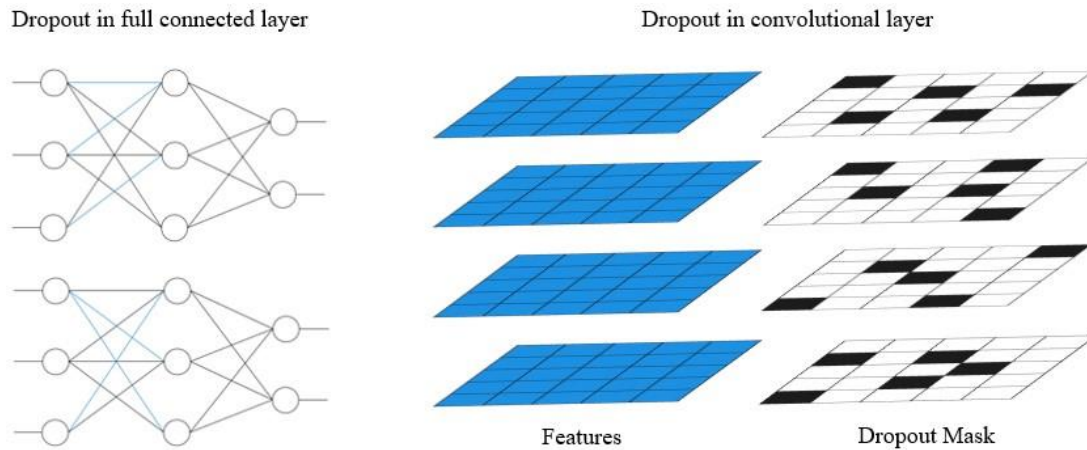


Figure 3. 13 Dropout in a fully connected layer and a convolutional layer.

The reason for applying dropout is that the updating of the weights in the network no longer depends on the combined action of hidden nodes with fixed relations, and each node is relatively independent of each other node. The principle of dropout is quite similar to that of bagging. The intent of both is to establish various sub-classifiers based on different subsets of the training data to prevent a situation in which certain features dominate the training process.

3.3.5 Normalization

Normalization refers to adjusting the location and scale of a data distribution to make the data fall into a specified region without changing the sample distribution so as to facilitate subsequent processing via machine learning algorithms. There are several different normalization methods. Among them, z-scores are commonly used. The processing is defined as shown in the equation below, where \bar{x} and σ represent the mean and standard deviation of input data samples, respectively. This equation adjusts the original distribution of x to a distribution with its central point at the origin and the variance is equal to 1.

$$\text{normalized } x = \frac{x - \bar{x}}{\sigma} \tag{3.35}$$

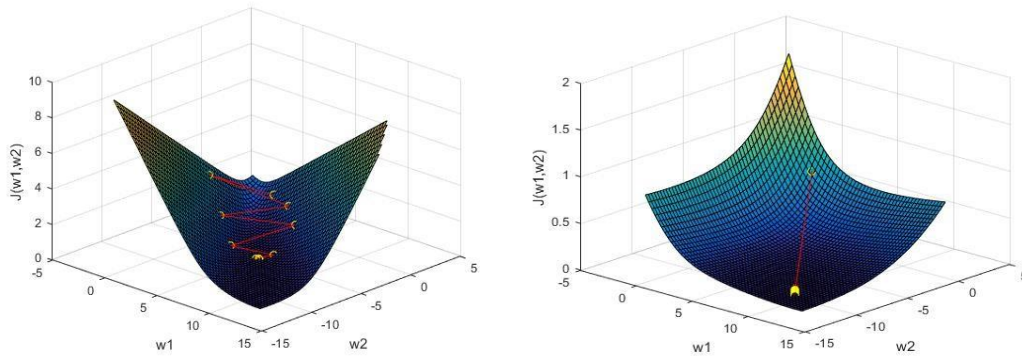


Figure 3. 14 The results of gradient descent without normalization (left) and with normalization (right).

For gradient descent, the use of non-normalized samples can lead to drastic changes in errors, meaning that only a low learning rate can be used to slowly search for the minimum. Normalized sample features vary more gradually, thus permitting a higher learning rate; consequently, optimization can be achieved in fewer iterations. In Figure 3.14, logistic regression is applied to both the original data and the normalized data. The learning rates were 0.04 and 0.5, respectively. Their initial weights are same. The update progress in both cases after 10 iterations is visualized below.

First, it can be seen that even though the learning rate in the former case is much lower than that in the latter, the direction changes dramatically during the initial stages of iteration, which is unfavourable when searching for the minimum. In the latter case, the direction remains stable. Second, the errors in the 10th iteration are 0.09 and 0.02, respectively, and the overall optimization trend in the latter case is much faster than that in the former, thus demonstrating that normalization helps to speed up the updating of the parameters.

Image data also need to be normalized before being used to train a CNN. The normalization method for image data is to subtract the average value among all training images from each input image. Scaling is unnecessary because the values of the pixels always distribute in the range of 0 to 255.

3.5 Typical CNN architecture

This section focuses on the current popular CNN structures, which are mainly used in image classification and regression.

3.5.1 AlexNet & ZFNet

AlexNet is the first deep convolutional neural network that has attracted the attention of researchers. In 2012, a team using a CNN model for the Large Scale Visual Recognition Challenge (LSVRC) won the first prize. They used seven improved CNNs designed to classify 1,000 objects [5]. The top-5 error rate for that method was 15.3%, which was far below the 26% error rate of the second-best method.

The whole network has eight trainable layers, consisting of 5 convolutional layers and three fully connected layers (Figure 3.15). This method contributed much to the development of CNNs. First, it extended the ReLU and dropout techniques to reduce gradient vanishing and over-fitting, thus making the training of deep CNNs feasible. Second, to save time, this network was trained with a GPU. This approach has also become standard for subsequently developed CNNs.

ZFNet [50] was developed based on AlexNet. Its contribution is the proposal of the deconvnet [51] technique, which aims to visualize selected internal features of a CNN. This technique can be regarded as a further application of backpropagation. The selected feature map is preserved, and the other feature maps are set to 0. Thus, the gradient of the input image corresponds only to the selected feature map, and the highest responses represent the most valuable contents of the image.

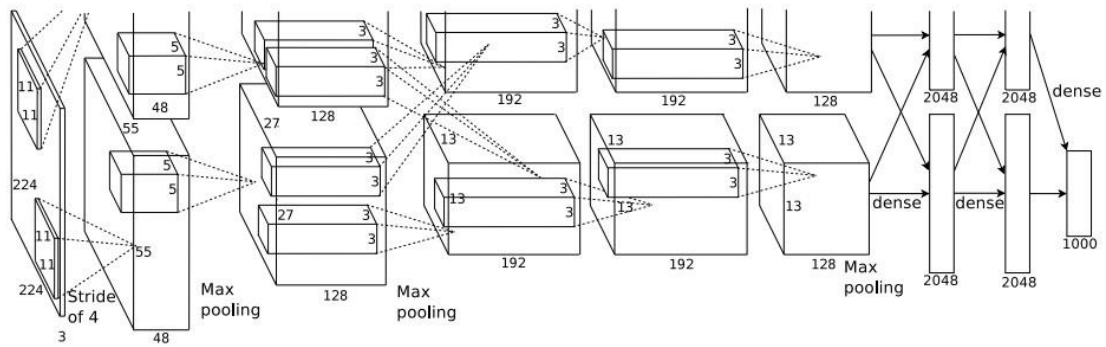


Figure 3. 15 The structure of AlexNet [5].

The structure of ZFNet is the same as that of AlexNet, but the kernel size and stride of the first layer are 7 and 2, respectively, rather than 11 and 4. This modification is based on observations of the visualized feature maps. These corrected parameter values allow the first and second convolutional layers to learn more features. It is a small development, but this idea has been applied in most subsequent CNNs. ZFNet participated in LSVRC 2013 and took first place, with a top-5 accuracy of 13.51%.

3.5.2 All convolutional network

All convolutional nets [52] are constructed entirely of convolutional layers and can achieve state-of-the-art results. The sub-sampling layers are replaced with convolutional layers with a larger stride (Figure 3.16). A fully connected layer or a specific convolutional layer is used to obtain 1×1 features; this is similar to the C5 layer in LeNet-5.

3.5.3 VGG16 network

VGG16 [53] can be regarded as an enlarged AlexNet. It was proposed in 2014. This architecture is based on a building block consisting of a pooling layer and several convolutional layers; the network is created by stacking

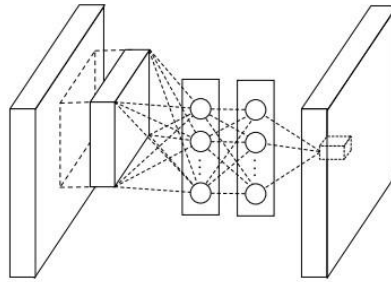


Figure 3. 16 NIN structure [54].

several such blocks. The whole network consists of fifteen convolutional layers and five pooling layers, divided into 5 blocks. In LSRVC 2014, this network earned second place for image classification, with a top-5 error rate of 6.7%.

3.5.4 Network in network

Network in Network (NIN) structure [54] is not a specific CNN but a design concept. First, to learn complex features, a cross-channel parametric pooling layer is added between two convolutional layers. This layer enables the creation of denser connections. In addition, this layer is equivalent to a 1×1 kernel, and it fuses and learns pixels from different channels in the same position.

Another contribution is that a global average pooling layer is used in place of a fully connected layer. The final 1×1 kernel fuses the current feature maps to make the quantity of feature maps correspond to the number of classes. Then, the global average value of each fused map is taken as the classification probability. The reason is that to feed the outputs of a convolutional layer into a fully connected layer, many weights must be assigned to connect all the elements, which can easily lead to over-fitting.

2.5.5 Inception

Inception, also called GoogLeNet [55], is the winner of LSRVC 2014. In its design, not only the depth but also the breadth of the network is considered.

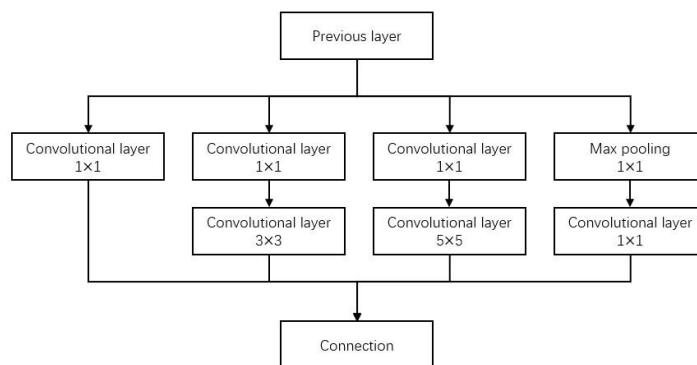


Figure 3. 17 The structure of inception.

Similar to VGG16, it performs feature learning through the re-use of a specific building block. In this block, different kernel or pooling settings can be adopted to extract different features (Figure 3.17). To avoid gradient vanishing, three loss functions are adopted at different depths. During testing, only the deepest outputs are retained as the results.

Inception represents the further development of a 1×1 kernel. Here, a 1×1 kernel is applied to reduce the dimensions of the feature maps, causing the subsequent larger kernel to contain fewer weights. Inception has more layers than VGG16 does, but the overall network size is smaller. Moreover, the 1×1 kernel can be regarded as an efficient way to increase the depth.

In subsequent work, several new methods have been developed based on Inception. The fundamental idea of these methods is inherited from the original Inception, but the original structure has been adjusted. In Inception V2 [56], a new module called batch normalization [57] was added to speed up training. In addition, in this network, the one large convolution kernel of the original is split into two smaller convolution kernels. In Inception V3, the overall convolution operation is replaced with a module consisting of row convolution and column convolution, and the original Inception structure is adjusted. In Inception V4 [58], Inception and ResNet are combined to build a deeper network. Xception [59] has also been developed based on Inception V3; in this network a depthwise separable convolution operation is used in place of the regular convolution operation in the Inception module.

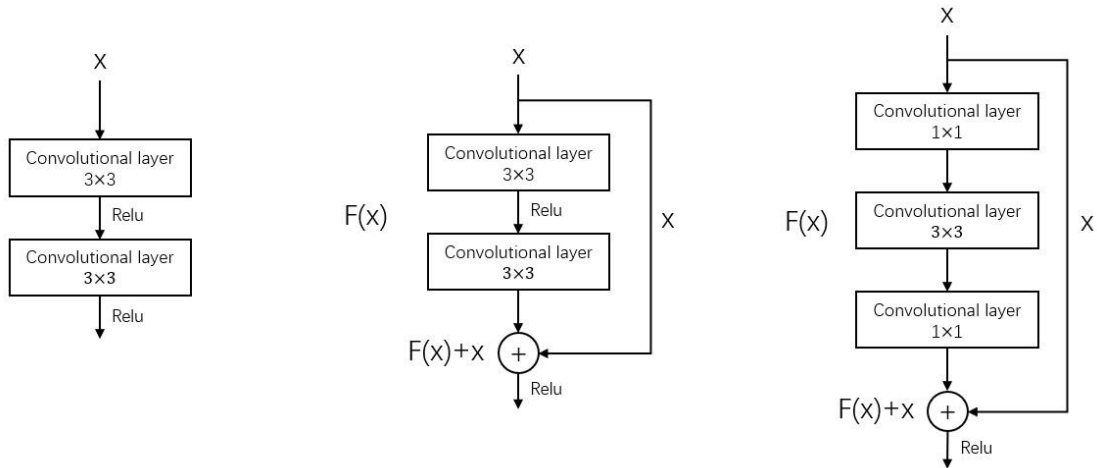


Figure 3. 18 The structure of a ResNet block. Left: regular convolutional block. Middle: ResNet block (<50 layers). Right: ResNet block (> 50 layers).

3.5.6 Deep residual network

The ResNet architecture [60] was presented in 2015 and represents an important turning point in the recent development of CNNs. The ResNet architecture was developed based on a new type of building block with fused inputs and outputs. The most popular implementations based on this technique are Res50, Res101 and Res152. The digits in these notations represent the number of convolutional layers in each network. These presented ResNets are created by stacking many building blocks. For a network with fewer than 50 layers, each block contains two convolutional layers, and the sizes of their feature maps are the same, as shown in Figure 3.18.

For a network with more than 50 layers, each block consists of two 1×1 convolutional layers and a 3×3 convolutional layer. As in Inception, the first 1×1 kernel is used to reduce the number of dimensions to save memory. The final 1×1 kernel aims to increase the complexity of the features learned by the 3×3 kernel. In 2015, ResNet-based ensemble methods earned first place in LSRCV, with a global top-5 error rate of 3.57%.

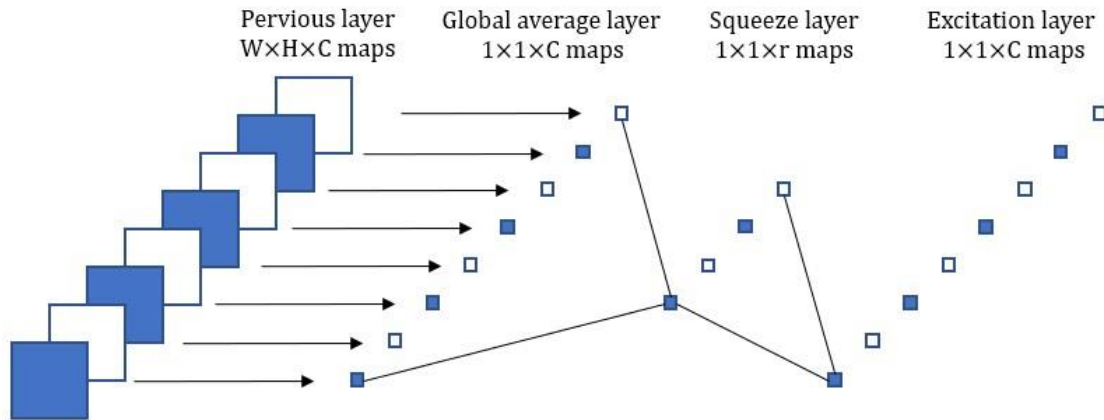


Figure 3. 19 The structure of SE module.

3.5.7 Squeeze and excitation networks

A squeeze-and-excitation network (SE-Net) was the champion of the last LSRVC (2017). This architecture [61] was proposed based on a module that aims to limit useless features as much as possible. It can be regarded as an upgrade to the cross-channel parametric pooling layer of the NIN architecture in which the fusion process focuses on the global feature map rather than single pixels.

An SE module can be added to any number of hidden layers. It consists of four layers (Figure 3.19): a global average pooling layer, a squeeze layer, an excitation layer and a scale layer. The squeeze layer is a convolutional layer with a $1 \times 1 \times r$ kernel and the ReLU activation function; r must be smaller than the number of input maps, which is why this layer is called the “squeeze” layer. There are two motivations for its use: First, it is used to add more nonlinear descriptors to fit the pattern between different channels. Second, similar to inception, it can reduce the number of feature dimensions to save resources. The excitation layer is also a convolutional layer with a 1×1 kernel, but it has a sigmoid activation function, and the number of outputs is equal to that of the previous layer (the number of inputs to the SE module). An output value nearer to 1 indicates that the corresponding feature from the previous layer is more valuable. Subsequently, these outputs are used to weight the features from the previous layer.

3.6 Summary

This section reviews the development of multilayer perceptron, backpropagation and convolutional neural network. It can be seen that there is essentially no difference between the multilayer perceptron and CNN. Both algorithms are composed of two parts, namely, the model and the optimization method. The CNN and its related methods merely further develop these two parts. For the former, the development of the CNN involves combining feature learning and ANN, so that feature extraction of images is not dependent on manual design. The follow-up approach focuses on how to complete feature learning more accurately or efficiently. For the latter, the CNN mitigates gradient dispersion primarily through ReLU and ResNet, allowing deeper structured networks to work. In addition, in order to optimize deeper and deeper models, new optimization methods are being continuously proposed.

Chapter 4. Review of image segmentation methods

4.1 Introduction

Image segmentation methods can be coarsely divided into methods using handcrafted feature spaces in which data aggregation/segmentation is taking place (these methods are sometimes called classical or traditional methods) and more modern methods utilising the so-called deep learning methodology. The former are based on techniques and methods mostly originate from digital image processing, whereas the latter are rooted in machine learning with the deep learning based on convolutional neural networks (CNNs). Since the recent introduction of end-to-end trained networks, deep learning methods are gradually replacing the more traditional methods, especially for semantic segmentation and instance segmentation.

This section introduces a selection of representative image segmentation methods. For the CNN base methods, the focus is placed on the two most successful deep segmentation networks, FCNs and U-Nets, and the methods developed based on these two network types. This survey serves as the basis for development of the methods proposed in this thesis.

4.2 Traditional image segmentation methods

This section is to introduce the representative traditional methods for segmentation. These methods are designed based on the typical image properties.

4.2.1 Thresholding

Thresholding is a simple image segmentation method that is especially suitable for images with large differences in colour (or greyscale values) between different foreground and background objects. For a binary segmentation, thresholding algorithm searches for a threshold T such that pixels with values smaller than T are assigned to one class and the rest to another class. Many thresholding methods exist, the most common of which is the histogram-based method and Otsu methods [62].

In the histogram-based method, it is assumed that the grey levels of the object and the background are different and that the image intensity distribution has two distinctive modes, with the value of between the modes selected to as the threshold. The threshold is usually selected by the user through manual selection from the image histogram. In the improved version of this operation, first, a threshold T_0 is selected, and the image is divided into r_1 and r_2 regions based on T_0 . Then, the average intensities μ_1 and μ_2 of r_1 and r_2 are calculated to obtain a new threshold $T_1 = (\mu_1 + \mu_2)/2$, and the process is repeated until μ_1 and μ_2 no longer change.

Otsu is a fully automatic threshold method for image segmentation. The percentages of foreground and background in the image are denoted by ω_1 and ω_2 , respectively, and the average pixel values within these two regions are denoted by μ_1 and μ_2 , respectively; then, the mean value for the image can be calculated as

$$\bar{\mu} = \omega_1 \times \mu_1 + \omega_2 \times \mu_2 \quad 4.1$$

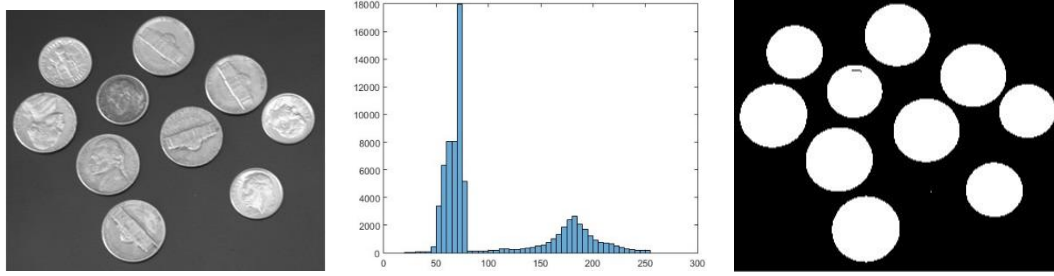


Figure 4. 1 Histogram-based method image segmentation method.

Where

$$\omega_1 + \omega_2 = 1 \quad 4.2$$

The inter-class variance is:

$$\sigma^2 = \omega_1 \times (\mu_1 - \bar{\mu})^2 + \omega_2 \times (\mu_2 - \bar{\mu})^2 \quad 4.3$$

The optimized value of the threshold is searched exhaustively so inter-class variance is maximised.

4.2.2 Clustering

Clustering was originally proposed to enable unsupervised learning. This type of learning method does not require ground truth. Instead, it analyses the distribution of the samples in the feature space and divides them based on some similar principles to achieve classification. Based on this idea, each pixel can be regarded as sample, and pixels with similar colours can be assigned to the same class using a clustering method, thereby completing the image segmentation task.

K-means [63] and mean shift [64], [65] are common image clustering methods, both of which require the specification of hyperparameters. The K-means method requires the number of clusters, that is, the number of regions that the samples should be divided into, to be specified. The mean shift algorithm requires more hyperparameters, including the type of kernel function, the number of iterations, and the bandwidth.

The mean shift algorithm is more flexible and can determine if the image regions are “independent” or should to be merged into a single region, the performance of the mean shift is highly dependent on the selected hyperparameters. Although the K-mean method also relies on hyperparameters, the adjustment of the hyperparameters is very simple.

In general, clustering-based segmentation methods can only be used on images whose background and foregrounds differ significantly, i.e. the regions can be considered to be homogeneous with respect to a specific characteristic e.g. intensity, colour or texture.

4.2.3 Region growing

Region growing is an iterative image segmentation method. It has three components: the initial point (seed) positions, growth criteria, and stopping conditions. First, it starts from a given seed points and incorporates similar corresponding regions into the same region in accordance with the growth criteria. Then, the identified neighbourhood regions are used as new seed points for continued growth until the stopping conditions are satisfied, for example, no more pixels are available that satisfy the growth criteria.

The performance of region growing depends primarily on the suitability of the growth criteria. The growth criteria have no fixed form. For example, multiple growth criteria can be designed based on thresholding. First, a threshold can be used to determine whether a pixel belongs to a specified distribution. Second, if the difference between two pixels is used to represent their similarity, a threshold can be used to determine the magnitude of similarity. Alternatively, when the average value in the neighbourhood of a certain point is used as a feature, a threshold value can be used to determine the approximate distribution of the pixel values. Growth criteria can also have more complex forms, such as probability values output by a classification method. The same is true for the stopping conditions.

The disadvantage of region growing is that it has a high computational load. In addition, simple growth criteria are sensitive to noise, such as shadows or illumination

variations, which can easily cause under-segmentation or over-segmentation. To avoid this problem, multiple growth criteria are sometimes used to control regions expansion.

4.2.4 Machine learning and image segmentation

In machine learning methods, image segmentation is usually performed via pixel-wise classification, where each pixel is treated as a data point to be classified. This approach consists of three components: sliding window, feature extraction and a machine learning algorithm. First, a window with a side length of size n is created and used to extract the neighborhoods centered on each pixel. Then, each extracted patch is described by means of a feature vector, and finally, these feature vectors are input into the machine learning algorithm. The class of each patch corresponds to the class of its center pixel. During training, a certain number of patches can be randomly extracted from the foreground and background of training images, as indicated by the ground truth. The class of each sample is determined by the class index of the corresponding position in the ground truth.

The size of the sliding window is a key hyperparameter. The larger the window, the more information it contains, and it becomes more complex to represent that information. The information in a small window is spatially well defined, but its relation to the overall foreground characteristics is low. This relationship may result in missing data and poor segmentation performance. Therefore, to mitigate the problems, feature vectors extracted with several windows of different sizes can be concatenated before processing with the machine learning method.

The performance of this kind of segmentation method depends on accurate features. The current feature extraction methods are often designed based on image characteristics, the representative algorithms are Histogram of Oriented Gradient [66], Local Binary Pattern [67], and histogram. However, the main drawbacks of these methods are:

1. Feature extraction and machine learning algorithm are completely independent. The selection of image features is often based on people's ideas. Machine learning methods actually learn features that people deem effective; consequently, features that are difficult for humans to observe but probably more effective will be ignored.
2. The features described by the method are relatively simple, and the objects described are often the lower-level features in the image. More complex semantic features can only be described by combining various kinds of feature vectors. However, this stacking method can hardly integrate features effectively and will undoubtedly increase the amount of computation.

4.3 Image segmentation with fully convolutional networks

The fully convolutional network (FCN) architecture was the first type of end-to-end network to be successfully used for semantic image segmentation based on deep learning [33]. FCN can process images of any size and obtain a full-size segmentation result without the need for additional step of pre-processing. The structure of an FCN can be divided into two parts, an encoder and a decoder. The former is used to extract low resolution, high-level features from the input image. The latter fuses these features and converts them into low-resolution segmentation results, then restores their size by means of up-sampling and cropping layers. The loss in the backward direction is determined by processing the full-scale segmentation result and ground truth. Then, the errors are propagated to each hidden layer that needs to be trained. This method not only simplifies the steps of image segmentation but also is more accurate than the traditional methods. For the PASCAL semantic segmentation challenge, FCN methods (and those related) occupy almost the entire leaderboard and continue to yield the best results.

Encoder

The encoder can be any CNN whose fully connected layer has been removed. It can be one of the existing CNN architectures or a custom built one. When designing an FCN, the choice of the encoder is usually determined by the complexity of the images and the performance of the hardware, with the goal of avoiding unnecessary calculations. It should be noted that when using FCN model, the final feature map is required to be of a certain size, otherwise, some smaller segmentation objects could be missed. Therefore, the rate of down-sampling should not be too large. For some composite tasks, FCNs have been developed that contain multiple encoders working in parallel; the outputs of these encoders are processed and then input into the decoder [37].

Decoder

The decoder consists of a pixel classifier, an up-sampling layer and a cropping layer. The pixel classifier is used to classify the pixels in the feature maps one by one. It is a convolutional layer rather than a fully connected layer. This definition is because the number of outputs of a fully connected layer is fixed, making it impossible to process images of different sizes. For general pixel classifiers, a 1×1 convolution kernel is used to fuse the feature maps and generate low-resolution segmentation result. Of course, larger convolution kernels can also be used, but additional padding is needed to ensure that the size of the feature maps is not further reduced.

In addition, to reduce the loss of segmentation details caused by down-sampling, feature maps of different resolutions can be extracted from convolution layers at different depths in the encoder, and corresponding pixel classifiers can then be designed separately. After that, the results can be fused through up-sampling. In an FCN, this structure is called skip structure (Long et al. 2015). Direct addition could be used as the fusion method. In some subsequently developed methods, such structures can be stacked, and the dimensions can then be reduced by using a 1×1 convolution kernel.

The up-sampling layer is a critical hidden layer in an FCN, and it serves as the basis for end-to-end training. The up-sampling layer is essentially a special convolutional

layer controlled by three parameters, namely, the size of the convolution kernel, the stride and the type of the initial weighted values. Among them, the stride size corresponds to the number and scale of previous down-sampling operations. Suppose that an image size is reduced by a factor of two in each pooling layer (under the premise that a convolution layer causes no change in the image size) and that this process is repeated five times in total (Figure 4.2). Then, the stride size of the up-sampling layer will be 2^5 . The size of the convolution kernel will be double the step size. The initial weighted values of the convolution kernel are bilinear interpolation, and they can be either trained or not. Finally, the up-sampled results are cropped to match the size of the ground truth. The output of each layer in the decoder is illustrated below.

For multi-category semantic segmentation, an FCN needs to generate multiple binary segmentation results corresponding to each class. Suppose that the entire database contains C types of foreground objects. First, the ground truth is converted into C binary images, where each binary image corresponds to only one category. Then, C segmentation results are generated, which are normalized via the softmax function, and then the loss is calculated using the cross-entropy loss function. This operation is similar to multi-category classification.

The original FCN architecture inducted three sub-architectures, namely, FCN32s, FCN16s and FCN8s (Figure 4.2). In all three, VGG16 was used as an encoder. The difference in the sub-architecture is that the sizes of the skip structures are different. FCN-8s performs classification after FC7, pool4 and pool3 and generates a corresponding segmentation result for each case. VGG16 contains a total of 5 down-sampling layers, and each output is reduced by a factor of 2. Therefore, the results of the last two down-sampling layers are required to be up-sampled and then merged with the result of the pool3 classifier to obtain the final segmentation result. Since the output of pool3 is only $1/8^{\text{th}}$ the size of the original image, the fusion result needs to be enlarged a factor of 8; this is the meaning of the notation 8s in FCN-8s. In FCN-16s classifiers are included only after pool4 and FC7, and their outputs are fused. The output of pool4 is $1/16^{\text{th}}$ the size of the original image, so the segmentation result

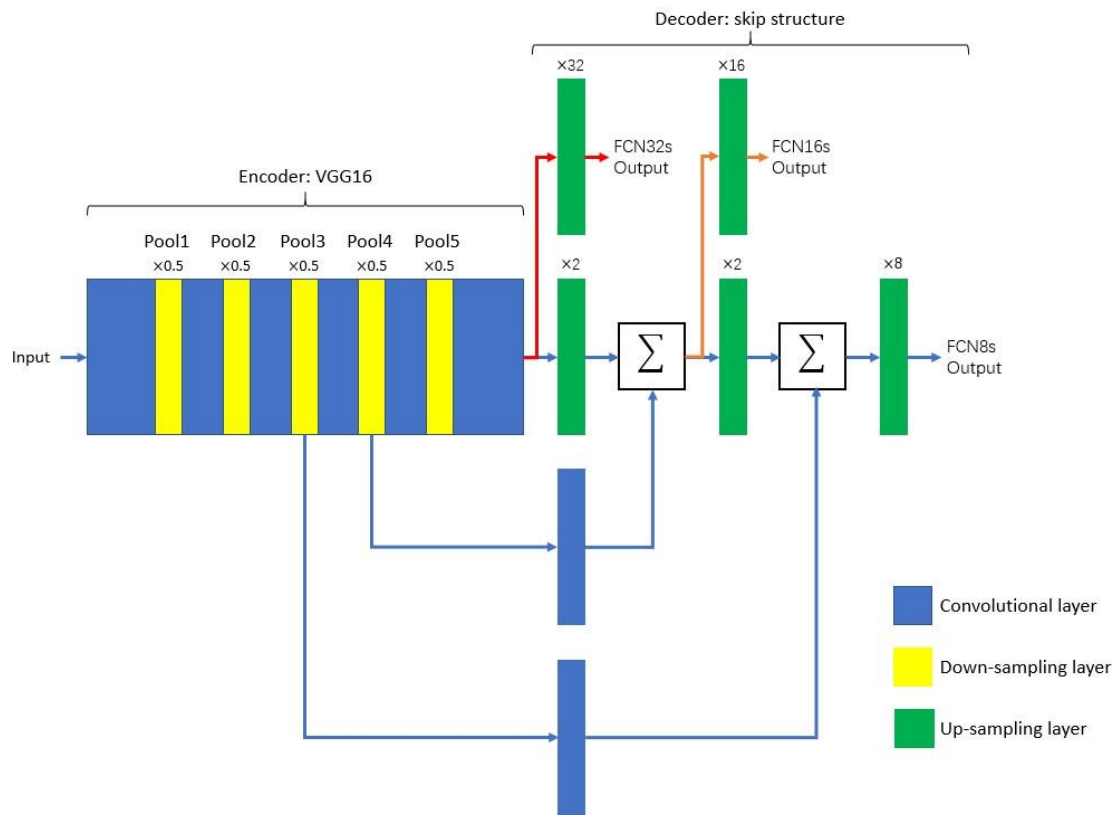


Figure 4. 2 The structure of FCN8s, FCN16s and FCN32s.

needs to be enlarged by a factor of 16. FCN-32s uses only the output of FC7 as the segmentation result.

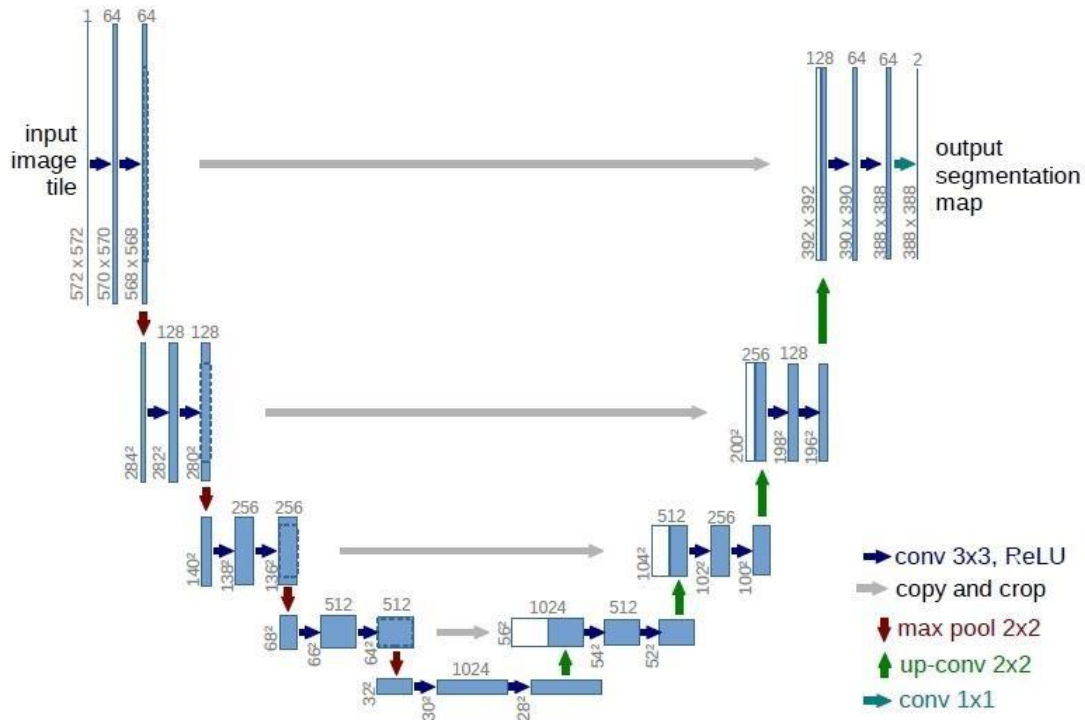


Figure 4. 3 The structure of Unet [35].

