Automatic Convolutional Neural Network Architecture Search for Breast Lesion Classification from Ultrasound Images: An ENAS Bayesian Optimization Approach

By

Mohammed Hussein Ahmed

School of Computing

The University of Buckingham

United Kingdom

A thesis

Submitted for the Degree of Doctor of Philosophy in in Computer Science to the School of Computing in The University of Buckingham

July -2022

Abstract

Breast cancer is one of the most common cancer types among women globally. Cancer detection/classification from medical images is a topic of vital importance because early cancer detection may allow patients to receive proper and timely treatment, significantly increasing their survival rates. Ultrasound imaging has been extensively used for various medical diagnostics by radiologists. Over the last ten years, sophisticated deep learning neural networks such as Convolutional Neural Networks (CNN) have been developed. Such deep learning neural networks tend to provide an "end-to-end" solution for image pattern recognition and have achieved impressive performance results for various applications. Deep convolutional neural networks have recently appeared in CAD systems due to their success in extracting effective image features.

CNN architecture design involves using many hyper-parameters. Creating a robust CNN architecture depends on finding an optimal combination between those hyper-parameters. Therefore, manually designing CNN architecture is time-consuming and leads to trial and error. Neural Architecture Search offers an alternative by automatically determining hyper-parameter settings for CNN architectures based on the dataset at hand. Efficient Neural Architecture Search (ENAS) is the efficient method for automatically designing CNN architecture. There has been no research in the literature - till that reported in this thesis - on using ENAS to search for CNN architecture for ultrasound images in general nor for breast lesion classification. In addition, there are only a few reported pieces of research on CNN architectures manually designed specifically for breast lesion classification from ultrasound images. These research works are based on only small datasets from one hospital in their modelling and testing processes.

This research aims to create an automatic designing CNN architecture approach for designing CNN architectures for classifying breast cancer from ultrasound images. This research investigates the effectiveness of one of the most popular methods, ENAS, for automatically searching for a convolutional neural network architecture. The research starts by adapting the ENAS framework to automatically search for optimal CNN architectures based on datasets of ultrasound images collected from different medical centres. The research then addresses the issue of model overfitting and generalisation of ENAS-based CNNs by using different data augmentation, reducing architecture complexity and training on an unbalanced number of images between benign and malignant classes. Furthermore, the ENAS framework is modified by expanding its search space by adding more operations suitable for ultrasound

images such as different convolutional operations with different filter sizes. This modification improved the overall performance of produced CNN architecture by ENAS. We further enhanced the design of final CNN model which are based on optimal cells obtained by ENAS by adding a high-way connection to compensate features from early layers to the final set of feature maps.

Furthermore, this research deploys the Bayesian Optimisation method to further develop an ENAS-B framework to address the limitations of the existing ENAS framework in optimizing the CNN architecture layers and trainable hyper-parameters, promoting an end-to-end automatic CNN search for the intended purpose of breast lesion classification using ultrasound images. The research concludes that a CNN model of 5 layers with optimised hyper-parameters is a robust model that can outperform the state-of-the-art CNN models designed for breast lesion classification such as VGG-16, ResNet50, Inception-V3, XceptionNet, NasNet mobile, EfficentNetB0 and mobile Net-V2 methods with transfer learning. The work presented in this thesis provides a good guideline for scientists to design a robust CNN model that can generalise beyond internal testing datasets.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this thesis entitled “Automatic Convolutional Neural Network Architecture Search for Breast Lesion Classification from Ultrasound Images: An ENAS Bayesian Optimization Approach” are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university.

This thesis is my own work and contains nothing which is the outcome of work done in collaboration.

Dedication

I dedicate my thesis to my family, my wonderful wife Khelan And my daughter Adan

Acknowledgement

First and foremost, I would like to thank Almighty Allah, the compassionate, the merciful, who has given me the courage and capacity to finish my study. I would like to sincerely thank my supervisors Mr Hongbo Do and Dr Alaa AlZoubi for their guidance, support and patience during my study.

I am grateful to my wife Khelan and my angel Adan for their patience and unending support.

I would also like to thank my parents, brothers and sisters for their love and emotional support.

Thanks for my friends, Aras Asaad, Jihad Anwar, and Hunar Ahmed for their supporting from my beginning of my study till now.

Special thanks go to Ten-D Innovations for providing financial support to my study and data for this research. I would also like to thank the School of Computing for providing a pleasant environment for my research.

Abbreviations

2D	2 Dimensions
3D	3 Dimensions
AI	Artificial Intelligence
ANN,	Artificial neural network
AUC	Area Under Curve
BO	Bayesian Optimization
BUSI	Breast Ultrasound Images
CAD	Computer-Aided Diagnostic
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CT	Computed Tomography
DAG	Directed Acyclic Graph
DARTS	Differentiable Architecture Search
DCNN	Deep Convolutional Neural Network
DL	Deep Learning
ENAS	Efficient Neural Architecture Search
FC	Fully-Connected
GAP	Global Average Pooling
GN	Group Normalisation
GP	Gaussian Process
GPU	Graphics Processing Unit
HW	High-Way Connection
LBP	Local Binary Patterns
LSTM,	Long short-term memory
LSVRC	ImageNet Large Scale Visual Recognition Competition
MRI	Magnetic Resonance Imaging
NAS,	Neural Architecture Search
PNAS,	Progressive neural architecture search
RL	Reinforcement Learning

RNN	Recurrent Neural Networks
ROI	Region Of Interest
SDV	Singular Value Decomposition
SGD	Stochastic Gradient Descent
SMBO	sequential model-based optimization
SOTA	State-of-the-art
SVM	Support Vector Machines
TNR	True Negative Rate
TNR,	True Positive Rate
US	Ultrasound
VGG	Visual Geometry Group

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

This thesis is broadly intended to develop and adapt advanced machine learning technology for improvements in health, and specifically focused on developing effective and efficient deep learning solutions in processing ultrasound images for accurate recognition of breast lesion type (benign or malignant). This chapter serves as a general introduction to the whole thesis. The chapter will first present the problem statement and research motivation. It will then specify the aim and objectives for the research covered by the thesis. The chapter will then describe the overall framework and the methodology followed by the research. Finally, the chapter will briefly summarize the research contributions and outline the structure for the rest of the thesis.

1.1. Problem Statement and Research Motivation

Cancer is one of the most serious and life-threatening diseases. Cancer can grow in many different parts of the human body. Among all forms of cancer, breast cancer is the most common in the world today [1]. According to the world cancer report in 2020, breast cancer had the highest incident with an estimated 2.26 million new cases in 2020, and accounted for 11.7% of the total number of malignancies [2]. Breast cancer was also the leading cause of mortality in 12 different locations throughout the world [3]. Previous studies have shown that early detection of breast cancers followed by appropriate treatment was responsible for 38% drop in mortality rate from 1989 to 2018 [4]. Digital Mammography (DM) and Ultrasound (US) are two commonly used imaging methods for breast cancer detection. Although DM is considered more effective, US imaging has the benefits of being safe without radiation, cheap, more sensitive to tumours located in dense areas and more versatile comparing to DM imaging [5].

Ultrasound imaging has demonstrated its usefulness in both detecting breast lesions and differentiating benign lesions from malignant ones. It plays a critical role in the diagnosis and the effective treatment of patients, and hence becomes a necessary procedure in many medical centres. However, US images suffer from many quality issues specifically related to use of sound signals in forming the image, such as speckle noises, poor contrast, blurry edges, image acquisition issues relating to the direction and pressure in placing the transducer during the

scan. These issues pose several difficulties in using US images in the diagnosis procedure. First, effective diagnoses heavily depend on the experience of the operators, i.e. radiographers and radiologists. Inexperienced operators often have difficulties in distinguishing between benign and malignant lesions, resulting in low diagnosis accuracy that will lead to unnecessary biopsies and even missed cases. Therefore, it is desirable to provide those inexperienced operators with supporting tools to improve their diagnostic accuracy and avoid tragic consequences to the patients. Secondly, there may exist a degree of inter-observer variability among radiologists, i.e. different diagnoses given by different radiologists for the same image. At the same time, there may also exist a degree of intra-observer variability, i.e. different diagnoses given by the same radiologist at different points of time. Such inter- and intra-observer variabilities are common place and well recognized issue in clinical practices [6]. Guidelines of different kinds such as Breast Imaging Reporting and Data System (BI-RADS) (see Chapter 2) have been used in clinics for maintaining consistency among US operators. However, the report generation using such guidelines is still based on the experience and the subjective judgement by the operator. So, the problem can only be reduced rather than avoided. Finally, there is always a limited supply of experienced radiographers and radiologists in medical centres. It normally takes years of training for the operators to become competent and reliable. Yet, the demand for experienced US operators is high, particularly in more populous and developing countries.

In recent years, Computer-Aided Diagnosis (CAD) systems have been applied to medical image analysis for various purposes [7]. A computer-based system for classifying ultrasound images of breast lesions provides decision support for the diagnosis, offer a second opinion besides the operator's observation, and minimize the dependency upon limited number of very experienced US operators. Such a system may also help in resolving the inter- and intra-observer variability issue by offering reliable and deterministic outcomes. Furthermore, CAD as an interdisciplinary technology that combines the strengths of advanced image processing techniques, machine learning algorithms and experts' domain knowledge, will ultimately improve accuracy of final diagnosis with great potential in reducing misdiagnosis rates.

Deep learning is considered a significant technology breakthrough in recent years as it has exhibited performance beyond the state-of-the-art in various machine learning tasks including object detection and classification from images. Contrary to conventional machine learning methods, which require a handcraft feature extraction stage (a challenging task as it relies on domain knowledge), deep learning methods automatically learn discriminative features from the input data with respect to the target outputs. This eliminates the tedious process of

investigating the discrimination ability of the image content-based features and engineering extraction algorithms. Consequently, there is a growing amount of interest in using deep learning to automatically learn useful features from ultrasound images of a breast lesion and diagnose the status of the lesion [8].

Convolutional Neural Network (CNN) is one of the most successful deep learning architectures for image classification. Most existing CNN architectures are manually designed by human experts. A CNN architecture consists of several types of hyper-parameters such as convolution layer, pooling layer, and fully-connected layer. As a result, manually choosing the right hyper-parameters for designing CNN architecture particularly effective for the problem at hand is a time-consuming and error-prone procedure [9]. For that reason, there is an increasing interest in automatic design of CNN architectures. Several approaches have been introduced including Reinforcement Learning (RL), Bayesian Optimization (BO) and evolutionary strategies. Neural architecture search (NAS) is a RL-based framework recently proposed [10], but NAS is a computationally intensive process due to the large search space and numerous permutations and combinations of architecture options. To address these limitations, the Efficient Neural Architecture Search through Parameter Sharing (ENAS) method was later developed [11] to reduce the amount of computation time significantly while maintaining comparable levels of accuracy.

Both NAS and ENAS have their origin in natural image classification, and hence there is a gap in the literature to fill in by utilizing and adapting ENAS and possible other approaches of automatic architecture designs for medical image analysis in general and ultrasound images for breast lesion classification in particular. In other words, there is a clear research question to ask: can automatic neural network search schemes (e.g. ENAS and other techniques) be adopted effectively for accurate and robust classification of breast lesion status from 2D ultrasound images? This research is set to answer that question.

1.2. Research Aim and Objectives

This research aims at developing effective frameworks and solutions for automatic search of optimal convolution neural network architectures and models for breast lesion classification from static 2D ultrasound images. In particular, the main research objectives are outlined as follows:

- To acquire knowledge and understanding on the state-of-the-art development of CNN architectures and technologies in the areas relating to medical image analysis in general

and in ultrasound image analysis in particular, and to acquire in-depth understanding of the principles behind automatic neural network search;

- To evaluate the effectiveness of existing CNN architectures in recognising benign and malignant breast lesions from US images obtained from clinical settings, and based on the results of the evaluation, to further modify and customize some existing CNN architectures with potentials for improving the performance of classification models;
- To investigate the effectiveness of adapting the ENAS framework in generating an optimal CNN architecture for breast lesion classification from US images;
- To investigate the issue of generalisation errors in CNN models and develop specific solutions to overcome the overfitting effect for the ENAS architecture-based models;
- To investigate the effects of structural modifications to the ENAS CNN architectures by introducing new highways (or skip connections) to improve the performance and the convergence of the model.
- To develop a novel hybrid automatic search solution that combines ENAS and Bayesian Optimization. In particular, Bayesian Optimization as a searching strategy aims at producing more accurate and more robust recognition models and at the same time reducing the amount of resource requirements for the recognition models
- To evaluate and compare our adopted ENAS and the hybrid ENAS-Bayesian models against the state-of-art CNN models for breast lesion classification using different datasets collected from different hospitals.

The outputs of this research benefit a CAD system by employing the optimal CNN architectures and the recognition models specifically targeting at the US images from clinics. Such an optimal model serves as a core component of the CAD system not only for lesion recognition but also for lesion detection and segmentation as a baseline model. Such a CAD system will provide decision supports to radiologists in accurate recognition of breast lesions.

It must be noted that this research is only concerned with the lesion recognition, *not* automatic lesion detection from an ultrasound image. Besides, although the same solutions developed from this research may well be applicable to US images of other types of lesions, the scope of this research is primarily set within the domain of breast lesion recognition.

1.3. Research Methodology and Research Framework

Our research follows a general approach from initial investigation to evaluation to prototyping to major development and at the same time uses empirical evidence from experiments to

support the development of novel ideas, algorithms and architectures and techniques. Our solutions are largely data driven. A mixture of deductive and inductive reasoning based on sound understanding is practised throughout the research lifecycle. This section therefore presents general methodology and research framework for this research.

The main focus of our research is automatic search for optimal CNN architectures for breast cancer classification from ultrasound images. Therefore, we start by evaluating one of most recent neural architecture search methods ENAS for its efficiency and accuracy. After selecting the optimal architecture, several methods have been examined for reducing generalisation errors for ENAS-based CNN models. We also modify the ENAS search space for operations and the search space in term of backbone structures by adding highway connections. Such modifications will provide an in-depth understanding towards the ENAS framework. Finally, we propose a new auto CNN design by combining ENAS and BO where BO strategy optimizes block structures and determine settings for trainable hyper-parameters for the final optimized CNN architecture. Figure 1.1 outlines our proposed solution framework. The process consists of four main stages of processing. The first stage takes a set of training US images and performs some image preparation tasks such as cropping the region of interest (RoI) and image resizing. The second stage is to use ENAS in searching the optimal cells (normal and reduction cells, see later chapters). The optimal cells are then used in the third stage to construct CNN layers and hyper-parameter searching through Bayesian Optimization. The final stage is to take the optimal CNN architecture and the training set to train the CNN model that will return the correctly predicted lesion status. Further details of each stage will be given in later chapters.

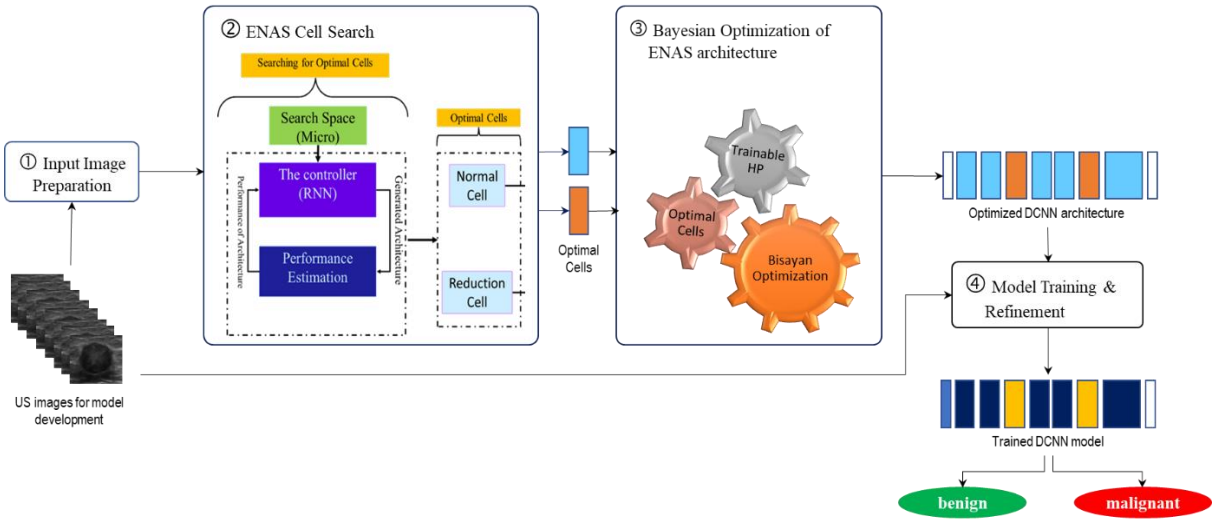


Figure 1-1:Proposed Framework of Automatically Optimizing CNN architecture for Breast Cancer Classification

Within the scope of this research, evaluation plays a central role not only for evaluating performance of various models produced by the finalized ideas, but also for validating and

testing alternative solutions. Therefore, the data we use becomes critical for the validity and soundness of the research outcomes. Data acquisition and collection criteria are in place to ensure (a) US images are of clinically acceptable quality and taken from US devices of multiple makes and models, (b) US images represent a variety of breast lesions of different pathologies and are taken from many patients from different medical centres, and (c) the image class labels should come from pathology reports particularly for borderline and malignancy cases. The roles of the image data and experimental protocols together with the performance metrics used will be further explained in later chapters.

1.4. Contributions of the Research

The contributions to knowledge made through this research can be summarized as follows:

- A good understanding about performance limitations with the existing handcraft deep learning CNN architectures and the performance limitations associated with manual modifications to such CNN architectures in the context of breast lesion recognition from ultrasound images. The thesis provides a comprehensive comparison between the ENAS-based models and the handcraft CNN models through a systematic evaluation.
- Adopt ENAS-based automatic search solution for building effective and efficient recognition models for recognizing breast lesions from ultrasound images.
- Explore a new scheme for reducing generalisation error of ENAS-based CNN models through uses of unbalanced dataset for searching and training, use of data augmentation, and reduction of architectural complexity.
- Develop an adaptive ENAS backbone architecture through skip connection “highways” of various kinds such as short, medium and long high way connections, leading to the conclusion that the long high way can improve model performance especially in terms of its generalisation power.
- Explore the effects of enriching ENAS operation search space and the conclusion that ENAS original operation set with small filter size convolutions is still most effective for the intended lesion recognition purpose.
- A novel search framework that uses the Bayesian Optimisation strategy to search for the optimal block structure of a CNN architecture based on cells optimized by ENAS, and to search for the optimal settings for trainable hyper-parameters to be adopted by the model training stage of the overall process.

- Evidence of the overall effectiveness of the ENAS-B method through extensive analysis and experiments using US breast images collected from different hospitals.
- Compare the ENAS-B performance with the state-of-the-art manually designed CNN models for breast lesion classification in US images.

During the course of this research, two papers have been peer-reviewed, accepted and published in two international conferences. The paper details are given as follows:

- Ahmed, M., Du, H., and AlZoubi, A. "**An ENAS based approach for constructing deep learning models for breast cancer recognition from ultrasound images**", International Conference on Medical Imaging with Deep Learning (MIDL 2020), Montreal, 6-8 July, 2020
- Ahmed, M., AlZoubi, A. and Du, H., "**Improving generalisation of ENAS-Based CNN models for breast lesion classification from ultrasound images**", In: Papież B.W., Yaqub M., Jiao J., Namburete A.I.L., Noble J.A. (eds) Medical Image Understanding and Analysis (MIUA 2021), Lecture Notes in Computer Science, vol 12722. Springer, Cham., pp438-453.

1.5. Ethics and Ethical Approval for the Research

The research takes place at the Ten-D Buckingham Research and Development Centre (TBRDC) under the Ten-D AI Medical Technologies Ltd (TenD) and University of Buckingham Research Collaboration Partnership scheme. TenD entered separate agreements with partner hospitals in Shanghai China regarding ethical uses of ultrasound images and acts as the third-party data provider for this research.

This research was granted ethics approval by the Research and Ethnics Committee of the School of Computing, the University of Buckingham before the start of the research by following the ethics approval procedure practised in the University of Buckingham. A limited number of images without patient identities are provided by TenD for feasibility tests and investigation purposes. The images are securely stored on the local share point created by the TBRDC with limited access only to the researchers involved in this research. Eventual training and testing of new architectures using a large number of images are performed on the local server behind firewall protection at TenD Shanghai office through remote access.

This research is intended for a good course and for benefiting patient health and effective use of resources. At the same time, it helps advancing the existing knowledge in utilizing deep learning technology for breast lesion recognition from ultrasound images.

1.6. Structure of the Thesis

The rest of this thesis is organised as follows. **Chapter 2** sets the background and context of this research by providing general information about breast cancer and use of ultrasound images in diagnosis of breast cancer. The chapter will also provide background knowledge in CNN architectures in general and neural architecture search in particular. **Chapter 3** presents a comprehensive literature review on the existing state-of-art approaches and techniques in medical image analysis especially CNN for breast cancer classification from ultrasound images as well as a thorough review of relevant existing works in our research domain, i.e. neural architecture search. C**hapter 4** provides details on the implementation of ENAS for generating optimal CNN architectures for breast cancer classification from ultrasound images. The chapter then presents a systematic evaluation and comparison of the implemented ENAS against the state of the art handcraft CNN architectures. This chapter further investigates generalisation errors of the ENAS-based CNN models for breast cancer classification from US images and proposes a set of solutions to overcome the generalisation issue. C**hapter 5** is designated for the adaptation of ENAS by modifying the ENAS block structure through the use of highways and evaluates the performance of the resulting recognition models built on the modified structures. Based on the investigation results of the previous two chapters, **Chapter 6** presents a novel idea of proposing a new CNN architecture search based on combining ENAS generated cells as a search space and Bayesian Optimization as a search strategy. **Chapter 7** concludes the thesis and outlines the future work.

Chapter 2. Backgrounds

This chapter presents general background information on the use of ultrasound imaging for breast lesion recognition and machine learning, particularly deep learning, for medical image analysis. It serves as a primer for knowledge and understanding upon which the rest of the thesis is built, and presents the needed application context for upcoming chapters.

The chapter consists of six sections. The first section is a brief overview of ultrasound imaging for breast lesion recognition, giving the application background where the problem for this research arises. The second introduces machine learning for medical image analysis in general. The third section describes specifically the concepts of CNN architecture, CNN building-blocks, and the main hyper-parameters for a typical CNN. Section 4 briefly reviews some well-known handcraft CNN architectures that have been applied to medical and even ultrasound image analysis. Section five describes the concepts and principles of neural architecture search and other optimization strategies for neural networks. Finally, Section six draws relevance of the background to the research undertaken for this thesis.

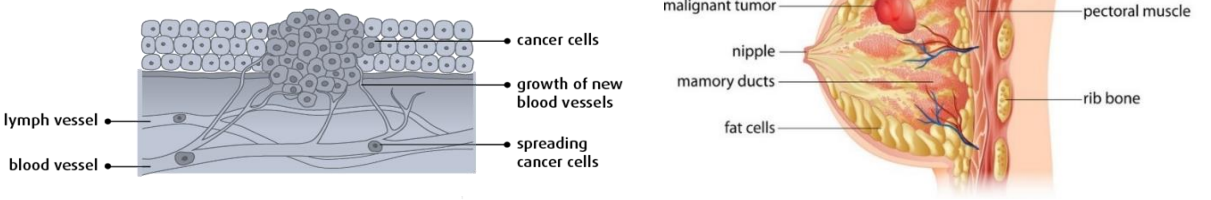
2.1. Ultrasound Imaging for Breast Cancer

2.1.1. Breast Cancer and Ultrasound Scan for Breast Lesion

The human body has trillions of cells growing normally through an organised process of generation, division and dying. Nonetheless, cells in any part of the human body, under certain unknown circumstances, may grow out of control. Instead of dying, they continue growing into abnormal and irregular cells some of which become cancerous. Figure 2.1a illustrates normal and abnormal cells. Cancer cells may not only appear as a lump shape known as a tumour, but also become invasive and spread through blood and lymph vessels to other parts of the body, destroying the normal tissues and eventually the entire cell re-generation process of the human body. Cancer is therefore considered as a life-threatening disease.

A mass concentration of cells in human breast is known as a breast lesion [12]. The formation of breast lesion may be caused by either cancer cell growth or by normal cell growth for other reasons such as accumulation of fat tissues, body fluid, etc [13]. Therefore, breast lesions can be classified as either malignant (i.e. cancer) or benign (harmless lump) [2]. Figure 2.1b illustrates the organ structure of a female breast. Mature woman breast contains connective

tissue, fat tissues and thousands of glands that produce milk for breastfeeding. Breast cancer is commonly formed in the inner lining of the tiny tubes, known as the ducts, which carry milk to the nipple. Cancer can also grow in the lobules, i.e. the glands that make milk [1]. Metastatic breast cancer cells can spread to the other organs of the woman body such as ovary or thyroid.



(a) Normal and Cancer Cells [14] (b) Breast Structure and Malignant Tumour [15]

Figure 2-1: Illustrations of Cancer, Breast Structure and Breast Cancer

There is no clear understanding on the causes of breast cancer. Some factors such as age, genetics, family history of cancer, body weight, radiation and hormone treatment received may make breast cancer more probable [2]. Several means including blood test, mammogram imaging and ultrasound imaging are often used for detecting breast cancer.

Medical images are digital images of organs inside a human body that are frequently used for regular health check-ups, monitoring, diagnosis of abnormality, etc. A specifically designed imaging device translates physical signals of a certain kind captured from sensors into images of a modality. Typical medical image modality devices include X-ray, Magnetic Resonance Imaging (MRI), Computed Tomography Scan (CTScan), and Ultrasound Scan (US). Although mammogram (a special type of X-ray) helps detecting abnormalities such as breast lesions and is widely used, the amount of radiation from the device received by the patient can be potentially harmful. Therefore, US scan that generates the image based on sound wave is often preferred. It is one of the safest medical image systems, and hence widely available in medical centres [16].