4.4 Unet

Unet [35] is an end-to-end trained semantic image segmentation network based on an FCN. In this network, the encoder and decoder have similar architecture, with a difference that the down-sampling layer in the encoder is replaced by an up-sampling layer in the decoder (Figure 4.3). In addition, the pixel classifier is shifted to the last layer for pixel-by-pixel categorization of the full-size feature maps.

The reason for adopting this structure in the U-Net is to preserve as much detail as possible in the segmentation results. First, the skip structure in an FCN8s can extract some additional details, but the fusion of different resolution segmentation results also add noise. For example, the edges of the foreground may become blurred, or the gaps between multiple foreground regions may be lost, causing them to merge rather than maintaining their respective shapes. Classifying the full-size feature maps will undoubtedly allow these details to be identified more accurately. Second, the up-sampled low-level feature maps and high-level feature maps are fused by convolution

operation. This fusion enables the extraction of more detail. Because of these advantages, U-Nets are often used for tasks requiring high levels of detail and shape accuracy, such as the segmentation of cells and neural structures.

4.5 Deep segmentation networks architecture

Since the introductions of FCNs and U-Nets, many new end-to-end trained architectures have been proposed. In this section, the most important FCN architectures are described.

4.5.1 DeepLab

DeepLab is a segmentation method that focuses on the atrous convolution. To date, four versions of DeepLab have been released. Among them, DeepLab V1 and DeepLab V2 presented in the same reference [68], they were developed based on FCN-8s. The authors chose to set the stride of the last two down-sampling layers to 1 to retain more detail in the encoder. In this way, the encoder can ultimately output a larger feature map.

One of the important concepts in design of CNN is the receptive field, which refers to the size of the area in the input image to which each unit in the output layer corresponds. A larger receptive field allows the output to contain more global features, what helps to improve the accuracy of the segmentation results. However, reducing the stride for down-sampling will make the receptive field smaller. In the example shown in Figure 4.4, when the pooling stride is 2 (Figure 4.4 (a)), the receptive field is 6 (a single output unit is connected to 6 input units). When the pooling stride becomes 1 (Figure 4.4 (b)), the output size is increased to 7, but only 4 input units are connected to a single output unit. In this case, there is no doubt that the output size is improved, but the amount information contained in each output unit is reduced.

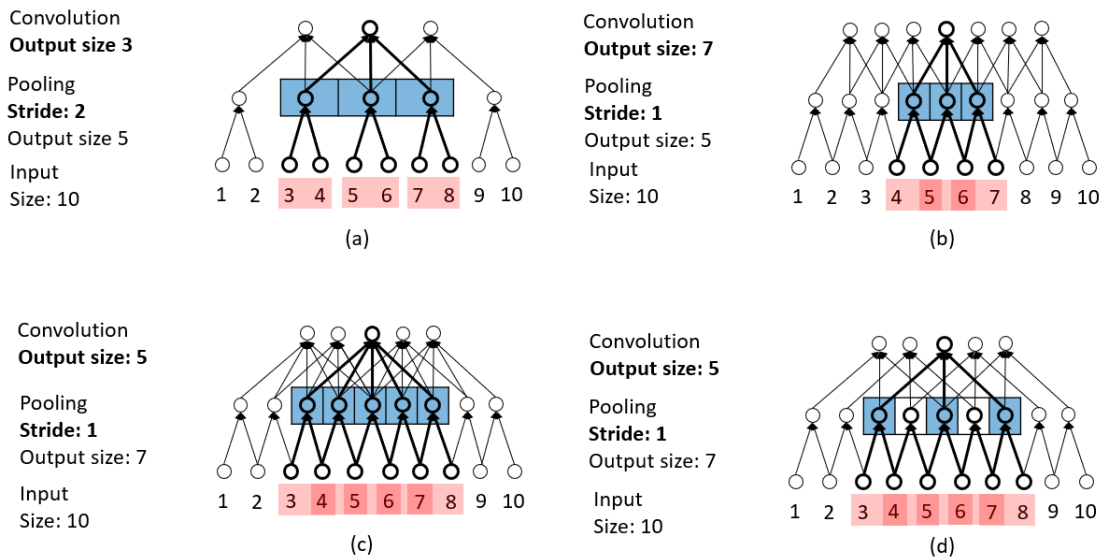


Figure 4. 4 Regular convolution (a), (b), (c) and atrous convolution (d). (a) Regular convolution, with pooling stride 2 and 1×3 kernel. (b) Regular convolution, with pooling stride 1 and 1×3 kernel. (c) Regular convolution, with pooling stride 1 and 1×5 kernel. (d) Atrous convolution, with pooling stride 1, 1×5 kernel and dilation 2; kernel size is 5 but only 3 weights are trainable.

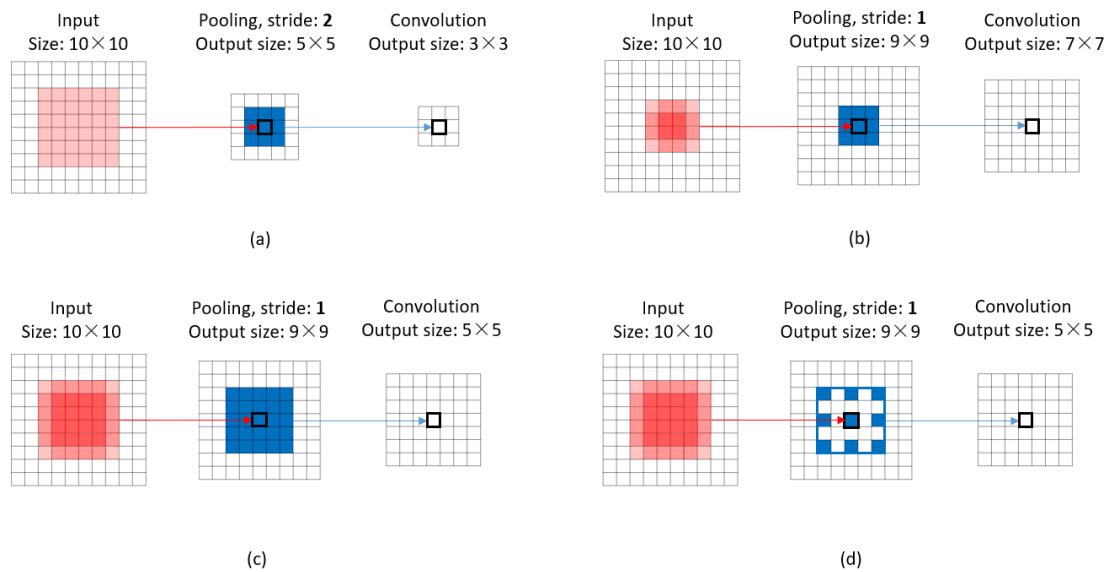


Figure 4. 5 Representation of 2d convolution layer, with regular convolution (a), (b), (c) and atrous convolution (d). Annotation follows the one introduced in Figure 4.4.

Using a larger convolution kernel can solve this problem, but it will increase the amount of computational cost and the number of parameters to be estimated, as shown in Figure 4.4 (c). To solve this problem more efficiently, DeepLab adds a convolutional layer built based on atrous convolution. Atrous convolution is also known as dilated convolution [69]. The underlying idea is to increase the size of the convolution kernel by adding 0s between the weights without changing the number of weights, as shown in Figure 4.4 (d). The definition of the atrous convolution is given as:

$$y[i] = \sum_{k=1}^K x[i + r \times k]w[k] \quad 4.4$$

where $y[i]$ is the output, $x[i]$ is an 1-D input signal, $w[k]$ represents the weight in a kernel. The parameter r is called dilation and it controls the stride between between each weight in an atrous kernel. Figure 4.5 shows the processing of atrous kernel for 2D input.

DeepLab V2 [68] was developed based on the DeepLab V1. It includes an atrous spatial pyramid pooling (ASPP) module, which was proposed to extract features from different receptive fields (Figure 4.6). This module consists of four parallel paths, each of which consists of an atrous convolutional layer and two 1×1 convolutional layers, where the last 1×1 convolutional layer is a pixel classifier. The dilation rates of the atrous convolutional layer on the four paths are 6, 12, 18, and 24. The ASPP module sums the outputs of all pixel classifiers to obtain an initial segmentation result. Finally, this initial result is post-processed using the conditional random field approach, and the output is the final segmentation result. DeepLab's authors compared DeepLab V1 and V2, and their evaluation showed that the latter's performance was significantly higher than that of the former.

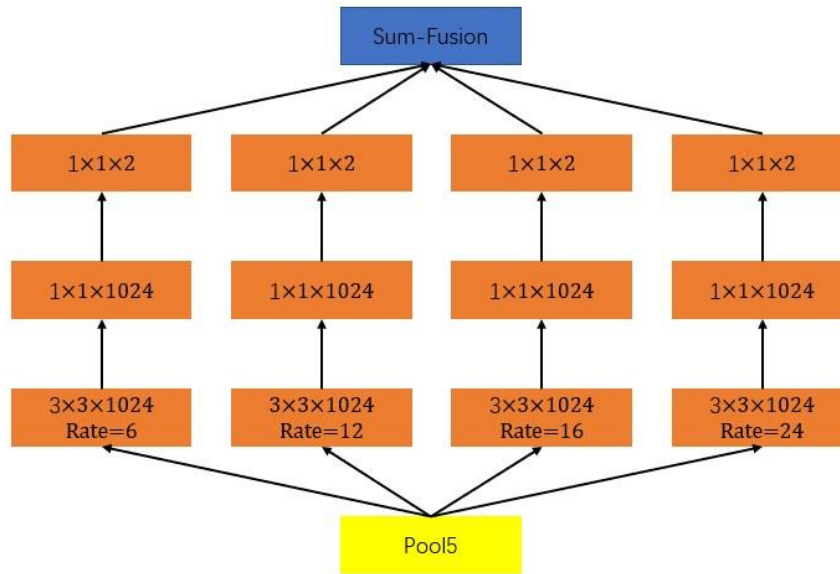


Figure 4. 6 The structure of ASPP module.

In DeepLab V3 [70], VGG16 (as used in the previous versions) is replaced with a ResNet structure. Both parallel structures in this network based on the atrous convolution and ResNet block have been investigated. The focus of the former investigation was to determine how to improve the depth of the network while retaining image details. The latter concerned the further development of the ASPP module. The new ASPP module has a total of five paths. It consists of three atrous convolutional layers, a 1×1 convolutional layer and a pooling layer, followed by another 1×1 convolutional layer to fuse the outputs.

DeepLab v3+ [71] is currently ranked number one on the PASCAL leaderboard (in cases where new training data are allowed). Based on DeepLab v3, this architecture includes an additional skip structure to fuse the segmentation results generated from the low-level features and the ASPP module. In addition, DeepLab v3+ learns features using a modified Xception block instead of a ResNet block. The test results show that this new structure demonstrates higher performance.

4.5.2 SegNet

The overall SegNet [72] structure is very similar to the Unet structure. The difference is that the SegNet architecture uses up-pooling layers instead of the up-sampling layers. Each up-pooling layer corresponds to a down-sampling layer in the encoder. Up-pooling is the same as down-sampling in backpropagation (Figure 3.11). If the down-sampling layer is a max pooling layer, then the corresponding up-pooling operation retains only the elements of the former that are non-zero in the reverse direction.

4.5.3 Global convolutional network

The authors [73] of this method found that a larger receptive field helps to improve the accuracy of image segmentation. Considering this, they designed a larger convolution kernel than that of atrous convolution and used it as the basis of their proposed global convolutional network (GCN). This convolution kernel has a composite structure with two paths, each of which consists of a single-column convolution and a single-row convolution, as shown in Figure 4.7. This approach has an advantage of increasing the receptive field without significant increase the number of kernel parameters.

Finally, the output results of the two paths are summed as the outputs of the module. ResNet-based feature learning [60] is adopted in this method. A skip structure is used in the decoder to fuse the features and output the results. There are 4 paths in the decoder, and global convolution is used as the pixel classifier on each path. By testing GCNs of different sizes, the authors concluded that a larger size could contribute to achieving more accurate segmentation results. However, the disadvantage of this method is that it still requires considerable memory.

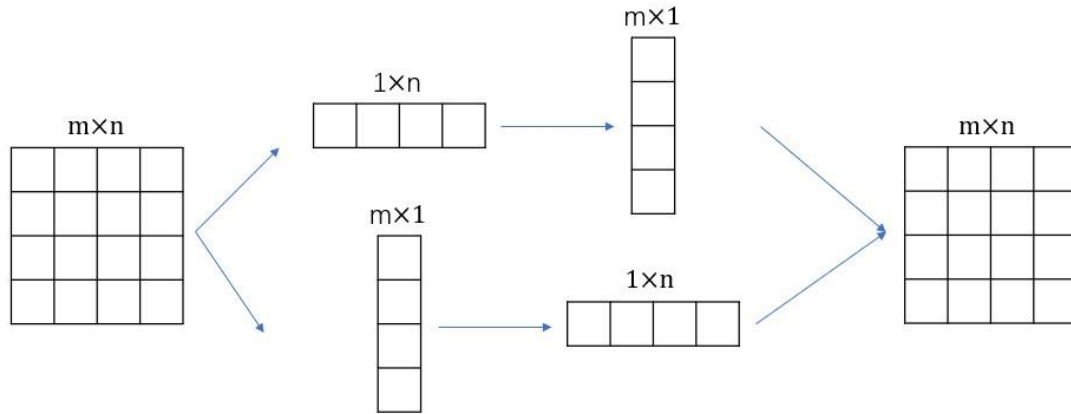


Figure 4. 7 The operation of global convolution.

4.5.4 Pyramid scene parsing network

The authors of the pyramid scene parsing network architecture [74] proposed a pyramid pooling module for learning multi-level features. This module was added after a feature-learning CNN. The module has N paths, each consisting of one down-sampling layer and $M/N \times 1$ convolution kernel. Here, M represents the number of original inputs to the module. These down-sampling layers can extract features from different receptive fields. Then, a 1×1 convolution kernel is applied for dimensionality reduction, and the features after dimensionality reduction are up-sampled to make their size uniform. Finally, these features are stacked with the original input to the module and then processed by the pixel classifier. This module is very similar to the ASPP module.

The authors tested the segmentation effects of pyramid pooling modules created based on max and average pooling. Their test results show that the latter is slightly better than the former. In addition, the test showed that dimensionality reduction also improves segmentation.

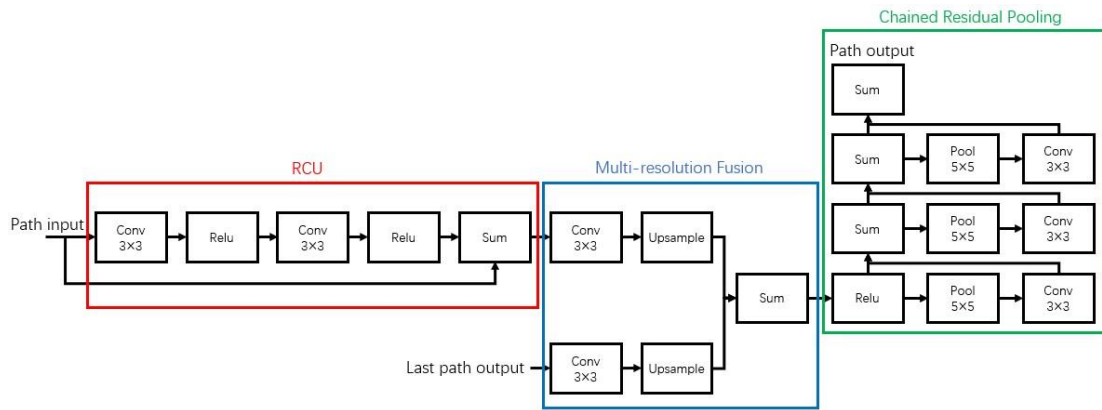


Figure 4. 8 The structure of RefineNet.

4.5.5 RefineNet

The overall structure of the RefineNet [75] is very similar to the GCN structure, but its skip structure is very complicated. Each path consists of a module called a RefineNet. Each module consists of three sub-modules connected in a sequence. The RCU is a conventional ResNet block with two sequential convolutional layers. The multi-resolution fusion module can be viewed as an extension of the fusion layer of an FCN to fuse features of two different resolutions. The chained residual pooling module is similar to a ResNet block but consists of pooling and convolutional layers. The structure of each path is shown below:

4.5.6 Deep contour-aware networks

To solve the problem of regions of the same foreground type merging together, a deep contour-aware network (DCAN) uses two decoders to segment the foreground and the contours of the foreground individually [76]. Then, it fuses the two sets of results to separate different foreground regions. Post-processing is required to remove some holes and small areas in the results. The ground truth of the contours can be obtained by inflating the foreground in the original ground truth. This method earned first place in the 2015 gland segmentation challenge [77].

4.5.7 Discriminative feature network

Discriminative Feature Network (DFN) proposed inter-class indistinction and intra-class inconsistency [78], the former means that the class of each pixel in the same segmented foreground should be same, and there should be no holes or pixels erroneously identified as other classes. The latter refers to the difference between the different class foreground should be increased. The above two ideas can be interpreted as how to reduce false positives and false negatives. DFN is designed as a composite structure consisting of three sub-networks: ResNet targeting at learning features, the smooth network for reducing false positives, and the border network aiming at increasing inter-class variation on the basis of contour segmentation. The overall structure of latter two is similar to skip structure, but the structure of their path is different.

The output of ResNet is input to the smooth network after global pooling. Smooth network consists of channel attention block (CAB) and Refinement residual block (RRB). CAB is used to fuse two feature maps with different resolution. Its structure is similar to the squeeze and excitation module, which is first to perform global pooling, and then use sigmoid to set a weight of 0 to 1 for each channel to suppress low-value features. RRB is a regular ResNet block. The Border Network consists only of RRBs and also receives features for each different resolution, but the way to fuse the features is addition. In addition, the smooth network uses the original ground truth, the loss function is cross-entropy, and the border network is the contour ground truth, which uses the focal loss [79].

In this method, Intersection over Union (IoU) is adopted to evaluate the segmentation results. The evaluation results and segmented images show the smooth network indeed reduces false positives in the segmented foreground and raises the averaged IOU 6.54%. However, the result is only 0.13% higher with border network, what hardly verifies that the module is essential. The idea of DFN and DCAN is similar, but the purpose is different. DFN is to reduce false negative, DCAN is to split the touched foreground.

It is possible that DFN is more suitable for use in tasks handled by DCAN. First, both can detect contour. Secondly, the result of DCAN mentioned above could return holes and small noisy objects, and the smooth network in DFN is very good at solving these two problems.

4.6 Summary

This section reviews the main traditional and recent learning image segmentation algorithms. It is shown that deep learning algorithms are become very popular in recent years, with continuously improving performance. However, this popularity does not mean that methods based on handcrafted features have been completely replaced. Therefore, the implementation of some deep learning methods is, in fact, inspired by handcrafted algorithms. For example, the FCN can be regarded as fusing the sliding window into the CNN and the DCAN extended edge detection into image segmentation.

Furthermore, the current research direction of deep learning methods can be divided into multi-scale feature extraction and multi-level features fusion. The former aims to improve the accuracy and integrity of end-to-end trained segmentation network, while the latter focuses on how to restore the details. Both research directions are considered in image segmentation within this thesis.

Additionally, IoU is used as a measure for evaluating segmentation results by many deep learning methods. However, there are currently several other evaluation methods that can be used to evaluate segmentation results, such as shape similarity and missing objects. When more measures are considered, it is difficult to say whether the afore-mentioned segmentation methods can guarantee their current performance. The DCAN evaluates its results using a variety of measures provided in the gland segmentation challenge [77], making it easier for other readers to understand the performance of their method. A similar branch-work approach has been adopted in this thesis.

Chapter 5. Proposed polyp deep segmentation methods

5.1 Introduction

The types of tasks and data used should always be considered before designing an image segmentation method, because the starting point for a natural image and a specific type of image are different. The former requires a segmentation method that treats different foregrounds equally and that guarantees stable performance across different datasets. The latter does not need to be applicable to all classes of data. Therefore, the unique properties of the foreground can be considered, and then a corresponding method is designed to identify such properties to solve specific problems.

Since polyp image segmentation is a specific task and as the contents of an image is relatively well defined, the latter idea has been adopted to design the polyp segmentation CNN, specific to segmentation of polyps in colonoscopy images. This chapter consists of five main sections. The first section introduces the polyp database. The second section examines properties of colonoscopy images. The third section describes the method designed for normalization of image borders. Following section

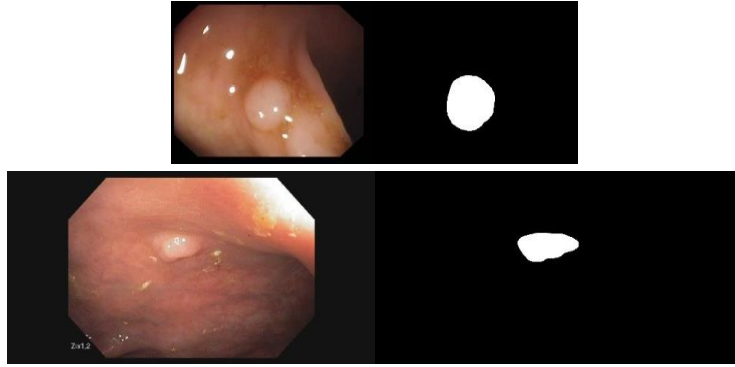


Figure 5. 1 An example of typical SD (top) and HD (bottom) training images and their corresponding ground truth.

describes different data augmentation methods adopted to increase the number of training images. Finally, the two novel deep network segmentation architectures, Dilated ResFCN and SE-Unet, are described together with the implemented test-time augmentation approach.

5.2 Polyp database

The polyp database, used in the reported research, was obtained from the GIANA polyp segmentation challenges⁸ which were organized as part of the 20th and 21st Medical Image Computing and Computer Assisted Intervention (MICCAI) conferences⁹. That polyp database consists of Standard Definition (SD) and High Definition (HD) endoscopy images (Figure 5.1). The SD database has two datasets: CVC-ColonDB (Figure 5.2) and CVC-ClinicDB (Figure 5.3). The first set (CVC-ColonDB) is used for training and consists of 300 low resolution, 500-by-574 pixels RGB images, which are accompanied by the corresponding ground truth segmented polyp binary images. The ground truth is composed of hand annotated/segmented polyps, with the annotation approved by trained colorectal endoscopists.

The second (CVC-ClinicDB) set has 612 RGB images each 288-by-384 pixels in size. That dataset does not include the corresponding ground truth segmentation and therefore is only used for testing. The images in the SD database are extracted from

⁸ <https://giana.grand-challenge.org/> [Assessed 20 Oct. 2019]

⁹ <https://www.miccai2018.org/en/> [Assessed 20 Oct. 2019]

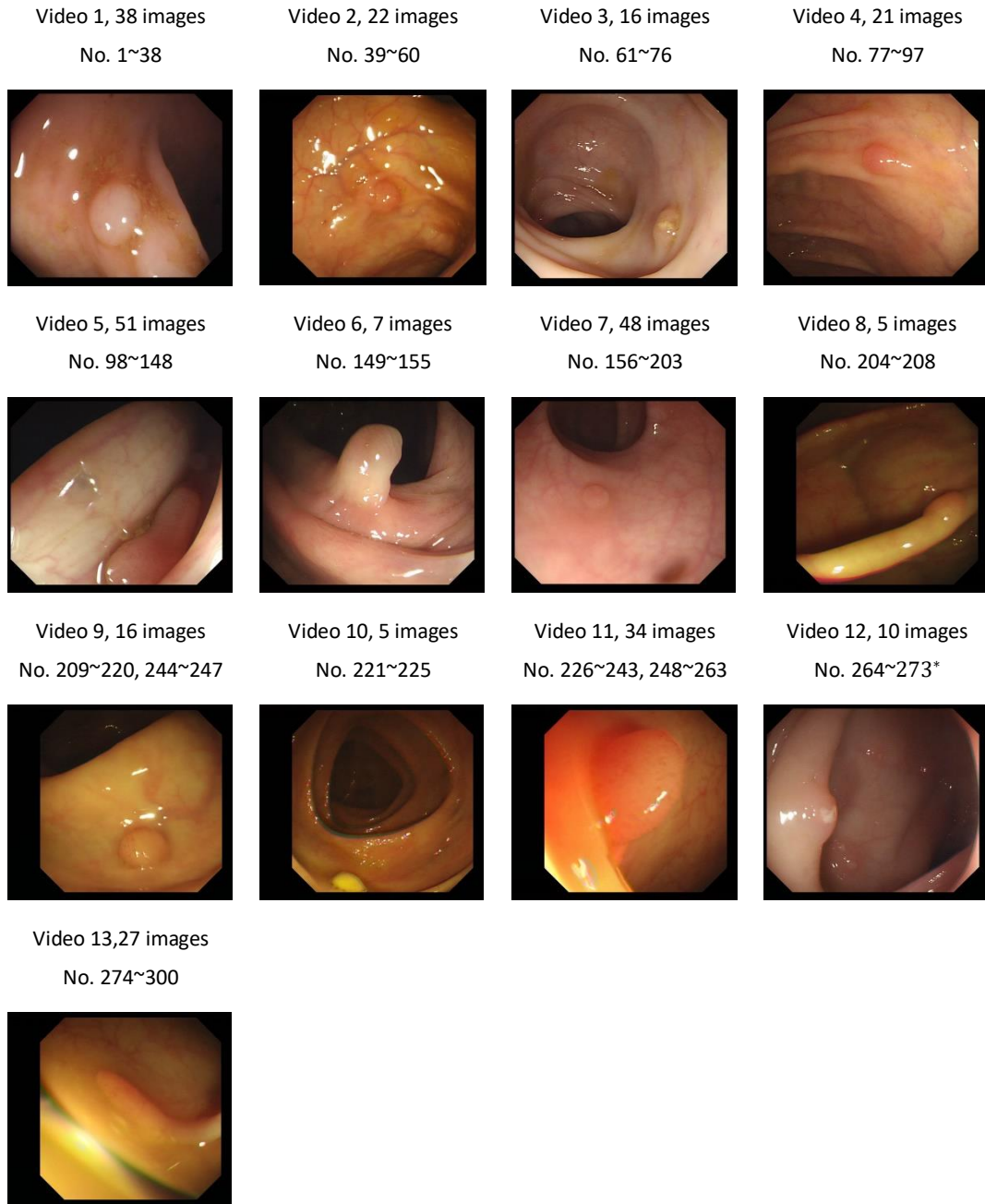


Figure 5. 2 A sample of images from the SD training dataset.

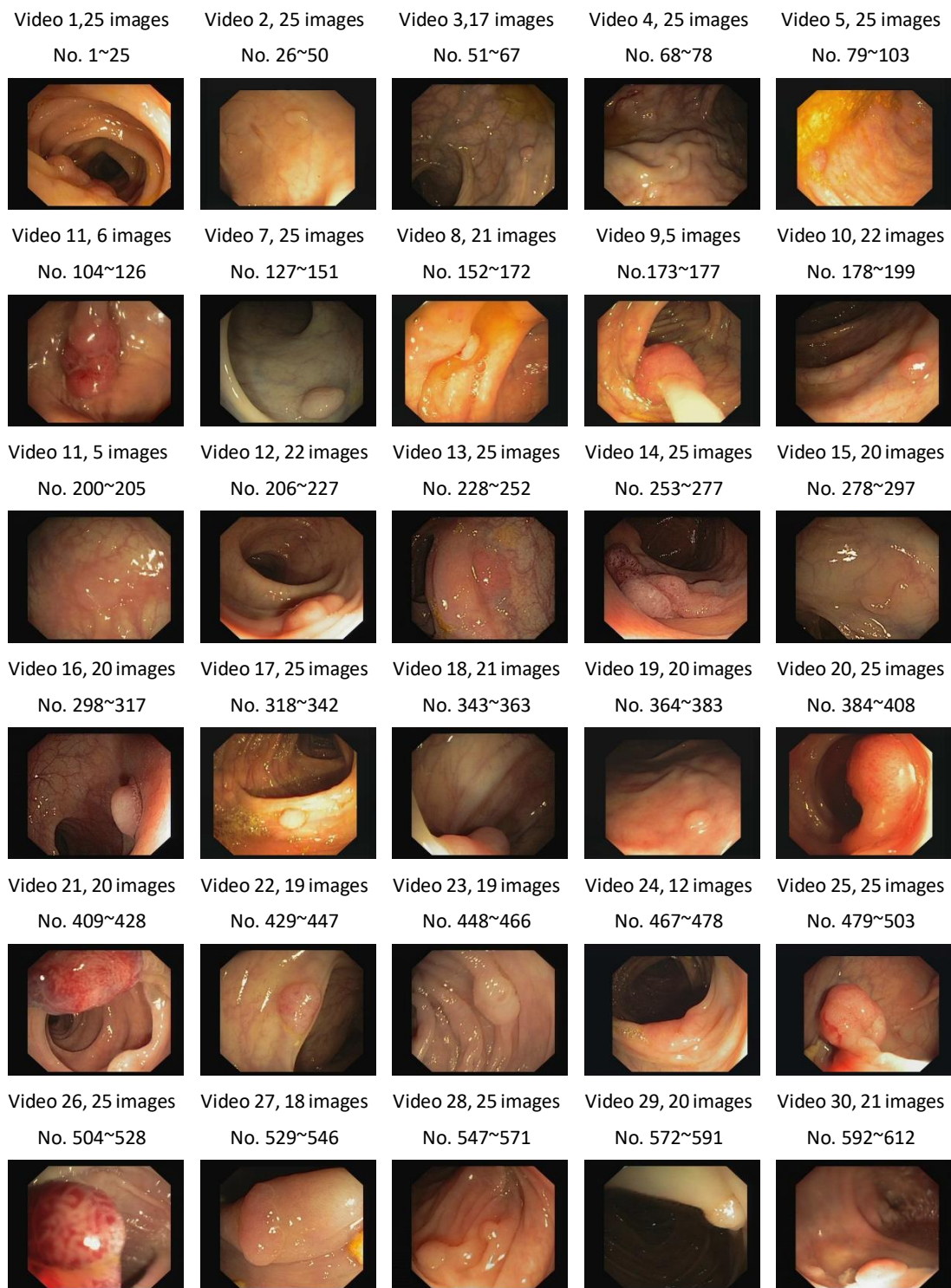


Figure 5. 3 A sample from the SD testing database.

a small number of colonoscopy videos. This method means that some of the images are highly correlated (show the same colon segment) if selected from the same video.

The HD database is composed of independent high-resolution RGB images of 1080-by-1920 pixels. The HD database includes 56 training images (with corresponding ground truth) and 108 images used for testing. All SD and HD images are framed by a black border, with the border being at a fixed position for the SD images extracted from the same video. The number of HD images is significantly smaller than that of SD images. The same video (in case of the SD images) was not used for selection of the training and testing image subsets.

5.3 Image analysis

Image analysis involves investigating properties of a polyp image that may affect the preformation of the segmentation method. This section covers two types of analysis: appearance and size analysis. The former measures image complexity while the latter measures the objective size.

Appearance analyses

In this section, the colour is analysed by K-means clustering to investigate the difference between the colour of the polyp and its surroundings or any hidden components that may affect segmentation.

Clustering involves creating a set of similar samples. In polyp segmentation, details on the polyp appearance of a polyp are represented by different colours so that colour clustering helps illustrate main components of the polyp. Moreover, clustering can also reveal similarities between polyp and its surroundings. When there is certain relationship between them, then the CNN should learn it by correcting the corresponding structure. While clustering can be performed using many methods, this section adopts k-means for two reasons.

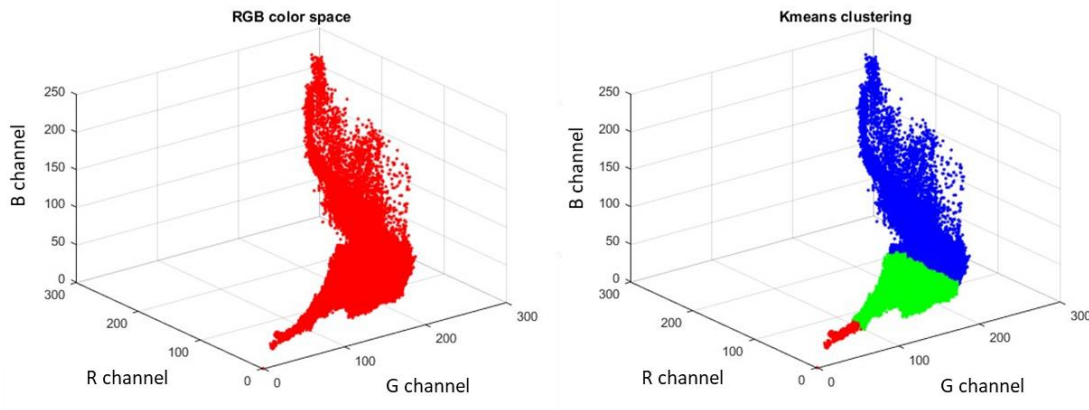


Figure 5. 4 The clustering result of Image No. 251.