In principle, an ultrasound scanner consists of several functional components including a transducer for scanning, a control console for control settings, and a display screen for showing the captured image. The transducer acts as a probe that is connected to a receiver and a transmitter. The transmitter sends ultra high-frequency sound wave pulses at the target organ in the human body while the receiver receives a reflected sound pulses from the organ. Based on the power of the reflection, frequency and timing of receiving sound pulses, an image of the

organ is constructed and displayed on the display screen [16]. According to the shapes and uses of the transducers, ultrasound scan can be categorised as external, internal or endoscopic scan [17]. For breast inspections, external scan is normally used. Figure 2.2 shows an ultrasound scan machine in a clinical breast examination (left), and the images obtained from the probe (right).

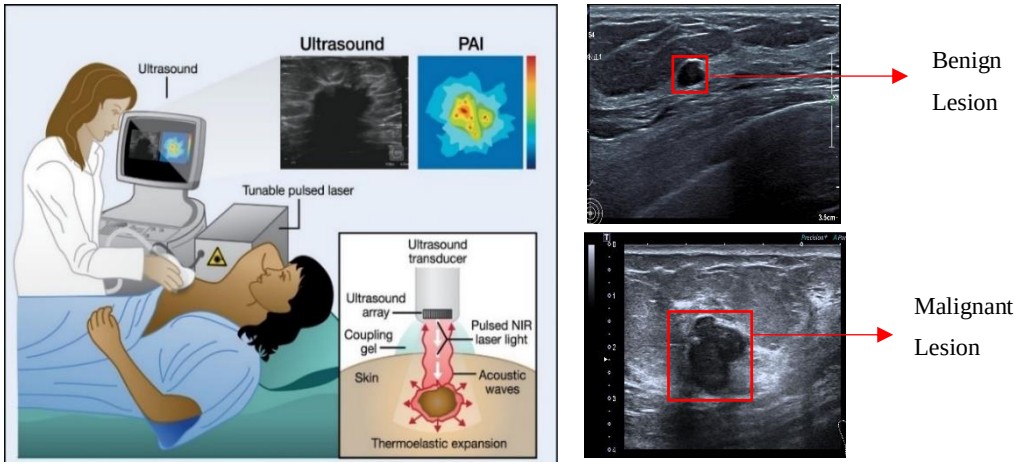


Figure 2-2: An example of scanning breast using ultrasound machine and breast lesion appearance[18]

2.1.2. Ultrasound Breast Lesion Image Description and Categorisation

Breast lesion is usually categorized as either being benign or malignant. Cells in a benign lesion are not cancerous; they do not spread to other parts of the human body, and thus are considered not life threatening. On the other hand, cancer cells inside a malignant lesion can invade tissues in the surrounding areas and spread to the rest of human body. Recognising malignant breast lesion at an earlier stage helps effective treatment and can save lives [19].

The shape and texture of a breast lesion in a US image are among the main indicators used by radiologists to classify the lesion into an appropriate class. Most benign lesions have regular shapes with smooth and regular margins, whereas malignant lesions have irregular shapes and unclear fuzzy margins. The ACR BI-RADS, as mentioned in Chapter 1, serves as one of several available guidelines for helping radiologists to better classify breast lesions and adhere to a common standard. The guideline specifies image characteristics on the lesion shape, orientation, margin, echo pattern, posterior features, and calcification. Several versions of the ACR BI-RADS have been published so far. The first version released in 1993 was used for diagnosing breast cancer from mammography images. The 2003 version introduced US and MRI imaging standards. The most recent version published in 2014 classifies breast lesion into six categories by a score assigned to an image according to the observed image characteristics as summarised in Table 1 [19]. The higher the category score is, the higher the risk of

malignancy becomes. BI-RADS 1 and 2 categories mean that the lesion is completely benign with no possibility of being malignant. Although BI-RADS category 3 is still considered as benign, it implies a small chance of malignancy and therefore may require additional diagnostic attentions after the initial ultrasound test. Category 4 indicates that the lesion is on the borderline between benignity and malignancy. To be more cautious, this category is further divided into three ordinal subcategories each of which has a higher degree of concern as shown in Table 2. Category 5 lesions have a very high probability of malignancy, and Category 6 lesions have their malignancy confirmed through biopsy.

Table 2-1:BI-RADS Categories [20]

BI-RADS Category	Assessment	Probability of Malignancy
0	Incomplete	Not enough information
1	Negative	0%
2	Benign	0%
3	Probably benign	0-2%
4	Suspicious	2-95%
5	Highly suggestive of malignancy	>95%
6	Known biopsy	Proven malignancy

Table 2-2: Sub-categories for BI-RADS Category 4. [20]

BI-RADS (4) Subcategories	Assessment	Probability of Malignancy
4a	Low suspicion for malignancy	2-10%
4b	Intermediate suspicion	10-50%
4c	Moderate concern	50-95%

It must be made clear that benignity or malignancy of a lesion can only be eventually confirmed through pathology examination like biopsy and that a BI-RADS score and the corresponding category are still assigned by doctors according to their visual inspections of the US image and hence subjective. As described in Section 1.1, human visual inspections of the US image, even with the help of the ACR BI-RADS guideline, still face with difficulties including the dependency on radiologist's experiences, the issue of inter- and intra-observer variability, and lack of supply of very experienced and reliable radiologists.

2.2. Machine Learning for Medical Image Analysis

Machine learning is a branch of artificial intelligence [21]. It is concerned with discovering various types of hidden patterns from data by executing learning algorithms on computers. Machine learning algorithms can be categorized into supervised, unsupervised and semi-

supervised learning [22]. This research is concerned with supervised machine learning which aims to learn, train and produce a reliable and accurate *model* from a set of training examples with known value for the target variable. The trained model will then take an unseen data record as input and predict correct output value of the target variable. The output value can be either categorical and then the model is known as classification model, or numerical and then the model is known as estimation model or regression model. In either case, the trained model by embedding the hidden pattern maps the input values (also known as features) to the output values (also known as labels) learnt from the training examples. Over the past several decades, machine learning has become an important technology for assisting medical diagnoses in general. The models learnt by the machine provide a second opinion on diagnostic outcomes and has shifted diagnosis decision-making from completely human-based to a hybrid form of humans and computers.

Medical images of the human body convey important information about organ structures and abnormalities of organs to specialised doctors (radiologists). Using functions provided by the imaging devices such as ultrasound scanner, the radiologists can observe the presence of certain anomalies such as calcification and measure the organ structure parameters such as size of the lesion and shape manually. This has been the clinical practice till this day for most, if not all, medical centres. Early efforts in CAD systems were focused on using computer vision and image processing techniques to automatically acquire the observed features and the measurements simply to relieve the manual workload on radiologists and avoid any intra/inter-observer inconsistency in the measurements [23]. Supervised machine learning techniques were then employed to build a diagnostic model based on the measurements and features that are automatically obtained [8].

The limitation of the approach described above is that the prediction model only considers those known domain features given by the domain experts. Medical images of various modalities, however, contain vast amount of pixel-based (i.e. image content-based) information such as intensity variation patterns, textures, shapes and contrast levels which may offer clues for detecting and diagnosis of diseases. Such patterns may or may not be known by the domain experts. Therefore, it is desirable to extract such information purposely from the images. In the same vein, images also have large amount of redundancy in data representation because many individual pixels have the same or similar color intensity values. Not all of these pixel values contribute equally towards the diagnostic decision-making. Therefore, another popular approach of utilizing machine learning and computer vision techniques is to develop effective algorithms in pre-processing the images first and then extracting useful features from the pre-

processed images, forming a richer basis of feature information for the supervised machine learning techniques to eventually build more effective models for diagnosis. Evidence has shown that such an approach does provide added value in terms of level of accuracy of prediction comparing to the previous approach [14].

Artificial neural network (ANN) is a major approach for machine learning, and has been used for supervised as well as unsupervised learning tasks [22]. The principle behind ANN is to emulate the working of human brain by constructing layers of interconnected artificial neurons and training the weights attached to the connections between the neurons [22]. Deep learning neural networks refer to the networks with many layers designated for different purposes. There are different types of deep learning neural networks such as Convolutional Neural Network (CNN), Deep Belief Networks (DBNs), Recurrent Neural Networks (RNN) and many more. Each of these techniques has more success in specific image, text, time-series or tabular-like data analysis tasks. For example, CNN achieved state-of-the-art performance in most of the image classification tasks [22]. The key concept of deep learning-based solutions is to let computers search and learn to discover features that optimally represent the data for the task in hand. This concept underpins many deep learning networks consisting of many layers that transform input data (e.g. images) to outputs (e.g. disease benign /malignant) by gradually learning basic (concrete) as well as higher-level (abstract) features. Successful deep learning networks to date are CNNs that exploit different convolution filters over many layers of the network [22]. Deep learning CNN models often outperform models built on extracted features by specifically designed algorithms, and hence become popular in machine learning in recent years.

2.3. Convolutional Neural Network: Building-Blocks

CNN is a special type of ANN specifically developed for image analysis tasks. CNN consists of a number of convolutional layers to extract features from the input image followed by a transformation layer using non-linearity functions, pooling layers for dimension reduction, and in the end fully-connected layers for classification [22]. Figure 2.3 outlines the structure of a typical CNN in the context of this research. We use ultrasound images in this example to show the relevance to work presented in this thesis. The network takes an input ultrasound image of a certain size, applies filters of certain sizes to extract some basic image features from local areas of the image by applying the convolution operation followed by the activation function, and then aggregates the basic features before the aggregated features are further processed by

the next layers. After several convolutional layers, transformation layers and pooling layers, the final feature maps (i.e. 2D extracted feature values) are flattened into a single feature vector. The feature vector is then fed into a conventional feed forward ANN containing the fully-connected layers to yield the final output values, i.e. either benign or malignant with class label with an associated probability. In this section, each key component of CNN will be briefly introduced. The purpose is to establish a general understanding for the terminology used in the following chapters, a necessary step due to the complex natures of various CNN architectures.

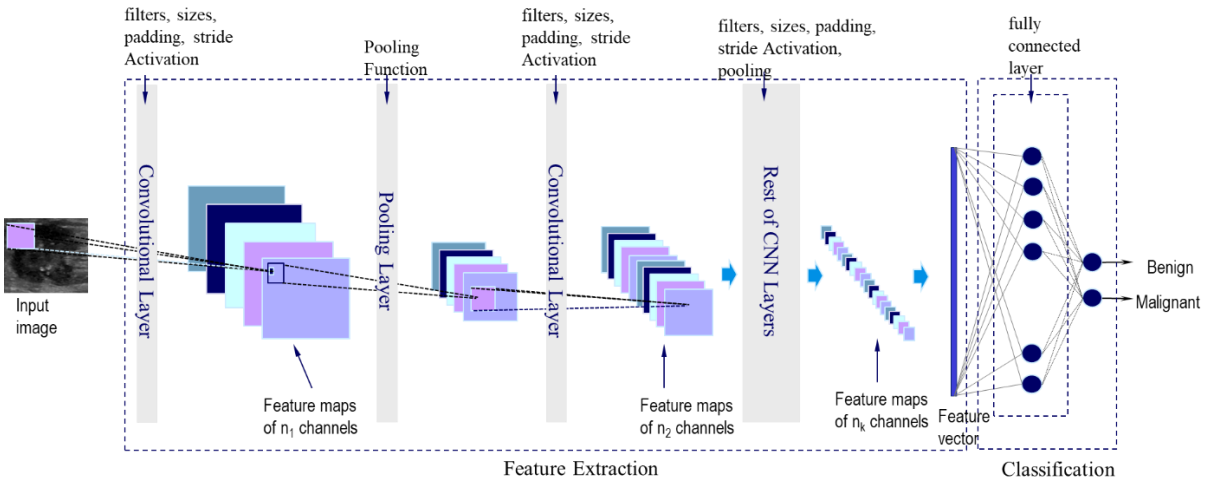


Figure 2-3: A Typical CNN architecture and components when used for lesion classification

Various characteristics of a CNN architecture can be described by variables known as *hyper-parameters*. In ANN, the variables that determine the neural network structure are known as structural hyper-parameters. Example structural hyper-parameters include the number of convolutional layers, activation functions to be used and so on. The variables that determine how learning/training must be conducted are known as learnable or trainable hyper-parameters. Example trainable hyper-parameters include learning rate, weight initialization, etc. Section 2.3.1 explains the structural hyper-parameters and Section 2.3.2 explains the trainable ones.

2.3.1. Structural Hyper-parameters

This section describes the main structural hyper-parameters of CNN architecture: i.e. the convolutional layer, non-linearity layer, pooling layer and fully-connected layer.

Convolutional Layer

The essential component of CNN is convolutional layer. The main purpose is to extract feature information from the input images by applying a convolution operation within a grid of discrete numbers (called a *filter* or a *kernel*). The output of the convolutional layer is a set of computed feature values forming a *feature map*. Figure 2.4 illustrates the principles of the convolution

operation across an example input image. A filter of size N×N (e.g. 2×2 in the figure) slides over the input image both horizontally and vertically at an interval known as a *stride* (i.e. the number of shifting pixels of the filter over the input image) till the whole image is covered [24]. Mathematically, the convolution operation works by crossing filter(s) over the input image and computing the weighted sum over the intensity values of the pixels covered by the filter where numbers within the filter are used as weights. In Figure 2.4, the weighted sum is calculated as: **7**×*2* + **2**×*0* + **3**×(*-1*) + **4**×*1* + **5**× *0* + **3**×(*-1*) + **3**×*1* + **3**×*0* + **2**×(*-1*) = 6 where the digits in bold represent the intensity value in the input image and the digits in italic represent the corresponding weights in the 2×2 filter. This convolution operation results in a matrix also known as a feature map. Feature maps will then be used as an input for the next and consequence layer.

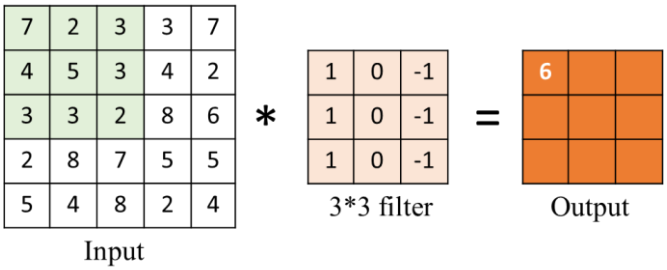


Figure 2-4: An example illustrating the steps of processing of applying the convolutional operation to an input image

Three hyper-parameters control the output volume of this windowing procedure: filter size, padding and stride. A filter of size N×N is a matrix of N×N weights, which can be initialized randomly and are then fine-tuned during the training process through back-propagation. The filter size defines a local neighbourhood of size N where feature information within the neighborhood of a specific kind can be extracted through convolution with the trained weights. Since the extracted feature information is represented by the result of the convolution operation, a feature map is normally of a smaller size (see [22] for the feature map size calculation). To maintain the spatial size of the input image, a process called padding may be used. The padding process involves adding additional values (such as zeros) around the input image's border [22]. The filter may scan by including padding values. The stride value *k* is the number of pixels a filter is shifted over the input image; the filter will move horizontally and vertically *k* pixels each time when the convolution operation is applied. Bigger stride values serve the purpose of reducing the dimensionality of the feature map [22], but may lose local image information. Different types of a convolutional layer such as Depthwise-seperable Convolution and Dilated Convolution, have been proposed [25]. We will explain those when needed in later chapters. To ensure extraction of various types of local image features, several

or even many filters may be applied, creating multiple feature maps from the same input image, stacked one after another along *channels*.

Activation Function (Non-linearity Layer)

The activation function takes each value of a feature map generated by the convolution operation, i.e. the weighted sum, and transforms it to another value. It is meant to introduce some non-linearity in this transformation. Sigmoid, Tanh, Elu and ReLU are a few examples of the commonly used activation functions in deep learning, but ReLU (Rectified Linear Unit) is the most commonly-used activation function in CNN [24]. ReLU is a simple and efficient method; it maps any negative input values to zero and keeps the positive input value unchanged [22]. Figure 2.5 illustrates the transformation of a feature map immediately after the convolution operation and the resulting feature map after the ReLU transformation.

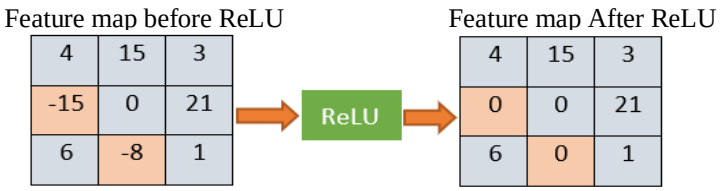


Figure 2-5: An example to show the effect of using activation function (ReLU) on a feature map

For conveniences, activation function is normally not treated as a separate layer. Rather, it is treated as a supplementary step to the convolution operation. In other words, the weighted sum is directly transformed using an activation function and then the feature map stores the transformed value before the convolution operation is considered complete.

Pooling Layer (Sub-sampling or Down-sampling)

A pooling layer is the place where a pooling function is applied to a local region of an input feature map, converting the values within the region into a single value, and hence reduce the dimensionality of the input feature map. There are several types of pooling functions such as max pooling, average pooling and sum pooling, which respectively result in the maximum value, the average and the sum of the values within the region. The pooling layer works by sliding a window of a certain size over the input feature map then feeding the values of the region within the window to one selected pooling function. Figure 2.6 illustrates an example of using Max pooling 2×2 over an input feature map.

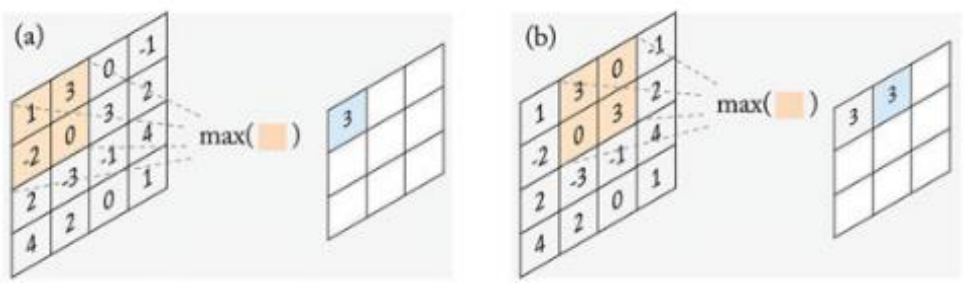


Figure 2-6: Steps (a) and (b) describes Max pooling operation [12].

Global Average Pooling

Global Average Pooling (GAP) is a pooling operation that replaces an entire feature map with the average of the values within the feature map. Figure 2.7 illustrates an example. In this example, each feature map of size 6x6 along 3 channels is replaced by the average of the feature map values, creating a feature vector of three components (i.e. three global averages). GAP appears replacing the traditional fully-connected layers in a CNN architecture; rather than implementing fully-connected layers on top of the final feature maps generated from the final convolutional layer, the average of each feature map is taken as a component of a final feature vector, which is then directly fed into the SoftMax layer. One of the advantages of GAP is that no parameter can be optimized in the global average pooling, therefore overfitting can be evaded at this layer. Besides, global average pooling totals the spatial information, meaning that it is more robust for spatialising the input translations [26].

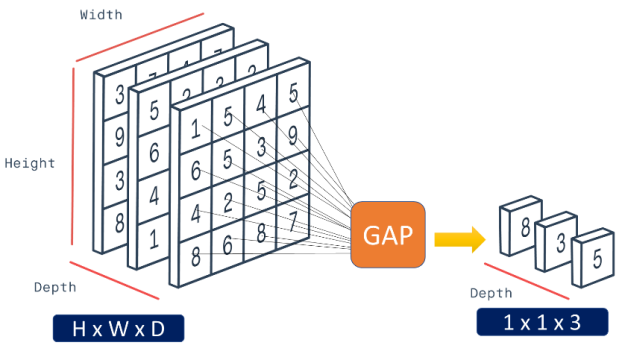


Figure 2-7: Global Average Pooling

Fully-connected Layer

Fully-connected layers are conventional feed-forward neural network layers where each neuron node is densely connected to all neurons in the next layer, as illustrated in the relevant part of Figure 2.3. Typically, fully-connected layers are placed at the end of the CNN network. They are used to map the last feature maps of the final convolutional layer to the output layer. The operations performed in each neural node include a weighted sum of all the inputs from the

previous layer followed by an activation function [22]. Because of the different structure of the fully-connected layers, the feature maps from the final convolutional layer need to be flattened into a one-dimensional feature vector before it is fed into the first fully-connected layer.

In CNN, SoftMax activation function is normally used in the last fully-connected layer (output layer) as a classification method. It calculates the relative probabilities by using the value of output nodes to determine the final probability value. The function is defined as follows:

$$softmax(Z_i) = \frac{\exp(Z_i)}{\sum_j \exp(Z_j)} \quad (2.1)$$

where, the Z_i represents the values from the i^{th} neuron of the output layer. The exponential acts as the non-linear as well as a normalization function to convert the values into probabilities.

2.3.2. Learnable Hyper-parameters of CNN

Most CNN layers consist of weights or biases which need to be tuned to extract robust features from input images. Several hyper-parameters such as learning rate and optimization function contribute to training the CNN model via the back-propagation process. This section describes various techniques used to set the weights in the CNN model such as weight initialization methods and regularization techniques.

Weight Initialization

The appropriate weight initialization in CNN plays an important role in avoiding vanishing gradients and reducing the time of convergence in training [27]. Therefore, initializing weight is one of the most important stages for designing a good CNN model [28]. To avoid similarity among hidden nodes of the same layer, weights should be initialized carefully. For example, setting all weights to zero leads to a model that is unable to learn any new features during training. On the other hand, if a weight value is too large, it will lead to an exploding gradient, while a small weight value will lead to a diminishing and vanishing gradient problem. Krizhevsky et al. [29] suggest the use of a random Gaussian distribution with a mean equal to 0 and a standard deviation of 0.01 for generating initial weights for their CNN model. Xavier is another random initialization method [30]. The authors generated weights by distribution with a mean equal to 0 and a variance equal to $2/(n_{in} + n_{out})$ where n_{in}represents the number of nodes feeding into it while n_{out} is the number of output nodes from the layer. In [31], He showed that using Xavier initialization with ReLU activation function in designing CNN architecture dose not perform well. Instead, he proposed a method, known as He initialization,

by setting mean = 0, standard division = sqrt(2/n) where n is the number of inputs to the node. An experimental study using an architecture with 30 layers showed that models with the He initialization converged while those with Xavier initialization did not [31]. To the best knowledge of this author, the reasons behind this difference in convergence have not been fully understood yet.

Regularization

Model overfitting is one of the main challenges facing deep learning. Overfitting means that a model performs well on the training data, but poorly on the validation or test data [22]. Having a huge number of parameters in a deep learning model trained from a small number of training examples is a major factor for causing the model to overfit the training data. To reduce overfitting, different techniques known as regularization techniques, such as L2 regularization, dropout, and data augmentation have been proposed in [28].

Dropout: Dropout is a method first proposed by Hinton et al. [32]. Dropout works by randomly setting some activations to zero. Mostly dropout is used in the fully-connected layers. Dropout reducec CNN model overfitting by not always tuning all weights all the time as shown in Figure 2.8. Dropout increases the accuracy of the CNN model even if certain information is missing.

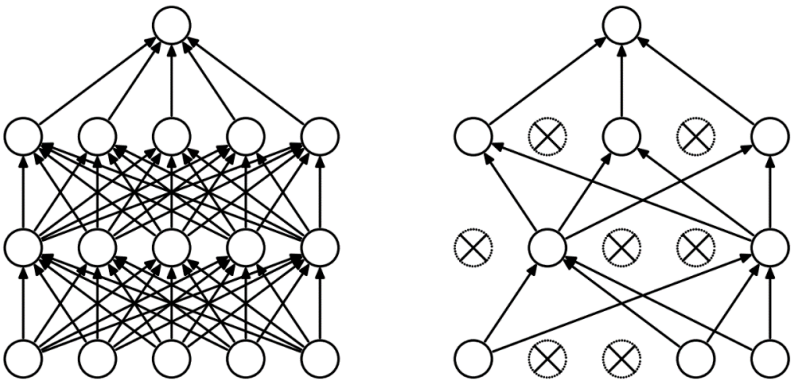


Figure 2-8: Fully-connected layer with and without Dropout [33]

Data Augmentation: Deep learning requires many training samples due to the massive number of weights that need to be "tuned". However, in most situations, we do not have large number of training examples, especially in the case of medical images. Therefore, data augmentation is a technique that artificially increases training examples to improve CNN model performance. For images, rotation, sampling, mirroring and cropping are some of the

techniques used for data augmentation purposes [29]. We will also show some of our own data augmentation techniques in later chapters.

L2 Regularization: L2 regularization consists of adding an extra term to the loss function that penalizes the complexity of the model as shown in the following equation. Choosing a suitable λ helps the model to perform better by finding small weights to minimize the loss function [22].

$$C = -\frac{1}{n}\sum_{xj}[yj\ ln\ a_j^L + (1 - yj)\ \ln(1 - a_j^L)] + \frac{\lambda}{2n}\sum_w W^2 \qquad (2.2)$$

The first part of the equation is in fact the cross-entropy loss function, and the second part is known as the regularization parameter which is adding the squared value to weight (*W*).

Batch Normalization: In a deep neural network, the input value changes layer by layer. However, this will cause a problem which is referred to as the Internal Covariate Shift [28]. Batch normalization (BN) is proposed by a Google researcher [34] that can be used to reduce that problem by normalizing the layer input. A batch normalization layer is often added between convolution layers to make the network learn better in terms of generalisation on a training and test set. This makes the CNN network converge faster. Indeed, batch normalization can reduce the problems that are faced throughout the training procedure with regard to a CNN model. Consequently, it is one of the most common methods that is used to help a model train more quickly and obtain better performance. Batch Normalization is standardizing the inputs of each layer (i.e. feature maps from previous layers). In each training iteration, BN produces batch (mini-batch **mean** and variance).

$$\mu = \frac{1}{n}\sum_i Z^{(i)} \qquad (2.3)$$

$$\sigma = \frac{1}{n}\sum_i (Z^{(i)} - \mu)^2 \qquad (2.4)$$

$$Z_{norm}^{(i)} = \frac{Z^{(i)} - \mu}{\sqrt{\sigma^2 - \epsilon}} \qquad (2.5)$$

$$Z' = \gamma * Z_{norm}^{(i)} + \beta \qquad (2.6)$$

First, the BN layer calculates the mean μ and the variance σ^2 of the feature map values across the mini-batch (3), (4). Then normalizes the activation vector $Z^{(i)}$ with (5). As a result, the output of each neuron follows a standard normal distribution across the mini-batch. It eventually computes the layer's output $Z'^{(i)}$ by using a linear transformation with γ and β, two trainable parameters (6).