1. Easy of use. In this thesis, colour clustering is not used for segmentation, as it is sufficient to determine the relationships of within-class or inter-class pixels. K-means only need to set the number of cluster centres. While other clustering methods are superior to K-means, they use more accurate hyper-parameters, which are difficult to secure.
2. Applicability. As the use of more hyper-parameters renders a method more sensitive, the method may sometimes only generate good results for specific data. The K-means approach is a relatively stable method because it only considers the distances of all samples. While the presence of different initial centres can cause results to vary, this problem can be avoided by clustering more images and identifying their similarities.

In the reported here experiment, the K-means method is used with three clusters. Figure 5.4 shows the K-means clustering result for image No. 251 in the RGB colour space. The red points in left figure represent all the pixels in the image No.251 before they are assigned to clusters. In the right figure, each image pixel has been painted in one of the three (red, green or blue) colours, to indicate to which of the three clusters it has been assigned by the K-means algorithm. Figure 5.5 and 5.6 shows the corresponding segmentation result for a sample of colonoscopy images. Appendix C shows the segmentation results obtained for different number of predefined clusters

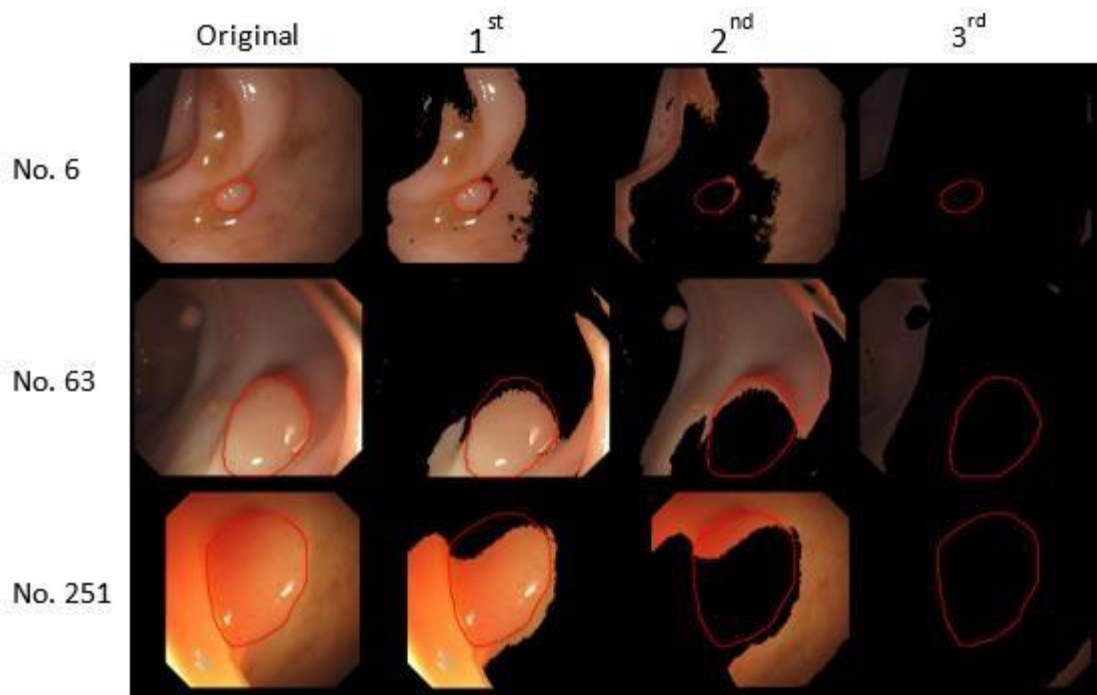


Figure 5. 5 Image No.6, 64 and 251 and their clustered results with three cluster centres in RGB colour space.

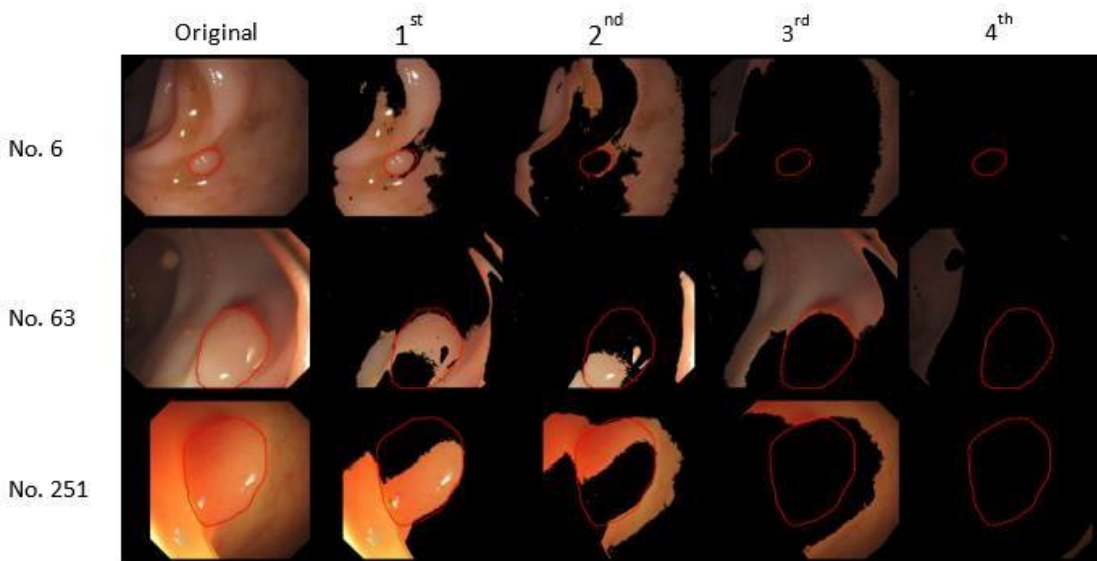


Figure 5. 6 Image No.6, 64 and 251 and their clustered results with four cluster centres in RGB colour space.

and pixels represented in different colour spaces. From these results two problems can be identified:

1. Intra-class difference: This problem denotes that pixels in the same foreground are different, which split the foreground into multiple regions. The colour is not fixed for the same polyp due to polyp morphology or uneven illumination. Taking No. 251 as an example, as the upper part of the polyp is darker than the lower part, this polyp is actually created from two parts. A similar problem is illustrated in No. 63. Based on three centres, the foreground of No. 63 has lost some contours. However, as the centre expands, the area with specular reflection is immediately removed from the foreground. This removal can lead to under-segmentation, which can result in hole formation, edge losses or even losses of targets.
2. Inter-class similarity. This problem corresponds to the previous one and occurs when pixels in the foreground and background are very similar, rendering it impossible to distinguish them by colour alone. All images shown in Figures 5.5 to 5.6 present severe levels of over-segmentation. Background and foreground pixels are even more similar than regions in the foreground. Even when the cluster centre is expanded, it can only continue to split the foreground and cannot remove the background.

These two issues can be further described as the same problem where each pixel presents an incorrect correlation with its neighbourhood. This problem may result when a polyp and background are similar in colour or when interference occurs during sampling.

Size Analyses

The size of a polyp is not fixed for each image several reasons. First, the size of a polyp changes in different stages. While polyps are initially very small, with time some

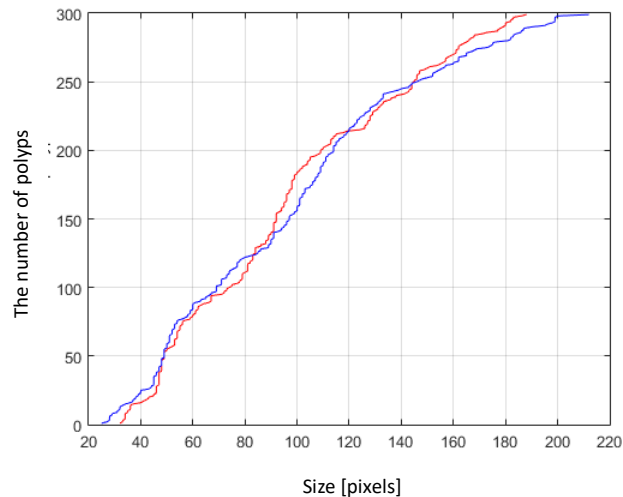


Figure 5. 7 The cumulative polyp size distribution of SD images for length (in blue) and for width (in red).

become malignant and increase to the point of occupying much of the surrounding area. Second, to further observe a polyp, an endoscope moves to positions close to the polyp.

Figure 5.7 shows cumulative histograms for number of SD image polyps as a function of their heights or widths. It can be seen that the polyps' heights and widths vary considerably. This large variation is a crucial issue for quality of image segmentation for both handcrafted and deep feature-based methods. For example, although the FCN can segment multi-size images, the scale invariance is not built into the network. Moreover, due to effects of colour and uneven illumination, the segmentation of a large polyp could lead to partial segmentation.

Furthermore, a presence of small polyps is a main cause for unbalanced data, i.e. the number of pixels representing polyps is significantly lower than number of pixels representing the background. The CNN is more inclined to select the class for which more training data is available. As polyps normally include far fewer pixels than the corresponding background, this causes CNN to learn features from the background rather than from a polyp.

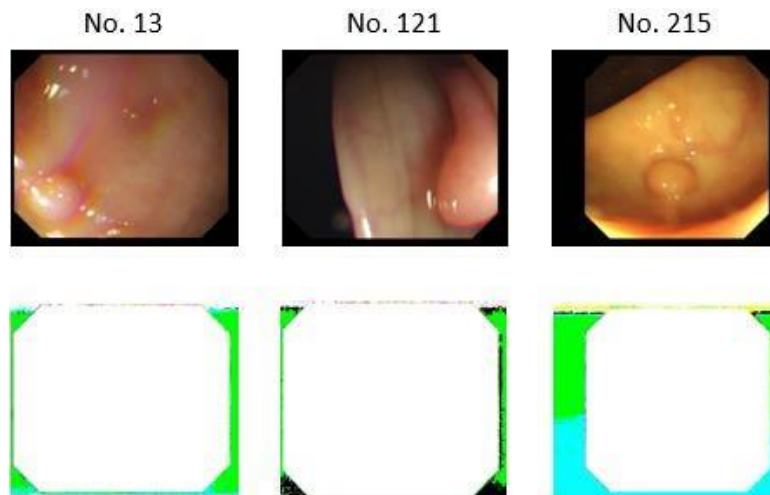


Figure 5. 8 Colonoscopy image with marked border. The values of different pixels are marked by different colour.

5.4 Pre-processing

CNN can learn relevant features automatically during the training process, however, on some occasions the method could learn from a spurious feature. This error may be because images can learn information which does not correspond to the problem needing to be solved. For example, in the colonoscopy images used in this work, there is a variable border region which does not carry any information about presence or characteristic of polyp (the appearance of polyp is independent of the appearance of the border). Nevertheless, the network can extract features corresponds to the border regions and try to use this in the polyp segmentation process.

Section 5.2 shows that all polyp images have a black border. However, pixel values on these borders are not all equal to 0 (Figure 5.8). For SD images, pixel values on the border are not the same.

To avoid the CNN learning from noisy regions, it is necessary to normalize the border by assigning value 0 to all pixels identified to be belonging to the border. Since the borders of images collected from the same video are fixed, standard deviations can be used to solve this problem (shown in Figure 5.9). First, all images are converted to the grey-scale and are combined into a 4-D tensor. Next, the standard deviation is calculated for each pixel location in the image. Since valid content in video images often changes (e.g. due to camera motion), its standard deviation will be large.

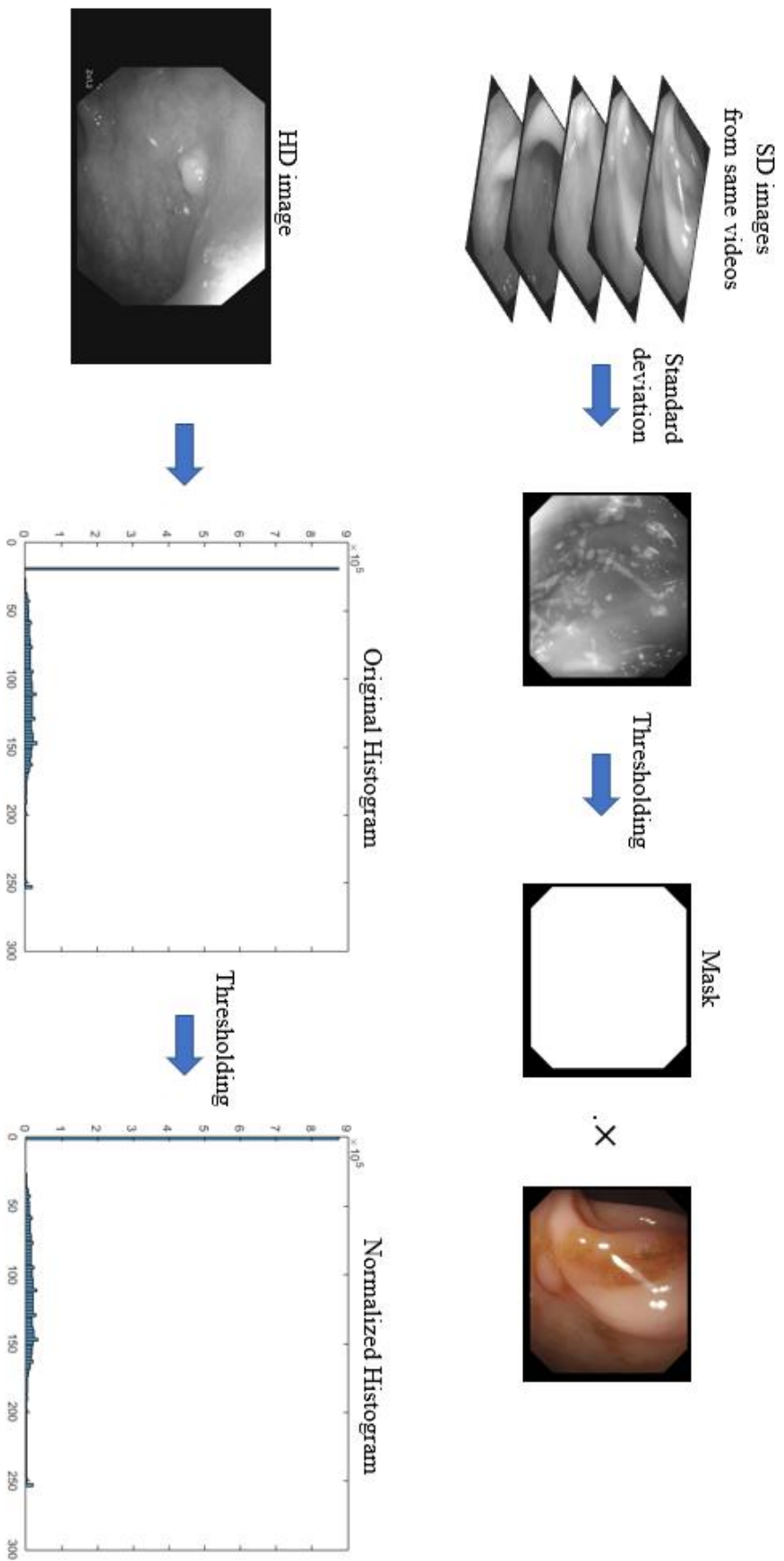


Figure 5. 9 Processing pipeline for border removal from SD and HD images.

However, the changes for the border pixels are small and therefore the standard deviation is small. This standard deviation map can be used to identify the pixels belonging to the border.

In the HD image, all pixels in the border for the given image have the same small value. There is similar error in HD images. The border of each image is a fixed value. The image to be measured can be converted to a grayscale image, and then the histogram is used to determine the threshold to normalize the border (Figure 5.9).

It should be noted that when normalizing the border of an SD image, some polyps near the border can lose small number of pixels. This operation does not affect training, but slight under-segmentation can occur when the border of a test image is normalized. Therefore, this operation is only implemented for SD images are used for training. As the position of the HD image border is estimated very accurately, the training and test images will be processed.

5.5 Data augmentation

Data augmentation is designed to provide more polyp images for CNN training. Although this method cannot generate new types of polyps, it can further highlight the properties of polyps based on modelling different image acquisition conditions (e.g. illumination, camera position, colon deformations).

The performance of the CNN-based methods relies heavily on the size of training data used. The whole available database includes only 355 images. Clearly, it is very limited at least from the perspective of a typical training set used in a context of deep learning. Moreover, for some polyp types, there are less than 10 corresponding exemplar images in the database. Therefore, it is necessary to enlarge the training set via data augmentation.

Regarding polyp segmentation, this thesis combines SD and HD images as a new database to obtain more types of polyps and applies six methods for data augmentation, including rescaling, shifting, rotation, colour jittering and contrast jittering. These are used individually or together.

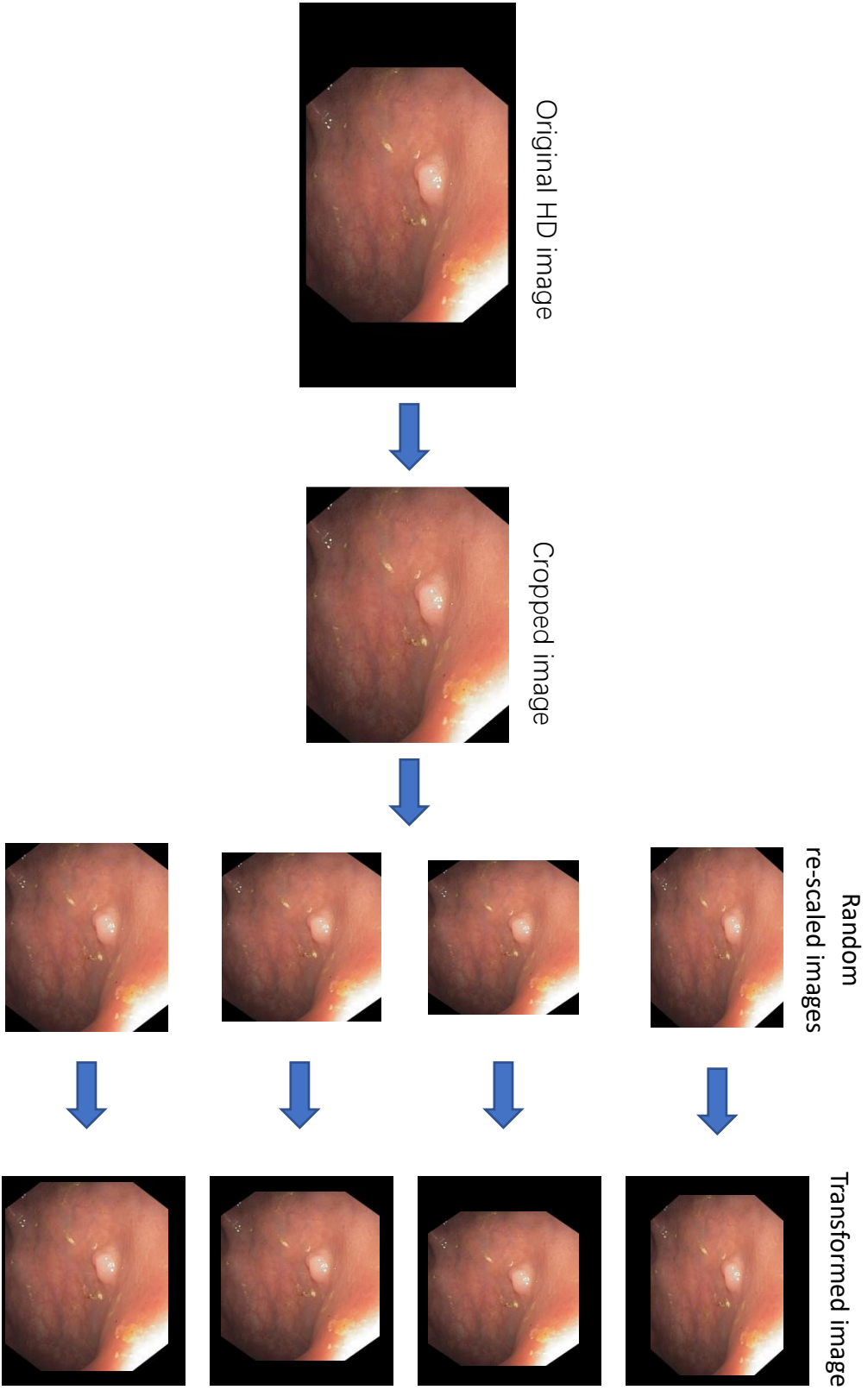


Figure 5. 10 The re-scaling processing of HD images.

Re-scaling

The deep network generates a large number of feature maps in each layer of the network. To reduce computation time, the current CNN framework saves the network parameters and feature maps into GPU-memory to avoid re-calculation. However, a higher-resolution image increases the size of a feature map, and thus, the network implementation can easily run out of the GPU-memory. The hardware used to support the reported research did not have enough memory to process the full resolution SD or HD images, therefore the images had to be reduced in size before they could be used on the available GPU hardware.

Whether an SD or HD image is involved, the image size is so large that CNN training requires the use of large memory and long duration for training. When a deeper network is necessary for polyp segmentation, it is difficult for regular a GPU-memory to support feature learning for a large image. Moreover, batch normalization must work for a large batch size, further increasing memory requirements.

First, HD images are transferred to the SD form by re-scaling and shifting. The whole processing method is shown in Figure 5.10. Since the border cannot provide any useful information, most of the border is removed before rescaling image. The cropped image is re-scaled to five random sizes, the height ranges from 400 to 480, and the width ranges from 400 to 554. The resulting image is slightly smaller than the original.

Subsequently, each single re-scaled image is embedded at random position into an all zero image of the same size as the normal SD image (500×574). Thus, the number of original HD images increase 5 times. In total, it includes 280 new images. This step actually combines re-scaling and shifting to generate a similar number of images present in the SD part of database. The corresponding ground truths are also processed in the same way. Next, all SD and augmented HD images and corresponding ground truths are re-scaled to 250×287. Because the original SD size is still very large, this new size decreases the amount of memory required to 75%. This rescaling allows the CNN framework to build a deeper network.

It should be noted that training and testing images differ in size, so the structure of the testing image changes when it is re-scaled to 250×287. As FCN can segment images of different sizes, the SD testing image are only used at their original size. This original size is similar to the re-scaled size of the training image.

Colour jitter

Colour jitter aims to generate more augmented images to train the learning-based algorithm. Without changing the content of an image, colour jittering can subtly change pixel values. Colour jittering model illuminations variance during acquisition of the original images. It also models differences in tissue pigmentation for different subjects.

Figure 5.2 shows polyps of different colours, but these colours are distributed within a relatively stable range of yellow, red and pink. The colour of an augmented polyp image should fall within the same range, as there are no new colours in the testing data. An incorrect colour may cause the CNN to learn the wrong patterns. Moreover, colour jittering should focus on the global image rather than on a single colour, as otherwise an image can be corrupted by uneven colours. Typical colour jittering involves four steps:

1. RGB image is transferred to the HSV colour space, which defines the colour by hue, saturation and value.
2. Determine the required range of these three variables. The colour in this range should be similar to the real colour the observed in the real colonoscopy images.
3. Select the random number between these ranges and then set these as parameters in the colour transformation equation.

$$\begin{aligned}H_{new} &= H + v3 \\S_{new} &= S^{v1} \times v2 + v3 \\V_{new} &= V^{v1} \times v2 + v3\end{aligned}$$

4. Transfer the HSV image to the RGB image as the final jittered image

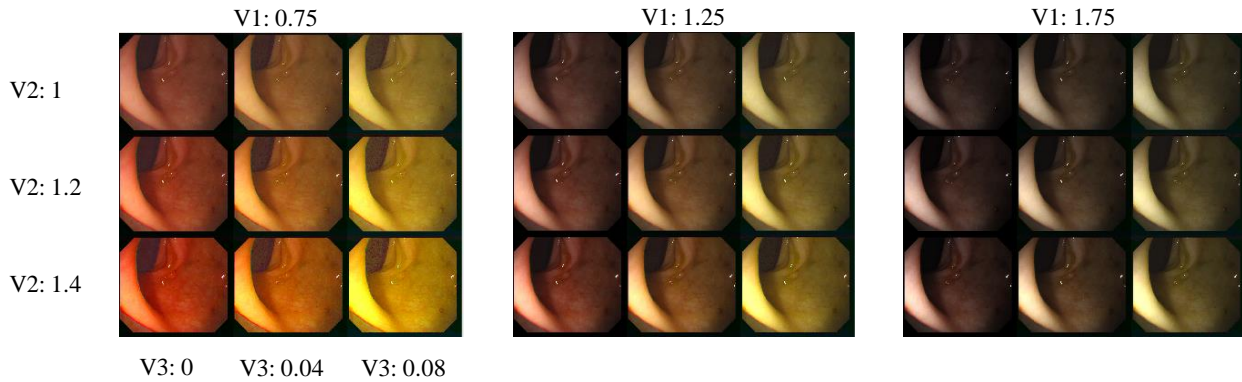


Figure 5.11 Selected examples of colour jittering experimental images.

Based on some experiments, these three ranges are selected as: v_1 (0.75-1.75), v_2 (1-1.4) and v_3 (0-0.08). Typical augmented images are shown in Figure 5.11.

An image is jittered in HSV space because this makes it easier to control the Hue and Saturation. In RGB, colour is defined by the proportion of red, green and blue present. The colour is changed by adding or subtracting value for each pixel in these three channels. However, when using this approach colour jittering, the transform of each pixel is independent. This transform destroys details and the consistency of an image, such as the gradients between neighbouring pixels. For HSV, each pixel is transferred along with a change in the hue, chroma and value of the global image. These can therefore remain as original details of the image.

Rotation

In this thesis, the angle of rotation is random to prevent learned features of CNN from depending upon a fixed angle or the positions of borders. First, the original image is padded to 100 zero pixels for the width and height to maintain the size of the image. Second, the image is rotated around its centre, and this processing achieves rotation together with shifting. Third, the regions falling outside of the original area are removed. The whole operation and a typical example are shown Figure 5.12.

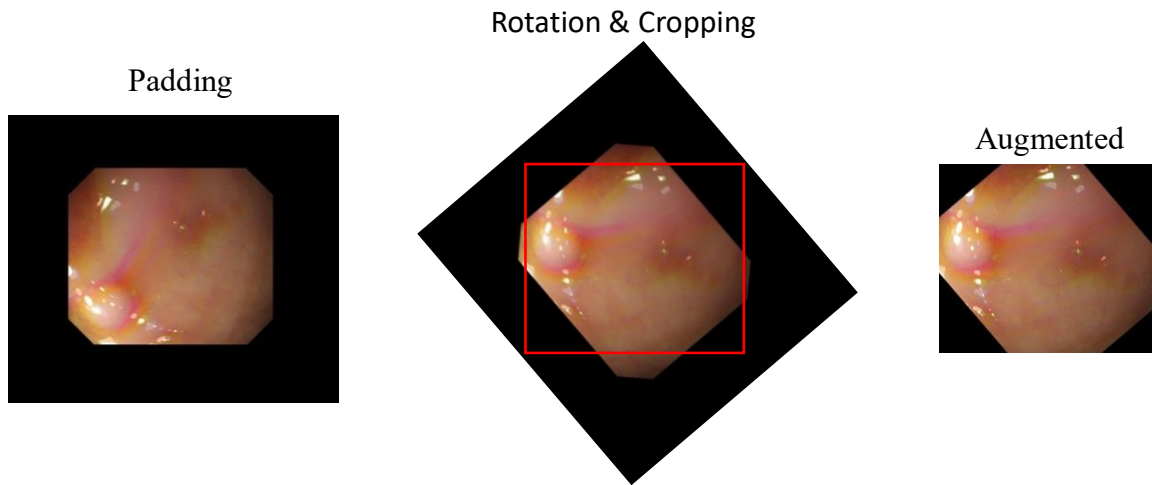


Figure 5. 12 The image rotation and corresponding augmented colonoscopy image.

Augmentation Implemented details

Components of the augmented training database are shown in the Table 5.1. The above data augmentation approach generates a total of 92640 new images. In particular, the base training data include 579 images (299 SD and 280 re-scaled HD images). Each image is rotated 50 times, is colour jittered with rotation 110 times. The hyper-parameters of the data augmentation method are random within the mentioned before range.

In validation, this augmented dataset is divided into four groups to perform the cross validation. More information on this point is given in Chapter 6 'Experiment and Results'. Subsequently, the whole dataset is used for segmentation of the test images. In addition, this thesis tests the segmentation results based on more different data augmentation methods in Chapter 6.

Table 5. 1 Number of augmented images using different augmentation method.

Training data	Rotation	Colour jitter & Rotation
SD data (299)	14950	32890
Re-scaled HD data (280)	14000	30800

5.6 Segmentation methods

5.6.1 FCN8s

Deep learning methods have generated many new end-to-end image segmentation networks. However, due to differences in data and settings, many such results are difficult to reproduce. Therefore, the earliest proposed FCN8s is still the most widely used. As mentioned above, FCN8s has been applied to some polyp segmentation tasks, and some have been compared with state-of-the-art handcrafted feature-based methods, showing that FCN8s segmentation offers performance advantages. Therefore, FCN8s can be used as a reference standard to test the performance of our proposed methods.

The original structure of FCN8s are described in section 4.3. The FCN8s version used here has some minor changes when compared with architecture described in Section 4.3. The output of pixel classifiers and up-sampling layers in the decoder are set to 2 and correspond to the foreground and background, respectively. With the original FCN8s, the two segmentation results are normalized by Softmax, and the residual is calculated by cross-entropy.

5.6.2 Proposed ResNet-FCN

In this section a deeper FCN is proposed and effects of the network depth on the quality of polyp segmentation is investigated. The feature extraction module of the FCN8s is replaced by a deeper CNN, the so-called ResNet50. In polyp segmentation, the ResNet architecture is used to build an FCN, based on three considerations:

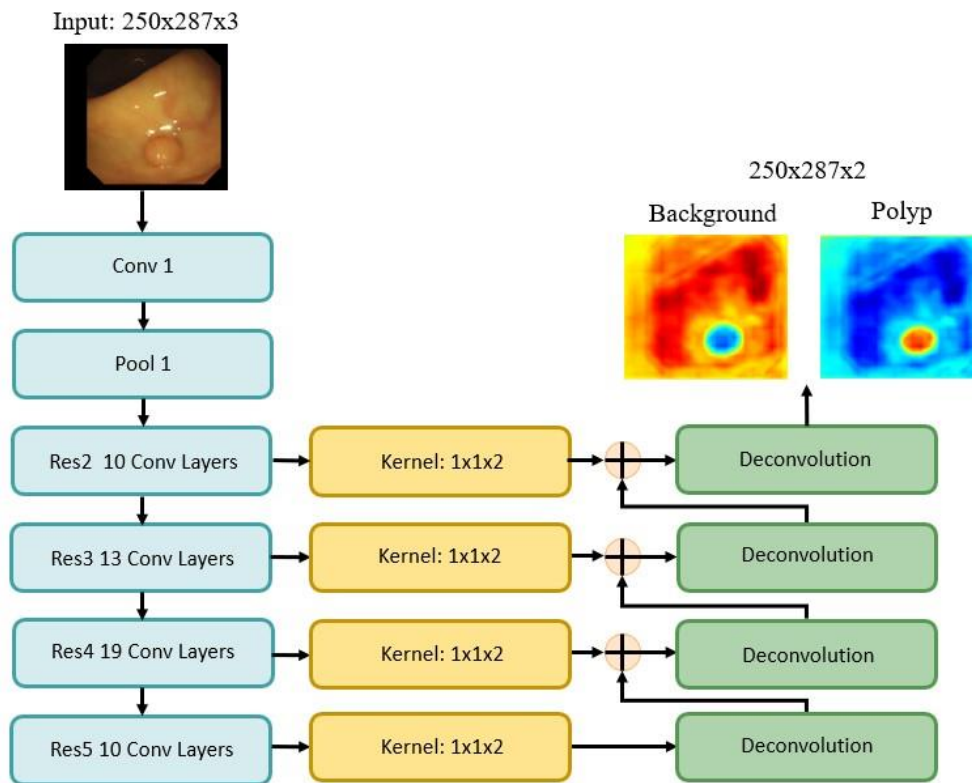


Figure 5. 13 The structure of the proposed network using ResFCN. Blue: Feature learning module. Brown: Pixel classifier. Green: Up-sampling.

1. Better feature learning ability. At present, the results of CNN development show that when there is no overfitting or gradient vanishing, the feature learning abilities of a deep network are stronger than those of a small network. As the appearance of a polyp is difficult to describe, increasing the network depth serves as the most straightforward means to improve FCN performance.
2. A deeper network has larger receptive field. A larger receptive field helps CNN to learn more global features as reflected in the discussion on Deeplab (section 4.5.1). The large receptive field is necessary for polyp segmentation, as the class of a pixel should be determined not only from the properties of its neighborhood but also from its surroundings or the context of the whole image. This process is similar to the fusion of multiple cells in the HOG feature.
3. Reasonable computational cost. Although ResNet is not the best performing CNN, it is the most widely used. This use is in a large part due to its relatively low computational cost effectively (compared to other CNNs). The deeper the network, the larger required memory source. When some new structures are added, a

network may only be able to run on high-performance hardware. ResFCN achieves a reasonable balance between gradient vanishing, performance and required resources.

ResNet is deeper than VGG16 and can reduce the effect of a vanishing gradient. Regarding hardware used for this thesis, the memory requirements for ResNet-101 and 152 are so large that they cannot reserve enough memory for other developments, rendering ResNet-50 the most suitable choice.

The architecture of ResNet FCN is shown in Figure 5.13. The last pooling layer of ResNet50 is removed, and the other parts can be divided into six sub components: Res1 – Res5. Res 0 represents the first convolutional and pooling layers. Res 2 – Res 5 represent the subnetwork with respectively 9, 12, 18, and 9 convolutional layers with 256, 512, 1024, and 2048 feature maps.

Each of these sub-networks operates on gradually spatially reduced feature maps downsampled with a stride of 2 when moving from sub-network Res_i to sub-network Res_{i+1}. The size of a corresponding feature map is 62×72, 31×36, 16×18, 8×9.