Alternatively, the authors of [35] proposed the Group Normalization (GN) method. For normalisation, GN splits the channels into groups and calculates the mean and variance within each group. Figure 2.9 illustrates the difference between BN and GN. Given the feature maps of shape (N, C, H, W), BN normalizes the N direction and GN normalize the C direction by divides the C channels into groups and normalizes the groups individually.

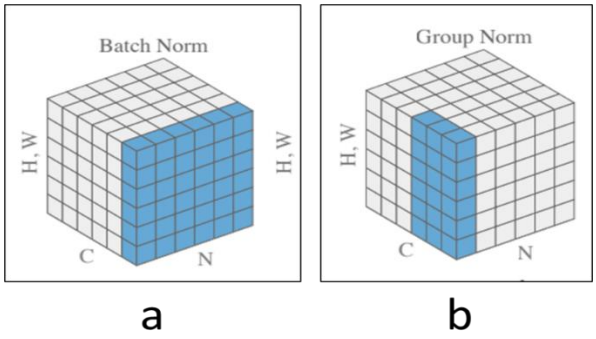


Figure 2-9: (a) describes Batch Normalization, (b) is Group Normalization [30]

The authors of [28] show that the accuracy of GN's computation is steady throughout a wide variety of batch sizes, regardless of batch size. When utilising a batch size of 2, GN has 10.6% lower error than BN; when using average batch sizes, GN is comparable to BN and beats other normalisation alternatives.

Optimization

Deep learning requires a huge number of parameters to be tuned during the training process. The most important parameters that need to be optimized are weights and biases. The CNN model's parameters will update themselves in each training step by applying optimization algorithms in order to minimize the loss and to make the model generalize well.

Stochastic Gradient Descent (SGD) is one of the most common optimization algorithms in neural networks. It lowers the target function $J(\theta)$ that takes a list of model's parameters $\theta \in R^d$ by tuning the parameters in reverse of the gradient of the target function $\nabla\theta J(\theta)$ with respect to the parameters. Another parameter is the learning rate η that defines the size of the steps to take towards reaching the minimum. A Gradient Descent has three variants - Batch Gradient Descent, Stochastic Gradient Descent (SGD), and Mini-batch Gradient Descent [22]. A Batch Gradient Descent also known as a Vanilla Gradient Descent calculates the gradient of the loss function with respect to the parameters θ for the whole training set [28]:

$$\theta = \theta - \eta \cdot \nabla\theta J(\theta) \qquad (2.7)$$

while SGD tunes the parameter for every train example x(i) and labels y(i) [28]:

$$\theta = \theta - \eta \cdot \nabla\theta J(\theta; x(i); y(i)) \quad (2.8)$$

However, the Mini-batch Gradient Descent combines both concepts and ensures the parameter update for each mini-batch of n training examples [36]:

$$\theta = \theta - \eta \cdot \nabla\theta J(\theta; x(i:i+n); y(i:i+n)) \quad (2.9)$$

In SGD, it's non-trivial to optimize a learning rate because the magnitudes of the diverse hyper-parameters change greatly, and throughout the training procedure, a modulation is required. To fix this issue, various SGD-based algorithms were proposed, including Adagrad, Adadelta, RMSprop, and Adam [28]. The goal of these SGD variants is to automatically adjust the learning rate to diverse parameters based on the gradient statistics. In addition, mostly they simplify the settings of the learning rate, leading to a quicker concourse. However, it is noted that the performance in terms of generalization tends to be considered not so good as that of SGD in several scenarios [28].

Loss Function

Always, a loss or cost function is used to estimate the error of the model's prediction during the training of the network classification model. It works by quantifying the difference between the model output prediction and the actual input (the labelled data should be used as an input for the CNN model). There are several types of loss functions such as Hinge loss, Softmax loss, Contrastive loss, Triplet loss and so on. Choosing a particular loss function depends on the type of situation such as detection or classification. For instance, the most common loss function used in the classification model is the cross-entropy loss function [28] which is defined as follows:

$$loss(p, y) = -\sum_n yn \log(pn) \quad (2.10)$$

where y is the ground truth output, n is the number of neurons in the output layer, and p is the probability for each output type.

2.3.3. Designing CNN Architectures

Designing CNN architecture is not an easy task as CNN architecture consists of many hyper-parameters. Example hyper-parameters include filter sizes, number of filters, stride, weight initialization, regularization methods, a number of convolution layers, pooling function,

learning rate, batch size and so on. There are no standard rules to follow for designing CNNs. There are two general strategies forwards. The first strategy is to manually stack CNN layers and tuning hyper-parameters. Most of the existing CNN models are designed by following this strategy. There are two commonly adopted alternative approaches under this strategy. For a specific application dataset, either we first design a shallow CNN architecture and then find the optimal depth and settings experimentally or we first evaluate several well-established CNN architectures on the dataset and then select the best performing one. Both approaches are time-consuming and require specific experience in medical image analysis as all the state-of-the-art CNN architectures are originally designed for classifying natural images rather than ultrasound images.

The second strategy of designing CNN architecture is automatically searching for an optimal CNN architecture for the dataset of interest. This technique requires defining a set of CNN hyper-parameters as a search space to be used by an optimization method. The method searches the space for the optimal combination of operations and input/output connections to form an optimal CNN architecture. Both strategies will be introduced in the next two sections, respectively.

2.4. Manually Designed CNN Architectures

Deep learning has recently achieved an impressive performance in different types of pattern recognition tasks such as speech recognition, natural language processing and image classification. In solving these practically challenging problems, several deep learning neural network architectures such as CNN and RNN have been developed [37]. This section presents several state-of-the-art CNN architectures, most of which have won the prize for the ImageNet Large Scale Visual Recognition Competition" (ILSVRC) challenge in recent years.

The first model that introduced the use of convolution network was LeNet [38]. This model was used for classifying handwriting digits with image size 32×32 pixels. The architecture consists of 3 convolutional layers followed by 1 pooling, and 2 fully-connected layers as shown in Figure 2.10. Due to the lack of high-power computers at the time, the researchers could not extend the LeNet architecture to a dataset of larger and more complex images.

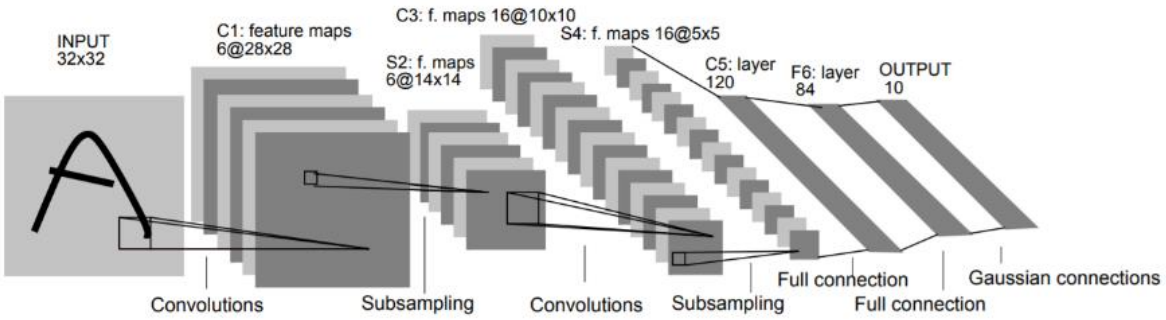


Figure 2-10:: Architecture of LeNet [33]

AlexNet is the first deep neural network architecture proposed in 2012 [29]. It is deeper than LeNet [38]. This architecture consists of 8 layers: 5 convolutional layers, 3 fully-connected layers followed by a non-linearity function ReLU. Three overlapping max-pooling layers have been used in first, second and fifth convolutional layers for down-sampling. Local normalization is used in the first and second convolutional layers after ReLU. Random Gaussian distribution with mean = 0 and Std = 0.01 was used for weight initialization. In addition, the bias value of zero is set for the first and third convolution layers, but the value of 1 for the rest of the layers. To reduce model overfitting, a dropout with a probability of 0.5 has been used in the first and second fully-connected layers. This architecture is trained on a large dataset which contains 1.2 million high-quality images with 1000 different classes entitled ImageNet LSVRC-2010. AlexNet remains as one of the most-popular state-of-the-art architectures that have been published until now. It is still widely used and adapted in various application domains for image classification. We also attempted it for classifying breast lesion in ultrasound images (see later in the thesis).

VGGNet was the runner up at the ILSVRC 2014 competition [39]. It was deeper and more accurate than the previous CNN architectures. Researchers working on VGGNet proposed 6 versions of deep learning architecture according to the number of hidden layers, which goes from 11 to 19 layers. VGG 16 and VGG19 are most commonly used for image classification purposes. To create a deeper model without exponentially increasing the number of parameters, a fixed 3×3 filter size has been applied to all convolutional layers. Figure 2.11 shows the VGG16 architecture. There are 5 blocks of 3×3 convolutional + ReLU + 1 MaxPooling layers. The first two blocks have two convolutional layers before the pooling layer, and the rest three blocks have three convolutional layers before the pooling layer. AlexNet, LeNet and VGGNet are similar in terms of basic block structures and different in terms of filters and the number and composition of each block.

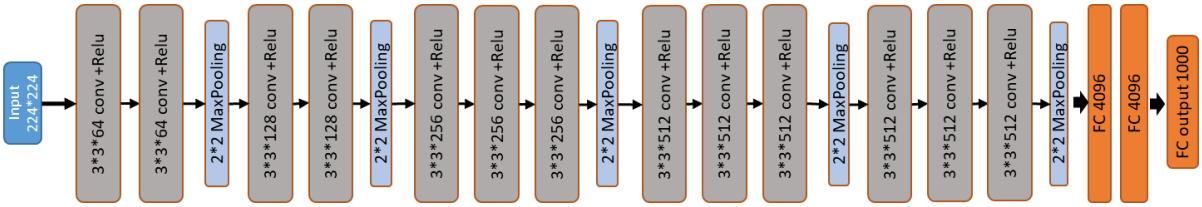


Figure 2-11: VGG16 architectures

GoogLeNet [40] was the winner of the ILSVRC 2014 competition [41]. It consists of 22 layers. In order to increase the depth of the CNN architecture without increasing the number of parameters, GoogLeNet replaced the classical design of the CNN model from a sequence of convolutional layers followed by pooling to stacked Inception Modules, as shown in Figure 2.12. An architecture of this kind can be seen as a Directed Acyclic Graph (DAG) where the nodes are a basic convolution or pooling operation, and edges are information flow from one unit to another. The number of processes from one layer to the next is not necessarily fixed. This interesting idea appears later in the NAS approach for architecture design (see Section 2.5).

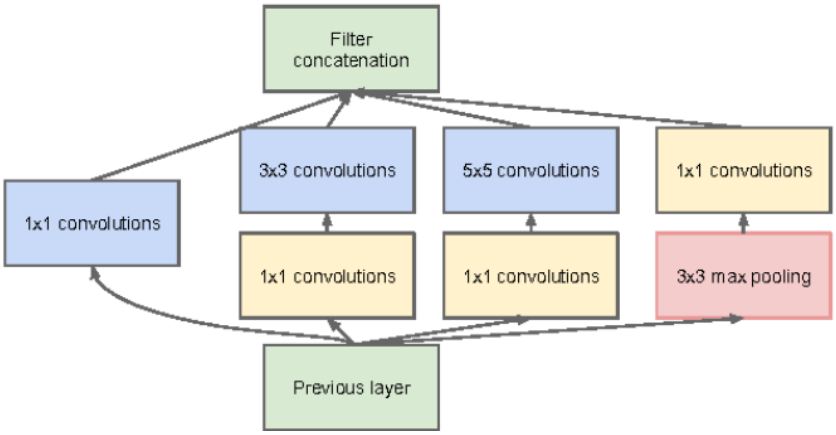


Figure 2-12:Inception module (GoogLeNet) [35]

The winner of the ILSVRC 2015 competition was Deep Residual Network (ResNet) [42]. One of the main contributions of this architecture is the residual connection which allows it to propose different versions of the network such as ResNet with (34, 50, 101 or 152) layers. ResNet uses the residual block (Figure 2.13A) to reduce the probability that vanishing gradient problem occurs, one of the well-recognised problems happening during backpropagation that prevents the network to learn further. Indeed, the strategy of designing modern CNN architectures often involves looking for higher accuracy and seeking lower training time. Moreover, DenseNet [43], same as ResNet, was proposed to overcome the gradient vanishing problem. Cross-layer connectivity is used to connect each earlier layer to the next layer in a feed-forward CNN. Consequently, feature maps of all preceding layers are reused as inputs to the next coming layers. In DenseNet the feature maps are concatenating instead of element-

wise adding. Therefore, this allows the DenseNet to clearly distinguish between information that is added to the network and information that is maintained [43].

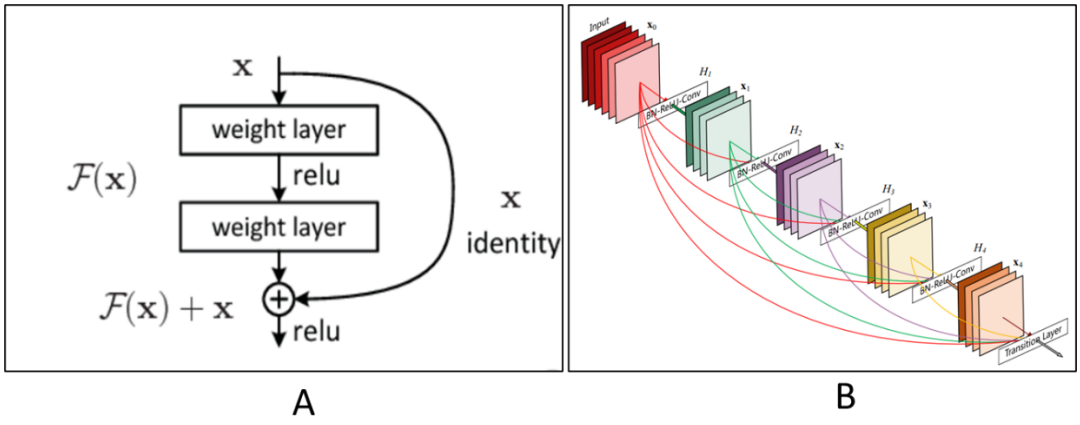


Figure 2-13: (a) Residual block [37], (b) Dense block of DenseNet [38]

More recently in 2016, a new version of GoogLeNet, known as Inception-V3, was proposed [44], which has a deeper architecture than the initial GoogLeNet. Later in 2016, Inception-ResNet, a deeper CNN model that combines inception blocks and residual connections, was also developed [45]. Xception [46] is the modified version of the Inception architecture, by making the Inception blocks wider and reducing its computational complexity of by exploiting depthwise separable convolution and replacing convolutional layers with filter size (1×1, 5×5, 3×3) with a single convolutional layer with size (3×3) followed by a 1×1 convolution.

Conventional convolution needs to perform multiplications extensively, which increases inference time and limits its usefulness for real-time applications with limited memory space such as autonomous cars, robots, healthcare, and mobile apps. As a result, many CNNs have been specifically proposed for such platforms, such as MobileNets [47] and MoblieNetV2 [48]. MobileNets [47] also replaces 3×3 convolutions with 3×3 depthwise separable convolutions and 1×1 pointwise convolutions to offer a fair balance between accuracy and training costs. The key contribution of MoblieNetV2 is a new layer module called the inverted residual with a linear bottleneck. This module accepts a low-dimensional compressed representation as input, which is then enlarged to high dimension and filtered by a lightweight depthwise convolution. In [48], they designed a new scaling approach that equally scales all depth/width/resolution dimensions using a simple yet extremely effective *compound coefficient*. They show how this strategy works for scaling up Mobile Nets and ResNet. To take it a step further, they used neural architecture search to create a new baseline network and scale it up to create the

EfficientNets family of models [48], which outperform earlier convolutional neural networks in accuracy and efficiency.

2.5. Automatic CNN Design

Due to the careful settings of so many hyper-parameters, creating an optimal CNN classification model by manually tuning these hyper-parameters can be a complex and prohibitive task if a trial-and-error approach is followed. Therefore, an automatic process of searching for the optimal CNN architecture based on the data at hand is an appealing proposition. Optimization of hyper-parameters is a significant study issue in machine learning and is commonly implemented in practice [49]. Despite their effectiveness, these approaches are still restricted to searching models from a fixed-length space. To put it another way, it is challenging to expect them to provide a variable-length configuration that details the connection and structure of a network [50]. Although there exist methods such as Bayesian optimization [51], [52] and evolution techniques that enable the search of non-fixed length designs, they are less generic and less adaptable than the Neural Architecture Search [10]. Neural Architecture Search (NAS) [10] is one of the most recent developments in answering this proposition. Zoph and Le proposed an initial NAS framework based on a reinforcement learning approach for training Recurrent Neural Network (RNN) to generate optimal CNN architectures [10]. The idea is to use RNN as the controller to generate parameters of CNN convolution layers. The NAS principal framework process consists of three fundamental components: search space, search strategy and performance estimation as outlined in Figure 2.14 and described in Section (2.5.1.). More details will be given in the next subsections.

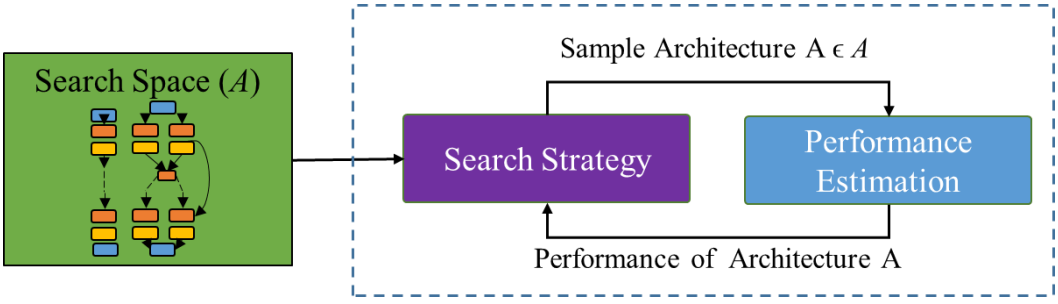


Figure 2-14: Overview of NAS Framework

2.5.1. NAS Components

Search Space

Search space contains a collection of basic deep learning architectural components that may be suitable for feature extraction from the provided images of a specific application domain. A basic search space may be thought of as a sequence of neural networks. A chain-structured convolutional neural network is composed of n layers, each of which gets input from the preceding layer i-1 and acts as an output for layer i+1. The search space can be therefore parameterized by the number of convolution layers, the type of layer operations, for instance, convolution layer, max pooling or average pooling, the connectivity between layers (Skip connection) and the hyper-parameters related to layer operation, for example, stride and number of filters [9]. Recently, block or cell-based search space was proposed. Instead of searching for the whole CNN architecture, the searching technique will search for cell blocks, and the final architecture will be built by stacking these blocks in a certain order. When compared to a comprehensive architectural search, this search space offers two distinct benefits. First, exponentially reducing the search space complexity because the cell is smaller compared to the whole CNN architecture. Second, Cells may be easily transferred to a different dataset by adjusting the number of cells utilised in the model [53].

Search Strategy

The search strategy is an algorithm that is used to explore the search space and construct candidate neural network architectures according to a set of given constraints. Using the correct search strategy algorithm helps to find an optimal architecture quickly. Several search strategies have been used to explore optimal CNN architectures, such as Bayesian Optimization, reinforcement learning (RL), evolutionary methods and random search and gradient-based methods. Below we describe two of the most commonly used methods used in architecture search techniques as search strategy algorithms [9].

A. Reinforcement Learning (RL)

RL is a machine learning area that is concerned with studying optimal decision making. The principle behind RL is how software agents decide to act, that a cumulative reward is maximized. Agents must find the best action to take in an environment that tasks are optimally performed. The environment sends observations to the agent in the form of a reward signal for any information or actions about the new state. The reward provides notice to the agent about the quality of the action [54].

RL can be considered a paradigm located at the boundary between supervised and unsupervised learning. It cannot be wholly supervised as it does not have a set of labelled data for training. However, it is also not unsupervised because of the cumulative reward which must be maximized by the agent. No supervisors are involved in the learning process, it is simply the reward signal that informs the agent how well it performs[54]. A crucial factor in RL is time; the learning process is performed sequentially with delayed environmental feedback. This feedback may be shortly delayed, therefor the agent might receive the feedback only after the successful completion of a task, e.g. if the agent's objective is escaping a maze, the feedback could occur at the end. Figure 2.15 represents RL framework.

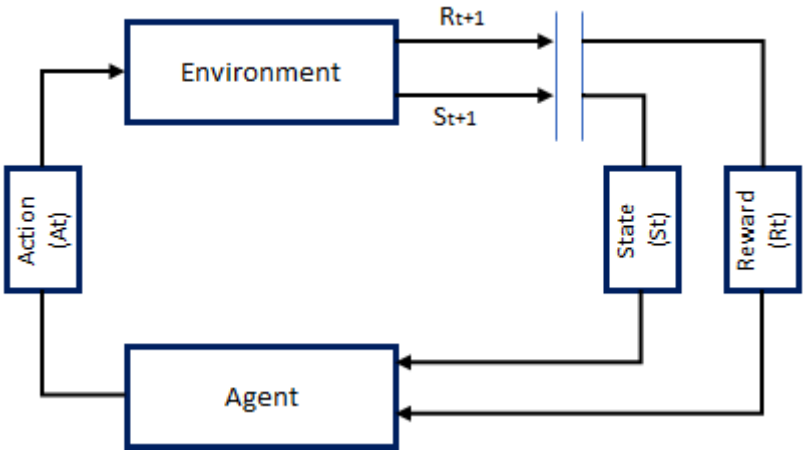


Figure 2-15: Reinforcement framework

The interaction between the agent and the environment, as shown in Figure 2.15, is performed over a sequence of discrete-time steps. At time step *t*, the agent receives an idea of the state S_t of the environment and consequently carries out an action A_t. At time step $t + 1$, the agent receives a reward *Rt* which is usually a real number and finds itself in the new state S_{t+1}. The set of sequences of the form state-action-reward is termed history. A naive attempt to pick the best action to take for some time step *t* would be completely based on history. However, this type of approach can fail in real-world problems because of its expansive history. Rather, the state, which encapsulates all current information, is used to take subsequent decisions. The environment state is a private representation of the environment based on when the next state and reward are issued.

B. Bayesian Optimization and Gaussian Process

As a method, Bayesian optimisation has been shown to successfully solve computationally expensive functions in order to find the extrema [55]. The method can be used to solve

functions without closed-form expressions, as well as for calculating expensive functions. When there is difficulty in evaluating the derivatives or the function is non-convex, the optimisation aims to determine the sampling point's maximum value for an unknown function.

$$X' = \arg\max_{X \in A} f(x) \tag{2.11}$$

where A represents the search space of x. Bayesian optimisation is derived from Bayes' theorem [56], i.e., from evidence data *E*, the model *M* posterior probability *P*(*M*|*E*) is proportionate to the probability of *P*(*E*|*M*) of overserving *E* given model *M* multiplied by the prior probability of *P*(*M*)

$$P(M \mid E) \propto P(E \mid M)P(M) \tag{2.12}$$

This formula represents the central concept of Bayesian optimisation, which is to combine the previous function distribution *f(x)* with the sample information to determine the function posterior. Next, the posterior information is used, according to a criterion, for finding where function *f(x)* is maximised. Utility function *u* represents the criterion, also known as the acquisition function. Function *u* determines the subsequent sample point to maximise expected utility. When performing a search of the sampling area, both exploitation (sampling from those with high values) and exploration (sampling from those with areas of high uncertainty) should be considered [55]. This will lower the sampling number. In addition, improvement to performance can be achieved even though the function may have many local maxima. As well as the sample information, Bayesian optimisation relies upon the previous distribution of the function *f*, which is essential in the statistical inference of the posterior function *f* distribution. A previous distribution need not be either entirely or partially objective but based on subjective belief. The usual assumption is that the Gaussian process is a good fit for the prior distribution of Bayesian optimisation since it is easy to handle and highly flexible.

In the machine learning field, the Gaussian process is a method produced based on Bayesian learning theory and the Gaussian stochastic process. The Gaussian process is a generalisation of the Gaussian probability distribution of random variables which are vectors or scalars (for a multivariate distribution). A stochastic process controls function properties [55]. The Gaussian process is stochastic in nature, as any finite sub-collection of random variables possesses a multivariate Gaussian distribution. The Gaussian process makes the assumption that similar inputs produce similar outputs, and therefore proposes a statistical model of the function. As with Gaussian distribution defined by covariance and mean, the Gaussian process is defined by

its covariance function k: $x \times x \rightarrow \mathbb{R}$ and its mean function m $x \rightarrow \mathbb{R}$. The Gaussian process is denoted here as

$$F(x) \sim GP\ (m(x), k(x, x')) \tag{2.13}$$

The Gaussian process is not the same as Gaussian distribution, i.e., the probability density function $f(x)$ for an arbitrary x is then not a scalar but rather a normal distribution function of all potential values of $f(x)$. For simplicity, suppose the mean function of the Gaussian process $m(x) = 0$, a popular choice for the covariance function k is the exponential square function.

$$k(x_i, x_j) = \exp\left(-\frac{1}{2}\ ||x_i - x_j||^2\right) \tag{2.14}$$

where x_i and x_j respectively represent the i^{th} and j^{th} samples. When x_i and x_j are proximal, the value of k (x_i, x_j) approaches 1; otherwise, it approaches 0. Thus, when two sampling points get closer, they have a mutual influence and a strong correlation; the mutual influence is weak when they are more distant from each other.

The following is how the posterior distribution of $f(x)$ is determined. Firstly, sample t observations as the training set $D_{1:t}=\{X_n,f_n\}^t_{n=1,}$ $f_n=f(X_n)$. Assume the function values f are drawn based on the multivariate normal distribution $f \sim (0,\mathbf{K})$, where the elements inside **K** are calculated by (14).

$$K = \begin{bmatrix} k(X_{1,} X_1) & k(X_{1,} X_2) & \cdots & k(X_{1,} X_t) \\ k(X_2, X_1) & k(X_2, X_2) & \cdots & k(X_2, X_t) \\ \vdots & \vdots & \ddots & \vdots \\ k(X_t, X_1) & k(X_t, X_2) & \cdots & k(X_t, X_t) \end{bmatrix} \tag{2.15}$$

Function k measures the degree of approximation between two samples. The diagonal element $k(x_i, x_i) = 1$ without considering noise effects.

Acquisition Function

After the posterior distribution of the objective function has been obtained, Bayesian optimisation utilises the acquisition function u to determine the maximum of the function f. It is usually assumed that the high acquisition function value corresponds to the large value of the objective function f. Therefore, maximising the acquisition function is the same as maximising the function f:

$$X' = arg \max_{X \in A} u(X|D) \tag{2.16}$$

There are several types of AF such as Probability of improvement, Expected improvement and GP upper confidence bound (GP-UCB) [57].

Performance Estimation

Performance estimation is a method for measuring the goodness in performance of the generated architectures by the search strategy algorithms. The performance estimation evaluates architecture and sends the evaluation metrics back to the search strategy algorithm, which in turn derive better and more improved architecture candidates for further evaluation. Assume that *S* is a search technique that samples an architecture *A* then the generated model train from scratch and then evaluates its accuracy on unseen data. The validation accuracy is used as a signal to update the search strategy [9].

2.5.2. Efficient Neural Architecture Search (ENAS)

Efficient Neural Architecture search has attracted a lot of interest because of its ability to generate high-performing CNN architecture in a very short period. According to Pham et al. [11], ENAS enhanced NAS [58] efficiency by forcing all sampled models to share weights, reducing the need to train each model from scratch to convergence by employing a similar search strategy but imposing a search space and parameter sharing constraint. This section will focus on describing ENAS components, its search space, search strategy and performance estimation strategy.

ENAS Weight Sharing Approach

ENAS [11] regards an CNN architecture as a subgraph of a single global directed acyclic graph (DAG), which may be thought of as a superset of all the ENAS-sampled child models. The graph's nodes represent local computations, while the graph's edges reflect information flow. As a result, if a calculation between two nodes is previously performed at one time while sampling an architecture, the computation or weight can be employed during the training of another network. As a result, two architectures can share the same parameters. This parameter sharing strategy is the driving force behind ENAS efficiency since it solves NAS's key shortcoming. When the nodes in the DAG are sampled again, ENAS preserves those weights and shares them. As a result, if an edge is shared by many architectures, the tensor is shared as well. One concern is that if we train the architecture independently, the weights will be different since the topologies of the architectures differ and sharing weights would result in inferior results. ENAS [11] argues that the purpose for this technique is multitask learning, in which

diverse tasks are assigned to the neural network and the sampled neural architectures tend to generalize well as a result of this strategy. Although there are no theoretical arguments that parameter sharing can identify the local optimum, it is nonetheless employed in the most recent neural architecture search approaches such as DARTS [59]. In practice, ENAS method training only one CNN network during the search stage which named Supernet. Supernet is pre-defined CNN architecture which includes all search space's operations and the generated CNN by ENAS method during the search stage are using trained weight of Supernet.

ENAS Search Strategy

For sampling CNN architectures, ENAS uses a Long short-term memory (LSTM) controller with 100 hidden units. The controller generates architectures via softmax classifiers in an autoregressive manner: the previous decision is used as input for the following step. The controller network is given an empty embedding as input in the first stage. In ENAS two types of learnable parameters exist, the LSTM parameters θ and the super net's parameter known as shared weight ω.

ENAS's training approach is divided into two interlocking sections. The first phase trains ω, which will share among child models on a full run through the training data set. The second phase of training is tuning the LSTM controller parameters, for a certain number of steps. In each epoch of LSTM, the controller generates 10 child model and evaluate each one on the validation set. Then the validation accuracy uses as a reward to update the LSTM parameters. To maximize the anticipated reward function, i.e. validation accuracy, a policy-based reinforcement learning technique is used. This parameter update aims to increase the controller's ability to generate better decisions with greater validation accuracies [11].

ENAS Search Space: Macro Search Approach

The authors proposed two search spaces used by the RNN controller trained with RL. The first search space is called Macro where the RNN controller searches for an entire network. The controller in this case makes two decisions: 1) which the previous node to connect, and 2) what computation operation to use. The first decision is whether to allows the model to form skip-connection, whereas the second decision is to select one operation from a predefined collection. There are 6 such operations, i.e. convolutions with filter sizes 3×3 and 5×5, depthwise-separable filter sizes 3×3 and 5×5, max pooling and average pooling with kernel size 3×3, to use for creating a particular layer in CNN architecture [11]. Figure 2.15 depicts an example of designing CNN model with 4 layers in Macro search space.

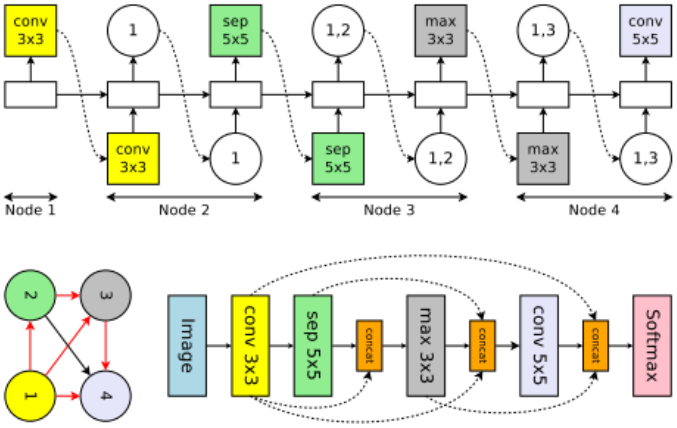


Figure 2-16:An example of designing CNN model with 4 layers in Macro search space. Top: the output of RNN [11].

As shown in Figure 16, the controller sampled only on decision which is convolution with a 3×3 filter for designing first layer. Since this is the first node, the controller only samples the operation because there is no other node to connect. To build the next layer, the controller sampled depth-wise separable with a 5×5 as an operation and node 1 to be connected. The controller designed node 3 by sampling MaxPooling 3×3 on the output of the Layer 2. Then, the result of this operation is concatenated along the depth dimension with Layers 1 and 2. This process repeat again for design last layer. As shown, the controller sampled nodes 1 and 2 with 5×5 convolution for generate layer 4, and hence the generated child model ends up with SoftMax layer.

ENAS Search Space: Micro Search Approach

In contrast, the Micro search space is cell-based where instead of searching for an entire architecture, the RNN controller generates cells, a unit that contains operations any the connections between them. It can be seen as a *micro* architecture. Two types of cell, i.e. normal cell and reduction cell, are generated by the controller. Normal cells are meant for feature extraction whereas reduction cells are meant for downscaling features. Each cell, either normal cell or reduction cell, consists of 5 nodes, and each node consists of 2 computation operations. Hence, for generating a cell, the controller selects two previous nodes as inputs and two operations from the collection of five operations in the search space, i.e. identity, separable convolutions with kernel size 3×3 and 5×5, and average pooling and max pooling with kernel size 3×3.

Figure 2.17 depicts the search for a convolution cell with 4 nodes. In particular, let assume that nodes 1 and 2 are already constructed. Let L1 and L2 be the outputs of these two nodes. For building node 3, the controller samples separable convolution 5×5 and identity as operations and the controller samples (node 2(L2) , node 2(L2)) as input for sampled operations

in node 3. This mean: L3 = sep conv 5×5(L2) + id(L2). Later, the controller designed node 4 by sampling average pool 3×3, and Sep convolution 3×3 as operation and node 3 (L3) and node 1 (L1) as input. This mean: L4 = average pool 3×3 (L3) + Sep convolution 3×3 (L1). Since all nodes except node 4 were used as inputs to another node, the only loose end, node 4, is addressed as the output of the cell. Once the optimal cells (normal and reduction) are found, the entire network consists of cells that are stacked one on top of another to form of the whole architecture.

$$R1 = num\_layers \,//\, 3 \quad (2.17)$$

$$R2 = (2 * R1) + 1 \quad (2.18)$$

$$Reduction\ Location = [R1, R2] \quad (2.19)$$

where R1 is the index of Reduction cell 2 and R2 is index of the Reduction cell 2, and num_layers is a predefined hyperparameter which is determine the number of repeating cells per each ENAS architecture including (Reduction and Normal) cells.

e.g.

$num\_layers = 15$

$R1 = 15 \,//\, 3 = 5$

$R2 = (2 * 5) + 1 = 11$

Thus, Reduction Location = [5, 11], which will determines the positions of two reduction cells in ENAS architecture which are layer = 5 and layer = 11. In the other word, all layers of ENAS will fill by Normal cell except layer (5 and 11) are Reduction.

In the end, in both Micro and Macro cases, after generating a set of CNN architectures, the architecture with the highest validation accuracy will be select, i.e. the optimal architecture trained from scratch on the dataset of interest.

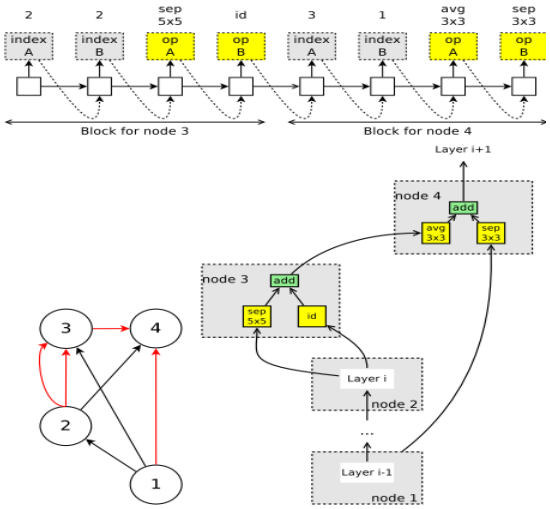


Figure 2-17: Illustrate of Micro search space. Top: The output of Controller. Bottom Left: corresponding DAG. Bottom Right: The Convolutional cell sampled by controller [11]

2.6. Summary

This chapter discussed the background of medical imaging and its application as a dependable tool for diagnosing, treating, and monitoring patients to provide readers with a fundamental grasp of the methods used in medical diagnosis. The ultrasound is a commonly used imaging modality for diagnosing breast cancer. Therefore, this chapter reviewed the essential background about ultrasound imaging in breast cancer diagnosis.

This thesis is concerned with developing a CAD system that relies on CNN models to improve breast cancer abnormality classification. In this vein, we introduced the building blocks of all components of CNN together with the difficulties of designing a manual CNN architecture design. In addition, we also introduced the fundamentals of designing automatic CNN architectures based on NAS and ENAS approaches in detail. This chapter serves as an introduction to the rest of the thesis by providing fundamentals of medical imaging and CNN architecture components as well as manual and automatic design on CNNs. In the next chapter, we will review existing work related to the thesis objectives.

Chapter 3. Literature Review

This chapter aims at reviewing the existing state-of-the-art solutions in the literature for breast lesion recognition from ultrasound images. We intend to conduct the review mainly from two dimensions: (a) existing solutions for breast lesion recognition from ultrasound images and (b) existing solutions regarding automatic CNN architecture design that is closely linked to the objectives of this thesis. We focus our attentions to both the application problem at hand and the advances of techniques in deep learning. In some sections, we will expand our scope of the review when there is insufficient published work in the literature.

The chapter is therefore organized into two main sections accordingly. The first section consists of the review of recent research work in breast lesion recognition from ultrasound images with specific attention to deep learning neural network solutions. The second main section focuses on particularly the most recent development in automatic CNN architecture design. At the end of the chapter, we summarize the related works, identify the gap in the literature, and make the link to our solutions presented in Chapters 4, 5 and 6.

3.1. Breast Lesion Recognition from Ultrasound images

As outlined in Section 2.2, medical image analysis has undergone three phases: the initial phase of automatically detecting known morphological features, the pattern recognition approach of extracting image-content features, and more recently CNN designs for feature extraction and classification. The development has also influenced ultrasound image analysis for breast lesion classification.

Prior to the detailed literature of using deep learning, we briefly review the key traditional machine learning techniques developed for breast lesion classification. Costa et al. [60] used a set of eight morphological features, i.e. convexity, elliptic normalized skeleton, proportional distance, elliptic normalized circumference, depth-to-width ratio, lobulation index, average distance, and normalized residual value, that were measured and recorded, and were then used for training a neural network. A small dataset of 100 images (50 benign and 50 malignant) was used for training and testing. The final configuration of the three-layer architecture of the ANN (5 neurons, 5 neurons and 1 neuron) was empirically determined. The overall accuracy of 95.55% was reported. After applying regularization and early stopping in the network training, the performance of the model was improved to 96.98%. Despite the seemingly impressive test

results, due to the small data set used, the investigation is only indicative, and the models obtained are likely to overfit. The proposed methods are therefore yet to demonstrate their effectiveness in real-life clinical practice.

Researchers have found that although morphological features have their advantage in feature explainability, classification models obtained from the features are unlikely to match the diagnostic accuracy made by experienced radiologists [61]. The ultrasound image is full of visual clues expressed by image textures that may not be directly mapped to the known morphological features or may yet be known by experienced radiologists. Therefore, utilizing machine learning and computer vision techniques to extract such image-content based features may have potential benefits and hence has become another popular approach. To ensure that useful feature information is extracted conveniently by specially designed algorithms, the input ultrasound images may need to undergo a pre-processing procedure. Evidence has shown that such an approach does provide added value [62]. The authors in [63] proposed a method to extract two types of features such as morphological, which is calculated from local characteristics of the tumour such as the margin and shape and named sonographic, and texture features, which were then used to train three classification models (Support Vector Machine (SVM), k-Nearest Neighbour (kNN), and ANN). A dataset of 321 ultrasound images was used for training and testing. The highest accuracy of 86.92% from the SVM classifier was reported. Similar works on other types of tumour classification have also been reported. A small scale investigation has been reported in [64] for ovarian tumour classification. The authors used Local Binary Patterns (LBP) for extracting texture features from ultrasound images and SVM as the classifier to distinguish benign and malignant ovarian tumours. The authors in [65] explored the effectiveness of a feature vector extracted from the Fourier Transform spectrum domain of the original ultrasound images when several different classifiers are used for solving the same problem. Another more recent attempt was made to evaluate the effectiveness of a range of texture-based features in distinguishing benign and malignant adnexal tumours [66]. Moreover, a number of existing methods presented in [67] and [68] that proposed for breast cancer classification by using machine learning algorithms. Researchers propose a number of CAD systems for breast cancer classification using ultrasound images [69], [70]. A few of them concentrated on the segmentation step (cropping RoI), followed by feature extraction, and a few extracted features from raw images [71].

Recently, many attempts have been made on cancer recognition from ultrasound images using deep learning algorithms, particularly CNN which provides an end-to-end classification model, i.e. all image pre-processing, feature extraction and classification are done by the same

CNN). As described in Section 2.3.3, existing CNN solutions can be categorised into *handcrafted CNNs* and *automatically searched CNNs*. As for breast lesion classification, the handcrafted CNNs can be further categorized into the adaptation of CNNs for natural images with transfer learning and specifically designed CNNs. These two subcategories of methods will be reviewed next. The automatic neural network search will be reviewed in full in Section 3.2.

Adaptation of Existing CNN with Transfer Learning

Due to the model complexity and lack of annotated datasets, most of the existing research efforts focus on adapting and customizing existing CNN architectures that were designed for classifying everyday objects from photographic images. Given the complex structures, classification models built on a CNN tend to have many parameters (i.e. weights on the connection links) that need to be adjusted during the training and validation process. To avoid model overfitting, such models must be trained on a large number of images of different variations. Although there are some publicly available repositories for medical images, image datasets in those repositories are limited in numbers comparing to the well-known ImageNet [72]. Acquiring sufficient and quality medical images (ultrasound images in particular) with reliable class labels can face practical difficulties. Due to these difficulties, many researchers have explored the use of Transfer Learning (TL) as a solution to overcome the image shortage issue ([32], [36] and [73]). Transfer Learning is a technique that initializes weights in a neural network with the values pre-trained by another neural network model on the images from the same domain of application or from completely different domains of application.

For breast cancer classification, most researchers used the CNNs with transfer learning, i.e. the trained model parameters (i.e. weights) inherited from the trained classification models for everyday objects. These models are then further trained using the breast lesion ultrasound images at hand. Han and Kang proposed adopted GoogLeNet with minor modifications (i.e. the removal of two Auxiliary classifiers) as the backbone architecture for training a classification model [74]. The average accuracy of 90% was reported from a 10-fold cross valuation over a dataset of 7,408 (4254 benign and 3154 malignant) images. In [75], ten commonly-used CNN architectures (i.e. ResNet101, ResNet50, ResNet18, InceptionV3, InceptionResNetV2, GoogleNet, MobilenetV2, Xception, DenseNet201, and SqueezeNet) designed for natural images were tested for classifying breast lesions over a public domain dataset of ultrasound images (133 normal, 210 malignant and 437 benign). The mentioned CNN architectures used as a transfer learning by modifying each one by replacing last fully-

connected layer with new fully-connected layer which consist of three nodes (number of classes). The result showed that ResNet101 outperformed the other pre-trained models. More recently, the authors of [76] designed a generic CNN model framework based on VGG19 with transfer learning for both breast and thyroid cancer classification from ultrasound images. Based on the same framework, the authors developed two separate CNN models (TNet and BNet) respectively for breast lesion and thyroid nodule recognition, and also tested a CNN model (TBNet) when images of breast lesion and thyroid module were combined. Test results on 672 breast lesion images and 719 thyroid nodule images show an overall accuracy of 89% for breast lesion classification and overall accuracy of 86.5% for thyroid nodule classification. The trained thyroid model (TNet) also performed well in classifying the breast lesions. The combined model (TBNet) achieved an overall accuracy of 82%. The TNet and BNet models even outperformed experienced radiologists when tested on an external data set.

In conclusion, adapting the pre-trained CNN architectures for breast lesion classification from ultrasound images has certainly shown their potentials, and reduced the severity of the problem faced by lack of training data. However, these existing architectures were originally designed for object classification from natural images which are very different in many ways from ultrasound images of internal organs of the human body. In addition, most of these networks are complex and contain large number of hyper-parameters that require sufficient number of images to tune. In other word, these challenges provided the motivation to other researchers to consider designing a customized CNN networks for breast lesion classification in US images.

Manual CNN Designed for the Purpose

Few attempts have been made to manually design CNN architectures for breast lesion classification in US images task. Byra et al designed a CNN architecture of three convolutional layers and two fully-connected layers [77]. Each convolutional layer uses 32 filters of filter size (3×3). Each convolutional layer is followed by Relu and 2×2 MaxPooling. Five-fold cross-validation was conducted on a dataset of 166 malignant and 292 benign images, and the average accuracy reached 83.0% (sensitivity 82.4) and Area Under Curve (AUC) was 0.912. Another specifically designed CNN architecture was reported in [78]. This architecture consists of four convolutional layers with different filter sizes and the number of filters (11×11(32), 7×7(64), 5×5(128), 3×3(256)), respectively for each of the four layers in that order as shown in Figure 3.1. To reduce model overfitting, several regularization methods (Batch normalization, data augmentation, dropout, and L_2 regularization) were used. Five-fold cross-validation was

conducted on a data set of 641images (413 benign and 228 malignant) for balancing the dataset, data augmentation was used. They randomly selected 185 and used flipping; thus, each class became (413 images). The average overall accuracy of 92.05% was achieved.

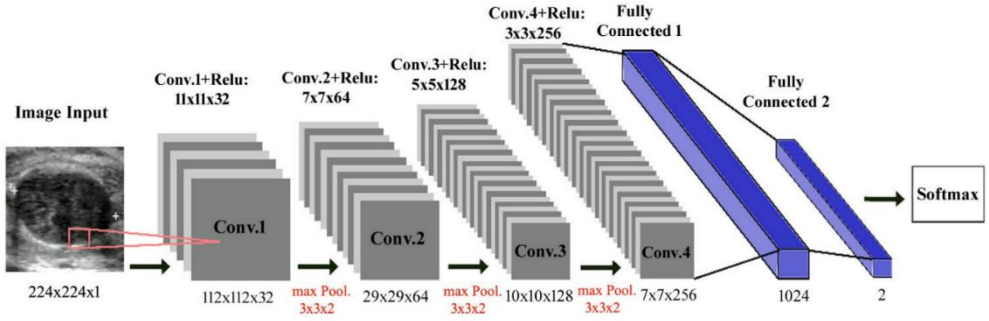


Figure 3-1: CNN4 architecture [78]

Recently, an architecture known as Fus2Net was proposed in [79]. The architecture starts with three normal convolutional layers with filter sizes and the number of filters (3×3(32), 3×3(32), 3×3(64)), respectively. Then, the rest of the architecture consists of block1 module and block 2 modules. Each block consists of one module, and the module consists of several convolutional layers with different filter sizes mainly (1×1, 3×3, 1×7, 7×1), more details about the architecture showed in Figure 3.2. Block 1 consists of three different modules with two branches, each branch includes different convolutional layers in terms of filter size and the number of filters. Both branched module 1 was designed by a 3×3 convolutional layer followed by the Max-pooling layer. While the branches of module 2 consist of 6 normal convolutional layers, including all mentioned filter sizes. However, module 3 is the same as module 1 only Max-pooling is changed to average pooling. The output of the Block 1 is the concatenation of all the feature maps of all the mentioned branches. Block 2 consists of four branches, three of them designed by different convolutional layers, while the fourth one is the residual layer from block 2 input to the output of the same block. Each convolutional layer is followed by batch normalization and Relu. They used a breast ultrasound dataset which consists of 100 images in each class of 50 cases. Data augmentation was applied for expanding the training set. Fus2net models achieved an accuracy of 92%, a sensitivity of 95.65%, and a specificity of 88.89%.

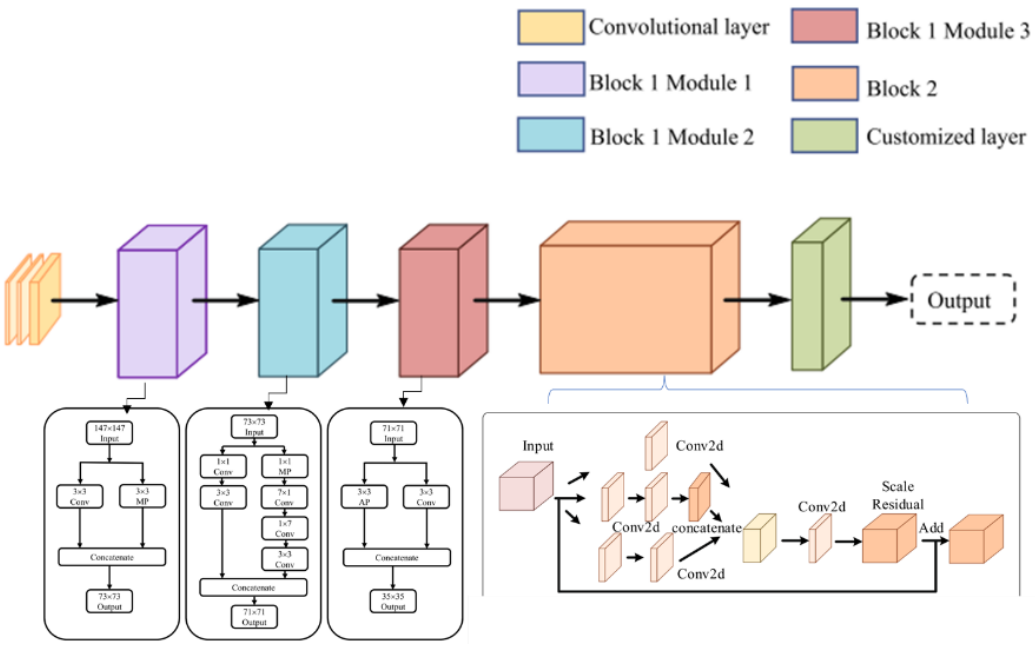


Figure 3-2:Fus2Net Architecture [78]

In a separate study [80], three pre-trained models based on Inception V3, ResNet50 and Xception architectures, respectively, the models trained on a specially designed CNN with three convolutional layers (CNN3), and the models trained on a collection of handcrafted features were compared. As presented in Figure 3.3, CNN 3 consists of three convolutional layers with filter sizes and the number of filters (3×3(32), 3×3(64), 3×3(128)) with stride one, followed by Batch normalisation and Relu. After each convolutional layer, there is the MaxPooling layer (2×2(stride 2)), and the final layer followed by fully-connected with (256 nodes) and followed by GAP layer. The handcrafted features used in [80] are as follow: (18 First-order texture features { Entropy, Minimum, Energy, Mean Absolute Deviation (MAD), 90th percentile, Maximum, Mean, Median, Interquartile Range, Range, 10th percentile, Robust Mean Absolute Deviation (rMAD), Root Mean Square (RMS), Standard Deviation, Uniformity, Kurtosis, Variance, Skewness }, 12 Texture features { Variance, Skewness, Kurtosis, Energy, Contrast, Correlation, Homogeneity, Variance, Sum Average, Entropy, Dissimilarity, Autocorrelation} and 8 Morphological features { Circularity, Maximum chord length, Second moment, Compactness, Roughness, Orientation, Radial distance standard deviation, Elongation }). To evaluate all models, 10-fold cross-validation over a dataset of 2,058 images (688 malignant and 1370 benign masses) was conducted. The three existing Inception V3, ResNet50 and Xception CNN models with transfer learning achieved an average accuracy of 85.13%, 84.94% and 84.06%, respectively. The overall accuracy of CNN3 reached 74.44%, and the highest overall accuracy among the models built on the handcrafted features

was 70.55%. With a reasonably large data set, this study's results are more reliable than some other better accuracies as mentioned earlier.