5.6.3 Proposed Dilated-ResFCN

This method involves developing a more advanced pixel classifier rather than single 1x1 convolutional kernel. The original idea is inspired by the architecture of DeepLab-LargeFOV, which aims to further limit effects of the problem discussed in the Image Analysis section. This section described one of the most important original contributions of this thesis.

Dilation convolution

Dilation convolution offers two benefits. First, it can provide more global information. DeepLab has shown that a large kernel is helpful for learning CNN-based semantic

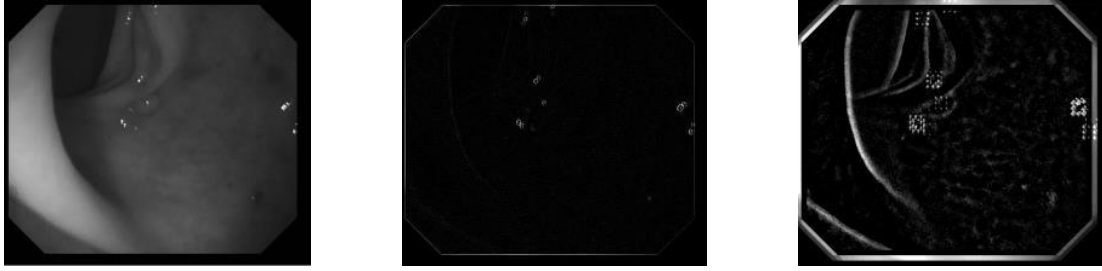


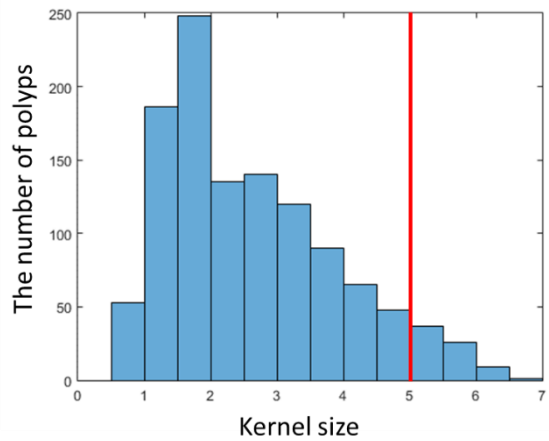
Figure 5. 14 The results of dilated Laplacian operator.

segmentation, as some properties of objectives depend on the surrounding area. A larger kernel can include more hidden patterns from the relationship between the polyp and background. Via K-means clustering (Figure 5.5 to 5.6) it can be observed that a polyp correlates with its surroundings, and this should be considered in CNN design.

Furthermore, it is helpful to generate more complete segmentation results on a polyp image. A small kernel is very sensitive to local features, which include significant and easy to learn patterns. For a large kernel, since global features change little, the responses of large kernel are more stable (Figure 5.14). Taking edge detection as an example, the detection of a regular Laplacian operator is not continuous, but that of dilated one (dilation rate: 2) is very stable.

Dilation Rate

First, a dilated convolutional kernel should be smaller than the current image. Otherwise, the weights of the kernel are wasted or reduced to a 1×1 kernel. This reduction narrows the range of dilation rates. Taking Res5 as an example, the resolution of the output is 8×9 . Figure 5.15 demonstrates different dilation rates and the histogram of polyp size. The table shows the 3×3 kernel different dilation rates. Although a 7×7 kernel can cover all polyps, polyps of between 5×5 and 7×7 are not too large, and this dilation rate too closely reflects the size of the current input, meaning that it cannot retain local details. A regular 3×3 kernel is too small. Therefore, a suitable dilation rate of 5×5 a good compromise (red line).



Dilation Rate	Kernel Size
1	3x3
2	5x5
3	7x7

Figure 5. 15 Number of polyps fully covered by an increasing kernel size.

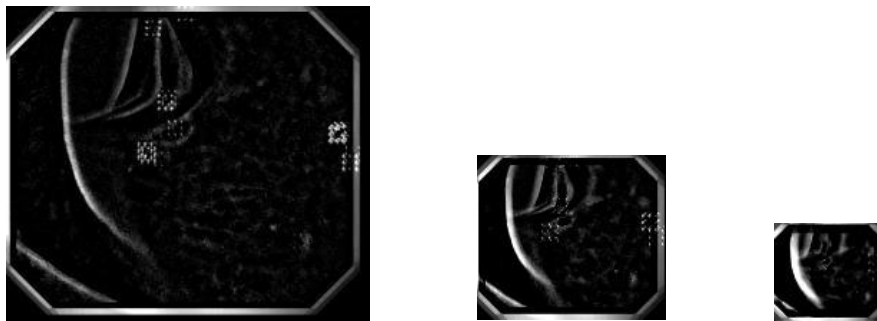


Figure 5. 16 The results of a dilated Laplace operator for images of different resolutions.

Since the resolution of internal features changes with depth, it generates varied types of features. Based on the properties of a CNN, a low-resolution feature map is more focused on high-level features, but it cannot retain enough details. In Figure 5.16, Laplace operator detection is altered by an image at a different resolution. It is always more heavily focused on the significant region and misses indistinct details.

To limit the effects of this problem, dilation convolution should be applied before all pixel classifiers. While excluding the regular connection of ResNet50-FCN, outputs from Res2 to Res5 in the proposed architecture are directed to a parallel classification path consisting of a dilation convolutional layer, 1x1 convolutional layer, dropout layer and final 1x1 convolutional layer with two outputs corresponding to the polyp and background confidence maps.

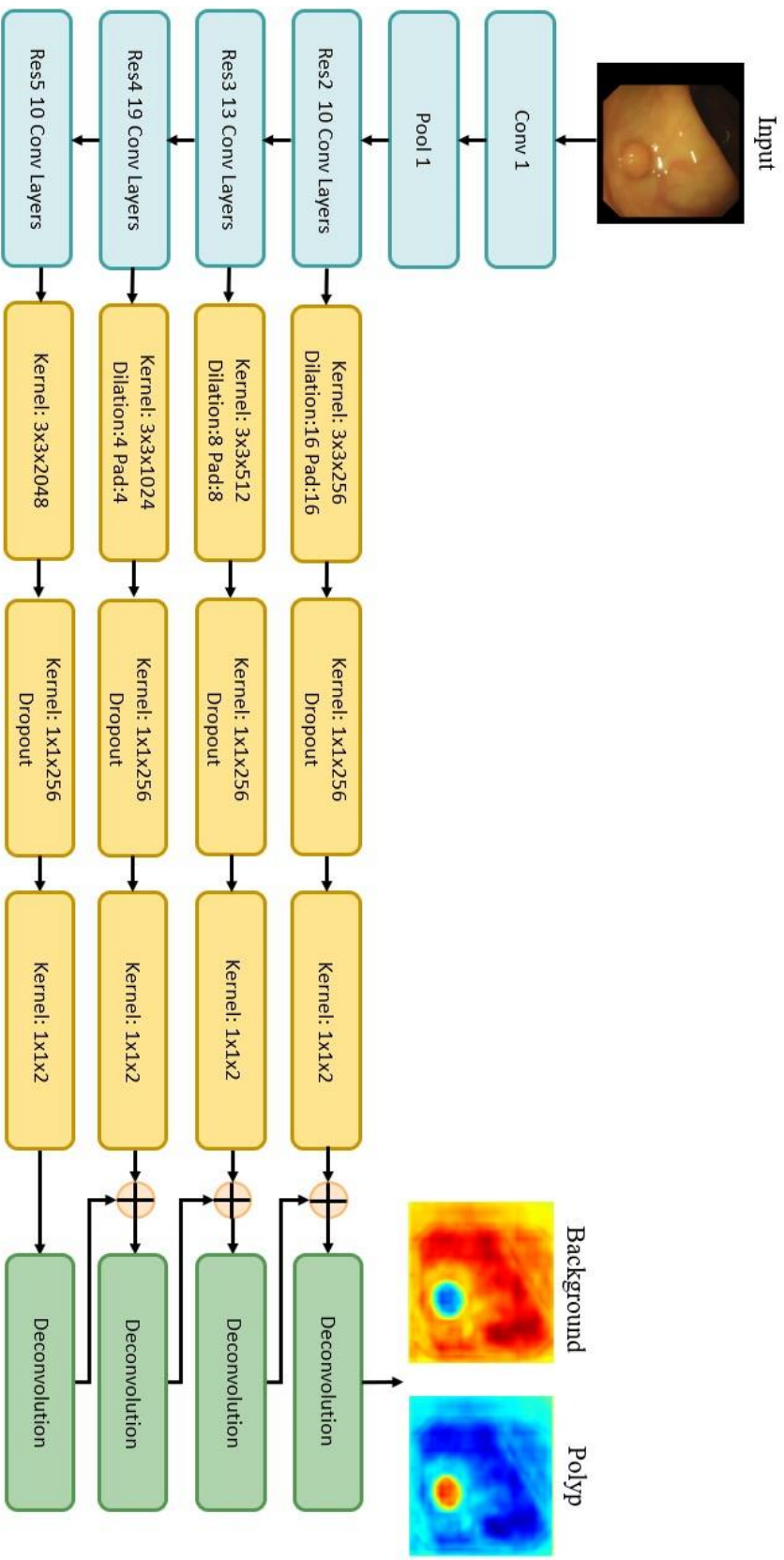


Figure 5. 17 The whole structure of Dilated-ResFCN.

As the region of each processed dilation convolution should be overlapping, dilation rates increase with resolution. Therefore, the rates of Res2-Res5 are 2, 4, 8 and 16 and corresponding kernel sizes are 5, 9, 17, and 33. The complete architecture of Dilated-ResFCN is shown Figure 5.17.

It should be noted that although the atrous spatial pyramid pooling (ASPP) of DeepLab V2 (Chen et al., 2018) has been proven to improve segmentation results, the memory required for this module cannot be supported. Moreover, the ASPP module has four different dilation rates. When this module is added to each sub-network of this method, 16 dilation rates must be determined. It is quite difficult to render these correct, and incorrect parameters increase the amount of noise in feature maps. Therefore, Dilated ResFCN only use single dilation convolution to increase the receptive field.

5.6.4 Proposed SE-Unet

This section described another key original contribution of the thesis. As indicated in the previous section the ResFCN and Dilated ResFCN focus on learning features using a larger receptive field. However, smaller polyps may be ignored by networks with a large receptive field, this is because smaller polyps may not excite, strongly enough, lower resolution feature map. To solve this problem, another novel network has been proposed. It has been designed specifically for detection and segmentation of small polyps missed by ResFCN and Dilated ResFCN.

Unet is used because it fuses all feature maps before using a pixel classifier to identify elements of a full-sized feature. This allows details of small or flat polyps to be preserved in the final feature map. The squeeze-and-excitation module can select the valuable features and to restrict unnecessary responses. The whole network can be divided into four parts: feature learning, up-sampling, atrous spatial pyramid pooling (ASPP) and the SE-module (squeeze-and-excitation). The complete structure of this method is shown in Figure 5.19. The roles of each component are discussed below:

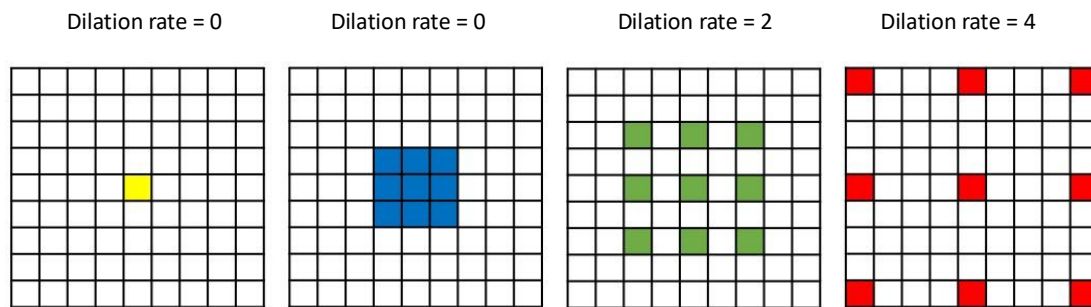


Figure 5. 18 Different dilated rates of the ASPP.

Feature learning and up-sampling

The U-net-based FCN requires more memory than FCN8s because it must save up-sampled features from the deep layer. To increase its efficiency, the feature learning module for this method is developed with VGG16.

The up-sampling module is a mirrored VGG16, but the pooling layers are replaced with deconvolutional layers. Moreover, there is a concatenation layer after each deconvolutional layer to combine low-level features and up-sampled features to maintain the details of object. As with the original Unet, the loss function involves cross-entropy with a sigmoid activation function.

Atrous spatial pyramid pooling

The ASPP learns multi-size features via the ASPP module. Since the last convolutional layer of VGG16 only outputs 256 feature maps, the required memory stores of the corresponding ASPP are very limited. Therefore, ASPP can be used for this method. The resolution of the last convolutional layer is 16×18 in the encoder. Based on the section 3.5.1, the ASPP of this method is shown in Figure 5.18 and it consists of 1×1 kernel, 3×3 kernel, and two dilation kernels with dilation rates of 2 and 4. These are used to detect features from a multi-sized feature.

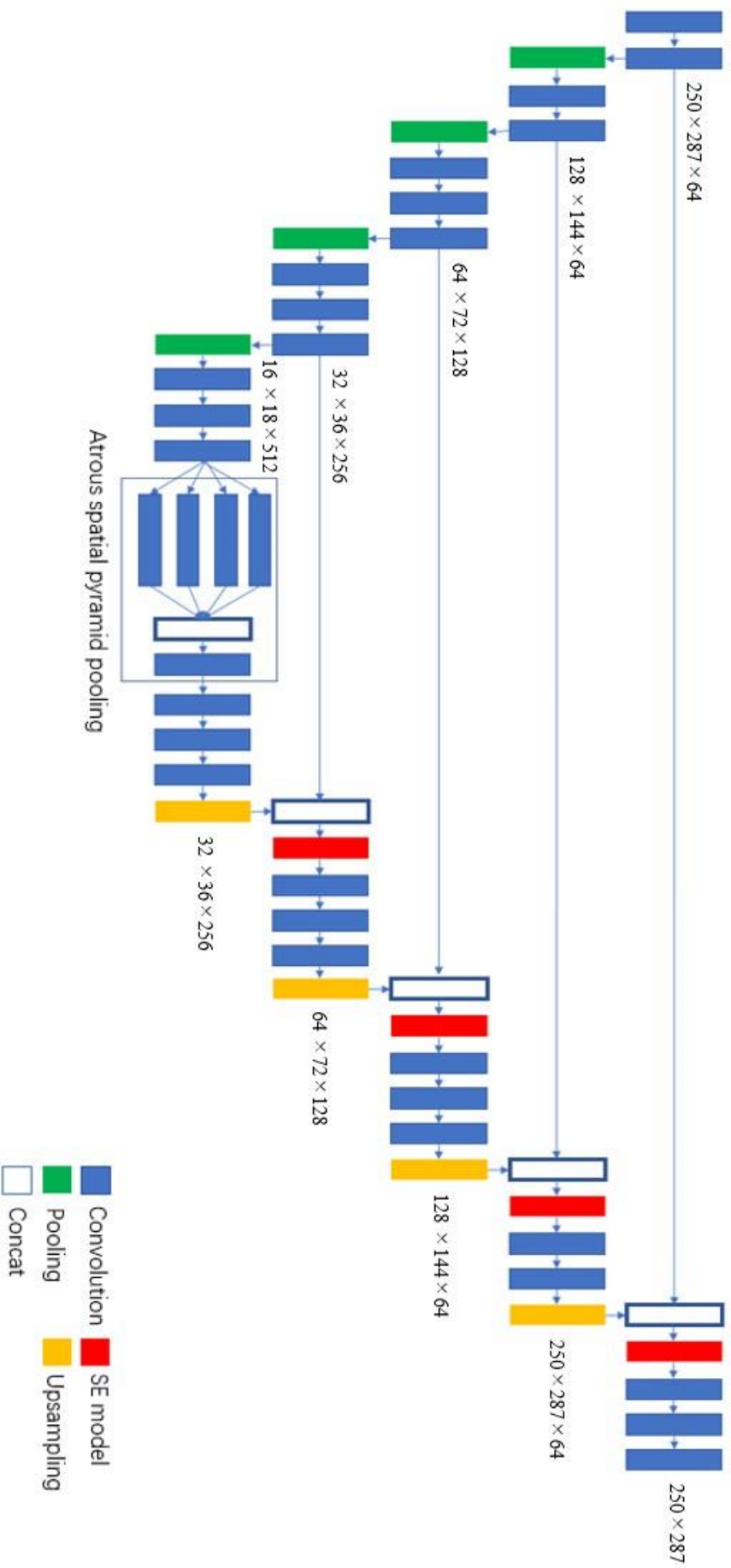


Figure 5. 19 he whole structure of SE-Unet.

Each component of the ASPP outputs 256 feature maps for a total number of maps of 1024. These features are saved in a concatenation layer. Pixels of the same position are fused by a $1 \times 1 \times 256$ kernel so that the number of final ASPP layers is 256.

Squeeze-and-excitation module

Shallow layers of the above three networks sometimes learn useless features from the polyp image such as the image border. These features do not provide any useful information and instead generate incorrect patterns and waste memory. This problem does not affect the FCN8s significantly because it can be restricted in deep layers. However, when these features are directly combined with the up-sampled features, they destroy the learned high-level features. Therefore, the SE-module is extended to U-Net, which is designed to mitigate this problem. An SE-module is added behind each concatenation layer in the up-sampling module. It assigns a coefficient for each feature map in the concatenation layer, and the region ranges between zero and one. The large coefficient represents corresponding features that are more valuable.

5.6.5 Test time segmentation

Since the CNN is not inherently invariant to image deformations, data augmentation aims at providing information that is more representative for various possible image changes, e.g. due to rotation, so the network can learn different image patterns created as a result of e.g. object rotation. In turn, when the data augmentation is employed at the test-time (so called test-time augmentation [53]), this may also help the network to make correct decisions as some of the augmented images may generate patterns which could resemble better image patterns learned during training. Based on this observation, for each input image, several rotated versions of the original image are generated and passed through the same network. The corresponding network outputs are rotated back to the original orientation and fused together.

Figure 5.20 shows the processing steps involved. As with training data, the rotation centre is assigned to the image centre. It is apparent that the segmentation results of

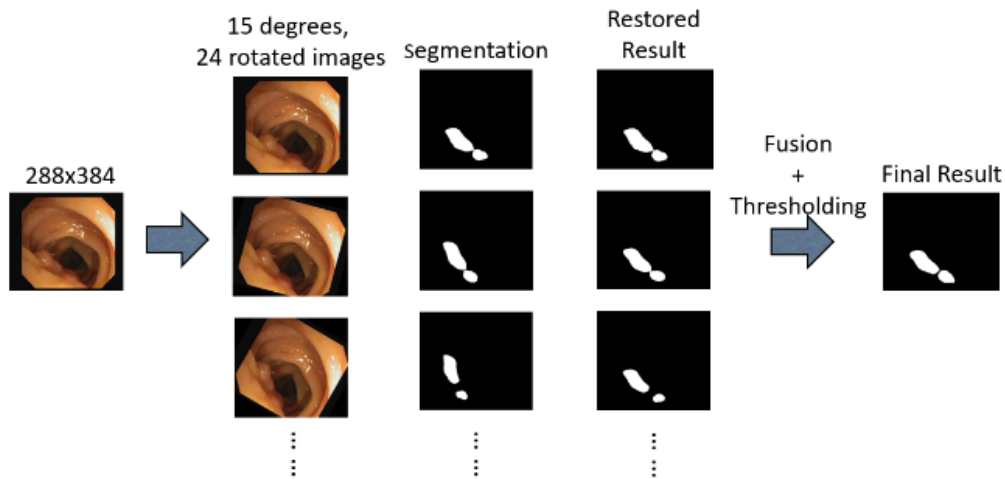


Figure 5. 20 Polyp segmentation based rotated test image.

each rotated image are different, meaning that the network applies different interpretations to rotated images. Moreover, it can further reduce border effects. It should be noted that the final segmentation result is fused using the binary segmentation results rather than confidence map

Similarly, to the rotation, the other test-time data augmentation models can change the segmentation results. However, rotation is the most obvious model and is easy to control, as the angle can only range between 0 and 360 degrees. For re-scaling, colour jittering or other methods, the range of parameters is broader than that of rotation, and using a wrong set of augmentation parameters may actually hinder the performance of such test-time augmentation. Therefore, for the reported results only rotation based test-time augmentation has been implemented.

5.7 Summary

This section analyses the polyp image and describes the design of the nearly proposed segmentation methods. Intra-class differences and inter-class similarities can occur in all image segmentation tasks (e.g., Discriminative feature network (section 4.5.7)). However, due to interference in image sampling and a high degree of similarity between polyps and backgrounds, this problem is particularly evident in cases of the

polyp segmentation. This section describes the proposed solutions to these two problems, Dilated ResFCN and SEUnet. Although these two problems are not fully solved, subsequent tests prove that they can be effectively controlled.

Chapter 6. Experiment design and results

6.1 Introduction

This chapter mainly focuses on describing the method used to select design parameters for segmentation methods proposed in Chapter 5 in reference to the polyp segmentation of the colonoscopy video dataset described in Chapter 4. Moreover, the chapter introduces various metrics used to evaluate and compare the performance of the proposed methods. Finally, it reports on overall results obtained from polyp video colonoscopy test data, including results derived from the 2017 and 2018 Gastrointestinal Image ANALysis (GIANA) challenges¹⁰.

This chapter is divided into five sections. The first section provides a brief overview of the experimental setup and hardware used for the experiments. The following section describes the validation framework based on the training dataset introduced in Chapter 4 while defining different training and validation data subsets. The third section defines different validation metrics, which are subsequently used for the methodological evaluation given in the rest of the chapter. The fourth section describes

¹⁰ <https://endovissub2017-giana.grand-challenge.org/> [Assessed 20 Oct. 2019]

Table 6. 1 Computer configurations.

	Usage	CPU	GPU	Memory	Hard Drive
The first computer	Training	I7-6900K	Quadro P6000, 24G	64G	SSD, 1T
The second computer	Training	I7-6900K	GTX1080Ti, 11G	64G	SSD, 1T
The third Computer	Testing	I7-3820	GTX 1080, 8G	16G	HDD, 2T

Table 6. 2 GPU divider, Cuda and Cudnn versions.

	Driver	Cuda	Cudnn
The first computer	384.130	8.0	6.0.21
The second computer	390.67	8.0	6.0.21
The third computer	375.39	8.0	6.0.21

the proposed testing methodology used for the selection of key design parameters. Finally, the last section presents the results of the 2017 and 2018 GIANA challenges, which were run as part of the Endoscopic Vision Challenge organized as part of the Medical Image Computing and Computer Assisted Intervention (MICCAI) conference.

6.2 Implementation details

For this work, all CNNs are implemented with Caffe. The Nvidia DIGITS is used to monitor training processing. In the testing stage, MATLAB is used as our Caffe interface. Two computers are used to train these networks, and another is used to do the testing. Their details are shown in Table 6.1 and 6.2.

For each experiment, all networks are optimized by Adam algorithm. They are trained with thirty epochs and an initial learning rate of 0.0001, then decreased by 0.1 at the tenth, twentieth and thirtieth epochs. It should be noted that these settings are only used to test the fundamental performance of the network, and so these settings may not be the best choice. The batch size of the training step is set to four.

There are two training datasets. The first one is used for cross-validation; information on this dataset is given in the following section. The second is the

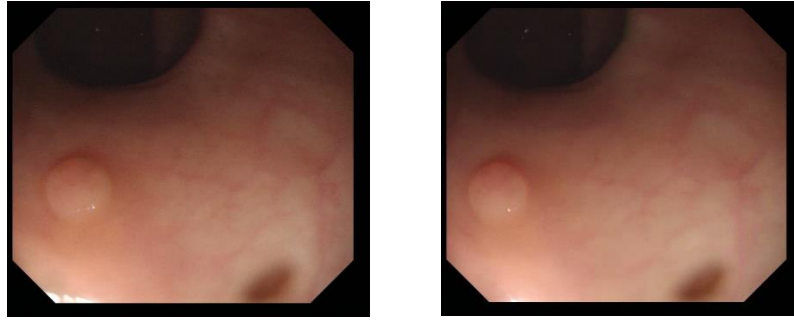


Figure 6. 1 Image No. 181 and No. 182 from vide 11.

complete augmented training database, and it is used to train the networks applied in the testing stage.

6.3 Validation data

The original SD and HD training images are divided into four cross validation folds (subsets). In each fold, part of the training data is used as the validation dataset, and the remaining images are used to train the networks. Since the HD database is integrated with the SD database, the first fold uses all SD augmented images for training and the re-scales HD images as the validation dataset. This is to check how well networks, trained on images acquired using one type of equipment (i.e. SD colonoscopy), can segment images acquired using different equipment (i.e. HD colonoscopy).

The SD subset consists of the images extracted from a few video sequences, with images from the same sequence being highly correlated (i.e. showing the same polyp with possibly only small appearance variations). For example, image No.181 and No.182 (Figure 6.1) are from the same video. These two images show the same polyp with similar light, shadow and surrounding tissue patterns, their cosine similarity [80] is 0.9982 which demonstrates that there is very little difference between them. If image No. 181 were to be used for training and image No.182 for validation, this would misrepresent the quality of the validation results as almost the same data would have been used for training and testing.

To ensure the validity and reliability of the methodology, validation data folds are

Table 6. 3 The details of four validation subsets. The number of images in the folds training subset is given of the augmentation.

		Fold Training Data	Fold Validation Data
V1	Image index	SD: 1~300 (Videos: 1~13) HD: 1~56	HD: 1~56
	The number of images	4784	56
V2	Image index	SD: 1~203 (Videos: 1~7) HD: 1~56	SD: 204~300 (Videos: 1~7)
	The number of images	4832	96
V3	Image index	SD: 98~300 (Videos: 5~13) HD: 1~56	SD: 1~97 (Videos: 1~4)
	The number of images	4821	97
V4	Image index	SD: 1~97, 204~300 (Videos: 1~4, 8~13) HD: 1~56	SD: 98~203 (Videos: 5~7)
	The number of images	4733	106

created based on the random selection of videos (section 5.2) rather than images. Frames extracted from the same video are not used for training and validation at same time. This approach aims to simulate the situation of real segmentation task. A random selection of images into folds' training and test subsets would lead to similar images (images representing the same polyp) present in both subsets. Further information on each fold and validation dataset is shown in Table 6.3.

6.4 Metrics

For the polyp segmentation challenge, the Dice and Jaccard indexes are designed as the standard metric for a single image. They are always used to measure similarities between the obtained binary segmentation map and corresponding ground truth.

Dice index

The Dice index, also called the Sørensen–Dice coefficient [81], was independently proposed by botanists Thorvald Sørensen and Lee Raymond Dice. It is defined as follows:

$$Dice = \frac{2|S \cap G|}{|S| + |G|} \quad 6.1$$

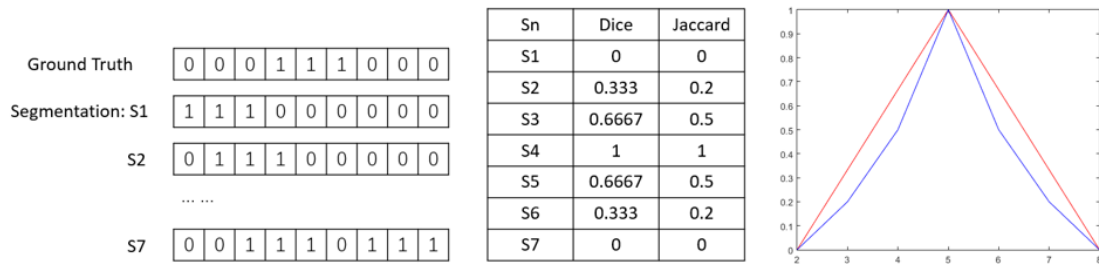


Figure 6. 2 The instance of Dice and Jaccard index. Left: The ground truth and segmentation results. Middle: The corresponding Dice and Jaccard metrics. Right: The visualization of these measures. Red: Dice index, Blue: Jaccard index.

In this equation, S represents the binary segmentation result. Normally, the segmented foreground is represented by 1, and the other parts are represented by 0. G represents the ground truth. $|S \cap G|$ represents the overlapping area between S and G . The value range of the Dice index is $[0, 1]$ where 1 denotes that the result and ground truth are exactly the same.

Jaccard index

The Jaccard index developed by Paul Jaccard is the proportion between the intersection and union of two sets. Therefore, it is also called the Intersection over Union for a specific task (e.g., image detection). As with the Dice index, segmentation results and ground truth are denoted as zero and one, and proportions range from zero to one. The Jaccard index is defined as follows:

$$Jaccard = \frac{|S \cap G|}{|S \cup G|} \quad 6.2$$

Figure 6.2 explains how differences affect the evaluation. The ground truth and result include two 1×9 vectors of 0 and 1. In different states, the position of 1 is moved to simulate correct and incorrect segmentation. The middle figure shows the value of the Dice and Jaccard indexes for different simulated segmentation results represented by S1-S7. However, when a partial overlap occurs, there are obvious differences in results. The right figure demonstrates their changes, and the change in the Dice index is linear while that of the Jaccard index is non-linear, and the latter is always smaller

than the former. The value of Dice and Jaccard are the same for the perfect results (equal to 1) and missed object (equal to 0).

Discussion of Dice and Jaccard index

The Dice and Jaccard can be also expressed in terms of the standard binary classification confusion matrix. For binary classification tasks, the results can be divided into four situations:

Table 6. 4 Definition of the confusion matrix.

		True condition	
		Positive (P)	Negative (N)
Predicated condition	Predicted Positive	True Positive (TP)	False Positive (FP)
	Predicted Negative	False Negative (FN)	True Negative (TN)

For image segmentation, the above components are as follows:

Condition Positive: Foreground of ground truth

Condition Negative: Background of ground truth

Predicted Positive: Foreground of segmentation results

Predicted Negative: Background of segmentation results

True Positive: Correctly predicted foreground of segmentation results

True Negative: Correctly predicted background of segmentation results

False Positive: Incorrectly predicted foreground of segmentation results

False Negative: Incorrectly predicted background of segmentation results

Segmentation (S)				
0	0	0	0	0
0	0	0	0	0
0	0	1	1	1
0	0	1	1	1
0	0	1	1	1

Ground truth (G)				
0	0	0	0	0
0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0

S + G				
0	0	0	0	0
0	0	0	0	0
0	1	1	1	1
0	1	1	1	1
0	1	1	1	1

FN
2TP
FP

Figure 6.3 The example of Dice index and its components.

$$Dice = \frac{2TP}{2TP + FP + FN} \quad 6.3$$

Figure 6.3 shows a Dice index only focused on the foreground. This work does not differentiate between FP and FN, corresponding to all the over and under segmentation. Dice index is equivalent to F1 score, and it can be expressed as functions of precision and recall.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad 6.4$$

Where

$$Precision = \frac{TP}{TP + FP} \quad 6.5$$

$$Recall = \frac{TP}{TP + FN} \quad 6.6$$

The Jaccard and Dice indexes use a similar equation and only differ in that the overlapping area is only counted once in the Jaccard index.

$$Jaccard = \frac{TP}{TP + FP + FN} \quad 6.7$$

These two measures can be expressed as function of each other:

$$Jaccard = \frac{Dice}{2 - Dice} \quad 6.8$$

$$Dice = \frac{2 \times Jaccard}{1 + Jaccard} \quad 6.9$$

Suppose D represents the Dice index, J represents the Jaccard index. The transformation between J and D can be defined as follow:

$$\frac{2(TP + FP + FN)}{2TP + FP + FN} + D = \frac{2TP + 2FP + 2FN}{2TP + FP + FN} + \frac{2TP}{2TP + FP + FN} = 2$$

$$\frac{D}{J} + D = 2 \quad \text{For } J \quad J = \frac{D}{2 - D}$$

$$\text{For } D \quad D + JD = 2JD(1 + J) = 2J \quad D = \frac{2J}{1 + J}$$

Validation:

$$J = \frac{D}{2 - D} = \frac{2TP}{2TP + FP + FN} \times \frac{2TP + FP + FN}{2TP + 2FP + 2FN} = \frac{TP}{TP + FP + FN}$$

$$D = \frac{2J}{1 + J} = \frac{2TP}{TP + FP + FN} \times \frac{TP + FP + FN}{2TP + FP + FN} = \frac{2TP}{2TP + FP + FN}$$

In this work, the results of Dice index are reported only, as the Jaccard index is monotonic with the Dice index as shown above. For the evaluation of results, the mean value and standard deviation of the Dice index are used. The mean value is used to evaluate the performance of the segmentation method, and the standard deviation is used to assess stability.

More measures

Reinke et al. [82] discuss the impact of different measures on the ranking of methods. One method may get the best results using one measure, but it may not perform well when assessed using a different metric. The other community used metrics are described below:

Accuracy

Accuracy is a common measure used for image classification and is defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad 6.10$$

It is apparent that TN is in the numerator. During segmentation, when the area of the object is very small, TN will be larger than TP. This renders the method highly accurate even when TP is close to zero. Therefore, its accuracy is not suitable for evaluating segmentation results.

Precision / Positive predictive value

Precision is a component of the Dice index or F1 score defined in Equation 6.5. During image segmentation, this measure can evaluate the ratio between correctly recognized pixels and all recognized pixels. We use it as a single measure to obtain final results from final polyp segmentation methods. In the context of segmentation, it could be used as indicator of over segmentation.

Recall / Sensitivity/Hit rate / True positive rate

Recall is another component of the Dice index designed to evaluate the ratio between correctly recognized pixels and true pixels. For precision, the measure is also the single measure used for polyp segmentation. In the context of segmentation recall could be used as indicator of under segmentation.