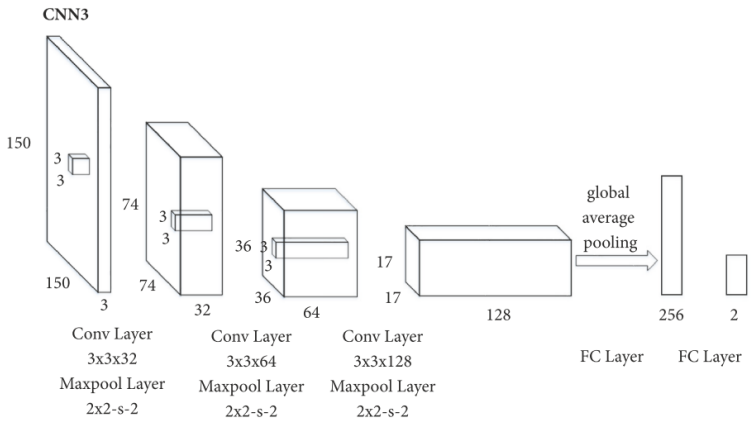


Figure 3-3:CNN3 architecture [79]

CNN Networks for Other Cancer Types and Medical Image Modalities

Ultrasound images frequently used in clinical diagnosis of different types of cancer. convolutional neural network powered by ultrasound images is a significant concern for the scientific community. Authors of [81] proposed CNN based method for thyroid cancer classification from ultrasound images. They proposed VGG16T model based on modifying VGG16 by moving Bach normalisation to before Relu and additional dropout layer. Thyroid dataset used in this paper includes 800 ultrasound images (400 Benign and 400 Malignant) for modelling part and 200 images used as external test. 10 cross-validation have been used in this paper and the result showed that VGG16T achieved 86.43% in overall accuracy which outperformed default VGG16 by around 5% on internal test. In [82] DCNN used for designing CAD system for liver cancer classification from ultrasound images. For designing this model, the authors used ultrasound dataset which consists of four classes (cyst (338 images), HCC (241 images), haemangioma (279 images), and metastatic liver cancer (122 images). Since the RoI size of liver cancer small they set input size 64×64 and modified VGG16 by reducing layers to 10 layers for avid input vanishing issue. Then they used 10-fold cross validation for evaluating their models and the model achieved 88% on overall accuracy.

More broadly, CNN has been attempted for other medical image modalities. In [83], cross-domain transfer learning was attempted to improve prostate cancer detection accuracy from ultrasound images. The Inception V3 architecture was pre-trained on two different publicly available datasets respectively: (a) ImageNet dataset of over 1.2 million images of natural objects of 1,000 classes, and (b) BrCa dataset of 8,000 cytology images of breast cancer. The

authors reported that the pre-trained models followed by further training with a prostate image dataset improved classification accuracy, and the model pre-trained with the images from the BrCa dataset improved the accuracy rate better than the pre-trained model with the natural images from the ImageNet dataset.

A comparative study was conducted upon three milestone CNN architectures (i.e. LeNet, AlexNet and GoogLeNet) for classifying medical images of various modalities [30]. Tests on an open-source dataset containing 37,698 cases of five different modalities (i.e. CT scan, MRI, X-ray, PET, and US) showed overall accuracies of 59%, 74% and 45%, respectively, for the three architectures. Based on the comparatively good performance of the AlexNet models, they modified the AlexNet architecture by dropping the last convolution layer and ignoring dropout in both fully-connected layers. Then the classification accuracy increased to 81%. In [17], a faster R-CNN architecture has been adapted for detecting thyroid papillary cancer. The improved faster R-CNN concatenated convolution layers 3 and 5, and a spatial constrained layer has been added before the output layer. The model was trained and tested on a set of 300 thyroid ultrasound images, and overall accuracy of 93.5% was reached.

Most of the summarised methods depending on transfer learning for designing CNN models for cancer classification in general and breast cancer especially. Moreover they only used only small dataset for evaluating their methods which is not enough for measuring the generalisation power of the model. Therefore, in our research, we aim to explore optimal CNN architectures specifically suitable for breast lesion classification from ultrasound images and compare their performances against those by the customized existing architectures with transfer learning.

3.2. Automatic CNN Architecture Search

After an extensive search, we have found very little existing work on automatic neural network architecture search for breast lesion classification from ultrasound images. Given the focus of this thesis is on developing automatic CNN architecture solution for ultrasound image classification, rather than expanding our scope of review from some anchor articles in the problem domain, we take a rather *top-down* approach by first reviewing some original works in this field for natural images and then investigating any existing work in the problem domain.

As mentioned in Section 2.5, Neural Architecture Search (NAS) is the most common automatically search technique proposed recently. In this approach, to limit the size of the search space, the overall structure of the network is manually predefined. The controller then searches for optimal hyper-parameters for the backbone network in the predefined search space

set such as filter size, stride, the number of filters in the convolution layer, and skip connection [10]. In order to increase the expected validation accuracy of the newly generated architectures, the search strategy depends on the feedback from the performance estimation. The performance estimation evaluates the goodness of the generated architecture based on the validation data set which is separated from the training set. The validation accuracy is then used as a reward to update the controller's RNN hyper-parameters. Some evidence shows that the NAS approaches have outperformed manually designed CNN architectures for image classification ([10] and [58]), which intrigued our interest in taking this approach for breast lesion classification.

Most of the modern CNN architectures such as GoogleNet and ResNet are based on a multi-branch structure (comparing to a single sequential structure of layers). Hence, most of the automatic search for CNN architectures is also based on multi-branch structures. To reduce the complexity of the search space, they are searching for optimal cells and then stack the cells together for building the whole CNN architecture. The controller determines the connectivity between blocks based on the skip connection concept proposed by ResNet [42] that we mentioned in Chapter 2.

Zoph et al [58] proposed an approach to search for optimal CNN architectures on the dataset of interest. This method search for cells or blocks rather than for the entire architecture. Furthermore, this approach optimizes two types of cells i.e. normal cell and reduction cell. For building the whole architecture, a list of these cells has been stacked in a predefined manner. The advantage of this approach is to reduce the complexity by searching only for optimal cells that can be transferred to another dataset by varying the number of blocks. The researchers reported that optimal cells were generated by using the CIFAR10 dataset. The optimized cells were then used to design a new CNN architecture, and the model was trained on the ImageNet dataset for image classification.

Progressive neural architecture search (PNAS) reported in [85] employs a heuristic search technique for searching block structures, starting from shallow models and evolving to complex ones. This approach defines a fixed number of cells, and each cell includes 2 operations among the 8 predefined ones. The search models start by building, training and evaluating all possible 1-blocks. Then expanding cell to 2-blocks and so on. The sequential model-based optimization (SMBO) strategy has been used for optimizing the search process by avoiding direct search in the whole space of cells. The PNAS approach achieved accuracy comparable to NAS [85] but much faster due to the fact that full training takes much more time than the performance prediction of designed cells.

As a further development of NAS and PNAS approaches, Efficient Neural Architecture Search (ENAS) proposed in [11] searches for an accurate CNN architecture efficiently. ENAS significantly improves on NAS [58] by requiring all child models to share weights instead to train each model to convergence. ENAS generate a CNN architecture that achieves a 2.89 per cent test error, which is comparable to NAS's 2.65 per cent, while reducing computational cost substantially less than NAS. Furthermore, the researchers modified the ENAS macro search by fixing skip-connection and searching only for operations [86]. Since automatically searching for CNN architecture has been successful in recognising objects in natural images, researchers are now attempting to apply ENAS to medical datasets, such as MRI segmentation [87]. In [88] the researchers used five publicly available datasets (Paediatric images, CXR images, OCT images, HAM 10000, and MESSIDOR) for searching and modelling by using the Google Cloud AutoML platform. Their findings show that automatic search methods can provide competitive classifiers when compared to manually created DL models.

Recently, the NAS approach was applied in medical image segmentation [89]. Three distinct basic operation sets on the search space were defined to discover two cell architectures (DownSC and UpSC) for semantic image segmentation. For searching for architecture PASCAL VOC2012 dataset has been used and the optimal architecture trained on three different medical datasets used (Promise12, Chaos, and ultrasound nerve), then the generated model achieves superior performance with far fewer parameters compared to the U-Net. Meanwhile, the authors in [90] proposed a framework to automatically search for the 3D deap learning models for classifying 3D chest CT scan images. This is named a differentiable neural architecture search (NAS). The result showed that the searched model outperformed manually designed 3D models for the same dataset. Another paper used NAS for optimising CNN architecture for skin lesion classification [91]. They used the same search strategy that was proposed in [92], which is a greedy search algorithm named hill-climbing, but they used different operations in search space. They used Net2WiderNet, Net2DeeperNet and operation for multi-branch. Net2WiderNet is enabled for expanding layers by searching for the number of filters per convolutional layer; Net2DeeperNet determine the depth of the network by the number of layers, while for the multi-branch they searched for skip-connection. Their proposed search approach generated CNN architecture for skin cancer classification which has 20 times fewer parameters compared to hand-crafted CNN. Table 3.1 summarizes the work and highlights some key characteristics of the existing work.

There is no study up to this day in applying ENAS to search for CNN architecture for ultrasound images in general nor for breast lesion classification. This means that there is a gap

in the literature to fill on the one hand, and an interesting research question to ask: *will ENAS success for natural image classification can be realized for breast lesion classification from ultrasound images?* If this is feasible, it means that a significant step of progress can be made towards automatic modelling of breast lesion ultrasound images, and even for other types of lesions beyond the breast lesions.

Table 3-1: Existing Research Work Using Deep Learning CNN Models

Methods	Dataset	CNN Designed	Training method	Accuracy	Search strategy
[74] 2017	US Breast	Hand-Crafted CNN	Transfer Learning	90%	-
[75] 2021	US Breast	Hand-Crafted CNN	Transfer Learning	96.27	-
[76] 2020	US Breast	Hand-Crafted CNN	Transfer Learning	89%	-
[77] 2017	US Breast	Hand-Crafted CNN	From Scratch	83%	-
[80] 2018	US Breast	Hand-Crafted CNN	From Scratch	74.44%	-
[79] 2021	US Breast	Hand-Crafted CNN	From Scratch	92%	-
[78] 2022	US Breast	Hand-Crafted CNN	From Scratch	85.98%	-
[10] 2017	ImageNet	Auto-CNN	From Scratch	96.35%	RL
[58] 2018	Cifar10 for and ImageNet	Auto-CNN	From Scratch	97.6%	RL
[85] 2018	Cifar10	Auto-CNN	From Scratch	96.59%	(SMBO)
[11] 2018	Cifar10	Auto-CNN	From Scratch	97.11	RL
[59] 2018	Cifar10 for search ImageNet for Modelling	Auto-CNN	From Scratch	97%	Gradient-based
[86] 2019	Cifar10	Auto-CNN	From Scratch	95.43%	Gradient-based
[90] 2020	CT scan chest	Auto-CNN	From Scratch	96.88%	RL
[93] 2021	microscopic images of blood	Auto-CNN	From Scratch	93.33%	BO
[89] 2019	Promise12, Chaos, US nerve	Auto-CNN segmentation	From Scratch	mIoU 99.2%	RL
[91] 2020	Skin Cancer	Auto-CNN	From Scratch	75.75%	hill climbing strategy

3.3. Summary

This chapter aimed at reviewing the literature on existing work on the computational aspects of analysing breast ultrasound images to develop machine learning algorithms that support

radiologists in their decisions with regard to diagnosing benign and malignant breast tumours. We started by reviewing the existing work on ultrasound image analysis in general. Then, we focused on reviewing the proposed approaches for breast cancer classification from ultrasound images. Hand-crafted machine learning approaches for breast cancer were reviewed first followed by surveying existing work using CNN for designing an accurate CAD system. One of the main differences between handcrafted approaches and CNN models is the fact that training CNN architecture required a large dataset of images. That is why most of the existing CNN models for breast cancer were designed based on transfer learning. Thus, we could only find a small number of papers that manually designed CNN architectures from scratch for the purpose of ultrasound image classification. Finally, the last section of this chapter was dedicated to discussing and reviewing the existing work on automatically searching and designing CNN architecture.

To the best of our knowledge, we have not encountered research work on the use of automatic searches for CNN architecture Ultrasound breast images. Even manually designed CNN architectures using ultrasound images are very few and only a small dataset from one hospital has been used in their modelling and testing protocols [78], [79] and [80]. However, [78] and [79] achieved a good overall accuracy on internal test which are 92.05% and 92% respectively, but using a small dataset from one hospital may result in producing overfitting and hence the CNN will not generalize well on data coming from different hospitals or images captured using different devices.

This thesis is going to utilise automatic CNN architecture search approaches such as ENAS to construct over a thousand CNN architectures and select the optimal model. Moreover, evaluating generated CNN on different breast Ultrasound image datasets, are collected from different hospitals and using different devices. We address the issues of the generalisation error (Chapter 4), of ENAS based models for breast cancer classification. Furthermore, we also contribute to improving the overall ENAS algorithm in terms of search space and search strategy (Chapters 4,5 and 6) to better optimise CNN architectures suitable for breast ultrasound images.

Chapter 4. Adapting ENAS for Breast Lesion Classification

The reviews of the background and the existing literature as presented in Chapters 2 and 3 have established two important observations. First, adapting existing CNN architectures originally designed for natural images with transfer learning to the ultrasound images of breast lesions may have its limitations in achieving satisfactory accuracy and robustness of the CNN models due to the specific characteristics of the ultrasound images. Second, designing an effective CNN architecture for classifying breast lesion in ultrasound images from scratch is difficult due to the complex nature of CNN models and a large number of hyper-parameters. Any handcraft attempts to tune the hyper-parameters to their ideal configuration will face a huge amount of challenge and prolonged period of time. Besides, medical image classification is critical task and requires careful network design. The classification models must therefore be reliable and generalizable on datasets acquired from different devices and from different medical centres.

Based on these observations and arguments, this thesis proposes a framework of automatic architecture search and automatic hyper-parameter search based on ultrasound images of breast lesions outlined in Chapter 1. As the first of three core chapters to present the framework in detail, this chapter is designated to the adaptation of ENAS for breast lesion classification from ultrasound images. This chapter consists of four main parts (Sections 4.1 to 4.4). The chapter first starts with a general preparation for the rest of the chapter and the following chapters. Section 4.1 presents the data sets used for supporting this research. The section then describes the pre-processing operations performed on the images before modelling, including image resizing, RoI cropping and data augmentation for expanding training sets. The section also outlines the experiment platforms used, experimental protocols followed, and the performance metrics to be measured. Section 4.2 presents the adaptation of ENAS and compares the performances of ENAS models against a wide range of existing CNN models (handcrafted and adopted). Section 4.3 specifically investigates the issue of model overfitting and generalization errors of ENAS models and develop suitable solutions to reduce the generalization errors. Section 4.4 reports our own attempt in adapting an existing CNN architecture for breast lesion classification. This attempt aims at further consolidating the validity of the ENAS-based approach for CNN design. Section 4.5 identifies and discusses some open issues derived from

the results. Section 4.6 finally summarizes the major findings and contributions made in this chapter.

4.1. Preparations

4.1.1. Breast Images Data Sets

The data sets used for supporting this research are all collected from clinics. To ensure generality of findings, we aim at collecting images[1] from multiple patients, devices of different makes, and different medical centres. By complying with the practice of clinical research, we use images from one medical centre for modelling (known as modelling data set or internal data set), and images independently sampled from the same medical centre or images from different medical centres for external testing (known as external data sets). These data sets are described as follows.

Modelling (Internal) Dataset: This dataset was collected from Pudong New District Renmin Hospital, Shanghai, China, and provided by TenD AI Medical Technology Ltd, the partner of this collaborative research. The data set consists of 1,102 image lesions in total, including 726 images of benign lesions and 376 images of malignant lesions. The images were captured using US machines of the following different makes: Siemens, Toshiba, GE, and Philips. The ground-truth annotation of each US image, i.e. benignity or malignancy, is based on pathology reports.

External Test Sets: Two datasets were collected from two different sources. The first data set, known as **External_A** (also known as BUSI dataset), is a public domain dataset originally collected from Baheya Hospital for Early Detection and Treatment of Women's Cancer, Cairo, Egypt in 2018 [94]. The dataset consists of 780 images of three classes: 133 images of normal breasts without lesions, 437 images of benign lesions and 210 images of malignant lesions. All images were taken using LOGIQ E9 ultrasound system and have an assigned class label provided with the images. For this research, 133 images for the normal breasts were ignored because they are irrelevant to the purpose of this research. 355 images of benign lesions were selected after removing the images with severe artefacts such as calibre points and lines. All 210 images of malignant lesions were kept. The final dataset contains 565 images.

[1] From this point onwards, unless specified otherwise, the term image refers specifically to breast ultrasound image.

The second dataset, known as **External_B**, was also collected from Pudong New District Renmin Hospital in Shanghai and provided by TenD. The dataset was independently sampled from the hospital's image database, separate from the Modelling dataset. It consists of 300 images of benign cases and 200 images of malignant cases. The images were again captured using US machines of different makes (i.e. Siemens, Toshiba, GE, and Philips). The ground-truth annotation of each US image is based on pathology reports.

Besides the class label, every image from all datasets also has the RoI specified by domain experts via a set of coordinates of the points placed on the boundary of a lesion.

4.1.2. Experimental Platforms, Protocols and Performance Metrics

Deep learning research in general and our investigation into CNN architectures in particular require sufficient computational powers and rich collections of software tools. Therefore, a desktop workstation with an Intel® Xeon® CPU E5-2670 v3 at 2.30 GHz (48 CPUs) with 64GB RAM memory, and 2 NVIDIA GeForce RTX 2080 GPUs was used to run all experiments reported in this chapter. The experiment scripts were composed in Python with TensorFlow and Keras library for machine learning tools and Python interface to neural networks.

The procedural framework for each experiment is fundamentally an iterative process. At each iteration, the cropped RoI input images are divided into separate training and testing sets. The appropriate resizing operation is then applied. The data augmentation is applied to the training images for the training set expansion. The training images are then either directly fed into a modified CNN architecture to train a classification model or used as basis for CNN architecture optimization and then modelling. The resulting models are then applied to the test images to obtain the performance metrics. At the end, the performance metrics are aggregated into overall performance metrics for analysis. For the modelling stage using the internal dataset, we follow a 5-fold cross-validation protocol in our evaluation of classification models, a common practice in machine learning. We follow the medical research testing protocol for external tests by using a selected model that has been internally tested during the cross-validation.

The performance metrics collected from the experiments are based on four counts with respect to the known image class labels:

- True Positive (TP): the number of test images of known malignant lesions that are classified as malignant;

- True Negative (TN): the number of test images of known benign lesions that are classified as benign;
- False Positive (FP): the number of test images of known benign lesions that are classified as malignant;
- False Negative (FN): the number of test images of known malignant lesions that are classified as benign.

Based on the counts, the following performance metrics are summarised:

- Overall accuracy rate:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (4.1)$$

- True Positive Rate (TPR) also known as Recall Rate or Sensitivity:

$$TPR = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \quad (4.2)$$

- True Negative Rate (TNR) also known as Specificity:

$$TNR = \frac{\mathrm{TN}}{\mathrm{TN+FP}} \quad (4.3)$$

4.1.3. Image Pre-processing

The ultrasound images of breast lesions usually have the lesions located close to the centre of the image. However, the images can be of various sizes and resolutions. Different makes of US machines and even different device parameter settings also mean that the US images can be of various levels of brightness and image quality. The images may also contain unrelated organs and tissues in the surrounding areas of a lesion. Besides, speckle noises embedded in ultrasound images make the image blurry and unclear by nature. On the one hand, we do not intend to apply image pre-processing techniques extensively before the images are fed into CNN due to the concern that such pre-processing may enhance but may also damage or distort image-based information useful for classification. Since our research is more concerned with the CNN architectural issues, effective and efficient pre-processing is in fact, out of the scope of this research. Nevertheless, some minimum pre-processing of the US images is still needed before training CNN architecture-based models.

Image Cropping

In this research, the RoI of a US image is a region of the image containing a breast lesion. Although automatic detection and segmentation of the correct RoI is interesting as a research topic by its own right, it is out of the scope of this specific research. We therefore required experienced radiologists to use a cropping tool developed by TenD AI Medical Technologies

(reported in [76]) to manually crop the RoI from the original US image. The tool enables the radiologist to upload an ultrasound image and manually identify the lesion boundary by placing coordinate points on the border of the lesion. We use the coordinates as reference to find the rectangular bounding box containing the lesion. We then expand the size of the bounding box by 8% to include some extra surrounding tissue areas around the lesion under the advice of domain experts. As for the External_A dataset, a mask with the coordinates on the lesion boundary is already given with each image. We therefore manually place the bounding box using the mask as a reference, and also attempt to include about 8% extra surrounding areas. Figure 4.1 shows four cropped RoI example images, two from the Modelling dataset (one image for a benign lesion and one image for a malignant lesion) and two from the External_A dataset (also one image for a benign lesion and one image for a malignant lesion. The bounding box is shown in red.

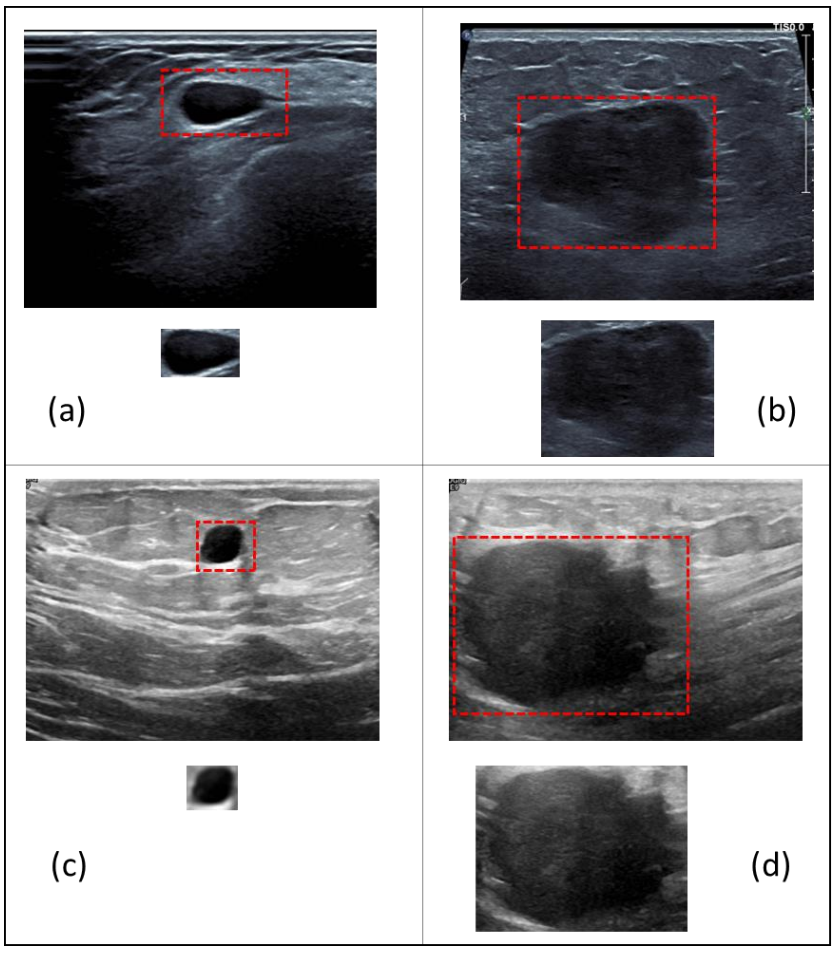


Figure 4-1: Cropped RoI example images (in red boxes): Modelling dataset ((a) Benign and (b) Malignant), External_A ((c) Benign and (d) Malignant)

Image Resizing

Most convolutional neural networks require all input images to have the same size [95]. Deeper CNN architectures usually require a specific input image size to avoid the vanishing gradient problem. Most of the state-of-the-art architectures are using relatively large image sizes such as 227×227×3 or 224×224×3 [37]. On the other hand, a large image size increases the training time and requires a large amount of memory to host model weights.

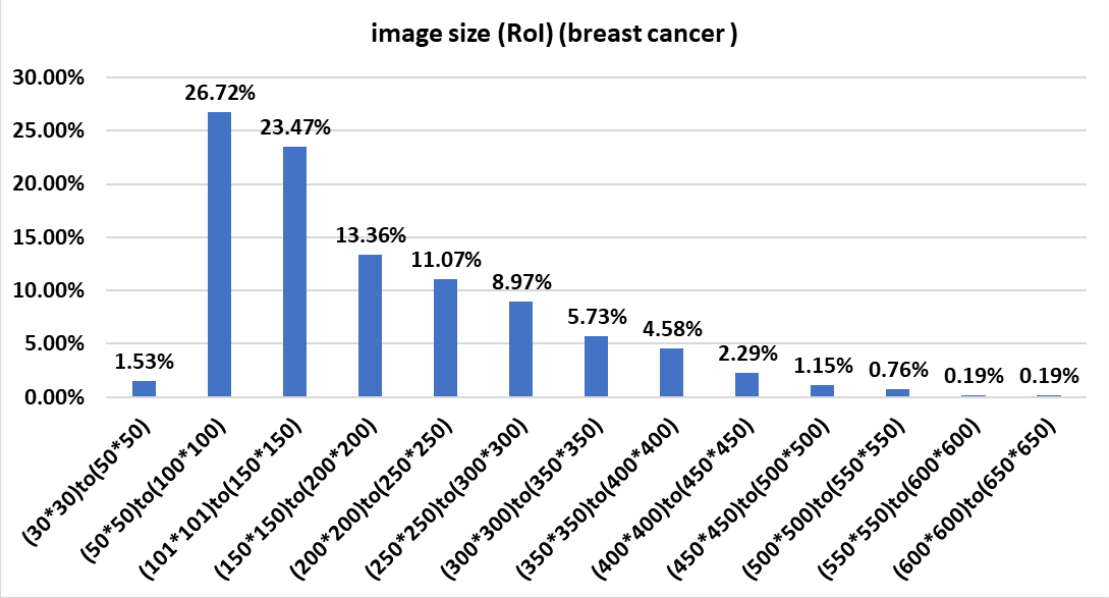


Figure 4-2: RoI Image sizes across Modelling Breast Lesion Datasets

An initial inspection of the different datasets at hand revealed that the RoI images of breast lesions have different sizes. Figure 4.2 shows the distribution of different RoI image sizes across the datasets. It appears that the majority of RoI images have a size between 50×50 and 200×200 pixels. This is only natural because breast lesions at different stages tend to have different sizes. Besides, the US images may be acquired at different scales; some images may contain amplified RoIs. Although padding has been used to deal with input images of variable sizes, in this research, we used the Bicubic method [96] to resize the input image to 100×100 pixels, i.e. the median of the most common RoI image sizes, for ENAS optimization.

Data Augmentation

As described in Chapter 2, training CNN models require a large number of training examples to overcome possible model overfitting. Yet, the Modelling dataset as described in Section 4.1 is still relatively too small (only a few hundred comparing with more than 1 million in the ImageNet for natural images [72]). Therefore, using data augmentation methods to enlarge the training dataset is unavoidable. Since developing new data augmentation methods is not of the

interest of this research, two types of augmentation methods as reported in [76], i.e. geometric methods and singular value decomposition methods, have been directly adopted. The methods are briefly outlined as follows.

A. Geometric Methods

Two geometric transformation methods are used: rotation and mirroring. These methods work by altering the original RoI image to a new position and orientation while preserving the shape of the class representation within the image:

- Rotation: we apply label-preserving image transformation called rotation. Each RoI image was rotated around the centre anticlockwise with respective degrees: 90; 180; and 270.
- Mirroring: We generate a reflected duplication of a RoI image by filliping the image across its vertical axis.

These computationally efficient methods generate four augmented images from each RoI image.

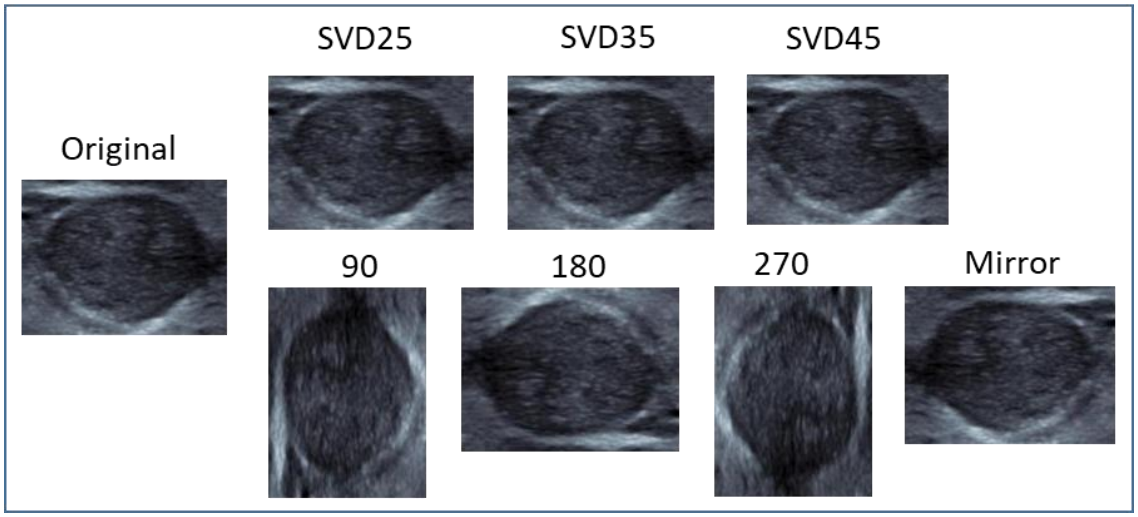


Figure 4-3: An Example RoI Image with its own Augmented Images

B. Singular Value Decomposition (SVD)

We used the Singular Value Decomposition (SVD) based image compression scheme to generate images that look like the original RoI image while preserving the geometry of the image, as proposed in [76]. The method generates three images with 45%, 35% and 25% ratios of the selected top singular values from each RoI image. This method approximately preserves the important information in the original US images, and at the same time reduces the amount of information redundancy. The method results in three augmented images from each RoI image.

Both Geometric methods and the SVD method generate seven augmented RoI images from the original RoI image. Figure 4.3 shows an example RoI image together with the 7 augmented images created from the original image.

4.2. ENAS CNN Architecture for Breast Lesion Classification

We adapt ENAS with micro search space [11] to search for optimal CNN architectures for breast lesion classification from ultrasound images. In particular, we search for the optimal CNN architecture for ultrasound image classification problems. We follow the two stages of automatic CNN design of the ENAS micro approach. More details are described in the next two sections.

4.2.1. ENAS CNN Architecture Search

This stage aims at searching for the optimal normal and reduction cells based on the validation accuracy. Therefore, from the available Modelling dataset (Section 4.1), a balance set of images was selected which consist of 262 images per class and named as BModelling dataset, is first split into training and testing sets according to a splitting policy. The training set was further split into super-net training set and validation set according to a separate splitting policy. To reduce the complexity of the search, a set of hyper-parameters are fixed. We inherit the ENAS default settings for these hyper-parameters. The backbone (super-net and child net) search architecture is set to 7 layers with the number of filters starting from 20. The batch size is set to 8, and the number of nodes per cell is set to 5. The dropout rate is set to 10% and the learning rate is initialised at 0.0005.

The number of epochs for training the super-net and the controller is set to 150 epochs. At each epoch, the controller generates a list of 10 candidate cells. and these architectures are evaluated using the validation set. Based on the validation accuracy, the optimal cells (normal and reduction) are selected. Then the final architecture is designed by stacking 17 cells in three Normal-Reduction cell blocks. Figure 4.4 shows the final architecture of 17 layers and the searched normal and reduction cells.

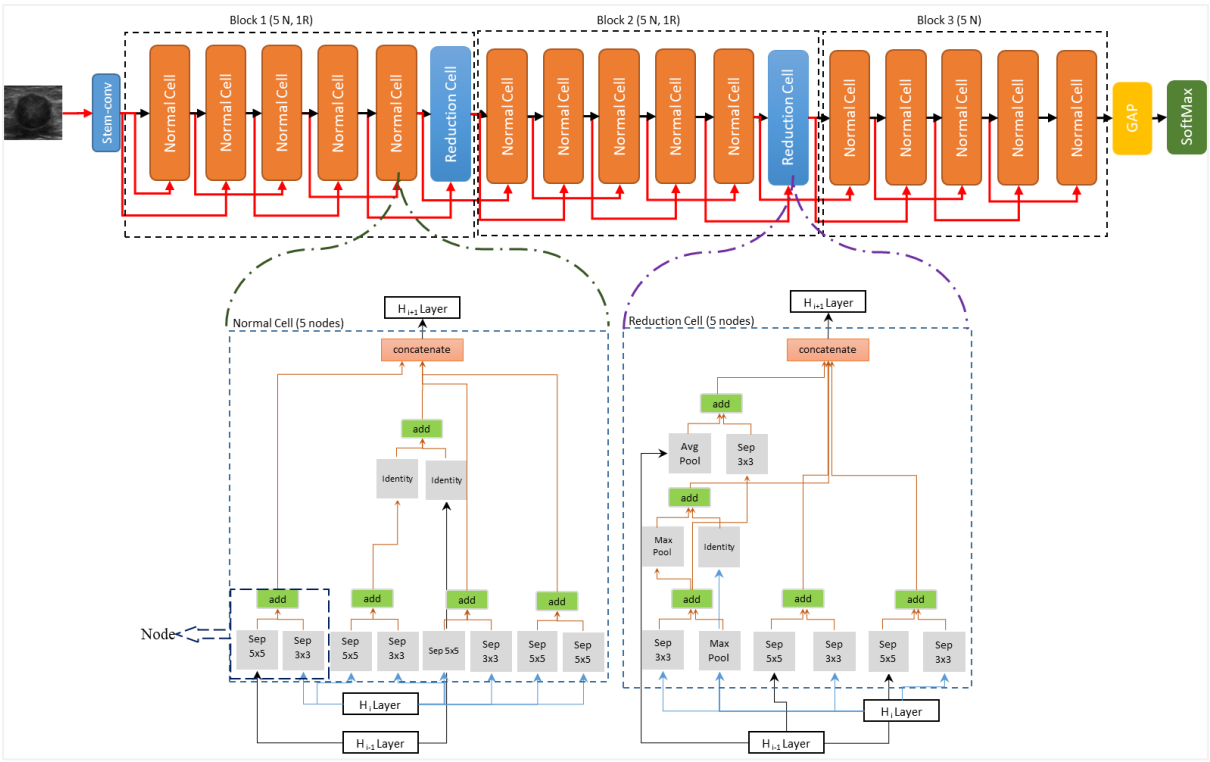


Figure 4-4:ENAS17 That designed by Normal cell and Redaction cell generated by ENAS for breast cancer classification.

4.2.2. ENAS CNN Modelling

The modelling stage is to use the BModelling dataset to train, from scratch, models based on the optimal CNN architecture obtained from the search stage. As described in Section 4.1.2, input image size is set to 100×100 pixels. In this modelling stage, the CNN models are trained by 100 epochs with a dropout rate of 20%, and the number of filters starts with 36, also the same setting of default ENAS method used for modelling stage. All the other hyper-parameters are set the same as the searching stage in Section 4.2.1.

4.2.3. Performance of ENAS CNN Models for Breast Lesion Classification

Sections 4.2.1 and 4.2.2 presented the adaption of ENAS approach to search for the optimal architecture and model. In this section, we will evaluate ENAS CNN models in an experiment conducted using a subset of the Modelling (internal) dataset (see Section 4.2.1). In this experiment, for searching, we split the BModelling dataset into 20% for testing and 80% for training/validation. The remaining 80% were further split into 90% for training super-net and 10% for validation, all through stratified random sampling. For modelling, we followed a stratified 5-fold cross-validation process (i.e. equal number of classes in each fold). At each iteration of the process, the data augmentation methods described in Section 4.1.3 were applied

to images in the training folds to expand the training set, but the images in the testing folds are not augmented.

We are intrigued with a question: *Will 150 epochs be appropriate for searching for optimal cells?*. The original article of ENAS [11] did not answer this question. We therefore applied a specific strategy: we selected the optimal cells from the first 50 epochs, the optimal cells from the second 50 epochs, and the optimal cells obtained from the last 50 epochs. Using the optimal cells from the three groups yielded three architectures up which we trained three sets of CNN models through exactly the same cross-validation process. Figure 4.5 shows the average true positive, true negative and overall accuracy of the CNN models.

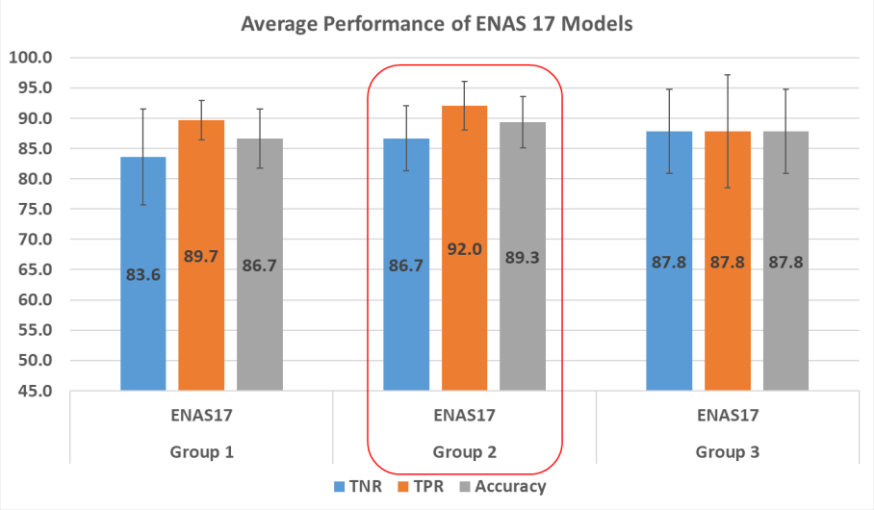


Figure 4-5:Performance of Optimal CNN architecture Generated by ENAS.

Figure 4.5 shows that ENAS CNN models achieved good levels of classification accuracy (between 86.7% to 89.3%) in general. The results also show that the optimal cells searched within the first 50 epochs are still relatively immature or sub-optimal. The optimal cells obtained from the final 50 epochs have the advantage of balanced performance on both the true positive rate and the true negative rate. In addition, the overall accuracy is still more than 1% higher than the cells obtained from the first 50 epochs. Surprisingly, the optimal cells obtained with the second 50 epochs (between 51 and 100) are the best, achieving an overall accuracy close to 90%, and has a 92% true positive rate and still a good true negative rate. This result seems to indicate that early stopping of the search may be advantageous. Another interesting point to note is that with balanced class dataset, the true positive rates are consistently higher than (at least equal to) the true negative rates. This observation may have an implication to the work presented later in the chapter about how to overcome the model overfitting issue.

4.2.4. Comparison between ENAS CNN Models and Other Existing CNN Models

It is interesting and useful to know how ENAS CNN models compare with models built on purposely designed CNN architectures for breast lesion classification and with models built on existing ImageNet-based CNN architectures with transfer learning. We, therefore, selected three manually designed CNN architectures for breast lesion classification from ultrasound: CNN3 [80], CNN4 [78], and Fus2Net [79] for comparison. Each one was trained and tested from scratch on the same dataset (BModelling). All the hyper-parameters were set as described in each original paper. Table 4.1 presents the comparison results of NEAS17 and the existing CNN architectures manually designed for breast cancer classification.

Table 4-1: Comparison ENAS17 and Existing CNN architectures Manually Designed for Breast cancer classification. trained on (BModelling) Balance Dataset

Models		Test sets	TNR %	TPR %	Acc %	No. Parameters
CNN for Breast Cancer US	CNN3	Internal	80.5	75.6	78.1	619,202
		External_A	53.6	97.7	75.7	
		External_B	27.5	98.3	62.9	
	CNN4	Internal	64.8	78.3	71.6	628,418
		External_A	36.2	92.6	64.4	
		External_B	56.9	88.6	72.8	
	Fus2Net	Internal	80.9	57.7	69.3	889,714
		External_A	61.2	60.4	60.8	
		External_B	67.7	57.5	62.6	
Our	ENAS17	Internal	**86.7**	**92.0**	**89.3**	**4,251,780**
		External_A	**65.1**	**93.3**	**79.2**	
		External_B	**60.7**	**97.5**	**79.1**	

At the same time, we include some well-established architectures originally designed for ImageNet. They include VGG16 [39], ResNet50 [42], InceptionV3 [44], MobileNet V2 [47], DenseNet [43], EfficientNetB0 [48], NasNet Mobile [58] and XceptionNet [46]. Only the last fully-connected layer has been changed from 1000 nodes to two nodes to reflect the class nature of the breast lesion classification. For training architectures, the number of epochs was set to 50 and the batch size was set as 16 for all the models, which is useful in term of converge and computation power. In this specific comparison, to maintain fairness of the comparison, all the network models were trained from scratch directly using the BModelling dataset under the exact same setting as the ENAS CNN models. In other words the same split of data used with the same data augmentation methods for expanding the training set. Table 4.2 shows the comparison result of ENAS7 and state-of-the-art architecture.

Table 4-2 Comparison of Various CNN Models trained on (BModelling) Balance Dataset without Transfer Learning

Models		Test sets	TNR %	TPR %	Acc %	No. Parameters
State-of-the-art Models	Vgg16	Internal	54.7	58.1	56.4	138,357,544
		External_A	47.8	59.6	53.7	
		External_B	50.6	59.2	54.9	
	Resnet50	Internal	54.1	95.8	74.9	25,636,712
		External_A	36.1	98.1	67.1	
		External_B	31.3	99.1	65.2	
	Inception_V3	Internal	85.5	82.6	84.0	23,851,784
		External_A	55.2	92.5	73.9	
		External_B	58.9	90.7	74.8	
	Mobile V2	Internal	66.8	92.4	79.6	3,538,984
		External_A	37.9	97.3	67.6	
		External_B	41.3	98.6	70.0	
	DensNet	Internal	79.7	82.2	81.0	8,062,504
		External_A	51.7	97.0	74.4	
		External_B	51.8	93.3	72.6	
	EfficientNetB0	Internal	70.5	89.7	80.1	5,330,571
		External_A	48.0	96.7	72.3	
		External_B	53.3	94.5	73.9	
	NASNet Mobile	Internal	20.4	78.5	49.4	5,326,716
		External_A	20.8	79.1	50.0	
		External_B	20.1	78.6	49.4	
	XceptionNet	Internal	84.7	89.0	86.9	22,910,480
		External_A	55.5	96.3	75.9	
		External_B	54.7	96.1	69.9	
Our	ENAS17	Internal	**86.7**	**92.0**	**89.3**	**4,251,780**
		External_A	**65.1**	**93.3**	**79.2**	
		External_B	**60.7**	**97.5**	**79.1**	

Tables 4.1 and 4.2 shows the results of the comparison. Among all methods, ENAS17 achieved the highest overall accuracy on the internal testing dataset, nearly 40% higher than the lowest-performance CNN model, and still more than 2% higher than its nearest challenger (XceptionNet). In particular, the difference between TPR and TNR remains small. Interestingly, the nearest challenger is also XceptionNet on this aspect. Surprisingly, all purposely designed CNN models do not perform better than others. External test results on all CNN models are generally lower than the internal test results with VGGNet, NasNet Mobile and specifically designed CNN for breast lesion ultrasound images as exceptions, although their overall accuracies are generally lower than the others. ENAS17 also suffers this type of generalization errors, but the external test accuracies remain the highest for both External_A and External_B datasets.

Another interesting finding is that ENAS17 models have a low complexity in terms of the number of weights in the network. Although ENAS17 models are significantly more complex than purposely designed handcrafted CNN models, they have the second lowest number of weights (less than 1 million weights than Mobile V2). ENAS17 achieves higher level of accuracy with a less complex model. ENAS17 remains the best in terms of accuracy achieved among the CNN models with the same degree of magnitude in the number of weights.

A similar trend that we noticed for ENAS17 earlier also exists among other types of CNN models. That is the TPR is generally higher than the TNR (Fus2Net is an exception). Some CNN models have gone to the extreme. For instance, Resnet50 achieved 99% for TPR at the expense of only achieving 31% for TNR for the External_B dataset. This finding can later become useful for dealing with ENAS generalization error issues. One caution that must be noted is that we did not fine-tune parameters for each CNN.

4.2.5. Comparison between ENAS Models and CNN Models with Transfer Learning

Tables 4.1 and 4.2 shows the comparison between ENAS models with other CNN models trained from scratch. Although it was a fair comparison, as mentioned in Chapter 3, most research adapting the existing CNN architectures for natural images with transfer learning. It would be interesting to know how ENAS models compare with CNN models with transfer learning (the current common practice as described in Chapter 3). There are two reported approaches of transfer learning: (a) using the trained model as feature extraction followed by only further training the final fully-connected layers [97] and (b) using pre-trained model weights to initialize the current CNN model weights followed by further training the whole CNN models with the images at hand [98]. For this comparison, we used method (b) to further fine-tune all the saved weights in the CNN model using ultrasound images of breast lesion. For training each CNN architecture, 20 epochs and batch size 16 have been used. The experimental set-up is the same as the previous comparison in Section 4.2.4; we first trained all selected CNN models on the BModelling dataset with training sets enlarged by using all data augmentation methods presented in Section 4.1 in a 5-fold cross validation process, and then evaluated the models on the two external datasets. Table 4.3 presents the performances of CNN models with transfer learning and the performances of ENAS17 models.

Although the transfer learning approach does achieve a better accuracy as reported in the literature [69] and [99], the results still show that the ENAS models outperform the state-of-the-art CNN models with transfer learning. Similar to the results in Tables 4.1and 4.2, ENAS17

models have the highest average overall accuracy on the internal test images, although the nearest contester Inception_V3, is only marginally behind. The external test results also show that ENAS17 models outperform the others on both external test sets. This finding is promising; it shows that ENAS models can outperform most, if not all, existing CNN models with or without transfer learning.

Table 4-3 Comparison of Result of Transfer Learning with ENAS-based Models on Balance Dataset.

Models		Test Data Sets	TNR %	TPR %	Accuracy %
SOTA That designed for natural images	Vgg16	Internal	45.9	92.8	69.3
		External_A	25.5	97.9	61.7
		External_B	27.5	95.5	61.5
	Resnet50	Internal	66.7	89.7	78.2
		External_A	44.7	93.2	69.0
		External_B	40.1	95.6	67.8
	Inception_V3	Internal	89.0	88.2	88.6
		External_A	58.5	95.8	77.1
		External_B	61.6	94.5	78.1
	Mobile V2	Internal	98.5	45.1	71.8
		External_A	85.9	55.3	70.6
		External_B	87.4	59.2	73.3
	DenseNet	Internal	75.1	81.4	78.2
		External_A	57.1	92.7	74.9
		External_B	58.4	96.1	77.3
	EfficientNetB0	Internal	89.3	80.1	84.7
		External_A	66.8	89.0	77.9
		External_B	60.9	95.1	78.0
	NasNet Mobile	Internal	72.3	46.8	59.6
		External_A	69.0	53.6	61.3
		External_B	73.6	51.7	62.7
	XceptionNet	Internal	87.4	76.7	82.1
		External_A	63.6	88.1	75.9
		External_B	65.4	93.0	79.2
Our	ENAS17	Internal	**86.7**	**92.0**	**89.3**
		External_A	**65.1**	**93.3**	**79.2**
		External_B	**60.7**	**97.5**	**79.1**

Similar findings from Tables 4.1 and 4.2 also appear in Table 4.3. For the majority of CNN models, there exists a degree of generalization error. While Mobile V2 and NasNet Mobile maintain the minimal generalization error, the rest shows a reduction of accuracy of around 10% between the internal test results and the external test results. In fact, ENAS17 models show serious generalization error issue than the others, an issue to address for the next section.

At the same time, a similar trend that TPR is higher than TNR still exist for several CNN models although transfer learning seems either has reduced the difference or completely reversed the situation, e.g. MobileNet V2, indicating some degrees of instability of the models based on those CNN architectures.

4.3. Reducing Generalisation Error for ENAS Models

Generalisation error is one of the common issues in machine learning, especially deep learning. The main goal of model development is to design models that generalize well on an external dataset. The performance of models on the internal test is also useful for fixing the issue of model overfitting, a reference primarily for a performance gap between the training accuracy and test accuracy. Reducing generalization is an broader issue, at the very least, the model that performs well for the internal testing also performs well on the external dataset collected from different data sources with the same data preparation protocol [100].

In this research, we used two different external datasets assembled from different sources as described in Section 4.1, and the difference between the performance of models on an internal and external test set was used as generalisation rate measurement. As shown in Table 4.1 (also in Tables 4.2 and 4.3), ENAS17 models have an average overall accuracy of 89.3% for the internal testing, but the overall accuracies drop by around 10% to 79.2% and 79.1%, respectively for the two external testing datasets. Figure 4.6 shows the problem more clearly. The main problem happens with TNRs; the TNR is reduced by 21% on the External_A dataset and by 26% on the External_B dataset. This problem might be caused by the use of the same data for searching and for modelling in the ENAS two stage approach. Existing work on this issue for ENAS is limited (on existing work [86] was identified in the literature). The authors suggested a method to reduce ENAS model overfitting and improving generalization rate by fixing skip connections and searching only for operation. The method in [86] was evaluated in ENAS Macro search space with a Cifar-10 dataset. Although we agree with the authors that modifying the network architecture can have an impact on model overfitting, we at this stage of investigation, are more interested in applying mature and established methods for solving the issue at the ENAS modelling stage rather than the search stage.

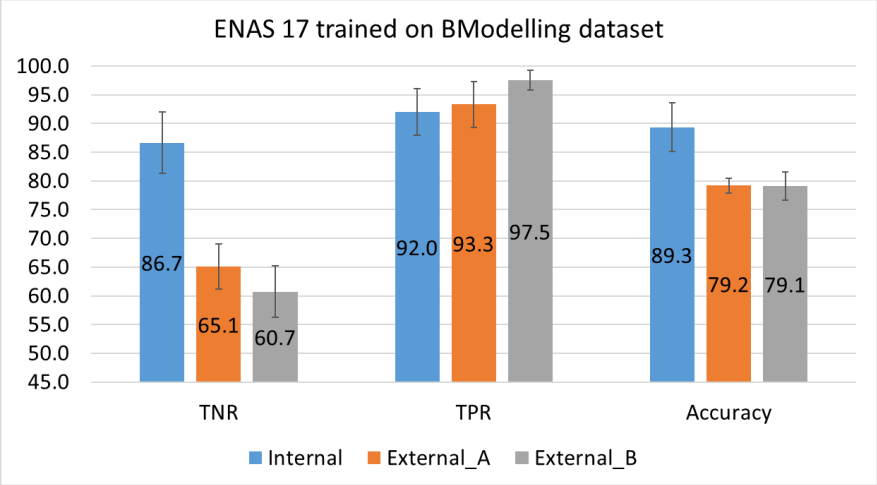


Figure 4-6: Performance of ENAS 17 Trained on BModelling Dataset on External Datasets.

4.3.1. Methods for Reducing Generalization Errors in ENAS Models

There are many methods for reducing model overfitting and improving generalization in deep learning such as data augmentation, regularization, dropout, reducing model complexity and early stopping. Deep learning requires a large amount of data for training, and overfitting can be caused by insufficient training images. Therefore, data augmentation can be an effective method to enlarge the training set and make the models less dependent on a few specific training examples. Regularization is a technique that adds a penalty term for the loss function, and L_1 and L_2 are the two most commonly used regularisation techniques. Dropout can be seen as another regularisation technique which drops neurons randomly at a fixed rate in training, which is mostly applied to the fully-connected layers. Simplifying model is also a technique used, which can be done by reducing the complexity of the model (e.g. removing layers, decreasing the number of filters or using small filter sizes). In this section, we investigate three techniques to improve the generalization of the ENAS model as detailed below.