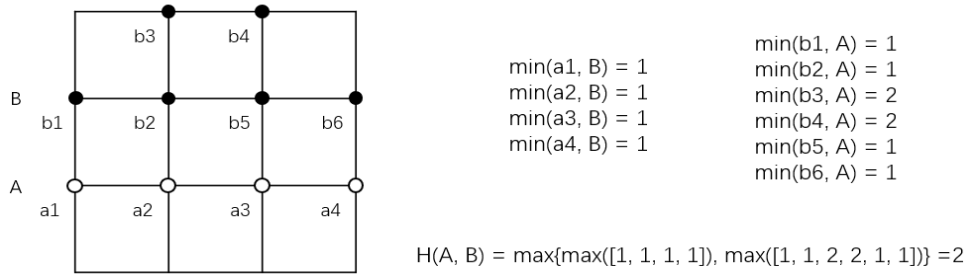


Figure 6. 4 The calculation of Hausdorff distance.

Hausdorff Distance

In this study, Hausdorff Distance [83] is used to determine the shape of the final segmented polyp for cross-validation. Hausdorff Distance is the measure used to determine similarities between the boundaries of two objectives. It is defined as:

$$H(G, S) = \max\left\{\sup_{x \in G} \inf_{y \in S} d(x, y), \sup_{x \in S} \inf_{y \in G} d(x, y)\right\} \quad 6.11$$

where $d(x, y)$ denotes the distance between pixels $x \in G$ and $y \in S$. \sup and \inf represent the supremum and infimum, respectively. In this equation, they can be regarded as the maximum and minimum. The method involves two steps. First, calculate the minimum distance between objective G to S and find the maximum value of these distances. Then, use the same approach to calculate and find the maximum distance from S to G. Finally, compare these two maximum values and choose the greatest value as the output. The best result of this measure is 0, which means that the shapes of two objectives are completely overlapping. The value for the missing data is 'Inf'

Figure 6.4 presents a simple example to explain Hausdorff Distance. The width and height of block is 1. A and B denote the boundaries of two objectives and are denoted by white and black points, respectively. The left part of the Figure 6.4 shows the calculation, and the result is 2. Figure 6.5 shows the value of Hausdorff distance for different segmentation results.

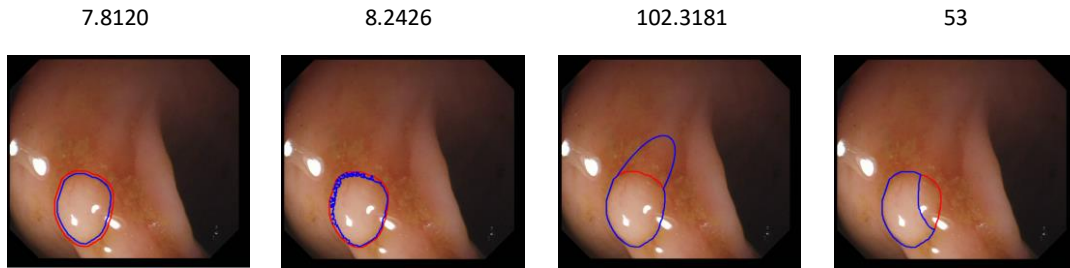


Figure 6. 5 The Hausdorff Distance of different segmentation results.

6.5 Experiment Results

6.5.1 Validation results of FCN8s, ResFCN, Dilated ResFCN and SE-Unet

This section shows the best results of four segmentation methods. The experiment aims to compare their fundamental performance, the results are not processed with any post-processing.

The four methods are trained with the above sub-sets. The pre-trained weights of FCN8s are retrieved from fcn.berkeleyvision.org. As ResFCN and Dilated ResFCN are new methods proposed in this study, their feature learning module is initialized by ResNet-50 which trained on the ImageNet database, and other remaining convolutional and up-sampling kernels are initialized by Xavier [84] and bilinear interpolation weights. SE-Unet is a special case for which initialization involves two steps. In the first step, SE modules are removed from SE-Unet, and then this simplified SE-Unet is trained on each sub-set. In the second step, SE modules are added and re-trained. Otherwise, SE-Unet cannot learn the features.

In Table 6.5 columns V1 to V4 represent the averaged Dice indexes of each sub-validation. (i.e. these four sub-sets are defined in Table 6.3) The last two columns list the mean values and standard deviations of the four validation results.

In Table 6.5, the mean value of Dilated ResFCN is ranked the best of the sub-validations. Figure 6.7 shows a visualization of Table 6.5 and demonstrates that Dilated ResFCN is well ahead of the other methods, as its shapes can completely cover those of all other methods. SE-Unet and ResFCN generate some similar results, but the

Table 6. 5 The overall Dice index of four methods (section 5.6).

Network	V1	V2	V3	V4	Mean	Standard Deviation
FCN8s	0.6938	0.5555	0.5891	0.6901	0.6322	0.0704
ResFCN	0.6298	0.7307	0.6211	0.8063	0.6970	0.0882
DilatedResFCN	0.7596	0.7825	0.6909	0.8824	0.7789	0.0792
SE-Unet	0.6302	0.7497	0.6702	0.7376	0.6969	0.0566

former generates the lowest standard deviation. Therefore, it should be considered the second-best method.

Figure 6.6 shows segmentation results of typical small, median and large polyps. The polyp occurrence confidence maps (POCM) show that FCN8s can determine the approximate position of a polyp, but it is very easy to generate FP and FN. For each POCM, the distribution of high response is diffused, and the shape is irregular. For the large polyp, FCN8s generate many strong responses outside of the polyp. These errors are difficult to remove because such an operation would generate more false negatives. For example, the response of some false positives is similar to the response of the polyp centre. The higher threshold would generate a hole within the segmented polyp. This problem illustrates that a large polyp is more complex to segment.

For the Dilated ResFCN, its POCM are more accurate than those of the other methods. There is a clear boundary in each POCM. Each response corresponds to a specific tissue, and the polyp always generates the highest value. Furthermore, polyp responses are more stable and uniform than those observed from the first two methods. As the Dilated ResFCN is developed from ResFCN, it also has a checkboard artefact.

Unfortunately, all the four methods do not cope well with the image border, and all of the POCM present relatively stronger responses near the border. Thus, the CNN uses the information around borders as a pattern to describe a polyp image. Although the latter two methods detected polyps more accurately, this does not mean that they fully model the relationship between borders and a polyp.

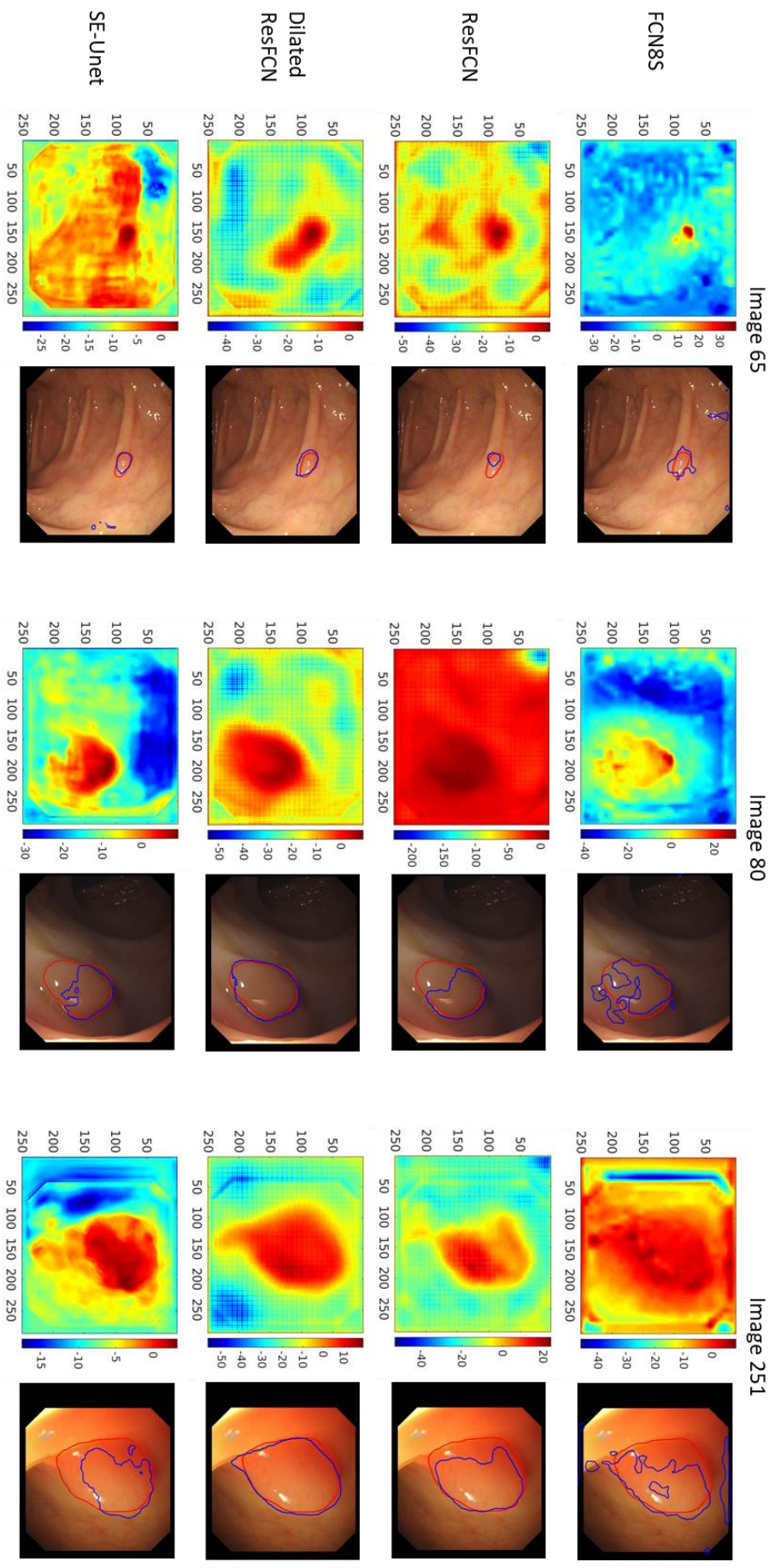


Figure 6.6 Typical results obtained for the SD images using FCN8s, ResFCN, Dilated ResFCN and SE-Unet networks. For each image: the left column shows the POCM with the red colour representing the high confidence and blue colour representing the low confidence of a polyp presence; the right column shows the original images with superimposed red and blue contours representing the ground truth and the segmentation results respectively.

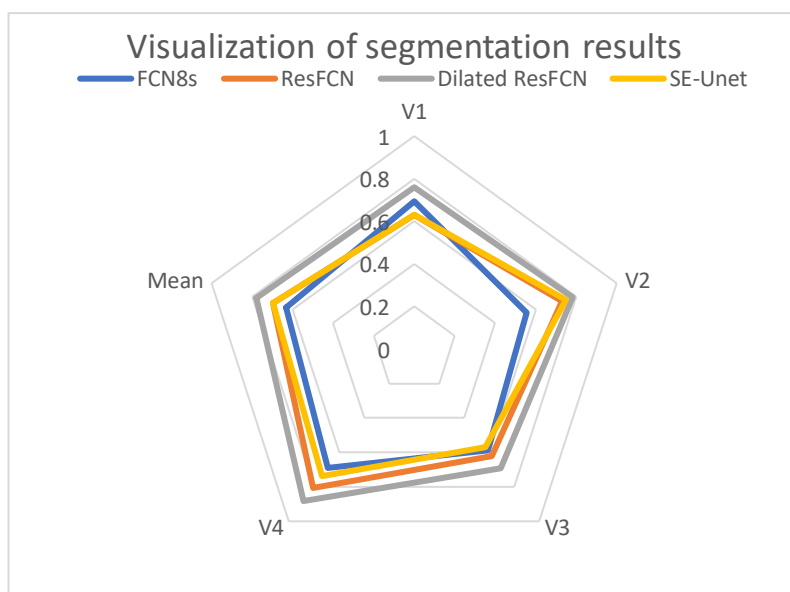


Figure 6. 7 Visualization of the Dice index from Table 6.5.

The POCM generated by SE-Unet are similar to those of the FCN8s, but the former generates fewer false positives and smoother boundaries. SE-Unet is also generates the fewest responses on the border. These responses are distributed across a colour image and not in black regions. Thus, unnecessary features have been suppressed.

Discussion of network size.

Table 6.6 and 6.7 shows resources required for the evaluated networks, including the number of weights, training and test time and network memory requirements. From this table it is clear that FCN8s requires more trainable parameters and training time than the other methods, but it performs the worst. This indicates that weights have not been used properly. Unexpectedly, ResFCN and Dilated ResFCN processing times are shorter than expected, showing that the 1×1 kernel are very efficient. However, the two networks still require considerable memory resources due to the presence of deep structures. Most of the CNN frameworks save internal feature maps in GPU memory to improve the processing speed of backpropagation. The more layers a network has, the more internal feature maps are saved. For SE-Unet, many up-sampled feature maps need to be saved, so this method also involves considerable memory requirements.

Table 6. 6 The training statistics for the tested four networks.

Networks	The number of weights	Training time	Required Memory
FCN8s	130 million	177min on GTX 1080Ti 201min on Quadro P6000	5.53 G
ResFCN	23 million	70min on GTX 1080Ti 61min on Quadro P6000	5.3 G
Dilated ResFCN	79 million	115min on GTX 1080Ti 121min on Quadro P6000	6.59 G
SE-Unet	24 million	135min GTX 1080Ti 157min on Quadro P6000	5.24 G

Table 6. 7 Processing mean times of a single image during prediction (testing) comparing performance of the CPU against GPU.

Networks	I7-3820 (CPU)	GTX-1080 (GPU)
FCN8s	8.0s	0.047s
ResFCN	0.70s	0.040s
Dilated ResFCN	1.80s	0.050s
SE-Unet	1.760s	0.0450s

In addition, this section also investigates whether deeper networks would help to improve the segmentation performance. To test this, the ResNet-50 in ResFCN is replaced by ResNet-101 and ResNet-152. Figure 6.8, shows the mean Dice index computed on the four validation folds (represented by the asterisk) and the corresponding variance (represented by the vertical bar) for three tested network configurations. It can be seen ResNet-152 is the worst one, because it has the smallest mean Dice index and largest variance. ResNet-101 and ResNet-50 have similar mean value, but ResNet-101 has the larger variance. This experiment demonstrates the performance gets worse when a deeper network is used. Therefore, the ResNet-50 has been selected as the feature extraction subnetwork for further investigation.

Discussion of missed polyps

Table 6.8 counts objects missed by the four methods. FCN8s detected most polyps, missing only six samples. The worst method is ResFCN, which missed thirty-two polyps. Dilated ResFCN and SE-Unet are superior to ResFCN but still lag behind the FCN8s.

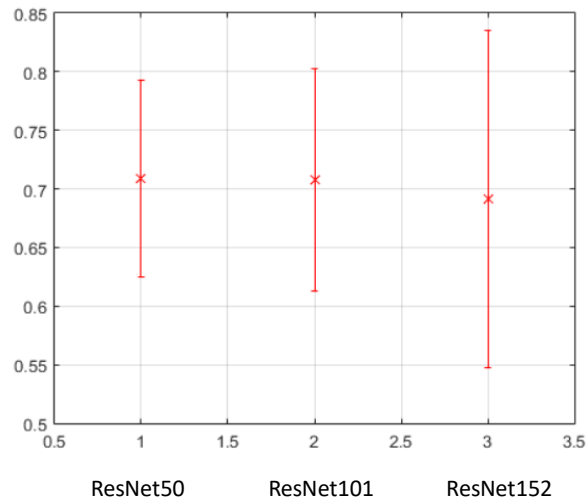


Figure 6. 8 ResNet50, 101 and 152 based FCN.

Table 6. 8 Data missed by each method (see section 5.6).

Networks	V1	V2	V3	V4	Total
FCN8s	0	0	3	3	6
ResFCN	3	4	23	2	32
Dilated ResFCN	2	3	15	0	20
SE-Unet	0	3	10	2	15

When strictly considering the results shown in Table 6.8, the FCN8s are indeed the best methods. However, segmentation methods should be selected based on their overall performance. Alternatively, although the FCN8s detected the most polyps, their mean value still lags behind those of the other methods. This lag indirectly illustrates that the other methods generated much more accurate segmentation results for the detected polyp. Table 6.10 and 6.11 list polyps missed by the Dilated ResFCN and corresponding segmentation results for FCN8s and SE-Unet. The results demonstrated that the missed polyp can be segmented by FCN8S and SE-Unet.

Figure 6.10 and 6.11 show that their segmentations are not very accurate. Figure 6.9 shows the polyp missed by FCN8s and SE-Unet. SE-Unet’s segmentation results are acceptable because corresponding Dice indexes are greater than 0.5. For the FCN8s, most of the segmentation results are not good with only two results exceeding a value



Figure 6. 9 The SD polyp missed by Dilated ResFCN as well as by FCN8s and SE-Unet.

Table 6. 9 The mean Dice index mean values of a re-segmented polyp generated by FCN8s and SE-Unet (see sections 5.6).

Networks	V1	V2	V3	V4	Mean	Mean*
FCN8s	0.1429	0.4066	0.2498	-	0.266	0.2891
SE-Unet	0.5319	0.6951	0.2424	-	0.4898	0.3684

of 0.5. It can be concluded that these FCN8 segmentation results are not useful. Table 6.9 lists the mean Dice index of the polyps missed by Dilated ResFCN for each sub-validation. Since there are no missing polyps in the fourth validation, the values are marked as '-'. The mean value is calculated as a mean from three validation and the mean* is averaged from the Dice index of all missed polyps regardless of the validation fold. Either way, SE-Unet is better than the FCN8s.

This comparison indicates that the SE-Unet can be used as a secondary segmentation method to segment polyps missed by the Dilated ResFCN. Table 6.12 lists the results of the hybrid approaches. In the hybrid approach, if the polyp is missed by the Dilated ResFCN, the corresponding images is subsequently processed by the FCN8s and SE-Unet. A comparison of these last results shows that the Dice index value was significantly improved in terms of single sub-validations and mean values.

Table 6. 10 Polyp missed by the Dilated ResFCN and successful segmented by the FCN8s.

Image Name	Sub-Validation	FCN8s Dice index	SE-Unet Dice index
33	V1	0.2256	0.1814
1	V3	0.1038	0
50	V3	0.2856	0.2856
51	V3	0.2520	0
56	V3	0.1919	0
83	V3	0.2992	0
88	V3	0.4918	0
91	V3	0.6065	0.0021
92	V3	0.0295	0.0096

Table 6. 11 Polyp missed by the Dilated ResFCN and successful segmented by the FCN8s.

Image Name	Sub-Validation	FCN8s Dice index	SE-Unet Dice index
43	V1	0.0601	0.8825
244	V2	0.3658	0.8953
278	V2	0.4332	0.6032
298	V2	0.4208	0.5869
10	V3	0.5085	0.6048
12	V3	0.4203	0.7843
27	V3	0	0.4816
29	V3	0.028	0.7079
84	V3	0.4810	0.6061

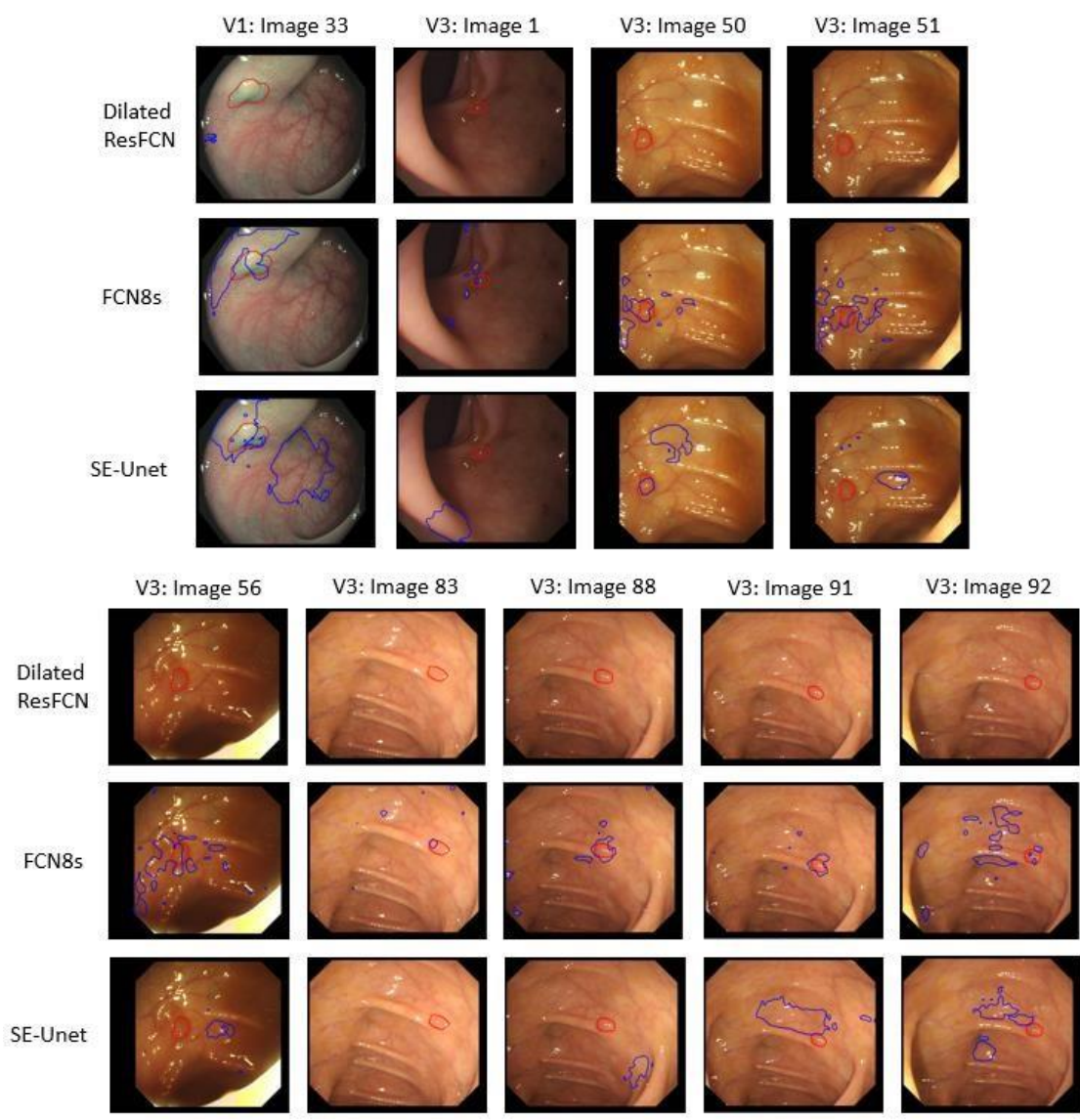


Figure 6. 10 The SD polyp missed by Dilated ResFCN and segmented by FCN8s.

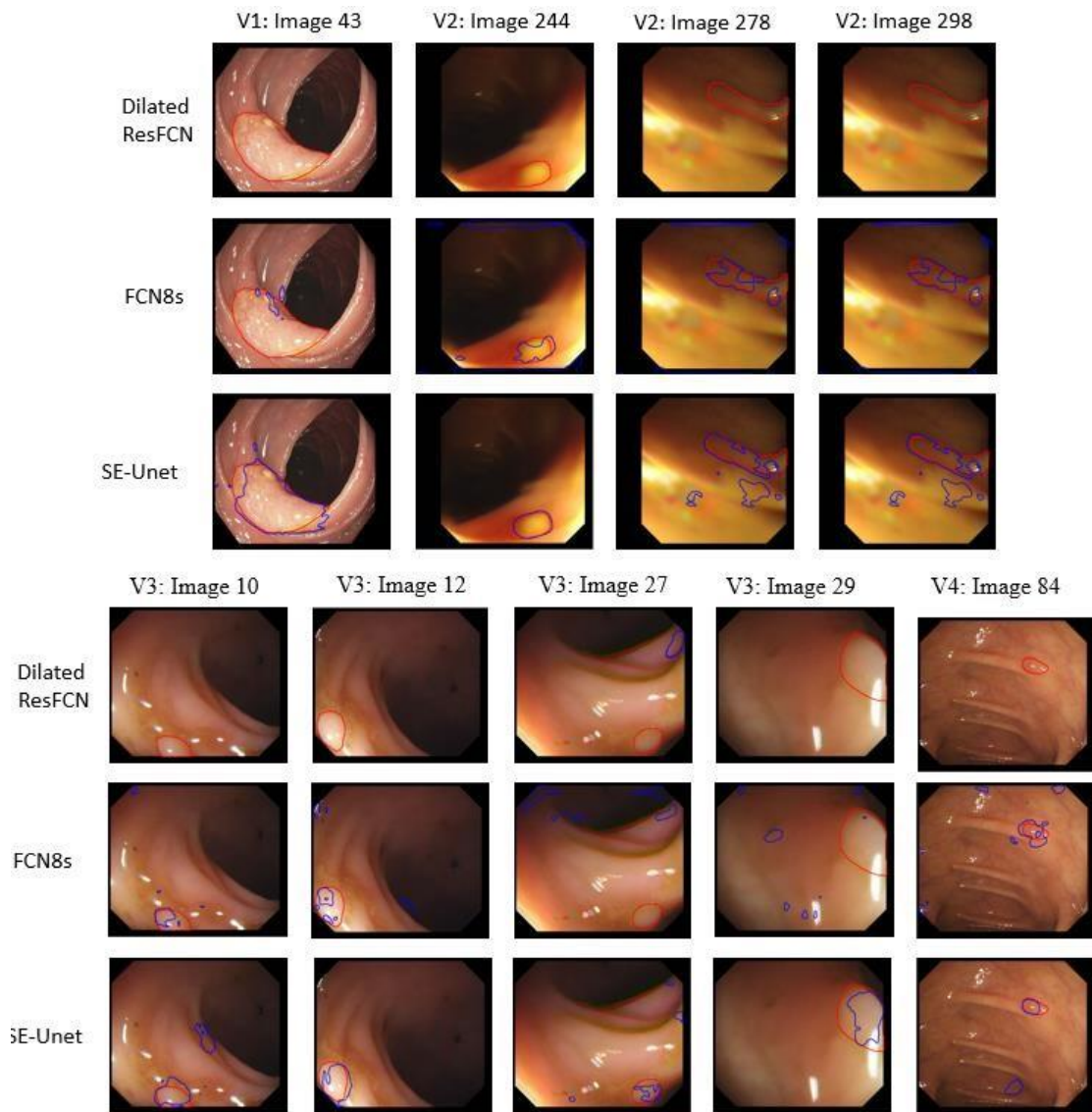


Figure 6. 11 The polyp missed by Dilated ResFCN and segmented by SE-Unet.

Table 6. 12 Overall results after combining Dilated ResFCN with FCN8s or SE-Unet (see section 5.6).

Combinations	V1	V2	V3	V4	Mean
Dilated ResFCN +FCN8s	0.7647	0.7952	0.7296	0.8824	0.7930
Dilated ResFCN +SE-Unet	0.7786	0.8042	0.7269	0.8824	0.7980

Table 6. 13 Results generated by the background with each method (see section 5.6).

Network	V1	V2	V3	V4	Mean	Standard Deviation
FCN8s	0.6777	0.5950	0.4965	0.7518	0.6378	0.0966
ResFCN	0.6764	0.7078	0.6257	0.8242	0.7085	0.0842
Dilated ResFCN	0.7668	0.7955	0.6979	0.8839	0.7860	0.0771
Hybrid	0.7668	0.8172	0.7378	0.8839	0.8014	0.0640

6.5.2 Validation results using background confidence map

In this section, the results of FCN8s, ResFCN and Dilated ResFCN are generated using the background confidence maps. No results for SE-Unet are not given in this section, as SEUnet is performed by the sigmoid cross entropy loss function, and its output layer only has one channel.

Table 6.13 shows the results. This table shows that Dice indexes are further improved. The standard deviation is also significantly reduced. Hybrid method can also be used for this section. The last row shows that the hybrid approach further improves performance, increasing the mean value to greater than 0.8 and reducing the standard deviation to 0.0640. Moreover, using background also reduces the number of missing polyps. Table 6.14 counts all data missing when using the background confidence map. For the first sub-validation, the Dilated ResFCN detected all of the polyps. Its averaged Dice index is smaller than that generated with the hybrid approach involving the Dilated ResFCN and SE-Unet. That reduction means SE-Unet generates more accurate segmentation results for these new polyps detected

Table 6. 14 Number of missing polyps using the background confidence map (see section 5.6).

Networks	V1	V2	V3	V4	Total
FCN8s	0	1	7	0	8
ResFCN	2	3	20	1	26
Dilated ResFCN	0	3	13	0	16

in the background. However, the background has better global mean value, so the Dilated ResFCN in the final hybrid method selects the background as the input.

Discussion of reasonable stopping epoch.

Regarding training data, it is quite difficult to determine the stopping epoch, as it is affected by many factors such as the number and types of training images used, initialization methods and learning rates. These issues are discussed in this section. The determined stopping epoch used for this work may not be the best one, but it presents certain rationalities using by cross validation.

Figure 6.12 shows the Dice index of each method for different epochs with thresholds equal to minus one. Figure 6.13 highlights improvements or setbacks occurring during training. A single line includes twenty-nine points, and each point represents the difference between two neighbouring epochs. When the value of a point is equal to zero, there is no change between two epochs. From these figures, two conclusions can be obtained:

First, the best method could be determined. Although some values continue to increase, improvements are so minor that the ranking is not changed. Figure 6.13 shows that the change in Dice indexes are close to zeros. This further shows that the Dilated ResFCN performs better than the other methods, and its success is not dependent on the use of specific data or number of training epochs.