Reducing CNN Architecture Complexity

Rather than simplifying the basic operations considered (due to the reasons given in the next Chapter), we purposefully maintain the optimal cells and reduce the number of layers of the searched CNN architecture from 17 to 7 by removing 10 normal cells. In other words, we designed an ENAS7 architecture that consists of 7 layers (cells). The ENAS method [11] identified optimal architecture depth as 17 layers for natural image classification task. This depth was identified experimentally. However, for breast lesion classification, the number of layers was reduced to 7 to reduce the architecture complexity and to maintain a good level of architectural depth for learning simple as well as complex features from the input image. The

architecture has a reduced number of weight parameters about 50% of the number for ENAS17. Figure 4.7 shows the ENAS7 architecture. It is interesting to observe whether this simple method of model complexity reduction can improve model generalization.

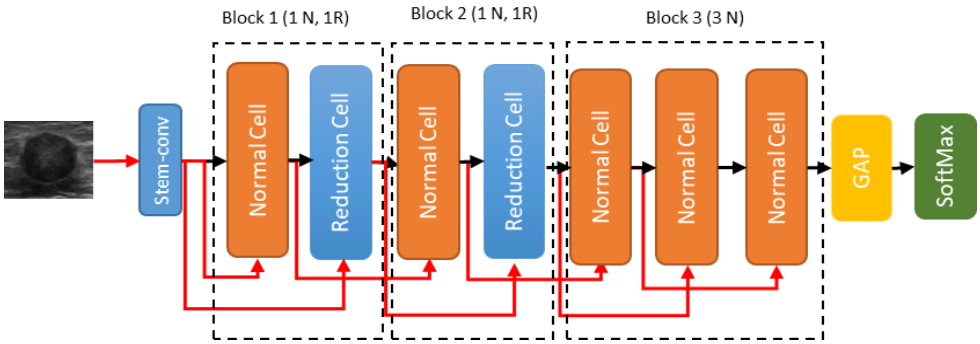


Figure 4-7: ENAS 7 Architecture

Impact of Data Augmentation

Although the principle of data augmentation is sound, since augmentation is based on the existing examples, the effect of data augmentation on generalization still needs to be tested. Therefore, we used the data augmentation methods presented in Section 4.1.3 to enlarge the training dataset. To investigate the effect of the methods on ENAS models in more detail, we will test ENAS17 (the original optimal architecture) and ENAS7 (the reduced complexity architecture).

Exploiting Unbalanced Datasets

Class imbalance problem may lead to models that can be more accurate on the majority class. A commonly adopted approach in classification in general is to use down-sampling or up-sampling techniques to bring the cases of different classes to a balanced level [101]. However, we have observed from Table 4.4 and 4.5 that using datasets of balanced benign and malignant classes often leads to models with higher sensitivity and lower specificity. This observation is backed up by published works in the literature (e.g. [76]), and our own experiences of developing DCNN models from ultrasound images [101]. In fact, class imbalance is more realistic in the intended area of application; the vast majority of lesions are benign and only a small minority of cases are malignant. However, using a dataset directly reflecting the reality is not desirable either; we may run the risk of classifying most testing images as benign. It is, therefore interesting to investigate upon which class the ENAS models tend to overfit more in both balanced and unbalanced class situations, and whether maintaining a reasonable unbalance of benign and malignant cases will bring about positive effect in improving model generalization.

4.3.2. Evaluating the Generalization Errors Methods for ENAS Models

This section describes the experiment set-up for evaluating the effects of the various methods, the experimental results, and result analysis.

Effects of Reducing Architectural Complexity on ENAS Models

As descried in Section 4.3.1, we simplified the original ENAS17 architecture into an ENAS architecture as shown in Figure 4.7. We then trained the ENAS7 architecture using the same BModeling dataset with the same hyperparameter settings as for ENAS 17 models so that we can focus on the effect of the model complexity change. The test results are presented in Table 4.3. The results are not as expected. In general, the difference in overall accuracy between the internal test results and the external test results remains within the range from 5% to nearly 10%. In particular, the simpler ENAS7 models have better performances on TNRs but at the expense of reduced TPRs across different sets. In other words, the simplification has not significantly reduced generalisation errors despite some marginal improvements in TNR on internal and external test sets. In certain circumstances of data augmentation such as no data augmentation was applied, the overall accuracy of ENAS17 models dropped 10.4% and 5.8% on External_A and External_B respectively, while the generalisation errors of ENAS7 models are 6.2% on External_A and 7.5% on External_B.

Effects of Data Augmentation on ENAS Models

To investigate the effect of the different data augmentation methods, we investigated different training set settings prepared for the different scenarios. Table 4.4 presents the performance metrics in different scenarios of augmenting training examples for both ENAS17 and ENAS7 models. There are again no clear-cut results. The table shows that the ENAS7 models without data augmentation are performing better on External_A in terms of TNR with a lesser performance drop of 6.8% for ENAS17 models and 10% lesser drop for ENAS7 models. In the second scenario where Rotation and Mirror data augmentation have been used, the average accuracies of the models from both ENAS17 and ENAS7 architectures are reduced to around 8% on both external test sets. ENAS17 and ENAS7 models trained on the enlarged training set using SVD and Mirror augmentation methods, the test accuracies of the ENAS7 models are reduced on both External_A (9.7%) and External_B (8.3%), while the accuracy reductions of ENAS17 models are around 5% on both external test sets. Although the generalisation error of ENAS 17 with SDV and Mirror reduced to 5.8% on External_A and 5.6% on External_B, still there is massive reduction in TNR. The results showed that ENAS17 and ENAS7 models

trained on balanced US images do not generalize well even with a different type of data augmentations.

Table 4-4 Effect of Data Augmentation on Generalization of ENAS 17 and ENAS 7

Models	Training scenario	Test sets	TNR	TPR	Acc.
ENAS17	Without any use of augmentation techniques	Internal	87.8	84.0	85.9
		External_A	81.0	70.1	75.6
		External_B	64.8	95.4	80.1
	With use of all the augmentation techniques	Internal	86.7	92.0	89.3
		External_A	65.1	93.3	79.2
		External_B	60.7	97.5	79.1
	With use of only SVD and mirroring techniques	Internal	83.9	87.1	85.5
		External_A	68.6	90.9	79.7
		External_B	63.3	96.5	79.9
	With use of rotation techniques	Internal	85.9	89.7	87.8
		External_A	64.7	93.7	79.2
		External_B	61.7	97.9	79.8
ENAS7	Without any use of augmentation techniques	Internal	86.2	86.3	86.3
		External_A	76.3	83.9	80.1
		External_B	63.9	93.6	78.8
	With use of all the augmentation techniques	Internal	90.9	86.7	88.8
		External_A	69.7	88.0	78.9
		External_B	65.0	96.7	80.9
	With use of only SVD and mirroring techniques	Internal	86.7	85.1	85.9
		External_A	63.3	89.0	76.2
		External_B	58.1	97.1	77.6
	With use of rotation techniques	Internal	87.4	85.5	86.5
		External_A	63.4	93.0	78.2
		External_B	63.0	97.4	80.2

Effect of Using Unbalanced Training Set for Generalisation

We intend to investigate the effect of an unbalanced training set on model generalisation. For this experiment, all modelling dataset (unbalanced) described in Section 4.1, i.e. 726 images of benign and 376 images of malignant lesions, were used as the training set for the ENAS17 and ENAS7 models, and ENAS17 and ENAS7 models were trained and tested through a 5-fold cross-validation. We want to examine the combined effects of an unbalanced dataset when different augmentation methods are used. The results are shown in Table 4.5.

Table 4-5 Reducing Generalisation error of ENAS 17 and ENAS 7 with Unbalanced dataset

Models	Training set scenario	Test sets	TNR	TPR	Acc.
ENAS 17	Without any use of augmentation techniques	Internal	90.3	65.3	77.8
		External_A	89.2	66.0	77.6
		External_B	79.8	89.7	84.8
	With use of all the augmentation techniques	Internal	89.5	72.6	**81.1**
		External_A	80.5	84.1	**82.3**
		External_B	79.8	94.8	**87.3**
	With use of only SVD and mirroring techniques	Internal	89.3	75.6	**82.5**
		External_A	82.4	84.8	**83.6**
		External_B	77.5	95.3	**86.4**
	With use of rotation techniques	Internal	89.7	65.0	77.3
		External_A	77.7	79.1	78.4
		External_B	81.8	92.3	87.1
ENAS 7	Without any use of augmentation	Internal	84.1	81.2	82.6
		External_A	84.6	68.0	76.3
		External_B	68.5	95.1	81.8
	With use of all the augmentation techniques	Internal	92.0	69.2	80.6
		External_A	83.8	75.4	79.6
		External_B	80.8	94.6	87.7
	With use of only SVD and mirroring techniques	Internal	92.3	64.6	78.5
		External_A	87.3	66.7	77.0
		External_B	81.2	86.3	83.8
	With use of rotation techniques	Internal	91.0	69.8	**80.4**
		External_A	81.0	80.8	**80.9**
		External_B	79.1	95.2	**87.2**

The results showed that using an unbalanced dataset as training set for ENAS architectures improves the performance of ENAS bases models on external data sets by significantly reducing the generalisation error rates. In particular, the generalization error rates of ENAS models in overall accuracy are reduced by 100% and TNR by around 50%, compared to the ENAS models that were trained on the balanced dataset. For the ENAS17 models, using SVD with mirroring and all data augmentation methods balanced the performance the best across the internal and external test sets. While, for the ENAS7 models, the scenario of using the rotation augmentation methods delivers more stable performances in both the internal and external test sets. As presented in Figure (4.6), our ENAS models do not generalize well for TNR. To reduce the generalisation error rate, we start using an unbalanced dataset to train ENAS models. Systematically we increased Benign cases to our US breast dataset to suit unbalanced dataset for training our ENAS models. First, we increased the number of Benign cases by 5% and 10% for our US dataset, then splitting the unbalanced US dataset for the test set and train. This Unbalanced dataset was used to train ENAS17 and ENAS7. We noted a slight reduction of overfitting, especially in TNR.

Table 4-6 Comparison CNN Models with ENAS models on UnBalance Dataset

Models	Test Data Sets	CNN Models from Scratch			CNN Models with TL		
		TNR	TPR	Accuracy	TNR	TPR	Accuracy
Vgg16	Internal	100.0	0.0	50.0	100.0	0.0	50.0
	External_A	100.0	0.0	50.0	100.0	0.0	50.0
	External_B	100.0	0.0	50.0	100.0	0.0	50.0
Resnet50	Internal	77.8	63.7	70.7	83.4	58.3	70.9
	External_A	80.5	59.9	70.2	74.4	65.2	69.8
	External_B	63.1	81.1	72.1	72.9	70.1	71.5
Inception_V3	Internal	92.9	36.8	64.9	97.1	32.7	64.9
	External_A	91.4	34.5	62.9	91.8	42.2	67.0
	External_B	87.7	52.5	70.1	91.1	52.3	71.7
Mobile V2	Internal	86.1	65.0	75.5	93.4	70.6	82.0
	External_A	79.0	70.2	74.6	68.6	75.5	72.1
	External_B	74.6	87.0	80.8	56.9	91.5	74.2
DenseNet	Internal	80.4	64.9	72.7	82.7	76.9	79.8
	External_A	74.5	73.6	74.1	74.5	79.3	76.9
	External_B	66.6	83.0	74.8	70.8	95.5	83.2
EfficientNetB0	Internal	84.8	71.3	78.1	89.8	75.0	82.4
	External_A	84.1	63.0	73.5	79.6	81.2	80.4
	External_B	72.7	85.7	79.2	77.8	96.5	87.2
NasNet Mobile	Internal	59.5	78.3	68.9	78.2	29.1	53.6
	External_A	60.2	82.5	71.4	77.8	30.4	54.1
	External_B	47.3	85.7	66.5	72.7	30.2	44.2
XceptionNet	Internal	88.0	64.7	76.4	92.0	72.9	82.5
	External_A	83.8	69.9	76.9	80.3	76.5	78.4
	External_B	77.5	86.7	82.1	79.7	93.8	86.8
CNN3	Internal	86.8	71.8	79.3	-	-	-
	External_A	71.7	80.8	76.2	-	-	-
	External_B	66.4	91.7	79.1	-	-	-
CNN4	Internal	74.2	66.7	70.4	-	-	-
	External_A	71.4	70.3	70.8	-	-	-
	External_B	63.1	79.4	71.2	-	-	-
Fus2Net	Internal	93.8	37.0	65.4	-	-	-
	External_A	86.4	44.8	65.6	-	-	-
	External_B	91.2	61.0	76.1	-	-	-
ENAS17	Internal	89.5	72.6	81.1	89.5	72.6	81.1
	External_A	80.5	84.1	82.3	80.5	84.1	82.3
	External_B	79.8	94.8	87.3	79.8	94.8	87.3
ENAS7	Internal	92.0	69.2	80.6	92.0	69.2	80.6
	External_A	83.8	75.4	79.6	83.8	75.4	79.6
	External_B	80.8	94.6	87.7	80.8	94.6	87.7

In summary, while data augmentation and simplifying the ENAS architectures may not improve the generalization, using unbalanced dataset does have a positive effect. Using unbalanced dataset for training the ENAS models achieves more stable and robust accuracies across internal and external datasets. As a final task of investigating the effects of using unbalanced data on generalization, we conduct another round of comparison between ENAS models and the CNN models based on other existing CNN architectures. All the settings are the same as the previous comparison (Sections 4.2.4 and 4.2.5), and only the training dataset is now replaced by the unbalanced dataset. Table 4.6 summarizes the performances of the models with and without transfer learning. The results showed that the use of unbalanced dataset with all data augmentation methods reduced the generalization errors across most CNN models comparing to the results reported in Tables 4.1 and 4.2, but ENAS models benefit from this use the most, creating stable and robust models although the overall accuracies on the internal testing are generally reduced. Appendix A includes detailed results of ENAS17 on unbalanced dataset.

4.4. AlexNet-based CNN Architecture Design for Breast Lesion Classification

The first two main parts of this chapter (Sections 4.2 and 4.3) focused on using the automatic neural architecture search and reducing the generalisation error for breast lesion classification task. It also included a comprehensive comparison against state-of-the-art networks. This section presents our own attempts to manually modify an existing CNN architecture and then train models on the modified architecture for breast lesion classification from ultrasound images. Our rational of this investigation are of two folds. First, we want to gain a first-hand experience in manually constructing a CNN architecture for breast lesion classification. Second, the experience gained from this exercise may provide insight into the next stage of research towards optimization of CNN architectures.

Therefore, we start by adopting the AlexNet [29] as the backbone architecture due to its maturity and popularity in the deep learning neural network research community. It has shown an outstanding performance in object recognition from natural images (e.g. animals, flowers and vehicles) [102]. Therefore, the effectiveness of this fundamental CNN architecture on ultrasound images of breast lesions would be of our interest. The first point of interest would be how effective when the basic AlexNet architecture is used to build a classification model. We purposely modify certain types of CNN hyper-parameters such as convolution layers, filter sizes, weight initialization schemes, Batch Normalization and Bias setting after an extensive

consultation of the existing literature on CNN based solutions for the intended application purposes. Figure 4.8 shows the workflow of this investigation.

As described in Chapter 2, AlexNet has 5 convolutional layers followed by RelU and 3 fully-connected layers. The hyper-parameters were fixed as the default ones, i.e. batch size = 128, learning rate = 0.0001, optimization using SGD, filter initialization using Gaussian Distribution, and epochs = 20. The number of nodes in the last fully-connected layer was reduced to 2 nodes based on the number of classes (Benign and Malignant).

In the first evaluation experiment, the balanced dataset BModelling was used in a 5-fold stratified cross-validation process (same split as Section 4.2), and TNR, TPR, Accuracy are collected for each fold. For the training folds at each iteration of the cross validation, all the data augmentation methods mentioned in Section 4.1 were applied. The overall average accuracy is 50.3% with TPR = 49.1% and TNR 51.5%. with a slight accuracy variation between folds. These results show that the basic AlexNet architecture designed for natural image classification seems to have no better performance in distinguishing benign or malignant breast lesions from ultrasound images than a random guess. The results also suggest that US image features are quite different from natural images in ImageNet, and the basic AlexNet is ineffective extract useful features from such images. Although training examples are limited, with data augmentation, the number of training images per class is close to that in the ImageNet dataset. So, after this evaluation, the following adaptation are carried out to AlexNet.

4.4.1. Structural Modification to AlexNet

Filter Sizes

The AlexNet architecture contains five convolutional layers with filter sizes 11×11, 5×5, 3×3, 3×3 and 3×3, respectively. Inspired by the VGGNet architecture which uses a fixed filter sizes (3×3) across different convolutional layers, the first modification to the AlexNet architecture is to fix the filter size across all five convolution layers. We then examine the effects of different filter sizes, i.e. 3×3, 5×5, 7×7, 9×9, 11×11, 13×13 and 15×15 systematically. Therefore, started from smallest filter size to possible largest filter size for extracting local and global features form the input images. All other hyper-parameters of AlexNet are set as the default values. Table 4.6 shows the average classification performances across all 5 folds for different filter sizes. For ease of comparison, we include the AlexNet with default filters (without change) in the first row of the Table 4.7.

Table 4-7 Performance of AlexNet with Different filter sizes(US breast)

Filter sizes	TNR	TPR	Accuracy
Defaults	51.5	49.1	50.3
3×3	52.4	50.6	51.5
5×5	51.0	49.8	50.4
7×7	51.9	51.8	51.9
9×9	59.2	60.2	59.7
11×11	69.7	69.9	69.8
13×13	75.0	75.3	75.1
15×15	76.3	79.3	77.8

The results show that the performances of larger filters are better than the default ones in the basic AlexNet, and as the filter size increases, there are improvements on all performance metrics. The results on the main metrics are fairly comparable with TPR marginally higher than the other metrics for filters larger than 7×7. Although the performances are not extremely high (maximum TPR 79.3%), they are significantly higher than the random guess as achieved by the default AlexNet filters. More details can be found in Appendix A.

One possible explanation for these results is that as the filter size gets bigger, provided the stride remains the same, larger overlapping regions of the image will be examined and local features are extracted at each convolutional layer and represented across the convolutional layers. Therefore, specific features of the US image are repeatedly extracted and summarised. Such repeated extraction may have assisted the discovery of subtle texture patterns that uniquely exist in US images of breast lesions.

Convolution Layers

Several architectures with different numbers of convolution layers have been developed for different types of datasets and different recognition tasks in the literature [29], [39] and [40]. To identify the optimal number of convolution layers required to build a suitable CNN architecture for breast lesion classification from ultrasound images, we tested several variants of the AlexNet by removing one or more layers with and without changing the filter sizes, which can be easily done due to the relatively simple structure of AlexNet. The approach of adding layers to AlexNet was excluded to avoid increasing the complexity and underfitting issue, because the results of original AlexNet showed that the model is underfit our breast dataset. Our strategies of removing convolutional layers can be described as follows:

- Removal of one convolutional layer:

- Remove the second convolutional layer with it is max-pooling layer.
- Removing the second convolutional layer without removing the max-pooling layer.
- Removing the third convolutional layer.
- Removing the fourth convolutional layer.
- Removing the fifth convolutional layer.

- Removal of two convolutional layers:
  - Removing the second convolutional layer with max-pooling and removing the fourth convolutional layer.
  - Removing both the third and the fifth convolutional layers.

This task is motivated by the research question of how many different levels of shape features are sufficient to distinguish one type of lesion from another. We hypothesise that the US image of the lesion may contain small differential details rather than large complex shapes as natural images do. It is, therefore, possible that fewer convolutional layers may make the classification results more accurate, and the models more robust.

Figure 4.9 shows the performance of 8 models including the original AlexNet. We first remove one layer at a time from the AlexNet architecture of five layers, i.e. Conv1, Conv2, Conv3, Conv4 and Conv5. At the same time, we need to examine the combined effects of removing a layer when filter sizes also increase. Detailed results of the experiment are presented in Appendix. The test results show that removing convolutional layers does have a significant impact on classification accuracy. The modified AlexNet architecture with its default filter sizes after removing the Cov2 layer with its max-pooling has resulted in a small improvement of average accuracy to 52.4%. By changing the filter size of the remaining layers to 11×11, we achieved an average accuracy of 81.2%, higher than the best result presented in Table 4.7. The results at least partially support our hypothesis that fewer layers (with increased filter size) will lead to better performance.

Continually, we remove pairs of convolutional layers from AlexNet. Firstly, we remove the Conv2 and Conv4 layers, and the average TPR, TNR and accuracy are increased to 54.2%, 53.9%, 54.1% receptively. Then, we examined different filter sizes for the remaining layers, and the overall accuracy is improved to 78.9% with 15×15 filter size. Afterwards, we remove another pair of layers (Conv3 and Conv5). The highest accuracy achieved was achieved with a filter size 15×15. The average TPR, TNR and overall accuracy are 76.8%, 76.7%, 76.7% receptively.

Removing one or two convolutional layers from AlexNet inevitably reduces the complexity of the trained model. Our experimental results consistently show that the modified architectures

with layers removed, the accuracy levels still increase from filter sizes 5×5 onwards. Therefore, the filter size has significant effects on our model performance for US image classification. We achieved an average accuracy 82.1% with a modified AlexNet (removing the second convolutional layer with a max-pooling layer with filter size 11×11). To study the effects of the max pooling on the overall model performance, we modified AlexNet by removing convolutional layer 2 without removing the max-pooling layer (i.e. the model consists of 3 max pooling). The average accuracy achieved is 81.2% while the filter size is set to 11×11. There is an optimal combination between the layer removed and the filter size fixed (see Figure 4.9 for details).

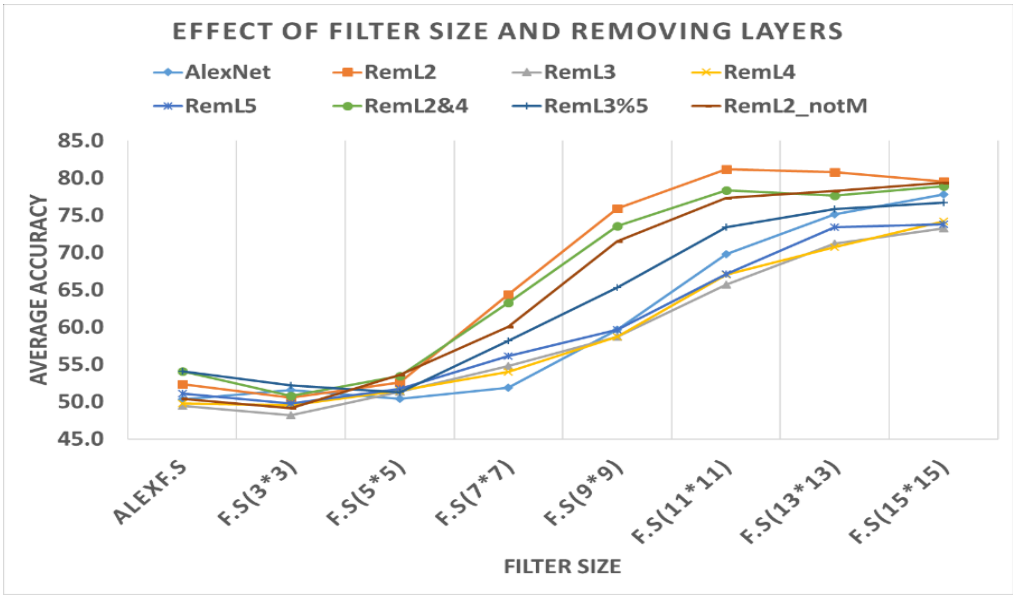


Figure 4-8: Results of Structurally Modified AlexNet

4.4.2. Modification of Training Hyper-parameters for AlexNet

Properly initializing weight is the key to training deep neural networks stably because ill-suited initializations lead to the vanishing or exploding gradient problem during the backpropagation in training. Both batch normalization and weight initialization are trainable hyper-parameters that we select for modifying the original AlexNet. We attempt to use different batch normalization and weight initialization techniques to suit ultrasound images. AlexNet uses Local Response Normalization (LRN) after the first and the second convolutional layers to normalize the pixel values in the feature maps among the local neighborhood. We propose to replace the LRN normalization method for the first and the second convolutional layers with the Batch Normalization (BN) method proposed in [34]. The BN method addresses the Internal Covariate Shift (ICF) problem, helping the CNN network to converge faster as well as reducing model overfitting [22]. For weight initialization, AlexNet uses Random Gaussian Distribution

(RGD) with mean 0 and standard deviation 0.01. The Xavier method is one of the common techniques for initialising CNN weights, and has shown to outperform the RGD initialization [30]. We attempt to use this initialiser instead of RGD.

Table 4.8 shows the results of modified AlexNet with different trainable hyper-parameters. Under the same setting as the initial evaluation of the basic AlexNet, the experimental results show that replacing LRN with BN achieved the average TPR, TNR and overall accuracy of 55%, 57.6% and 56.3% receptively, higher than 48.5%, 51.6% and 50% by the basic AlexNet. Applying Xavier initialiser to replace RGD produces the average TPR, TNR and overall accuracy are increased to 67.8%, 59.5% and 63.7% receptively. Combining BN and Xavier weight initialization further increased the average TPR, TNR and overall accuracy to 74.1%, 71% and 72.5% receptively. The result showed that using Xavier and BN together helps the model to learn well and perform better than the basic AlexNet.

We further examined the effect of Bias. When the bias is set to 1 in all convolution layers, the overall accuracy is increased to 72.9%. When the bias is set to 0 (i.e. no Bias) in all layers, the overall accuracy is further increased to 80.4% (TPR 82.5% and TNR 78.3%). Moreover, to further improve the performance of the trained model, we changed the order of BN in the first and second convolution layers. In the original AlexNet architecture, normalization is placed between the convolutional layer and max-pooling layer. Then we changed the order of BN from before the max-pooling to after the max-pooling in both the first and second convolutional layers. This simple modification further increased model overall accuracy from 80.4% to 84% as well as TPR and TNR are respectively increased to 85.5% and 82.5%. Table 4.7 shows the progressive improvements with more modifications.