Second, the reasonable learning rates of each method can be determined. As noted earlier, the processing method can be divided into three stages. The learning rate between the first and ninth epochs is $1e-4$, and changes occurring during this training

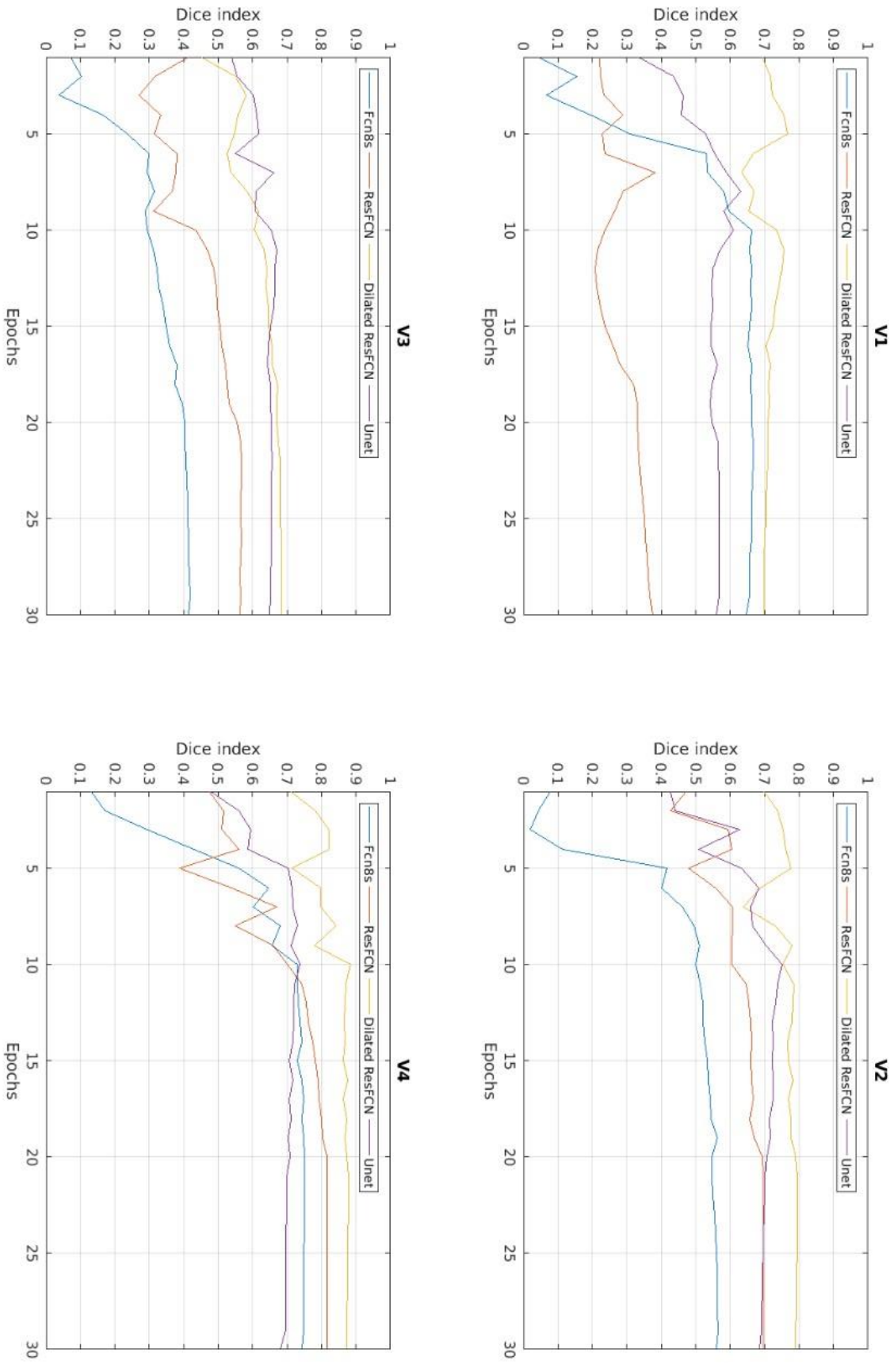


Figure 6. 12 The Dice index of each method during the training

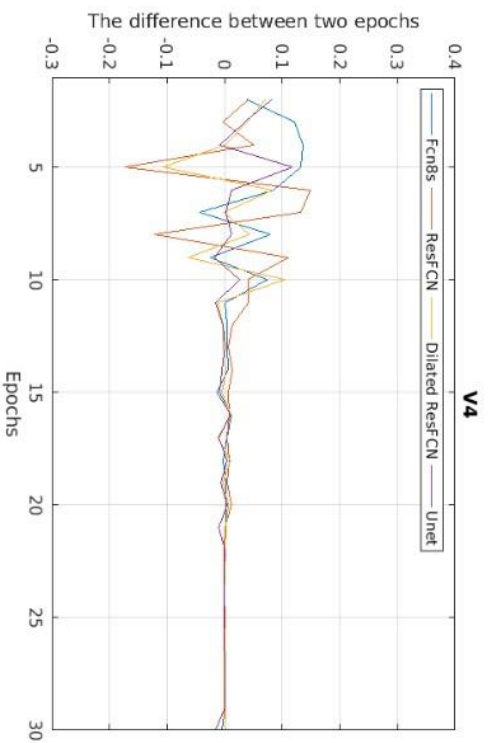
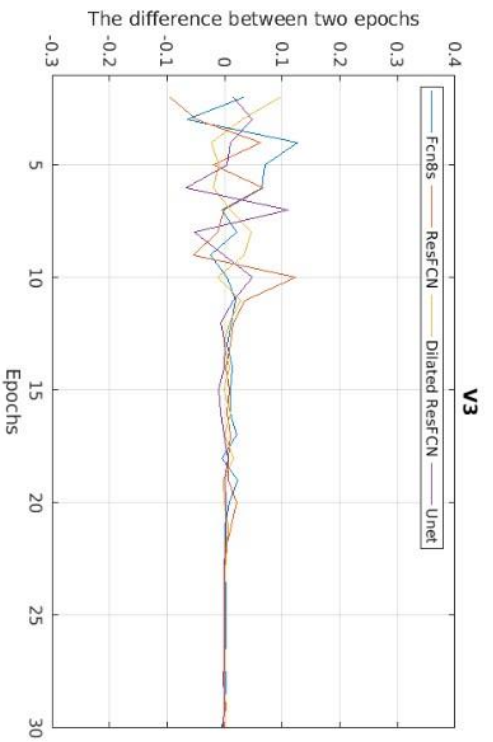
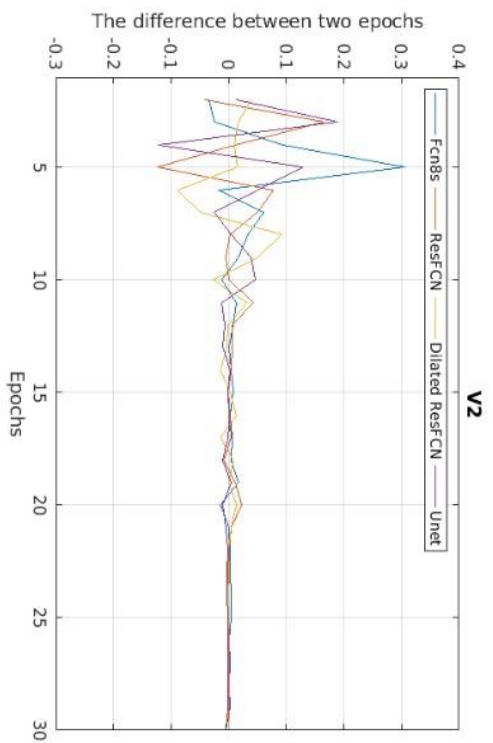
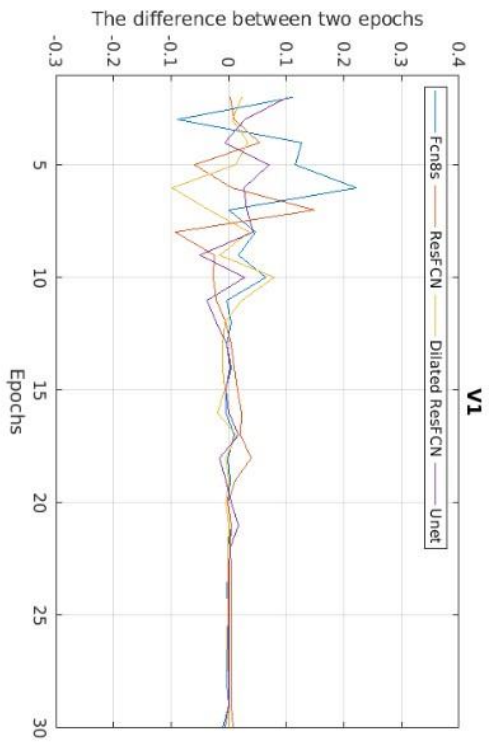


Figure 6. 13 The difference between two neighbouring epochs.

phase are considerable. All of the methods present an obvious increasing trend in this stage, but they are not stable. The second stage occurs between the tenth and nineteenth epochs. It is evident that the learning gradually become stable. Figure 6.12 shows that SE-Unet always generates its best results near the tenth epoch and that $1e-5$ is last valid learning rate. The learning rate of the twentieth to twenty-ninth epochs is $1e-6$. For the Dilated ResFCN, results for V2, V3 and V4 first slightly improve, and then these results become more stable and the changes in the Dice index less significant. That stability means the network has converged in this stage. When training continues, network over-fitting results.

From these two findings, the last learning rate of the Dilated ResFCN is designated as $1e-6$, and the network weights are selected twenty-fifth epochs in the testing stage. Although V1 and V4 generated their best results for the earlier epoch, there is a risk of selecting the worse weights. Therefore, while this strategy may not the best one, it is the least risky. Similarly, SE-Unet can select a stopping epoch between the fifteenth epochs at a learning rate of $1e-5$.

Regarding the best results of the Dilated ResFCN for V1 and V4, that of explanation involves transfer learning. The feature extraction network has learned some common features from several natural images, and some of the features also appear in the polyp (e.g., as similar colours or gradients between different pixels). Recently, some proposed methods have employed a pre-trained feature extractor to describe new types of data without training.

Discussion of validation results without transfer learning

For this experiment, methods are trained without the pre-trained model to investigate the significance of the transfer learning for polyp segmentation. Weights of the convolutional kernel are initialized by Xavier [84] and the up-sampling kernels are initialized by bilinear interpolation weights. Other settings used are the same as those used for the previous experiments. Corresponding results are shown below:

Table 6. 15 The results of networks without performance without the transfer learning (see section 5.6).

Networks	V1	V2	V3	V4	Mean	Standard Deviation
FCN8s	0.5202	0.6252	0.3914	0.6173	0.5385	0.1091
ResFCN	0.6021	0.3404	0.3236	0.4003	0.4166	0.1280
Dilated ResFCN	0.4903	0.3544	0.2666	0.3412	0.3631	0.0932

It should be noted that SE-Unet does not use transfer learning, so there is no result in Table 6.15. This table shows that the FCN8s rank first while the results of the other two networks are too low. This finding shows that the pre-trained weight is quite important for the Resnet-based FCN.

6.5.3 Validation results using precision, recall, and Hausdorff distance metrics

This section presents the results of precision, recall, and Hausdorff distance tests and further compares the performance of the four methods (FCN8s, ResFCN, Dilated ResFCN, and SE-Unet) based on these results.

The first metric is precision. As can be observed from Equation 6.5, the higher its value, the lower false positives are, the fewer errors there are in the results. Table 6.16 illustrates the precision score of each method. Hybrid method outperforms the other methods and results are achieved the best average performance. Second place goes to Dilated ResFCN, as its V1 and V4 are the same as for the Hybrid model, as no polyp is missed in these two validations. Third place goes to SE-Unet, which trails by 7% in terms of average performance, but which presents the lowest standard deviation. Fourth place goes to ResFCN whose average value is very close to that of SE-Unet but with less stability. The FCN8s rank last, only surpassing ResFCN for V1. The main observation to consider here is the result obtained for V3. Table 6.14 shows that this method can detect 90 (a total of 97) polyps in V3 but Table 6.16 shows the detection with less precision than the other methods.

Table 6. 16 The precision metric for each method (see section 5.6).

Network	V1	V2	V3	V4	Mean	Standard Deviation
FCN8s	0.7528	0.6724	0.5346	0.7471	0.6767	0.1016
ResFCN	0.7382	0.7560	0.6588	0.8288	0.7454	0.0698
Dilated ResFCN	0.7865	0.8336	0.7467	0.9035	0.8176	0.0674
SE-Unet	0.7594	0.7744	0.6813	0.7758	0.7477	0.0449
Hybrid	0.7865	0.8573	0.7925	0.9035	0.8350	0.0588

Table 6. 17 The recall metric for each method (see section 5.6).

Network	V1	V2	V3	V4	Mean	Standard Deviation
FCN8s	0.7222	0.6132	0.5015	0.7729	0.6524	0.1207
ResFCN	0.7118	0.7584	0.6368	0.8532	0.7401	0.0906
Dilated ResFCN	0.8298	0.813	0.6878	0.8927	0.8058	0.0858
SE-Unet	0.6278	0.7604	0.7124	0.7523	0.7132	0.0607
Hybrid	0.8298	0.8338	0.7277	0.8927	0.8210	0.0685

The second metric used recall rate (Table 6.17). This metric could be used as indicator of under segmentation. Ranked first in this category are Hybrid and the Dilated ResFCN. Third place goes to ResFCN, but it is ranked 4, 4, 4, and 3 for the four sub-validation results. It is only its stronger results for V4 that improved its ranking. This high level of instability is reflected in its standard deviation. The fourth place is SE-Unet, which ranks second in V3. Although it lost more targets than FCN8s (Table 6.14), it generated far more effective results than FCN8s and remained the most stable.

The above two tables further prove that Hybrid's method is superior when correspond to other tested methods. The tables also show that the use of SE-Unet as a secondary segmentation was quite reasonable, as it detected the polyp missed by the Dilated ResFCN. The FCN8s' false positive and false negative are very high, resulting in very few effective segmentation results. SE-Unet is not only superior to FCN8s for these two metrics but also exhibits considerable levels of stability in general.

Table 6. 18 The Hausdorff Distance for each method (see section 5.6).

Network	V1	V2	V3	V4	Mean	Standard Deviation
FCN8s	101.0584	273.6480	164.1729	233.2572	193.0341	76.1774
ResFCN	148.9207	365.0675	129.5731	161.0291	201.1476	110.0451
Dilated ResFCN	84.8172	46.5405	52.1660	34.4450	54.2422	21.6548
SE-Unet	147.6407	83.2507	109.7959	97.1535	109.4602	27.6662
Hybrid	84.8172	49.2468	73.2521	39.4450	61.6903	20.9620

Table 6.18 shows the Hausdorff distance obtained for each method. If the polyp is not detected, the metric returns Inf as a value of the Hausdorff distance, making it impossible to calculate the average. Therefore, we could only evaluate the shape of the detected polyos. The Dilated ResFCN takes first place with mean values clearly superior to those of the other methods. Hybrid takes second place, it finds more polyps but with potentially inaccurate contours. In addition, unlike image detection, image segmentation can fit the shape of a polyp. From this point of view, the advantages of the Dilated ResFCN are undeniable.

6.5.4 Data augmentation ablation tests

This section investigates various proposed data augmentations methods on the Dilated ResFCN architecture. Table 6.19 shows the mean Dice index obtained on each cross-validation fold along the overall mean dice index averaged across all the four folds. It can be seen the rotation is the most useful data augmentation for Dilated ResFCN. Deformation and colour jitter provide similar Dice index which is somewhat smaller than the one obtained for rotation. It is also evident that the combination of different augmentation methods improves overall performance.

Table 6. 19 Mean Dice index obtained on 4-fold validation data using Dilated ResFCN network.

Network	V1	V2	V3	V4	Mean
Combination	0.7583	0.7420	0.6086	0.8518	0.7402
Rotation	0.7602	0.7146	0.6145	0.8361	0.7314
Deformation	0.6772	0.7058	0.5917	0.7483	0.6807
Color jitter	0.6241	0.6957	0.5696	0.8019	0.6728
Scale	0.6536	0.6368	0.4742	0.7817	0.6366

6.5.5 Significance test

In the experiment reported above, each method is evaluated using different metrics by calculating the corresponding means and standard deviation on population of validation images. These results may reflect the performance of the methods to some extent but are not completely reliable. The dataset used only accounts for part of the whole population, and mean values and standard deviations be affected by a random selection of the test sample. Therefore, the method ranking based on these evaluations involve a certain level of risk.

The experiment reported here is inspired by [7]. To eliminate related risks, we carried out a significance tests of the five segmentation methods based on Friedman test [85]. This method is a nonparametric test method that can compare multiple methods where the distribution of samples is not required to satisfy the normal distribution. However, it requires all test groups submit same number of results. The significance of the reported results for metrics are tested.

The five methods (FCN8, ResFCN, Dilated ResFCN, SE-Unet and Hybrid method) were compared. Friedman test results are ranked in descending order (i.e. the smallest mean rank is the best one). The original H0 hypothesis is: there is no significant difference between the tested methods. The alternative hypothesis H1 is: there is a significant difference between the two methods. The significance level is set to 0.05. The p-value of each method is shown below:

Table 6. 20 p-value: Dice.

	FCN8s	ResFCN	Dilated ResFCN	SE-Unet	Hybrid
FCN8s	1	0.000138	1.19e-7	3.58e-7	1.19e-7
ResFCN		1	1.19e-7	0.757487	1.19e-7
Dilated ResFCN			1	1.19e-07	0.941865
SE-Unet				1	1.19e-7
Hybrid					1

Table 6. 21 p-value: Precision.

	FCN8s	ResFCN	Dilated ResFCN	SE-Unet	Hybrid
FCN8s	1	9.92e-09	9.92e-09	9.92e-09	9.92e-09
ResFCN		1	9.92e-09	0.988121	9.92e-09
Dilated ResFCN			1	9.92e-09	0.927061
SE-Unet				1	9.92e-09
Hybrid					1

Table 6. 22 p-value: Recall.

	FCN8s	ResFCN	Dilated ResFCN	SE-Unet	Hybrid
FCN8s	1	9.92e-09	9.92e-09	9.92e-09	9.92e-09
ResFCN		1	9.92e-09	0.898167	9.92e-09
Dilated ResFCN			1	1.15e-08	0.947832
SE-Unet				1	9.93e-09
Hybrid					1

These five methods can be divided into three groups. The first group consists of Dilated ResFCN and hybrid method. The second group consists of ResFCN and SE-Unet. The first group is FCN8s. There is significant difference between different groups. Figure 6.14 shows the mean rank of each method based on different measures.

Firstly, the hybrid method is the best method, because it obtained lowest mean rank. That means it always has higher precision, recall and Dice index than other methods. However, since the segmentation result of hybrid and Dilated ResFCN results are same, there is no significant difference between them. Especially for V1 and V4, the hybrid and Dilated ResFCN results are completely coincident. This coincidence creates a strong correlation between them.

In the second group, the resulting p value shows no significant difference between them. In terms of mean ranking, most intervals of the two methods overlap, and SE-Unet is slightly better than ResFCN. However, average Dice and Recall values indicate that ResFCN is better than SE-Unet (Table 6.23), which is the opposite of what is reflected by the mean rank.

This shows that SE-Unet generates better results than ResFCN in the four cross validations. When adopting one of these two networks for polyp segmentation, the ranking is more reasonable than the average. After testing, Dice, Precision and recall values do not satisfy the normal distribution, and so their average value cannot reasonably predict their distribution. Rankings can reflect the difference between a method and other methods when processing different images. It is easier to understand the type of polyps that a given method is good at segmenting, acting as an estimate of the total distribution. For the above reasons, while the two methods do not significantly differ, SE-Unet's performance is slightly better than ResFCN.

Finally, the tested methods are significantly different from FCN8s (marked in blue), and their mean ranking are better than those of FCN8s. This difference shows that the improved results of the proposed methods are not attributable to accidental factors.

Table 6. 23 The mean value of ResFCN and SE-Unet (The summary of Table 6.13, 6.16, 6.17).

	Dice	Precision	Recall
ResFCN	0.7085	0.7454	0.7401
SE-Unet	0.6969	0.7477	0.7132

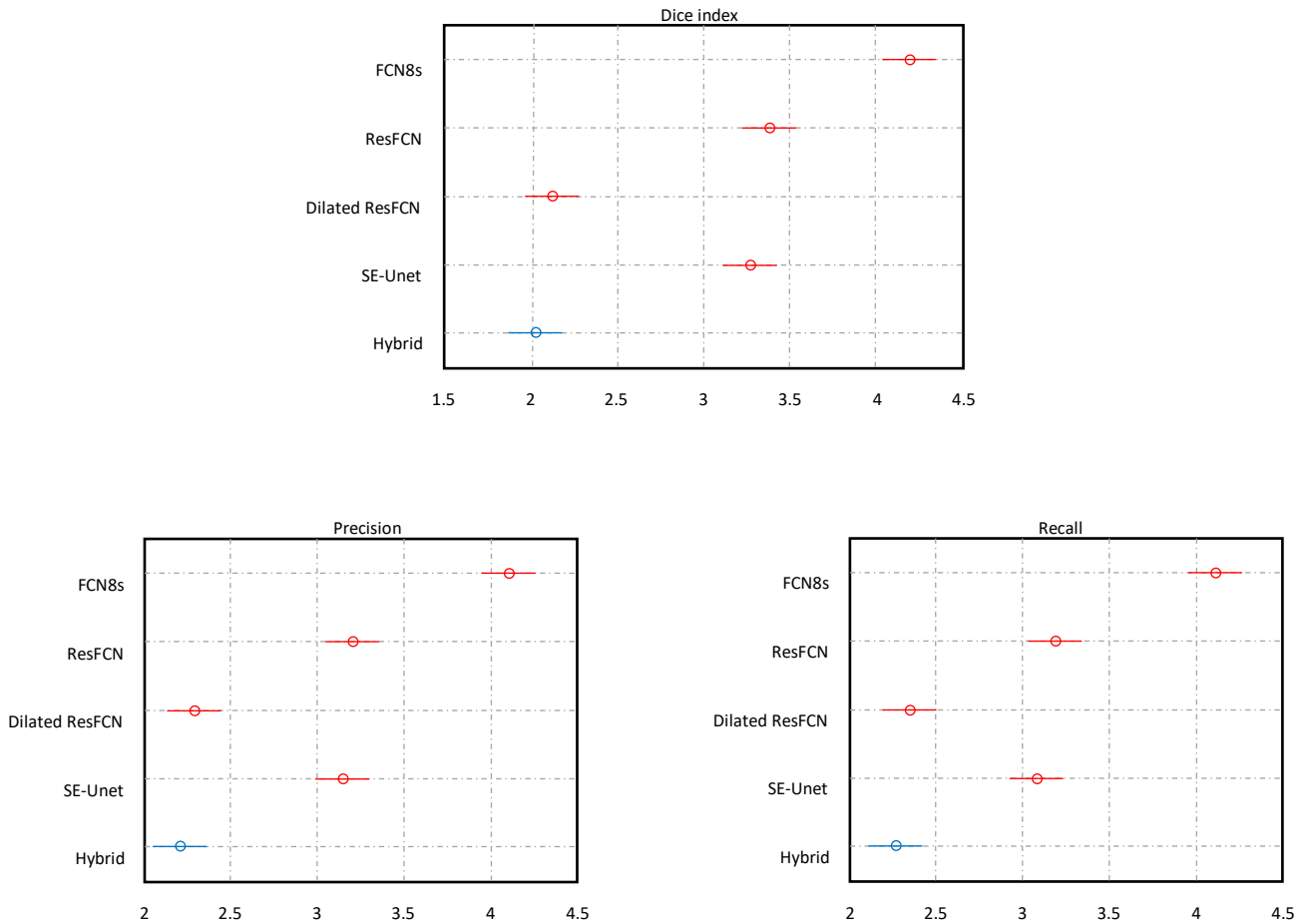


Figure 6. 14 The mean rank of each method, with the blue segment indicating the best result.

6.6 Results on testing dataset

This section introduces segmentation results obtained for the test data. The Dilated ResFCN and Unet were tested on the GIANA polyp segmentation challenge organized as part of the MICCAI'2017 and MICCAI'2018. Although the ground truth images for the test data are not published, the evaluation results for that data were obtained through the challenge.

Table 6. 24 Results obtained on the test data using different architectures and networks outputs.

	Foreground			Background			With test-time data augmentation		
	Mean	Std	Missing	Mean	Std	Missing	Mean	Std	Missing
Dilated ResFCN	0.7717	0.2394	17	0.8126	0.2043	9	0.8293	0.1956	9
SE-UNet	0.8019	0.2240	14	-	-	-	0.8102	0.2207	13
Hybrid	0.7825	0.2204	6	0.8169	0.1904	4	0.8343	0.1837	3

Table 6. 25 The definition of different qualitative measures of the segmentation accuracy.

Level	The corresponding range of Dice index
Very Bad	[0, 0.04)
Bad	[0.4, 0.6)
Normal	[0.6, 0.8)
Good	[0.8, 0.9)
Very good	[0.9, 1]

6.6.1 Test data results

Table 6.24 shows the standard deviation and missing data for results generated by the foreground, background and test time augmentation. The best results, highlighted in blue, have been achieved using the hybrid method with averaged Dice index of 0.8343, standard deviation of 0.1837 and only three polyps missed.

Figure 6.15 shows more detailed results. The results are divided into five levels (Table 6.25). This figure also shows that the use of background confidence map and the rotation testtime augmentation improves the results of the Dilated ResFCN. After these two operations, the number of “very bad” was reduced by 50%, with only 23 segmented polyps in that category. This demonstrates that the use of background confidence map and the test-time augmentation is important and necessary to improve overall results.

The example of segmentation results for the hybrid approach are shown in Figure 6.16 with the blue counter representing obtained segmentation results. From this image it can be concluded that for the “very good” segmentation, i.e. for the Dice index

value between 0.9 and 1, there is no obvious errors in the segmentation results. The position and shape of the contour correctly represents detected polyps. Some polyps are so “flat” that they can be easily confused with the adjacent tissue, however the Dilated ResFCN still can correctly outline them (e.g., images 271, 434 and 464).

When considering the “good” results with the Dice index between 0.8 and 0.9 (see Figure 6.17), two issues are identified in the segmentation results. The first refers to small and separated false positives (e.g., image 157). The second concerns inaccurate contours. Some results show that segmentation contours cannot completely fit polyp shapes. This misfit results in false positive and false negative detections. However, this type of result is still acceptable. The two problems can be solved through the use of larger training datasets or through additional post-processing.

For the “normal” results corresponding to the Dice index range of [0.6, 0.8), the results (see Figure 6.18) demonstrate that the segmentation error is visible to the naked human eye. Some post-processing may reduce such errors. In practical applications, detection a dedicated polyp detection network can be used to replace or aid the segmentation network. When considering the “bad” results corresponding to Dice in the region between 0.4 and 0.6, the main issue relates to missing data (see Figure 6.19). The Dilated ResFCN has missed some polyps that are though correctly detected and segmented by the SE-Unet – this result is marked with ‘*’. The correct detection of polyps cannot be guaranteed, as the CNN experiences difficulty in distinguishing polyps from the background. While post-processing could to some extent improve these values, a significant improvement is unlikely.

The results corresponding to the Dice index range of [0, 0.4] cannot be improved by conventional post-processing (see Figure 6.20). Three polyps were missed by the Dilated ResFCN and SE-Unet. The number of false positives and false negatives is always greater than the number of true positives. A possible improvement could be achieved by adding more polyps having similar characteristic to the training database.

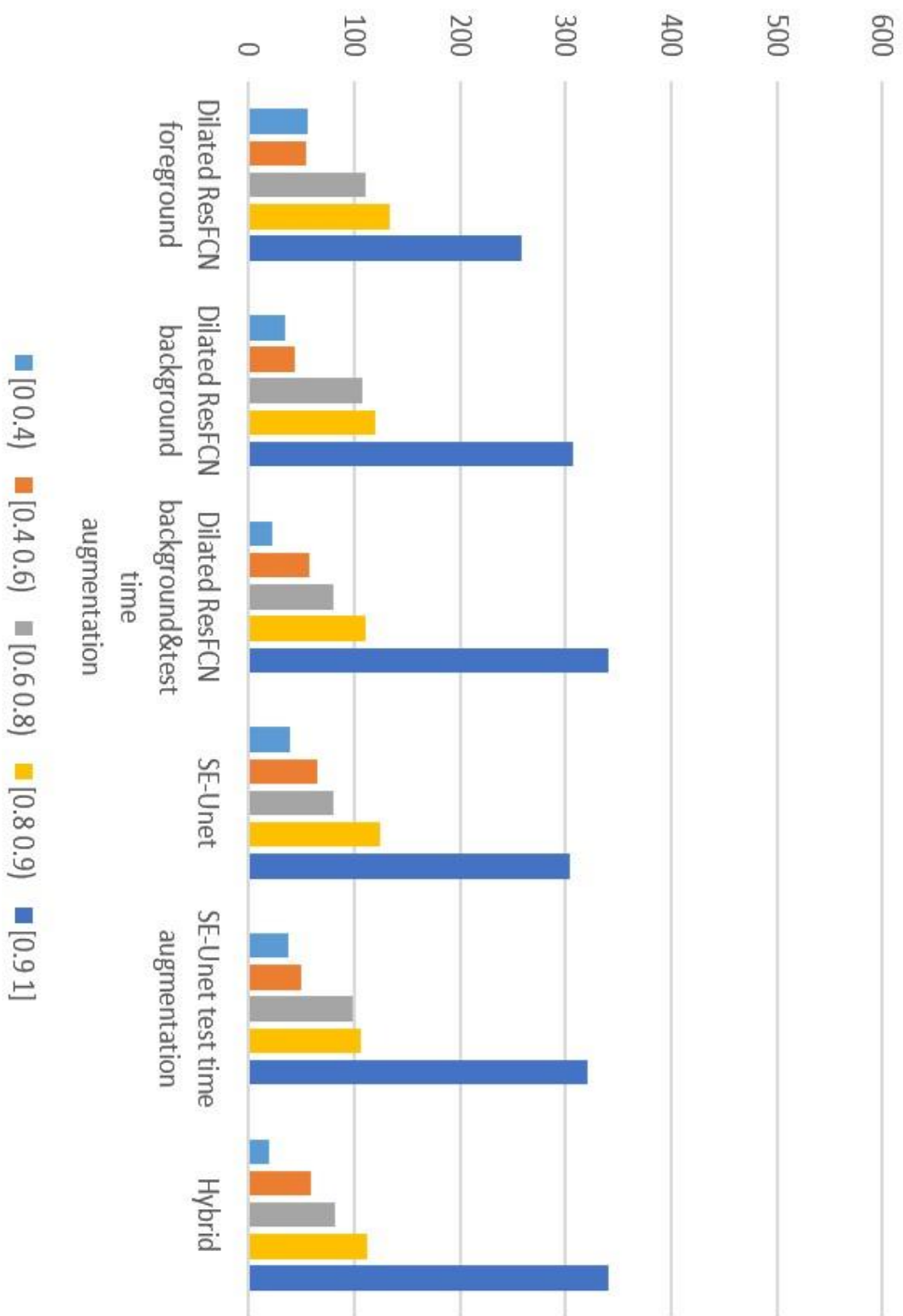


Figure 6. 15 The Dice of each method in different value regions.

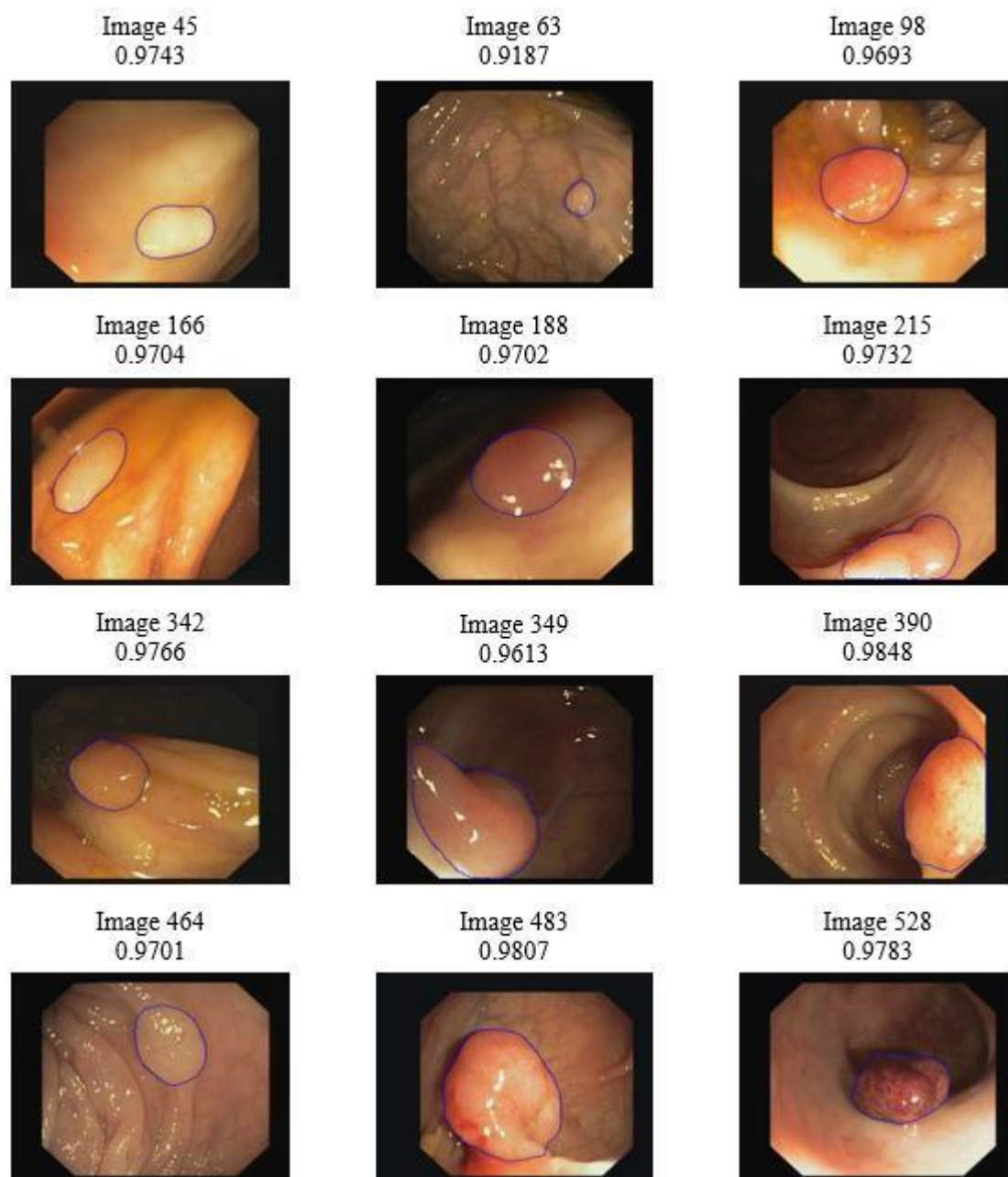


Figure 6. 16 Example of segmentation results obtained for SD images with the Dice index within the range of [0.9,1].

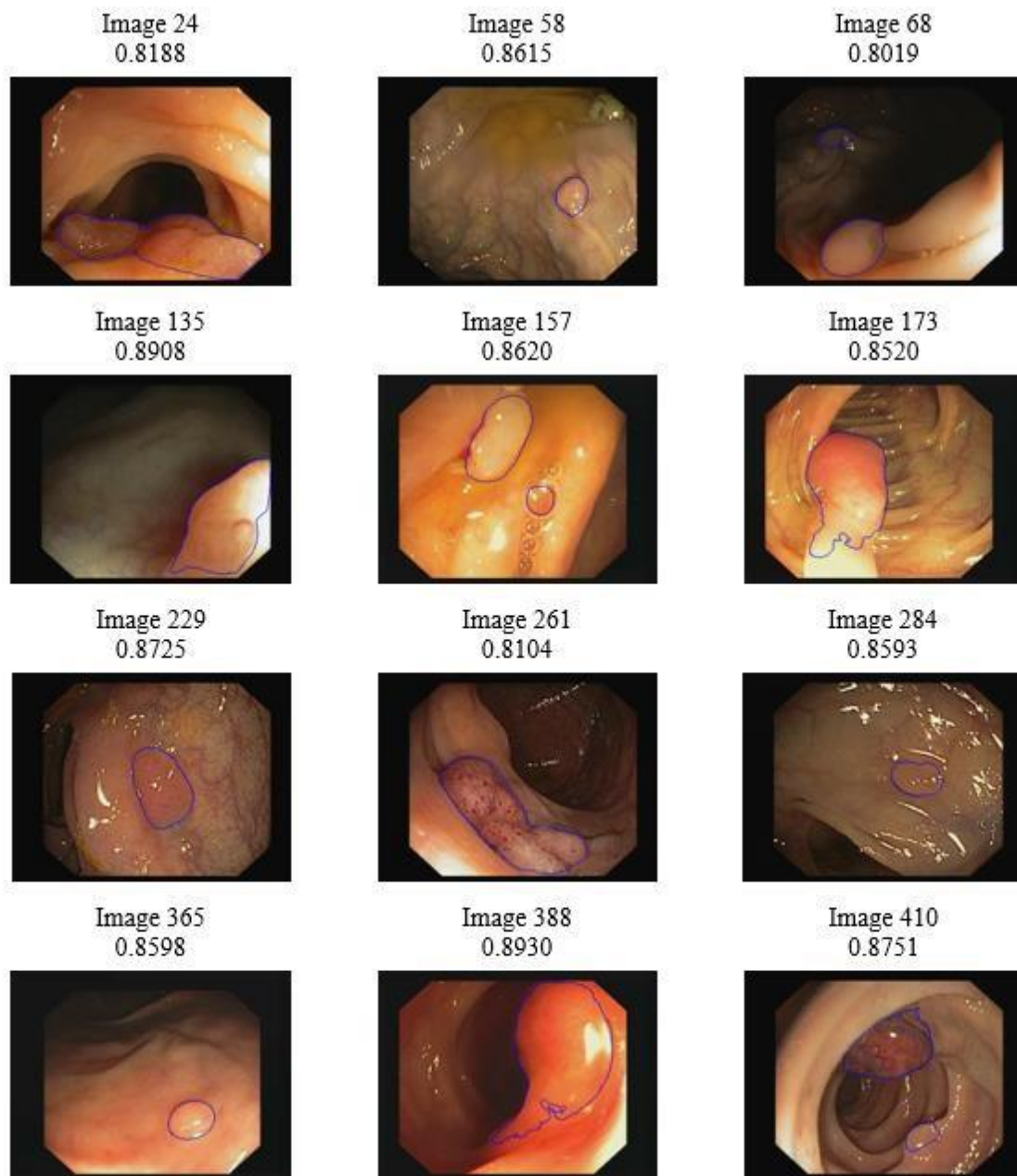


Figure 6. 17 Example of results obtained for SD images with the Dice index within the range of [0.8, 0.9).

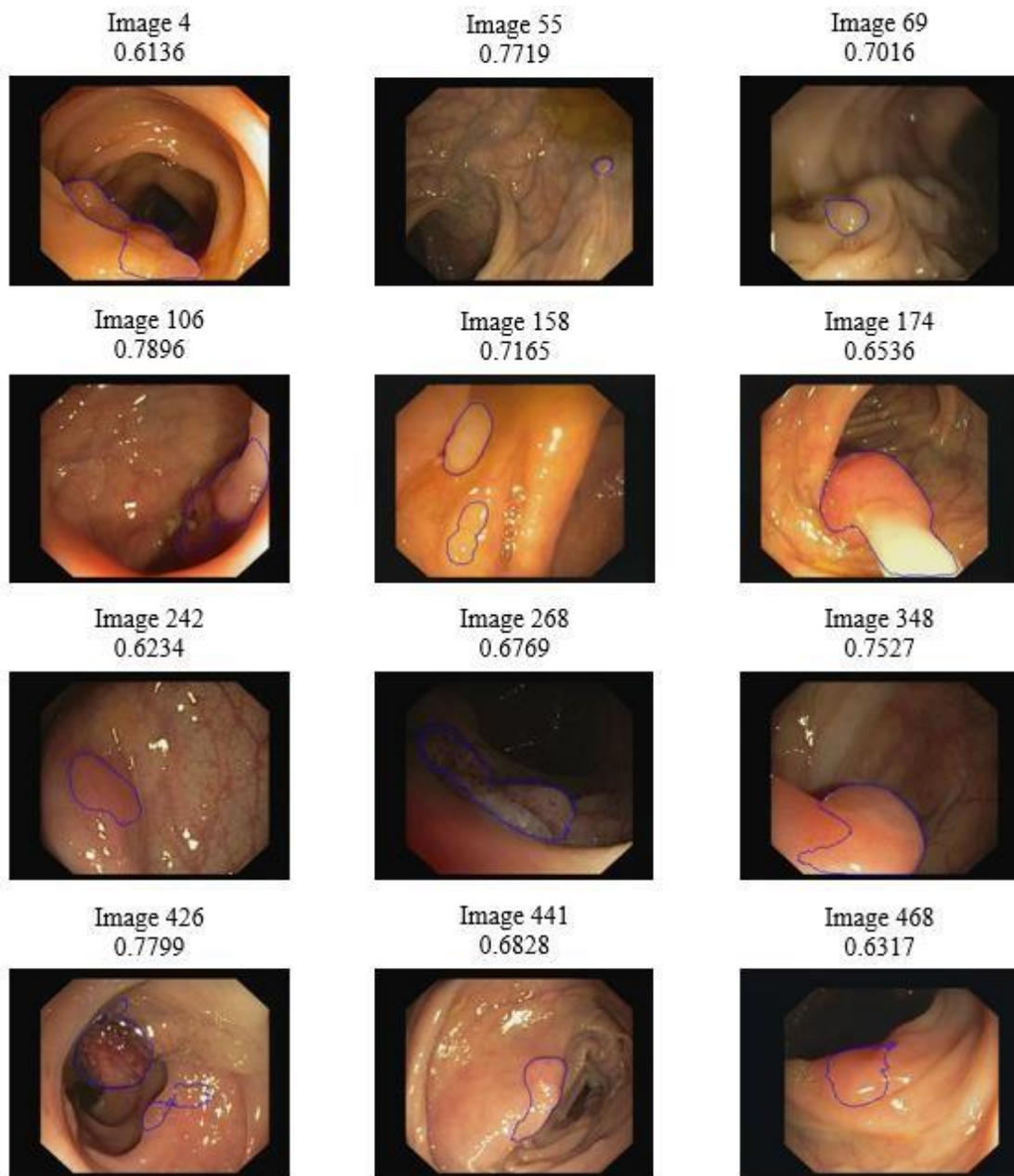


Figure 6. 18 Example of results obtained for SD images with the Dice index within the range of [0.6, 0.8).

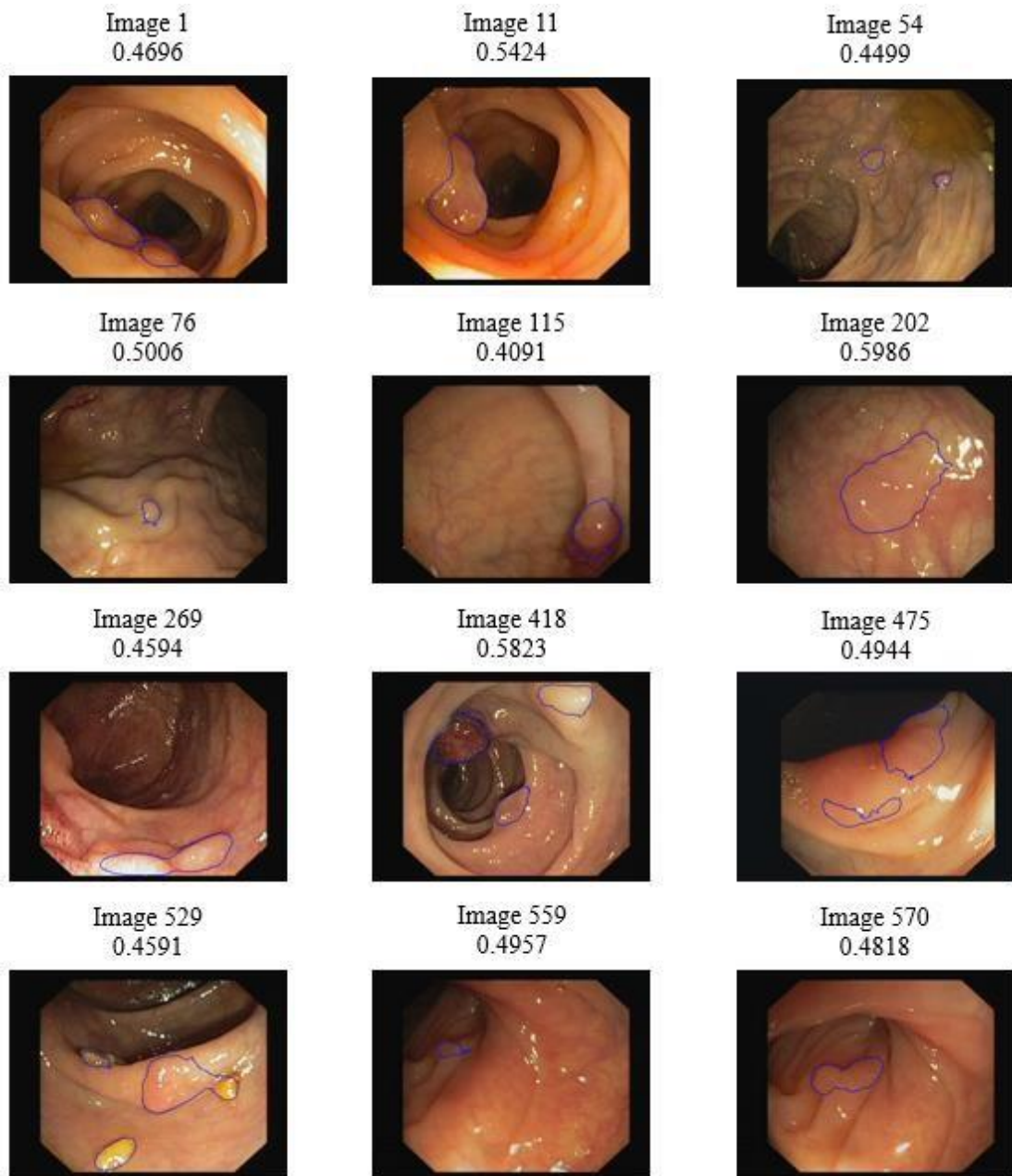


Figure 6. 19 Example of results obtained for SD images with the Dice index within the range of [0.4, 0.6].

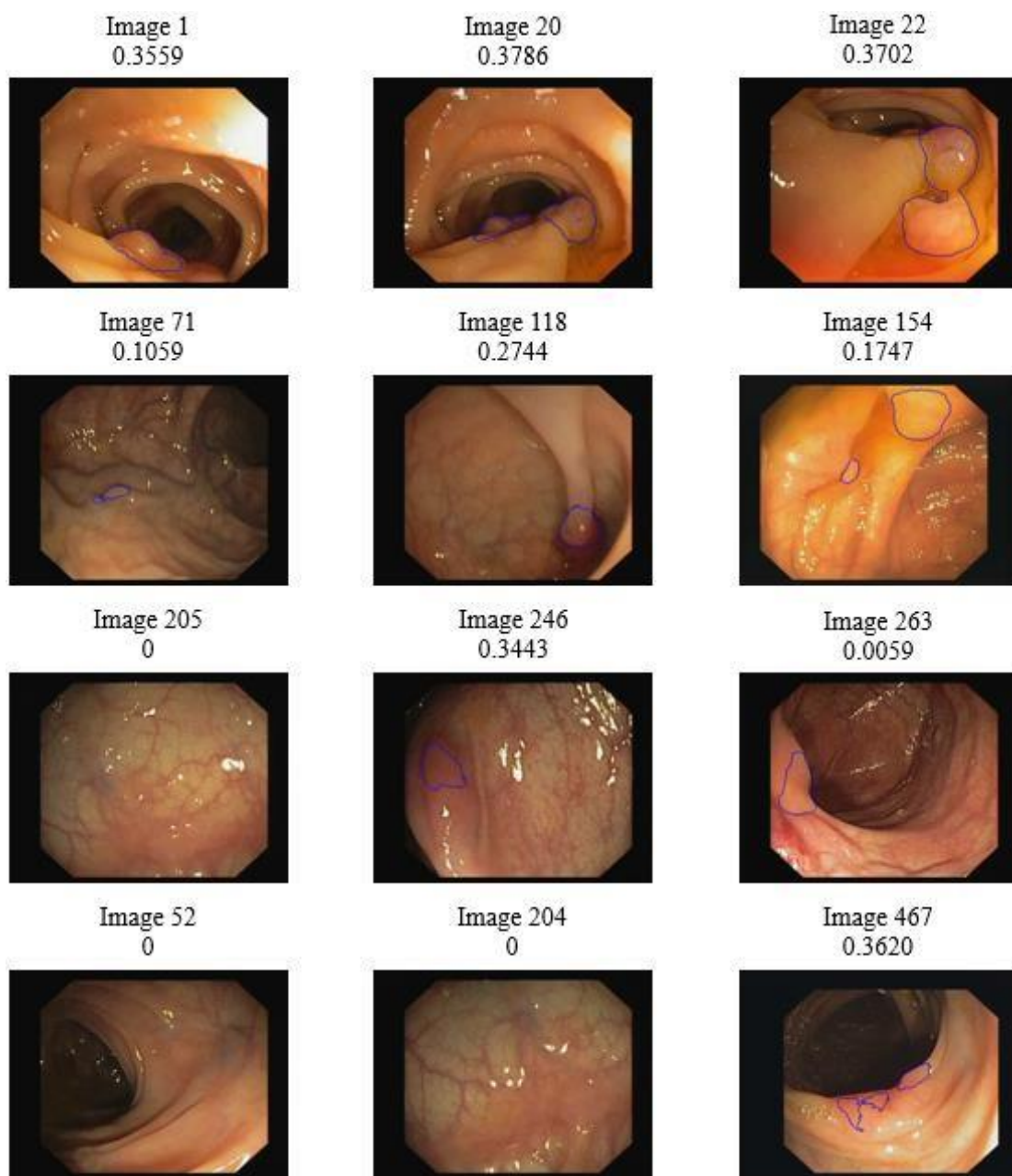


Figure 6. 20 Example of results obtained for SD images with the Dice index within the range of $[0, 0.4)$.

Table 6. 26 Ranking of the SD segmentation task.

Method	Jaccard		Dice index	
	Mean	Std	Mean	Std
CVML	0.72	0.22	0.81	0.21
Team 2	0.61	0.24	0.72	0.25
Team 3	0.67	0.25	0.77	0.22
Team 4	0.27	0.21	0.39	0.25
Team 5	0.47	0.35	0.54	0.38
Team 6	0.48	0.29	0.59	0.32

Table 6. 27 Ranking of the HD segmentation task.

Method	Jaccard		Dice index	
	Mean	Std	Mean	Std
CVML	0.74	0.20	0.83	0.18
Team 2	0.40	0.25	0.52	0.28
Team 3	0.70	0.24	0.80	0.20
Team 5	0.64	0.28	0.73	0.28
Team 6	0.53	0.24	0.39	0.23

6.6.2 Ranking of submitted results in the second GIANA challenge

For the 2017 GIANA challenge, the task was to segment SD and HD polyp images. Six teams participated in the SD image segmentation task and five teams participated in the HD segmentation task. As submitted, the results consisted of binary polyp segmentation maps for each test image. The submitted binary maps were used by the challenge organisers to calculate the mean values and standard deviations of the Dice and Jaccard indexes, so in total eight sub-evaluations were conducted.

Our results were generated using two stages. First the Dilated ResFCN network was employed to generate a probability map for each test images. Subsequently these probability maps were segmented using hybrid level set [86], [87]. The ranking for this

approach is shown in Table 6.26 and 6.27, with our submission clearly providing the best segmentation results. It should be noted that the use of the level set in the postprocessing did not have a significant effect on the resulting Dice/Jaccard metrics but was nevertheless implemented as it provide a convenient mechanism to control smoothness of the estimated polyp contours.

For the HD image segmentation task, our methods performed the best across all sub evaluations. Our methods are thus superior and more stable than the other submitted methods.

6.6.3 Ranking of submitted results for the third GIANA challenge

For the 2018 GIANA challenge, twelve and eleven teams participated in SD and HD segmentation tasks, respectively, using the same test images as those of the previous challenge. During that challenge our results were generated using four stages:

1. Application of the Dilated ResFCN with rotation-based test-time augmentation.
2. Application of the SE-Unet with rotation-based test-time augmentation.
3. Applying threshold-based segmentation to the obtained probability maps.
4. In case Dilated ResFCN network did not detect any polyps the SE-Unet was used and the resulting segmentation was return as the final result.

Regarding SD tasks, our teams were ranked in second place.

6.6.4 Comparison of segmentation methods

The purpose of this section is to provide a quantitative comparison of the methods proposed in the thesis with other reported polyp segmentation methods. However, such a comparison is a very challenging task. Firstly, some published methods use polyp segmentation as a way to perform polyp detection and localization, and the quantitative results for this intermediate segmentation stage are not reported. Secondly, the implementation details for some of the methods are not provided (e.g.

values of the method's design parameters, a number of used image samples, or the training stopping condition), and therefore, it is difficult to reproduce evaluation results for these methods. Furthermore, the training and test data used by different methods are not always the same. This makes it impossible to compare their performance under the same test conditions. Finally, when reporting on the methods' performance, some of authors use test data which have been already used for training, therefore such evaluation results do not reflect the real performance of these methods.

The comparison only includes polyp segmentation methods for which Dice index evaluation results have been published (the averaged Dice index is selected as a metric for methods comparison).

Table 6.28 lists the results of the comparison for the selected methods. The rows in yellow represent cases where the training and test data do overlap, whereas the blue rows represent uncertain cases when it is not clear if the same data was used for training and testing. Some of the methods reported in the table provide results for different configurations. In these cases, the best performing configuration is indicated by the average Dice index in bold, subsequently these configurations are used in a direct comparison with the proposed method. The gPb-OWT-UCM and Depth of Valleys methods are the only methods which use handcrafted features.

The two reported handcrafted feature-based method have the worst performance as measured by the mean Dice index and therefore are excluded from further consideration. The remaining methods use various deep learning approaches. It can be seen that the proposed hybrid method has the best performance. The second-best method is Multiple Encoder-Decoder network (MEDN), with a comparable performance. However, for the reported results the hybrid method used only 355 images for training, whereas the MEDN method used 612 for training. Furthermore, the mean Dice index results reported here used 612 test images, whereas reported MEDN results used only 196 test images. This situation illustrates that the hybrid method obtained better results under more adverse test conditions and therefore it

Table 6. 28 The comparison of existing polyp segmentation methods.

Methods		Dice indexes	Training Data	Testing Data	References
Proposed Hybrid methods		0.8343	CVC-ColonDB	CVC-ClinicDB ¹¹	
Proposed Dilated ResFCN		0.8293	CVC-ColonDB	CVC-ClinicDB	
FCN8s		0.810	CVC-ColonDB		[32]
		0.516	CVC-ColonDB, CVC-ClinicDB		[8]
ResNet-50 FCN8s		0.6906	CVC-ClinicDB	CVC-ColonDB	[88]
		0.3230	CVC-ClinicDB	ETIS-Larib	
	Resized testing image	0.4623	CVC-ClinicDB	ETIS-Larib	
		0.5853	CVC-ColonDB	CVC-ClinicDB	
	Pre-processing	0.6787	CVC-ClinicDB	CVC-ColonDB	
Mask-RCNN	ResNet50	0.716	CVC-ColonDB	CVC-ClinicDB	[89]
	ResNet50	0.804	CVC-ColonDB ETIS-Larib		
	ResNet101	0.704	CVC-ColonDB		
	ResNet101	0.775	CVC-ColonDB ETIS-Larib		
Multiple Encoder-Decoder network		0.889	CVC-ClinicDB		[90]
		0.829	CVC-ClinicDB	ETIS-Larib	
gPb-OWT-UCM		0.52~0.53	St. Marks Hospital and Academic Institute, Oxford Radcliffe Hospitals, Indiana University		[17]
		0.44~0.49			
Depth of valleys		0.55	CVC-ColonDB		[9]

¹¹ CVC-ColonDB and CVC-ClinicDB are described in section 5.2

could be expected that if the test condition would be the same for the both methods the proposed hybrid methods would outperform the MEDN method by a large margin.

The Dilated ResFCN, ResNet50 FCN8s and Mask-RCNN are all created based on ResNet50 for the deep feature extraction. As with the Multiple Encoder-Decoder network, the latter two methods are trained based on the database with more colonoscopy images. However, the proposed Dilated ResFCN is still the best method. This could be because: (i) the design of Dilated ResFCN is more suitable for polyp segmentation, and (ii) the proposed image pre-processing and augmentation methods used with the Dilated ResFCN are better than the ones used with the other methods.

Finally, it can be seen from the results that although the Mask-RCNN: ResNet-101 has a deeper architecture than ResNet-50, its segmentation performance is worse. This is consistent with the experimental results in this thesis (section 6.5.1), and this further confirms the rationale behind the selection of the ResNet50 as the base for the feature extraction subnetwork for the proposed Dilated ResFCN network.

Chapter 7. Summary, contributions and future work

This section summarizes the research reported in this thesis and lists the key contributions made as part of the development of new deep learning networks proposed in the thesis to solve the polyp segmentation problem. In addition, possible future advances resulting from this work and other deep learning methods for polyp segmentation are discussed.

7.1 Summary

This thesis provides a set of solutions for implementing polyp segmentation on small training data sets. This includes the method for removing the noise in the border of polyp images. The original dataset, with 356 training images, is expanded to 90,000 images using a number of augmentation techniques, so that the deep learning algorithm can obtain enough data to complete training.

Subsequently, in Chapter 5, the polyp images are examined by the K-means clustering method. This analysis finds that the successful completion of the polyp segmentation task is dependent on the solution of two problems. The first one is the

under-segmentation caused by intra-class differences, such as non-uniform polyp appearance caused by varying illumination. The second one is the over-segmentation caused by inter-class similarities, such as comparable colour distributions. To solve these two problems, two polyp segmentation convolutional neural networks are proposed: Dilated ResFCN and SE-Unet, which are based on a fully convolutional network model. Regardless of whether these two proposed methods work independently or are combined, their accuracy, robustness and effectiveness are better than FCN8s.

Chapter 6 investigates different metrics and evaluates the performance of the proposed methods from different perspectives. In order to further reduce the impact of accidental factors on assessment of the segmentation results, statistical significance tests are carried out to examine the rank of various segmentation methods. It is shown that the proposed methods outperform the previously proposed ones. These tests confirm the results from the GIANA 2017 and 2018 challenges (held at the Medical Image Computing and Computer Assisted Interventions conferences) where Dilated ResFCN achieved the best results for the SD and HD polyp segmentation task (at the GIANA 2017) and the hybrid method gained the second place for SD polyp segmentation (at the GIANA 2018).

The primary novel contributions reported in this thesis are the two end-to-end trained networks for polyp segmentation. The Dilated ResFCN has a larger receptive field due to implemented dilated convolutions. The purpose of this approach is to alleviate polyp under- and over- segmentation problems identified for the FCN8s. The SE-Unet enhances the detection of small flat polyps, through multi-resolution feature fusion. This type of polyp could be missed by the Dilated ResFCN network. When these two networks are combined more types of polyps can be successfully segmented. Furthermore, these networks can be efficiently deployed on a standard desktop computer, with an affordable graphics card (e.g. GTX1080) allowing for real-time image segmentation.

The validity of the proposed networks has been justified and tested against other

reference polyp segmentation methods. For these tests a number of metrics have been used, including: Dice index, precision, recall, polyp detection false negative (missed polyps), and Hausdorff distance, with some of these metrics used in the context of polyp segmentation for the first time. The performed comparisons have demonstrated that the proposed hybrid approach outperforms other methods. To decrease the risk in method ranking caused by a small test dataset, the statistical Friedman test has been used to further validate significance of the results. The significance of different data augmentation methods has been evaluated using comparative ablation tests.

The main finding reported in this thesis can be summarized as follows:

- The ablation tests have demonstrated that rotation, local deformation and colour jitter are the most important augmentation techniques.
- The use of all the augmentation methods significantly improves the performance of the method.
- Dilation kernels can improve performance of polyp segmentation on multiple evaluation metrics. The value of Dice index, precision and recall has improved by 11%, 10% and 9% respectively. The Hausdorff distance has decreased by 73%.
- Based on the Friedman test, the performance of the proposed method is statistically significantly better than other assessed methods, whereas the performance of these other methods is statistically comparable with respect to each other.
- The proposed hybrid method (combining the Dilation ResFCN and SE-Unet methods) improves overall performance by performing better on the small polyps.
- The processing time of proposed Dilation ResFCN and SE-Unet are 0.05s and 0.45s respectively, therefore they can operate in real-time.
- The results indicate (Table 6.24, small number of missed polyps) that the proposed hybrid segmentation method can be also used for the polyp detection task.

- Dilated ResFCN matches the shape of the polyp well, with the smallest and most consisted value of the Hausdorff distance.
- The proposed techniques proved their performance by archiving 1st place at the 2017 GIANA challenge (SD and HD segmentation) and 2nd place at the 2018 GIANA challenge (SD segmentation).

7.2 Future work

Polyp detection and localization

Polyp segmentation provides a detailed description of polyp shape. However, clinicians are often more interested in identifying and locating polyps in images (i.e. outcomes of polyp detection and localization). Based on the performed tests, the segmentation methods proposed in this thesis certainly have potential to be adopted for polyp detection and segmentation. For example, for some of the test images the calculated Dice index (between segmentation results and the ground truth) is within 0.3 to 0.6, which is considered a poor segmentation result. However, from the detection and localisation perspective such values of the Dice index are sufficient to consider this an accurate polyp detection/localisation. Therefore, in future work, the proposed methods can form the basis for further development to solve polyp detection and localization problems. However, this would require network re-training and comprehensive testing on images without polyps to assess robustness with respect to false positive.

Addition of Temporal Information for Polyp Segmentation

The polyp images in the GIANA SD dataset are extracted from videos, and the same polyp is often visible in multiple consecutive frames. Therefore, it is possible to consider the correlation between the neighbouring frames. Such an approach would help to reduce the impact of various disturbances on the CNN. For example, the image frames at time $t-1$ and $t+1$ can be used as supplementary information for polyp segmentation in the image frame at time t .

This idea can be implemented by using two key approaches. The first one is to design a CNN with multiple input layers to receive multiple consecutive input images, and then merge their features (e.g. using a so call slow fusion) to segment polyps. The second approach is based on the so-called recurrent networks (e.g. LSTM) where the temporal information is directly learned.

New Kernel

The reported research has identified the need for dilated convolution where the dilation rate is determined by polyps' size. It should be noted that a fixed dilation rate may not be flexible enough to process images with a larger variability of polyp sizes. Therefore, the dilation rate can be considered as a trainable parameter, which is adjusted by the CNN based on the available data through the network training. Recently, two types of convolution kernels have been proposed which could be instrumental in achieving that objective, namely Learning Dilation Factors [91] and deformable convolution [92]. The spatial transformer network [93] also can be considered.

Developed hybrid method

The proposed hybrid method is a combination of two independent FCNs, namely: Dilated ResFCN and SE-Unet, which undoubtedly take up extra resources and time to complete training. Therefore, in the future work, it is worth considering fusing these two methods into a single FCN. The current idea is to use ResNet as the encoder. The feature maps of different resolutions are learned by dilated convolution or other new kernels. The features are then up-sampled and pixel-wise classified by the decoder in SE-Unet.

Transfer learning

The current deep learning framework often requires large GPU memory when training big CNNs, but they require relatively little memory in the prediction mode. Therefore,

it is possible to extract low-level or middle-level features using CNNs that have been trained on another database, and the CNN does not need to be retrained. Next, these features could be filtered and fused by a specific CNN structure. These operations can greatly reduce CNN's dependence on large available memory and speed up the training. The idea is currently immature, and it needs to be further investigated.

Publications

Korsuk Sirinukunwattana, Josien PW Pluim, Hao Chen, Xiaojuan Qi, Pheng-Ann Heng, Yun Bo Guo, Li Yang Wang, Bogdan J Matuszewski, Elia Bruni, Urko Sanchez, et al. Gland segmentation in colon histology images: the glas challenge contest. *Medical image analysis*, 35:489–502, 2017.

Yun Bo Guo and Bogdan J. Matuszewski. GIANA Polyp segmentation with fully convolutional dilation neural networks, In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 632-641, 2019.

Yun Bo Guo and Bogdan J. Matuszewski. Polyp Segmentation with Fully Convolutional Deep Dilation Neural Network Evaluation Study. In *23rd Conference on Medical Image Understanding and Analysis*, pages 384-395, 2019

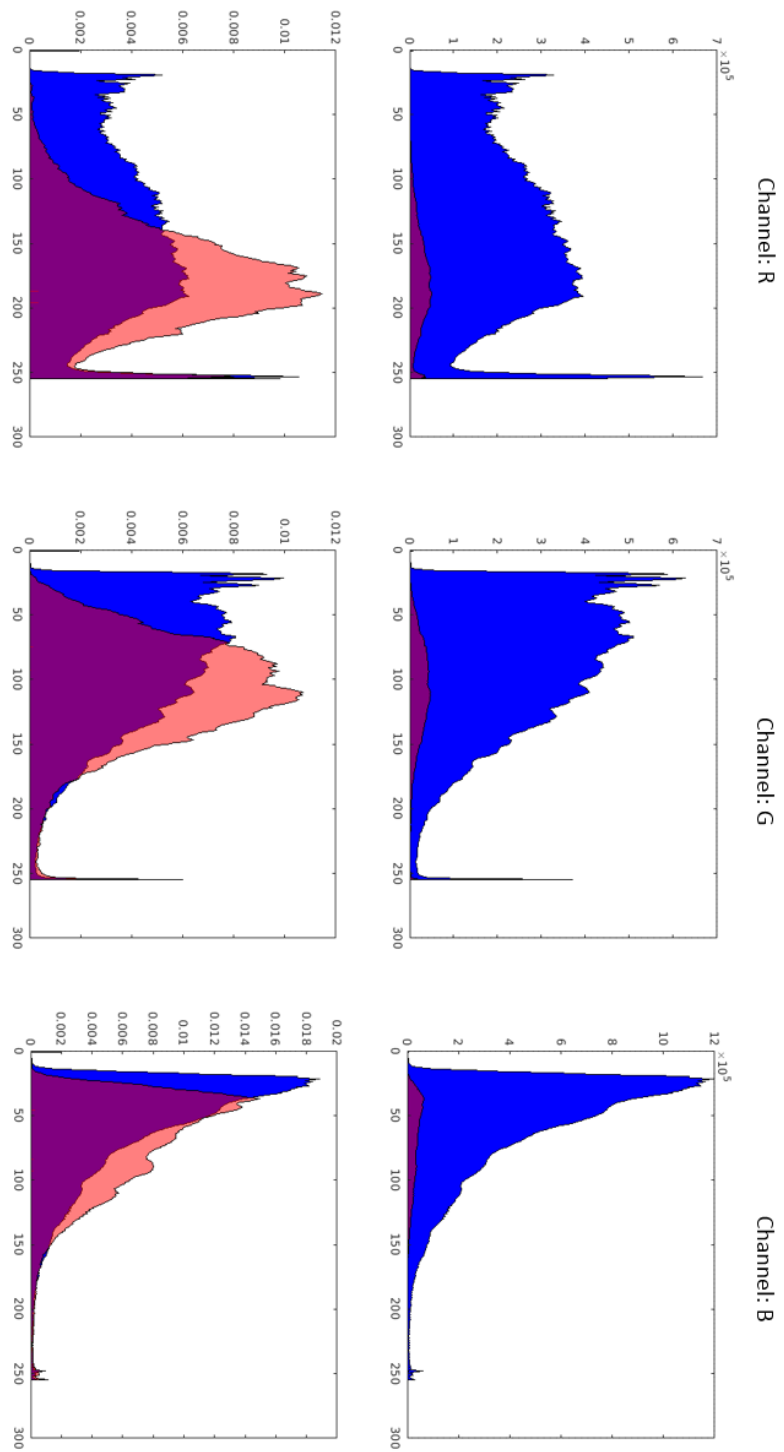
Challenge Participation

2015 GlaS@MICCAI'2015: Gland Segmentation Challenge Contest (5th place).

2017 Gastrointestinal Image ANALysis polyp segmentation challenge (1st place).

2018 Gastrointestinal Image ANALysis polyp segmentation (SD) challenge (2nd place).

Appendix A: Histogram of polyp (SD) and background.



Blue: Histogram of background (without border). Red: Histogram of polyp. Vertical axis (Top): The number of pixels for each value. Vertical axis (Bottom): The number of pixels divided by total number of pixel. Horizontal axis: Pixels value, 0~255.

Appendix B: ResNet-50 network

No.		Layer name	Layer type	Connection	Parameters	Output size	Output size (Seg)
0		Input	Input	NAN	NAN	224×224×3	250×287×3
1		Conv1	Conv	0	K:7×7×64, S:2, P:3	112×112×64	125×144×64
2		Pool1	Pool	1	K:3×3, S:2	56×56×64	62×72×64
3	Res2	Res2a branch1	Conv	2	K:1×1×256, S:1, P:0	56×56×256	62×72×256
4		Res2a branch2a	Conv	2	K: 1×1×64, S:1, P:0	56×56×64	62×72×64
5		Res2a branch2b	Conv	4	K: 3×3×64, S:1, P:1	56×56×64	62×72×64
6		Res2a branch2c	Conv	5	K: 1×1×256, S:1, P:0	56×56×256	62×72×256
7		Res2a	Fusion	3, 6	Sum	56×56×256	62×72×256
8		Res2b branch2a	Conv	7	K:1×1×64, S:1, P:0	56×56×64	62×72×64
9		Res2b branch2b	Conv	8	K: 3×3×64, S:1, P:0	56×56×64	62×72×64
10		Res2b branch2c	Conv	9	K: 1×1×256, S:1, P:0	56×56×256	62×72×256
11		Res2b	Fusion	7, 10	Sum	56×56×256	62×72×256
12		Res2c branch2a	Conv	11	K:1×1×64, S:1, P:0	56×56×64	62×72×64
13		Res2c branch2b	Conv	12	K: 3×3×64, S:1, P:0	56×56×64	62×72×64
14		Res2c branch2c	Conv	13	K: 1×1×256, S:1, P:0	56×56×256	62×72×256
15		Res2c	Fusion	11, 14	Sum	56×56×256	62×72×256
16		Res3	Res3a branch1	Conv	15	K:1×1×512, S:2, P:0	28×28×512
17	Res3a branch2a		Conv	15	K: 1×1×128, S:2, P:0	28×28×128	31×36×128
18	Res3a branch2b		Conv	16	K: 3×3×128, S:1, P:0	28×28×128	31×36×128
19	Res3a branch2c		Conv	17	K: 1×1×512, S:1, P:0	28×28×512	31×36×512
20	Res3a		Fusion	16, 19	Sum	28×28×512	31×36×512
21	Res3b branch2a		Conv	20	K:1×1×64, S:1, P:0	28×28×128	31×36×128
22	Res3b branch2b		Conv	21	K: 3×3×64, S:1, P:0	28×28×128	31×36×128
23	Res3b branch2c		Conv	22	K: 1×1×256, S:1, P:0	28×28×512	31×36×512
24	Res3b		Fusion	20, 23	Sum	28×28×512	31×36×512
25	Res3c branch2a		Conv	24	K:1×1×64, S:1, P:0	28×28×128	31×36×128
26	Res3c branch2b		Conv	25	K: 3×3×64, S:1, P:0	28×28×128	31×36×128
27	Res3c branch2c		Conv	26	K: 1×1×256, S:1, P:0	28×28×512	31×36×512
28	Res3c		Fusion	24, 27	Sum	28×28×512	31×36×512
29	Res3d branch2a		Conv	28	K:1×1×64, S:1, P:0	28×28×128	31×36×128
30	Res3d branch2b	Conv	29	K: 3×3×64, S:1, P:0	28×28×128	31×36×128	
31	Res3d branch2c	Conv	30	K: 1×1×256, S:1, P:0	28×28×512	31×36×512	
32	Res3d	Fusion	28, 31	Sum	28×28×512	31×36×512	
33	Res3	Res4a branch1	Conv	32	K:1×1×512, S:2, P:0	14×14×1024	16×18×1024

34		Res4a branch2a	Conv	32	K: 1×1×128, S:2, P:0	14×14×256	16×18×256
35		Res4a branch2b	Conv	34	K: 3×3×128, S:1, P:0	14×14×256	16×18×256
36		Res4a branch2c	Conv	35	K: 1×1×512, S:1, P:0	14×14×1024	16×18×1024
37		Res4a	Fusion	33, 36	Sum	14×14×1024	16×18×1024
38		Res4b branch2a	Conv	37	K:1×1×64, S:1, P:0	14×14×256	16×18×256
39		Res4b branch2b	Conv	38	K: 3×3×64, S:1, P:0	14×14×256	16×18×256
40		Res4b branch2c	Conv	39	K: 1×1×256, S:1, P:0	14×14×1024	16×18×1024
41		Res4b	Fusion	37, 40	Sum	14×14×1024	16×18×1024
42		Res4c branch2a	Conv	41	K:1×1×64, S:1, P:0	14×14×256	16×18×256
43		Res4c branch2b	Conv	42	K: 3×3×64, S:1, P:0	14×14×256	16×18×256
44		Res4c branch2c	Conv	43	K: 1×1×256, S:1, P:0	14×14×1024	16×18×1024
45		Res4c	Fusion	41, 44	Sum	14×14×1024	16×18×1024
46		Res4d branch2a	Conv	45	K:1×1×64, S:1, P:0	14×14×256	16×18×256
47		Res4d branch2b	Conv	46	K: 3×3×64, S:1, P:0	14×14×256	16×18×256
48		Res4d branch2c	Conv	47	K: 1×1×256, S:1, P:0	14×14×1024	16×18×1024
49		Res4d	Fusion	45, 48	Sum	14×14×1024	16×18×1024
50		Res4e branch2a	Conv	49	K:1×1×64, S:1, P:0	14×14×256	16×18×256
51		Res4e branch2b	Conv	50	K: 3×3×64, S:1, P:0	14×14×256	16×18×256
52		Res4e branch2c	Conv	51	K: 1×1×256, S:1, P:0	14×14×1024	16×18×1024
53		Res4e	Fusion	49, 52	Sum	14×14×1024	16×18×1024
54		Res4f branch2a	Conv	53	K:1×1×64, S:1, P:0	14×14×256	16×18×256
55		Res4f branch2b	Conv	54	K: 3×3×64, S:1, P:0	14×14×256	16×18×256
56		Res4d branch2c	Conv	55	K: 1×1×256, S:1, P:0	14×14×1024	16×18×1024
57		Res4f	Fusion	53, 56	Sum	14×14×1024	16×18×1024
58	Res4	Res5a branch1	Conv	57	K:1×1×512, S:2, P:0	7×7×2048	8×9×2048
59		Res5a branch2a	Conv	57	K: 1×1×128, S:2, P:0	7×7×512	8×9×512
60		Res5a branch2b	Conv	58	K: 3×3×128, S:1, P:0	7×7×512	8×9×512
61		Res5a branch2c	Conv	59	K: 1×1×512, S:1, P:0	7×7×2048	8×9×2048
62		Res5a	Fusion	58, 61	Sum	7×7×2048	8×9×2048
63		Res5b branch2a	Conv	62	K:1×1×64, S:1, P:0	7×7×512	8×9×512
64		Res5b branch2b	Conv	63	K: 3×3×64, S:1, P:0	7×7×512	8×9×512
65		Res5b branch2c	Conv	64	K: 1×1×256, S:1, P:0	7×7×2048	8×9×2048
66		Res5b	Fusion	62, 65	Sum	7×7×2048	8×9×2048
67		Res5c branch2a	Conv	66	K:1×1×64, S:1, P:0	7×7×512	8×9×512
68		Res5c branch2b	Conv	67	K: 3×3×64, S:1, P:0	7×7×512	8×9×512
69		Res5c branch2c	Conv	68	K: 1×1×256, S:1, P:0	7×7×2048	8×9×2048
70		Res5c	Fusion	66, 69	Sum	7×7×2048	8×9×2048
71		Pool5	Pool	70	K:7×7, S:1 P:0	1×1×2048	Removed
72		FC1000	FC	71	1×1×1000	1×1×1000	Removed

Appendix C: K-means clustering based HSV and lab colour space

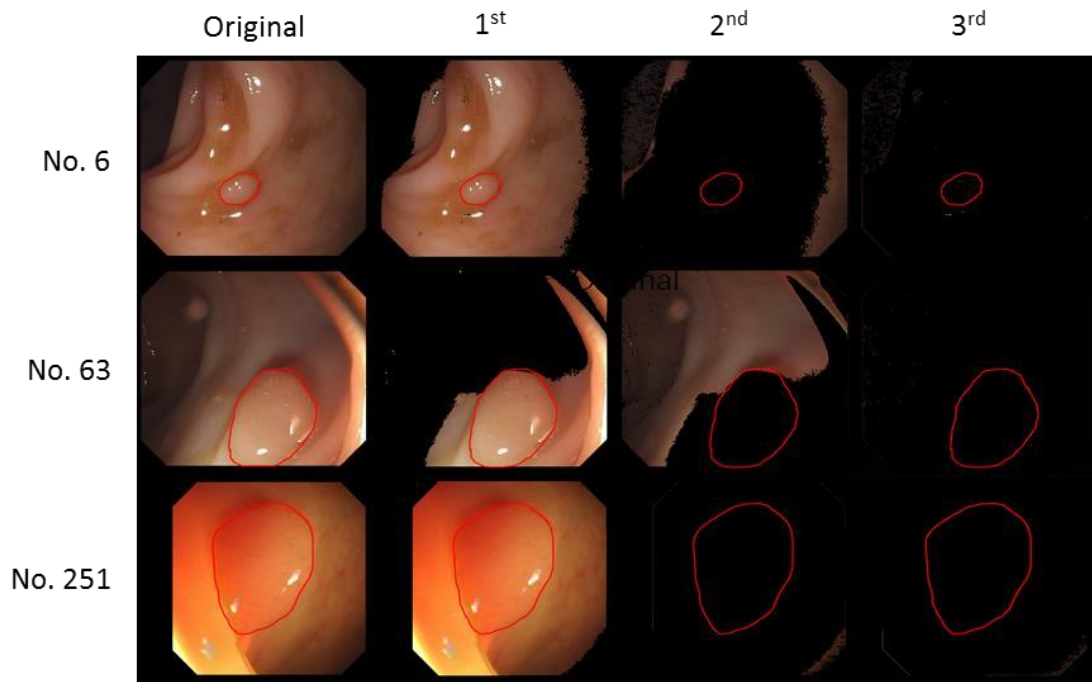


Image No.6, 64 and 251 and their clustered results with three cluster centres in HSV colour space.

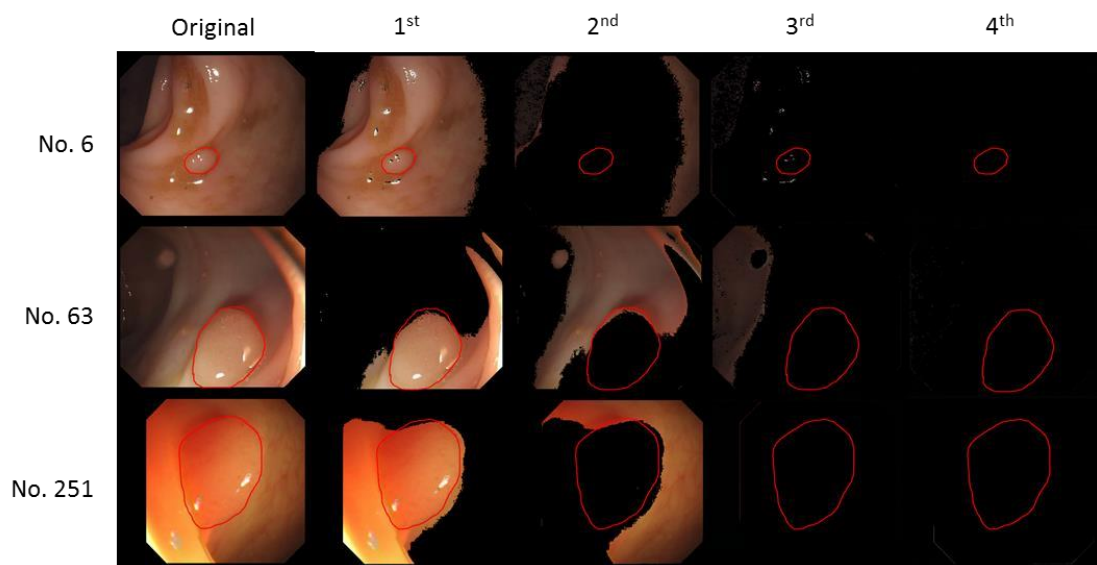


Image No.6, 64 and 251 and their clustered results with four cluster centres in RGB colour space.

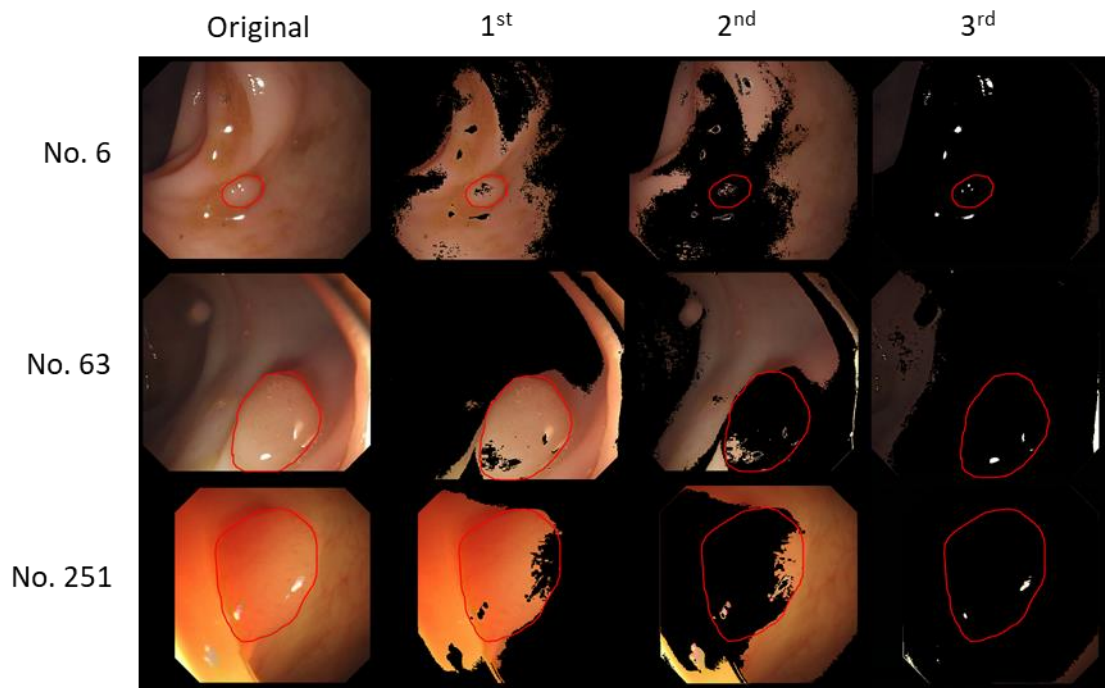


Image No.6, 64 and 251 and their clustered results with three cluster centres in Lab colour space.

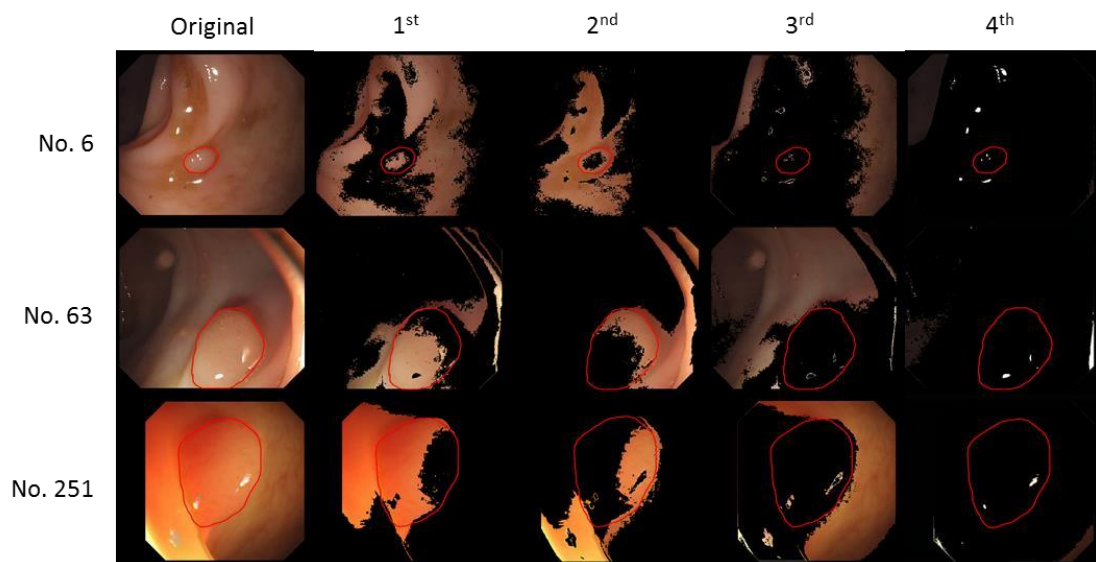


Image No.6, 64 and 251 and their clustered results with three cluster centres in Lab colour space.

Appendix D: Gradient descent algorithms

In recent years, some new gradient descent algorithms are proposed. Most of them aim to search for the minimum more quickly. This section first reviews the classical gradient descent, then introduces the representative new algorithms.

Classical gradient descent

The method employed above is the classical batch gradient descent (BGD) algorithm. In this algorithm, the parameters are updated using the average gradient of all training samples in each iteration. Its advantages are that the gradient direction is stable and accurate, and the convergence speed is fast. Its disadvantage is that all training samples are required for every parameter update; therefore, when the numbers of samples and parameters are large, considerable memory is needed to store the gradients, and the update efficiency is low.

Stochastic gradient descent (SGD) [94] refers to the random extraction of samples (without replacement) to update the parameters. Once all samples have been extracted, the next iteration is performed. It has the advantages of low memory requirements and a fast speed for backpropagation. In addition, SGD supports online learning, with the only requirement being to add the new samples to the training data. The disadvantage is that the direction of the gradient changes frequently during updating, so it is difficult to reach convergence at a local minimum.

Mini-batch gradient descent is a fusion of the two methods discussed above. The total samples are divided into several training subsets, each subset called mini-batches. Each time, a subset is randomly extracted to calculate the gradient and update the parameters. When all training subsets have been used, the next iteration begins.

Obviously, this method has two advantages. The first is that it offers great flexibility in choosing the appropriate batch size in accordance with the performance of the equipment being used. The second is that it is more stable than SGD. Therefore, it is a commonly used optimization method in deep learning.

Momentum

The momentum approach [95] can be used to stabilize the update direction of SGD by accumulating previous weighted gradients, which facilitates the model's convergence. Let the current parameter be θ_{t-1} and the updated parameter be θ_t ; then, the steps of the update process can be written as follows:

$$v_t = \gamma v_{t-1} + \eta \frac{\partial J(\theta)}{\partial \theta}$$

$$\theta = \theta_{t-1} - \gamma v_t$$

γ is a coefficient, whose value is usually set to 0.9. Therefore, v_t is actually a weighted sum of the previous gradient and the current gradient. This can be seen as imposing an inertia on the current gradient so that it does not deviate too much from the direction of previous updates, and the more similar it is to the previous direction, the faster the model converges; this is where this method gets its name. Next, v_t can be used to update θ_{t-1} , resulting in θ_t .

Nesterov accelerated gradient

The Nesterov accelerated gradient (NAG) method [96] was developed based on the momentum approach. For updating θ_{t-1} , this method considers not only the previous γv_{t-1} but also a temporary gradient $\hat{\theta}_t$ obtained along the same trend, which is derived via the following equation:

$$\hat{\theta}_t = \theta_{t-1} - \gamma v_{t-1}$$

This method is equivalent to taking a tentative step towards updating the parameters, obtaining $\hat{\theta}_t$ and then summing the gradient $\hat{\theta}_t$ and the previous gradient γv_{t-1} together to serve as the gradient needed for updating θ_{t-1} . This

approach can be expressed as follows:

$$v_t = \gamma v_{t-1} + \eta \frac{\partial J(\hat{\theta}_t)}{\partial \hat{\theta}_t}$$
$$\theta = \theta_{t-1} - \gamma v_t$$

With this additional update based on γv_{t-1} , the NAG method is more strongly affected by the previous gradient in the weighted sum than the momentum approach is, so the convergence speed for SGD is faster. Of course, this method also requires additional computation.

Adagrad

As mentioned above, as the number of iterations increases, the learning rate needs to be reduced to reduce the error, because too large a step will cross the local minimum. However, the update speed is different for each parameter, so the required learning rates are also not the same. In some large neural networks, the number of parameters often reaches tens of millions, so it is obviously impossible to set the learning rates manually. To this end, AdaGrad [97] was proposed to enable adaptive adjustment of the learning rate. The AdaGrad update formula is defined as follows:

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\sum_{i=1}^t (g^i)^2 + \epsilon}} g_t$$

The denominator in this formula is the square root of the sum of the squares of all previous gradients. The sum of the squares is calculated to avoid gradient offset. The square root is used to prevent the learning rate from decaying too fast due to an excessively large denominator. ϵ is a constant that is included to ensure that the denominator will not be zero. With an increasing number of iterations, the learning rate will gradually decrease. However, it should be noted that this reduction is not constrained. If the gradient accumulates too fast, it is likely that the model will not be

able to obtain enough gradient information and will never reach convergence or will need more iterations.

Adaptive moment estimation

Adaptive Moment Estimation (Adam) [98] is a fusion and further improvement of the RMSProp and momentum methods. It has two main features. First, it considers both the first moment (average) and the second moment (uncentred variance) of the previous gradient, with the former acting as a gradient and the latter acting as a denominator to cause the decay of the learning rate. As in RMSProp, the previous gradient is stored by means of an exponential average, i.e.,

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

The authors recommend that the values of β_1 and β_2 should be set to 0.9 and 0.999, respectively. The second feature of this method is that the deviations of the first and second moment are corrected when updating the parameters. This correction is because in the first iteration t , the values of m_{t-1} and v_{t-1} are 0, and the weight is large. This leads to a large deviation of m_t from g_t during the initial stage of training. Another parameter v_t has same problem. Therefore, adjustments are needed.

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\widehat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

Finally, the gradient and denominator of RMSProp are replaced with \widehat{m}_t and \widehat{v}_t , respectively, resulting in

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t$$

Ruder [99] compared the convergence rate of this method with those of other methods. When logistic regression is used to classify MNIST, Adam converges twice as fast as AdaGrad. When a multi-layer perceptron is used, Adam is also much faster than the other methods mentioned above. Therefore, the method designed in this paper will also be optimized using Adam.

References

- [1] Freddie Bray, Jacques Ferlay, Isabelle Soerjomataram, Rebecca L Siegel, Lindsey A Torre, and Ahmedin Jemal. Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians*, 68(6):394–424, 2018.

- [2] T.M. Mitchell. *Machine Learning*. McGraw Hill, 1997.

- [3] David H Hubel and Torsten N Wiesel. RECEPTIVE FIELDS OF NEURONES IN THE CAT'S STRIATE CORTEX. *The Journal of Physiology*, 148(3):574–591, 1959.

- [4] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceeding of IEEE*, vol. 86, pp. 2278-2324, 1998.

- [5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing systems*, pages 1097–1105, 2012.

- [6] The Paris endoscopic classification of superficial neoplastic lesions: esophagus, stomach, and colon: November 30 to December 1, 2002. *Gastrointestinal Endoscopy*, pp. S3–S43, 2003.

- [7] Jorge Bernal, Nima Tajkbaksh, Francisco Javier Sánchez, Bogdan J Matuszewski, Hao Chen, Lequan Yu, Quentin Angermann, Olivier Romain, Bjørn Rustad, Ilangko Balasingham, et al. Comparative Validation of Polyp Detection Methods in Video Colonoscopy: Results From the MICCAI 2015 Endoscopic Vision Challenge. *IEEE Transactions on Medical Imaging*, 36(6):1231– 1249, 2017.

- [8] David Vázquez, Jorge Bernal, Francisco Javier Sánchez, Gloria Fernández-Esparrach, Antonio M López, Adriana Romero, Michal Drozdal, and Aaron Courville. A Benchmark for Endoluminal Scene Segmentation of Colonoscopy Images. *Journal of Healthcare Engineering*, 2017, 2017.
- [9] Jorge Bernal, Francisco Javier Sánchez, and Fernando Vilariño. Towards automatic polyp detection with a polyp appearance model. *Pattern Recognition*, 45(9):3166–3182, 2012
- [10] Jorge Bernal, Francisco Javier Sánchez, Gloria Fernández-Esparrach, Debora Gil, Cristina Rodríguez, and Fernando Vilariño. WM-DOVA maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians. *Computerized Medical Imaging and Graphics*, 43:99–111, 2015.
- [11] Jorge Bernal, Javier Sánchez, and Fernando Vilariño. Impact of Image Preprocessing Methods on Polyp Localization in Colonoscopy Frames. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 7350 – 7354. IEEE, 2013.
- [12] Sae Hwang, JungHwan Oh, Wallapak Tavanapong, Johnny Wong, and Piet C De Groen. POLYP DETECTION IN COLONOSCOPY VIDEO USING ELLIPTICAL SHAPE FEATURE. In *2007 IEEE International Conference on Image Processing*, volume 2, pages II–465. IEEE, 2007.
- [13] Sebastian Gross, Manuel Kennel, Thomas Stehle, Jonas Wulff, Jens Tischendorf, Christian Trautwein, and Til Aach. Polyp Segmentation in NBI Colonoscopy. In *Bildverarbeitung für die Medizin 2009*, pages 252–256. Springer, 2009.
- [14] Matthias Breier, Sebastian Gross, and Alexander Behrens. Chan-Vese Segmentation of Polyps in Colonoscopic Image Data. In *Proceedings of the 15th International Student Conference on Electrical Engineering POSTER*, volume 2011, 2011.

- [15] Matthias Breier, Sebastian Gross, Alexander Behrens, Thomas Stehle, and Til Aach. Active Contours for localizing polyps in colonoscopic NBI image data. In *Medical Imaging 2011: Computer-Aided Diagnosis*, volume 7963, page 79632M. International Society for Optics and Photonics, 2011
- [16] Tony F Chan and Luminita A Vese. Active Contours Without Edges. *IEEE Transactions on image processing*, 10(2):266–277, 2001
- [17] Melanie Ganz, Xiaoyun Yang, and Greg Slabaugh. Automatic Segmentation of Polyps in Colonoscopic Narrow-Band Imaging Data. *IEEE Transactions on Biomedical Engineering*, 59(8):2144–2151, 2012.
- [18] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour Detection and Hierarchical Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2010.
- [19] Nima Tajbakhsh, Suryakanth R Gurudu, and Jianming Liang. A Classification-Enhanced Vote Accumulation Scheme for Detecting Colonic Polyps. In *International MICCAI Workshop on Computational and Clinical Challenges in Abdominal Imaging*, pages 53–62. Springer, 2013.
- [20] Nima Tajbakhsh, Changching Chi, Suryakanth R Gurudu, and Jianming Liang. AUTOMATIC POLYP DETECTION FROM LEARNED BOUNDARIES. In *2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI)*, pages 97– 100. IEEE, 2014.
- [21] Nima Tajbakhsh, Suryakanth R Gurudu, and Jianming Liang. Automatic Polyp Detection Using Global Geometric Constraints and Local Intensity Variation Patterns. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 179–187. Springer, 2014.

- [22] Stavros A Karkanis, Dimitrios K Iakovidis, Dimitrios E Maroulis, Dimitris A. Karras, and M Tzivras. Computer-Aided Tumor Detection in Endoscopic Video Using Color Wavelet Features. *IEEE Transactions on Information Technology in Biomedicine*, 7(3):141–152, 2003.
- [23] Robert M Haralick, Karthikeyan Shanmugam, et al. Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics*, (6):610–621, 1973.
- [24] Dimitrios K Iakovidis, Dimitrios E Maroulis, Stavros A Karkanis, and A Brokos. A Comparative Study of Texture Features for the Discrimination of Gastric Polyps in Endoscopic Video. In *18th IEEE Symposium on Computer-Based Medical Systems (CBMS'05)*, pages 575–580. IEEE, 2005.
- [25] Luís A Alexandre, Nuno Nobre, and João Casteleiro. Color and Position versus Texture Features for Endoscopic Polyp Detection. In *2008 International Conference on BioMedical Engineering and Informatics*, volume 2, pages 38–42. IEEE, 2008.
- [26] Stefan Ameling, Stephan Wirth, Dietrich Paulus, Gerard Lacey, and Fernando Vilarino. Texture-Based Polyp Detection in Colonoscopy. In *Bildverarbeitung für die Medizin 2009*, pages 346–350. Springer, 2009.
- [27] Sungheon Park, Myunggi Lee, and Nojun Kwak. Polyp detection in Colonoscopy Videos Using Deeply-Learned Hierarchical Features. *Seoul National University*, 2015
- [28] Eduardo Ribeiro, Andreas Uhl, and Michael Häfner. Colonic Polyp Classification with Convolutional Neural Networks. In *2016 IEEE 29th International Symposium on Computer-Based Medical Systems (CBMS)*, pages 253–258. IEEE, 2016.
- [29] Ruikai Zhang, Yali Zheng, Tony Wing Chung Mak, Ruoxi Yu, Sunny H Wong, James YW Lau, and Carmen CY Poon. Automatic Detection and Classification of Colorectal Polyps by Transferring Low-Level CNN Features From Nonmedical Domain. *IEEE Journal of Biomedical and Health Informatics*, 21(1):41–47, 2016.

- [30] Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine learning*, 20(3):273–297, 1995.
- [31] Lei Zhang, Sunil Dolwani, and Xujiong Ye. Automated Polyp Segmentation in Colonoscopy Frames Using Fully Convolutional Neural Network and Textons. In *Annual Conference on Medical Image Understanding and Analysis*, pages 707–717. Springer, 2017.
- [32] Mojtaba Akbari, Majid Mohrekesh, Ebrahim Nasr-Esfahani, SM Reza Soroushmehr, Nader Karimi, Shadrokh Samavi, and Kayvan Najarian. Polyp Segmentation in Colonoscopy Images Using Fully Convolutional Network. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 69–72. IEEE, 2018.
- [33] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.
- [34] Qiaoliang Li, Guangyao Yang, Zhewei Chen, Bin Huang, Liangliang Chen, Depeng Xu, Xueying Zhou, Shi Zhong, Huisheng Zhang, and Tianfu Wang. Colorectal Polyp Segmentation Using a Fully Convolutional Neural Network. In *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 1–5. IEEE, 2017.
- [35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional Networks for Biomedical Image segmentation. In *International Conference on Medical image Computing and Computer-assisted Intervention*, pages 234–241. Springer, 2015.
- [36] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. U-net++: A Nested U-net Architecture for Medical image Segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 3–11. Springer, 2018.

- [37] Ahmed Mohammed, Sule Yildirim, Ivar Farup, Marius Pedersen, and Øistein Hovde. Y-net: A deep Convolutional Neural Network for Polyp Detection. *arXiv preprint arXiv:1806.01907*, 2018.
- [38] Miao Fan, Jared Vicory, Sarah McGill, Stephen Pizer, and Julian Rosenman. Features for the Detection of Fat Polyps in Colonoscopy Video. In *Annual Conference on Medical Image Understanding and Analysis*, pages 106–117. Springer, 2018.
- [39] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.
- [40] Fernando M Silva and Luis B Almeida. Acceleration techniques for the backpropagation algorithm. In *European Association for Signal Processing Workshop*, pages 110–119. Springer, 1990.
- [41] Vinod Nair and Geoffrey E Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.
- [42] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep Sparse Rectifier Neural Networks. In *Proceedings of the fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.
- [43] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical Evaluation of Rectified Activations in Convolutional Network. *arXiv preprint arXiv:1505.00853*, 2015
- [44] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *arXiv preprint arXiv:1511.07289*, 2015.

- [45] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.
- [46] Anders Krogh and John A. Hertz. A Simple Weight Decay Can Improve Generalization. *Advances in Neural Information Processing Systems*, pages 950-957, 1992.
- [47] Jake Bouvrie. Notes on Convolutional Neural Networks. 2006. Available: http://cogprints.org/5869/1/cnn_tutorial.pdf [Assessed 21 Oct. 2019]
- [48] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research*, 15(1):1929– 1958, 2014
- [49] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient Object Localization Using Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–656, 2015.
- [50] Matthew D Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014.
- [51] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Robert Fergus. Deconvolutional Networks. In *CVPR*, volume 10, page 7, 2010.
- [52] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for Simplicity: The All Convolutional Net. *arXiv preprint arXiv:1412.6806*, 2014
- [53] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, 2014.

- [54] Min Lin, Qiang Chen, and Shuicheng Yan. Network in Network. *arXiv preprint arXiv:1312.4400*, 2013.
- [55] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper with Convolutions. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.
- [56] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.
- [57] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [58] Saining Xie, Ross Girshick, Piotr Dollr, Zhuowen Tu, and Kaiming He. Aggregated Residual Transformations for Deep Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2017.
- [59] François Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1251–1258, 2017.
- [60] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [61] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-Excitation Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.
- [62] Nobuyuki Otsu. A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

- [63] Stuart Lloyd. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.
- [64] Dorin Comaniciu and Peter Meer. Mean shift: A Robust Approach Toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (5):603–619, 2002.
- [65] Keinosuke Fukunaga and Larry Hostetler. The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.
- [66] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, 2005.
- [67] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (7):971– 987, 2002.
- [68] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic Image Segmentation with Deep Convolutional nets, Atrous convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017.
- [69] Fisher Yu and Vladlen Koltun. MULTI-SCALE CONTEXT AGGREGATION BY DILATED CONVOLUTIONS. *arXiv preprint arXiv:1511.07122*, 2015.
- [70] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

- [71] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 801–818, 2018.
- [72] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017
- [73] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large Kernel Matters—Improve Semantic Segmentation by Global Convolutional Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4353–4361, 2017.
- [74] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid Scene Parsing Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2881–2890, 2017.
- [75] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path Refinement Networks for High-resolution Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1925–1934, 2017
- [76] Hao Chen, Xiaojuan Qi, Lequan Yu, and Pheng-Ann Heng. DCAN: Deep Contour-Aware Networks for Accurate Gland Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2487–2496, 2016.
- [77] Korsuk Sirinukunwattana, Josien PW Pluim, Hao Chen, Xiaojuan Qi, Pheng-Ann Heng, Yun Bo Guo, Li Yang Wang, Bogdan J Matuszewski, Elia Bruni, Urko Sanchez, et al. Gland segmentation in colon histology images: The glas challenge contest. *Medical image analysis*, 35:489–502, 2017.

- [78] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a Discriminative Feature Network for Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1857–1866, 2018.
- [79] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for Dense Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.
- [80] Hieu V Nguyen and Li Bai. Cosine Similarity Metric Learning for Face Verification. In *Asian Conference on Computer Vision*, pages 709-720, Springer, 2010
- [81] Lee R Dice. Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26(3):297–302, 1945.
- [82] Annika Reinke, Matthias Eisenmann, Sinan Onogur, Marko Stankovic, Patrick Scholz, Peter M Full, Hrvoje Bogunovic, Bennett A Landman, Oskar Maier, Bjoern Menze, et al. How to Exploit Weaknesses in Biomedical Challenge Design and Organization. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 388–395. Springer, 2018.
- [83] Daniel P Huttenlocher, William J Rucklidge, and Gregory A Klanderman. Comparing Images Using the Hausdorff Distance. In *Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 654–656. IEEE, 1992.
- [84] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.
- [85] Janez Demšar. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.

- [86] Yan Zhang, Bogdan J Matuszewski, Lik-Kwan Shark, and Christopher J Moore. Medical Image Segmentation Using New Hybrid Level-set Method. In *2008 Fifth International Conference Biomedical Visualization: Information Visualization in Medical and Biomedical Informatics*, pages 71–76. IEEE, 2008.
- [87] Yan Zhang and Bogdan J Matuszewski. MULTIPHASE ACTIVE CONTOUR SEGMENTATION CONSTRAINED BY EVOLVING MEDIAL AXES. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 2993–2996. IEEE, 2009.
- [88] Willem Dijkstra, André Sobiecki, Jorge Bernal, and Alexandru C Telea. Towards a Single Solution for Polyp Detection, Localization and Segmentation in Colonoscopy Images. In *VISIGRAPP*, 2019
- [89] Hemin Ali Qadir, Younghak Shin, Johannes Solhusvik, Jacob Bergsland, Lars Aabakken, and Ilango Balasingham. Polyp Detection and Segmentation using Mask R-CNN: Does a Deeper Feature Extractor CNN Always Perform Better? In *2019 13th International Symposium on Medical Information and Communication Technology (ISMICT)*, pages 1–6. IEEE, 2019.
- [90] Quang Nguyen and Sang-Woong Lee. Colorectal Segmentation using Multiple Encoder-Decoder Network in Colonoscopy Images. In *2018 IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pages 208–211. IEEE, 2018.
- [91] Yang He, Margret Keuper, Bernt Schiele, and Mario Fritz. Learning Dilation Factors for Semantic Segmentation of Street Scenes. In *German Conference on Pattern Recognition*, pages 41–51. Springer, 2017.
- [92] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable Convolutional Networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 764–773, 2017.
- [93] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial Transformer Networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

- [94] Herbert Robbins and Sutton Monro. A stochastic Approximation Method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.
- [95] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1):145–151, 1999.
- [96] Yurii Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. A METHOD OF SOLVING A CONVEX PROGRAMMING PROBLEM WITH CONVERGENCE RATE $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.
- [97] John Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.
- [98] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [99] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.