Table 4-8 Results of Modified AlexNet with Different Trainable Hyper-parameters

Modified AlexNet Architectures	TNR	TPR	Accuracy
AlexNet	51.6	48.5	50.0
AlexNet +BN	57.6	55.0	56.3
AlexNet +Xavier	59.5	67.8	63.7
Alex +BN+ Xavier	71.0	74.1	72.5
AlexNet + BN + Xavier +bias=1	71.0	74.8	72.9
AlexNet + BN + Xavier +bias=0	78.3	82.5	80.4
AlexNet + BN after_Pooling + Xavier +bias=0	**82.5**	**85.5**	**84.0**

The aforementioned analysis and architecture building resulted in networks manually designed for breast lesion classification. To evaluate such manually designed networks with our models in Section 4.2 and 4.3, two of the best-performed models based on the modified AlexNet architectures are selected to compare with the ENAS models. For conveniences, the modified

AlexNet structure without the second convolutional layer and max-pooling but with the filter size setting to 11×11 achieved good classification performance with the average accuracy of 81.2%, and is named as *SM-AlexNet*. The modified AlexNet by using Xavier (bias = 0) as a weight initialization method, replacing LRN with BN and changing BN from before pooling layer to after pooling layer boosted the accuracy further to 84%, and is named as *TM-AlexNet*. However, the results shown in Table 4.9 demonstrate that the automatically designed CNNs (ENAS 17 and 7) still outperform the manually designed Modified AlexNet. Moreover, ENAS-Based models are much lighter than the modified models in terms of the number of weights.

Table 4-9 Comparison between Manually Modified AlenNet and ENAS Models

Models	TNR	TPR	Accuracy	No. of Weights
SM-AlexNet	81.7	80.6	81.2	89,051,834
TM-AlexNet	82.5	85.5	84.0	56,858656
ENAS17	**86.7**	**92.0**	**89.3**	**4,251780**
ENAS7	**90.9**	**86.7**	**88.8**	**2,342484**

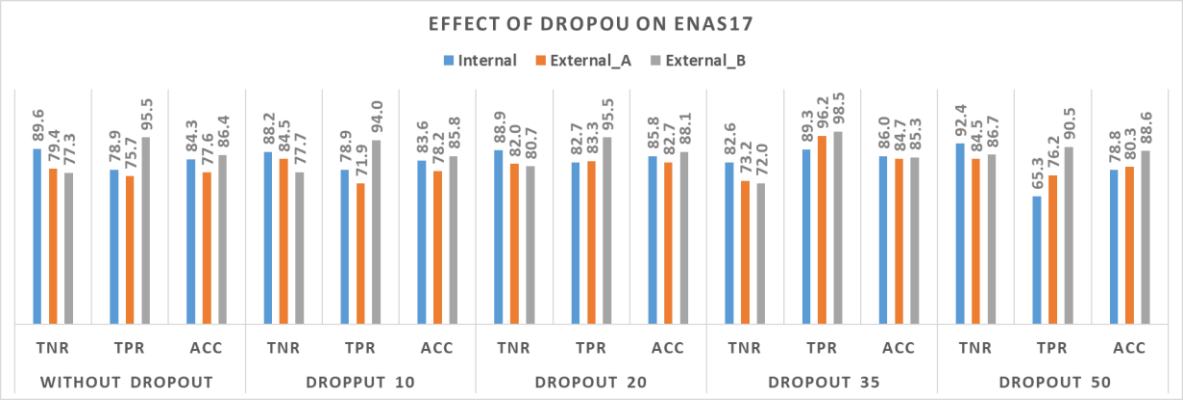
4.5. Discussions

In order to improve the generalisation rate of ENAS based models on external datasets, three methods have been examined. Each of these methods has a distinct influence on the generalisation of the ENAS based models. By reducing ENAS17 complexity nearly 50%, ENAS7 models achieved a slightly reduced accuracy, but generalisation errors not significant reduced by simpler ENAS model. On the impact of data augmentation methods, external test results demonstrated that the ENAS models trained on training set expanded by the rotation data augmentation method performed similarly to those models trained on dataset obtained by other data augmentation methods. In some specific cases, some models such as ENAS7 models trained on unbalanced data set, performed even better for external tests than the models based on other methods of data augmentation.

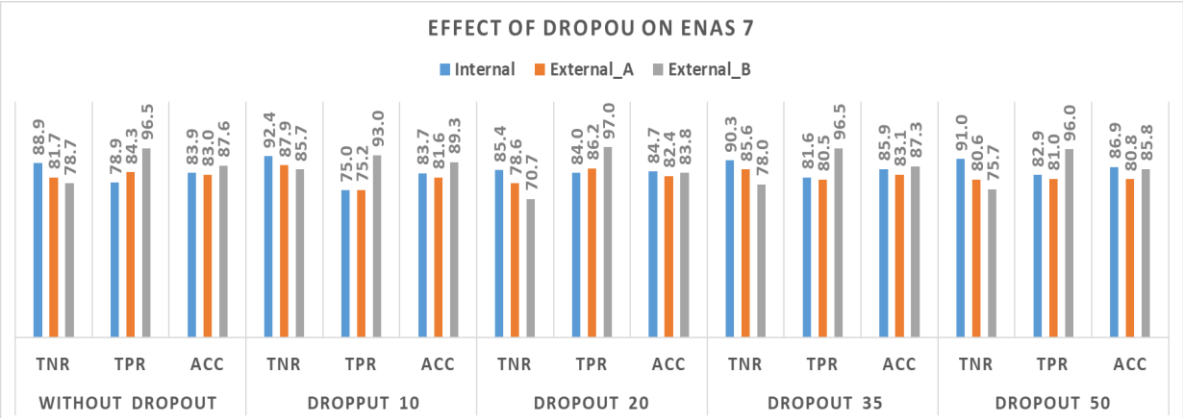
The most effective technique for reducing the generalisation error of ENAS models is to use the unbalanced dataset at the modelling stage of the ENAS method. When the unbalanced data set is used for training classifiers, the models typically over-learn from the majority class due to increased priority and significance. We exploited this negative effect to balance out the loss of accuracy by increasing the number of benign cases for our dataset. To further explore the effect of different class balance ratios on reducing generalisation error, we conducted a few controlled tests on various benign vs malignant ratios. Training the ENAS models with training examples at the ratio of 1.20:1, the generalisation error did not improve. Then at the ratio to

1.40:1, we noted that the generalisation error of TNR reduced by around 3% on External_A and 8% on External_B. Therefore, we were encouraged in using a higher ratio of 1.93:1 in the modelling data set (see Section 4.1), and a good generalisation on the external data sets was shown in Table 4.5. The optimal ratio is still an open research question for the time being.

The work in this chapter attempted only a few methods for improving ENAS model generalisation rate. Several other methods may also help. For example, the dropout rate as mentioned before is also one of the commonly used techniques. Within a small scale, we conducted an experiment to evaluate the effects of the dropout rates on ENAS17 and ENAS7 models, and determine the optimal dropout rate for the models. Both models trained from scratch with different dropout rates such as 0%, 10%, 20%, 35%, or 50%. It is worth noting that the dropout rate of 20% on the fully-connected layer was defined as the default in the ENAS modelling stage. In this experiment, unbalanced data sets with all augmentation methods were used for training ENAS17 and ENAS7 models.



(a) Effects of Drop-out Rates to Generalization of ENAS17 Models



(a) Effects of Drop-out Rates to Generalization of ENAS7 Models

Figure 4-9:Effect of different Dropout rates on ENAS 17 and ENAS7 models

We applied an 80% - 20% single split for training and testing for the internal test, whereas External_A and External_B sets were still used as external testing data sets. The test results show (Figure 4.9) that different dropout rates only have marginal effects on generalisation

errors. The results also showed that the TPR and overall accuracy decreased significantly more for the ENAS17 models trained with dropout rates of 10% and 50% than those of the ENAS17 models trained with the default dropout rate of 20%. We, therefore, decided to maintain 20% as an optimal dropout rate in ENAS17 models. The results also show that the ENAS7 models with a dropout rate 35% perform better than ENAS7 with 20%. Figure 4.10. More investigation is still needed on this aspect of reducing generalisation errors for ENAS models. Figure 4.10 show the effects of various drop-out rates towards the model accuracy for ENAS17 models (Figure 4.9(a)) and for ENAS7 models (Figure 4.9(b)). The ideal drop-out rates are either the default 20% or 35%.

4.6. Summary

This chapter presented different approaches for developing deep convolutional neural networks (DCNNs) for breast lesion classification in ultrasound images. In particular, this chapter consists of three main parts. The first part adapted the ENAS method for searching for the optimal CNN architecture for breast cancer classification and selecting the optimal cell. The second part focused on testing generated ENAS models on external datasets and investigating different techniques for reducing the generalisation error of ENAS models for breast cancer classification from US images. The third part of this chapter compared our ENAS models with existing state-of-the-art CNN models.

We first started by using the ENAS technique for automatically searching for optimal CNN architecture design for breast cancer classification from US images. Then, the best cell is selected (based on validation accuracy) among generated cells to design a CNN architecture which is ENAS 17. After training ENAS 17 from scratch the average test accuracy of ENAS 17 reached 89.3%. While, the evaluation accuracy of the ENAS17 model on external test datasets was reduced by around 10% in overall test accuracy and more than 20% in TNR.

We then evaluated several techniques to reduce performance drop in ENAS 17, by reducing ENAS 17 layers and proposing ENAS7, examining different types of data augmentation using an unbalanced dataset technique, and altering the dropout rate setting. The most effective technique for reducing overfitting in our ENAS models is the unbalanced dataset technique which is reduced overfitting to100% in overall testing accuracy and 50% in TNR in both ENAS 17 and 7 models. The ENAS 17 model that trained on unbalanced data achieved good classification performance on internal, External_A, and External_B sets with an average accuracy of 85.8%, 82.7%, and 88.1%, respectively. The possible explanation for these results

is that some of the important features of benign cases would be lost during training ENAS models, therefore increasing benign cases for the training set will avoid that issue.

On the other hand, we made several manual changes to the AlexNet CNN architecture including increasing filter sizes, reducing convolutional layers and exploring various hyperparameter tunings such as weight initialization, batch normalization and bias. Overall, this exploration into CNN architecture in general and especially ENAS guides us for modifying ENAS architecture to achieve better accuracy and robustness of the models, with this limited amount of data, which is a common problem in medical image analysis

The main findings of this chapter can be summarised as follows:

1- The study in this chapter demonstrated that the ENAS approach to CNN model design is a promising direction for classifying ultrasound images of breast lesions.
2- ENAS based CNN models outperformed the handcrafted CNN architectures in a transfer learning setting and training from scratch for breast lesion classification using US images.
3- Reduced generalisation error of ENAS based CNN models by using unbalanced datasets in training. We demonstrated that using unbalanced dataset in the training set is better than data augmentation and complexity reduction for the purpose of generalisation error.
4- Using more than one method of data augmentation to increase training samples is better than using a single augmentation approach to design a good ENAS based CNN.
5- The manual adaptation of well-known CNN architecture (e.g. AlexNet) is time consuming and requires large number of trails. This highlights the need for the automatic network design.

AlexNet architecture with large filter sizes and fewer layers is better than original AlexNet for lesion classification using US images. Also, using Xavier weight initialisation and batch normalisation with original AlexNet model is improving the performance of lesion classification by 30% overall accuracy in comparison with standard default AlexNet architecture.

Intrigued by the ENAS performance, Chapters 5 and 6 present our approaches of adapting ENAS for accurate classification task. In particular, in the next chapter, we modified the ENAS structure by adding a new highway connection to the architecture and using several operation sets in search space. Chapter 6 presents a novel approach of using Bayesian optimisation with ENAS for building CNN architectures.

Chapter 5. Structural Modifications of ENAS Architecture

Chapter 4 presented our successful adaptation of the ENAS approach for automatically designing CNN-based models for classifying breast lesions from US images with performance higher than the handcraft architectures and transfer learning ones. However, whether further customization of the ENAS architectures can lead to more accurate and robust network remains an open question. The aim of this chapter is to answer this research question by investigating into possible modifications to the CNN architectures from the ENAS framework.

The proposed approaches to modify ENAS can be summarized as follow. First, the ENAS search space for operations is modified by adding additional suitable operations guided by the existing literature and our research experience, aiming to give the RNN controller a wider range of operations to select from when searching for an optimal cell. In addition, we will modify the layer structures of the backbone architecture by adding skip connections. Intrigued by existing research in skip connections such as DenseNet [43], we aim at introducing different types of skip connections within the default architecture. At the conceptual level when the design of a CNN architecture as searching for a subgraph of a directed acyclic graph (DAG), the first modification is made to the internal structure of vertices whereas the second modification is concerned with adding edges between vertices of the DAG.

This chapter is therefore composed of the following sections. Section 5.1 focuses on the proposed approach of enriching the ENAS operation list based on our own research results described in Section (4.4) and thoughts reported in the literature. Section 5.2 outlines three principles of adding skip connections and introduces the concept of highways of various kinds and the rationales behind them. Section 5.3 then evaluates the proposed modifications to the ENAS framework through a set of experiments and analyses the results. Section 5.4 will discuss some issues raised through this investigation before Section 5.5 summarizes the main findings from the research reported in this chapter.

5.1. Expanding Operation Set for ENAS Search Space

There are several types of convolutional operations, such as normal convolution, depthwise separable convolution, and dilated convolution [103]. Each of these operations includes several hyper-parameters that directly have impacts on the performance of the CNN architecture. Since

the ENAS – Micro scheme searches for optimal cells, the RNN controller selects the optimal operations from the provided set of operations and the connection between the operations for generating the internal structures of cells. As a result, adding sensible alternative operations will widen the choices that the controller may take in designing optimal cells. In order to explore the useful operations for ultrasound image analysis, we will consider adding new operations to the default operation list already used by the ENAS framework. However, increasing the size of the operation search space will also increases searching time and hence require additional computational power. Consequently, this modification has to be performed with care.

In NAS [58], 13 different operations were introduced in the operation search space. The follow-up approaches of NAS (PNAS and ENAS) reduce the operations to 8 and 5, respectively but still work well (see Chapter 4) despite the fact that the operations were selected explicitly for natural images. This observation, therefore, leads to two conclusions. First, it is not the more operations but which operations to include that matter. Second, the diversity of operations of different types in the search space may give the controller a chance to generate better cells.

In the ENAS operation search space, there are 2 small (3×3 and 5×5) depthwise separable convolutions, 2 alternative pooling operations from local (3×3) neighbourhoods, and the identity operation, with stride 1 for normal cell and 2 for reduction cell for all operations. While identity operation provides opportunities for skip connections among the selected operations, the two convolutions aim at extracting local features within the local neighbourhoods. The pooling operations are meant to reduce dimensions gradually. We have therefore concluded that the focus of attention should be at increasing alternative convolution operations as such an operation enables extracting additional local textures. We therefore keep the pooling and identity unchanged, and systematically enlarge the operation search space with additional convolutions. Table 5.1 summarizes the original operations and the additional ones to be added under three different settings.

To understand the effects of the added operations systematically, we have decided to introduce these operations in three different settings. In the first setting, Operations of Set A include 5 existing ones plus 2 extras: 7×7 depthwise separable and a 9×9 standard convolution. The addition is based on our own handcraft adaptation of AlexNet (as reported in Section 4.4.2) where using convolutional layers with larger size filters results in models with better classification accuracy for breast cancer classification from ultrasound images. However, A convolutional layer with a large filter size means an increased number of weight parameters in a CNN architecture, leading to complexity and over-parameterisation. Therefore, in the second

setting, Operations of Set B include more efficient convolutions, i.e.7×7 depthwise separable and 9×9 depthwise separable instead, reducing the number of weight parameters and multiplication by 23x. However, different types of convolution operations affect CNN architecture performance and complexity. Therefore, in the third setting (Operations of Set C), three different types of convolutions with varying filter sizes are added to the default ENAS operation set: 7×7 depthwise separable, 3×3 standard Convolution and 3×3 dilated convolution with dilate rate 2.

Table 5-1:Added list of operations to default ENAS operations in 3 different scenarios.

	Operations of Set A	**Operations of Set B**	**Operations of Set C**
Operations of Original ENAS search Space	3×3 depthwise separable	3×3 depthwise separable	3×3depthwise separable
	5×5 depthwise separable	5×5 depthwise separable	5×5depthwise separable
	3×3 Avg-Pooling	3×3 Avg-Pooling	3×3 Avg-Pooling
	3×3 Max-Pooling	3×3 Max-Pooling	3×3 Max-Pooling
	Identity	Identity	Identity
New Operations Added to the Search Space	7×7 depthwise separable	7×7 depthwise separable	7×7 depthwise separable
	9×9 normal Convolution	9×9 depthwise separable	3×3 normal Convolution
	-	-	3×3 dilated convolution

5.2. Designing High-way Connections for ENAS Backbone Architecture

Small details or information in the input image may be lost after passing through many layers (convolutional and pooling). This problem may be caused by weights in earlier layers poorly trained during the "vanishing gradient" issue in the backpropagation process. Various methods have been attempted to reduce the problem. Highway network [104] and ResNet [42] proposed adding skip connections, i.e. passing feature maps from one layer directly to the following layers, to reduce this problem and training deep CNN networks with hundreds of layers. Recently, DenseNet [43] proposed an architecture that connected all layers to the final convolutional layer to ensure that maximum information passed among the layers. In the neural architecture search approach (e.g. NAS, ENAS and PNAS), skip connections are used as one of the searchable hyper-parameters optimized by the controller. Most of the existing skip connections pass the output of the early layer as input to deeper layers. Two operations commonly used for combining skip connection features with the features of the abstract layer are addition [42], [104] and concatenation [43].

In the original ENAS backbone architecture, there are two types of skip-connection. The first is inside the cells generated by the controller, and the second skip-connection is between the cells (see Section 2.5.2). In our study, we introduce a new type of skip connection, known as Highway, into the ENAS backbone architecture. This highway connection aims to contribute a fixed amount of the features extracted in early layers into the network's final decision, aiming at combining low-level features extracted from the early layers with more abstract features from later layers can assist the classification decisions. Therefore, the feature maps of early layers are directly combined with the last layer's output using the concatenation operation. In other word, the concatenation operation joins a collection of inputs together. It accepts a list of tensors with the exception of the concatenation axis that are all the same shape as inputs and outputs a single tensor that is the concatenation of all inputs.

In our proposal, the starting point of the highway and the operation used on the highway are the important hyper-parameters for designing a highway connection. We consequently propose four different highway connections, which will be explained in more detail next.

Short Highway

In general, each layer of a CNN architecture extracts specific type of features. In general, the early layers mainly focus on low-level and basic features such as local shapes, edges, salient points, and corners, whereas deeper layers extract higher level and more abstract features such as texture patterns, overall shapes [105]. Both basic and abstract features can be useful for medical image classification [106]. The first proposed highway connection can be seen as Short Highway, also known as “within block” highway, bringing the output of the second reduction layer of ENAS17 CNN architecture to the final layer. The intention is to combine the feature maps of the second reduction layer with the feature maps of the last layer before applying global average pooling. Based on the operation used to combine the feature maps, we can sub-categorise this scheme into:

(a) short-highway-identity-addition (SHIA) where we add the features of the second reduction layer with the last layer,
(b) short-highway-identity-concatenation (SHIC) which concatenates the second reduction layer features with that of the final layer,
(c) short-highway-with-25%-features (SH25) that concatenates the feature maps from the second reduction layer with the feature maps of the last layer but reduces the number of feature maps by 25% using 1×1 convolution followed by Batch Normalisation and Relu. The rationale behind SH25 is to balance the amount of feature maps passed by

short highway and the feature maps of the final layer. In other words, in ENAS17, the output of each normal cell (layer) the concatenation of four nodes (see Figure 4.4), thus in this highway the amount of passed features reduced to the same amount of features in one node, also for controlling increase the model's complexity. The three short highway structures are shown in Figure 5.1. In this study, we first design ENAS17 CNN cells using the original ENAS architecture search (see Section 4.2).

Medium Highway

Another proposed highway connection is Medium Highway showed in Figure 5.2, which passes the feature maps from the first reduction layer to the final set of the feature map. The output (feature map) size of the first reduction layer is 50 × 50, while the output size of the last layer is 25 × 25. Therefore, in the Medium Highway, max-pooling (3×3, stride 2) has been used to reduce the size of feature maps and 1×1 standard convolutional layer with a 25% ratio of feature maps, followed by Batch Normalization and Relu used to reduce the number of output channels.

Long Highway

We propose the Long Highway to pass the low-level features from an early layer to the final layer as shown in Figure 5.3. This skip connection starts from the first layer to the last layer. In other words, skipping both reduction layers of the ENAS17. Therefore, as an alternative for reduction layers, max-pooling (5×5, stride 4) is used for reducing output size, 1×1 convolutional, followed by Batch Normalization and Relu.

Combined Highway

A combined Highway is simply the combination of short, medium and long highways as shown in Figure 5.4, providing an opportunity to combine features of different levels of abstraction at the end of the feature extraction process. In particular, the outcome from an early convolutional layer, the outcomes from two reduction layers are combined with the final outcome of the final convolutional layer via concatenation. The purpose of proposing this highway is to investigate the effect of the mixture of highway connections on the performance of ENAS17 models in differentiating malignant tumours from their benign counterparts. The outputs of the highways and the outputs of the final convolutional layer can be combined in a weighted scheme. For instance, each output counts for 25%, an equal weighting in the final concatenated feature map output. For example, the output of final layer is 576 feature maps and the number of feature

maps passed by short-highway(25) is 144 channel which is equal to 25% of the output of the final layer. The ideal weight distribution can be experimentally investigated.

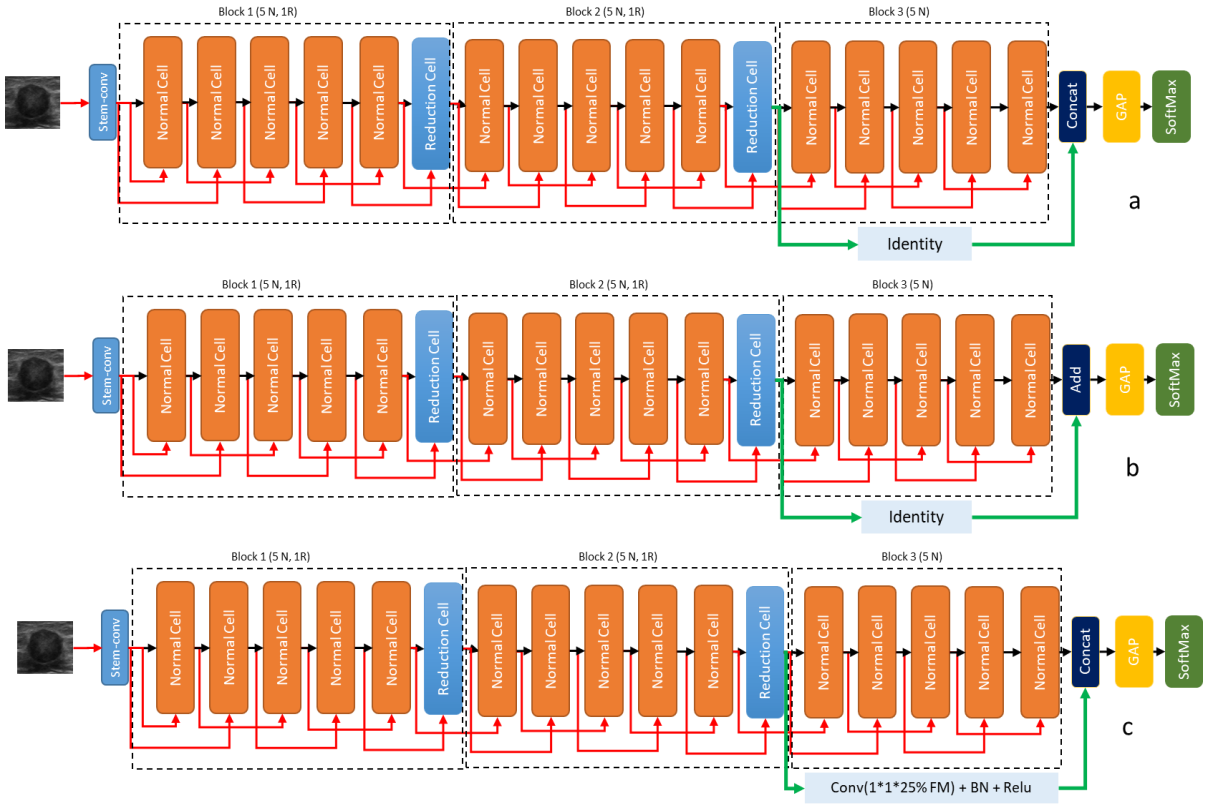


Figure 5-1:Protocols of Short Highway connection applied on ENAS17. (a)Identity- Concatenation. (b) Identity-Addition. (c) Short-highway (25%)

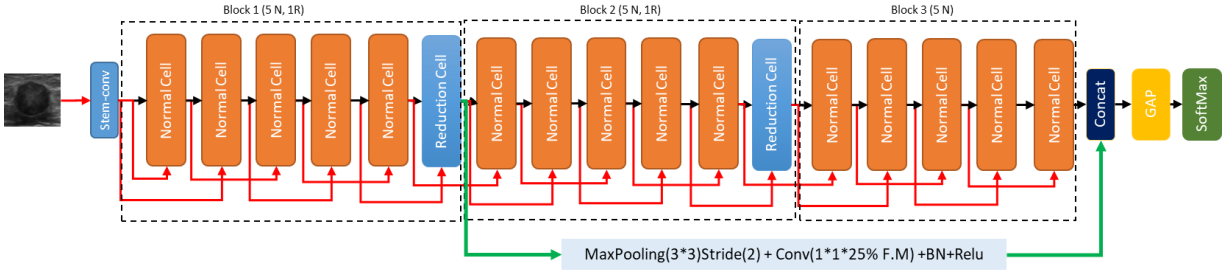


Figure 5-2:Modified ENAS17 architecture by adding Medium Highway

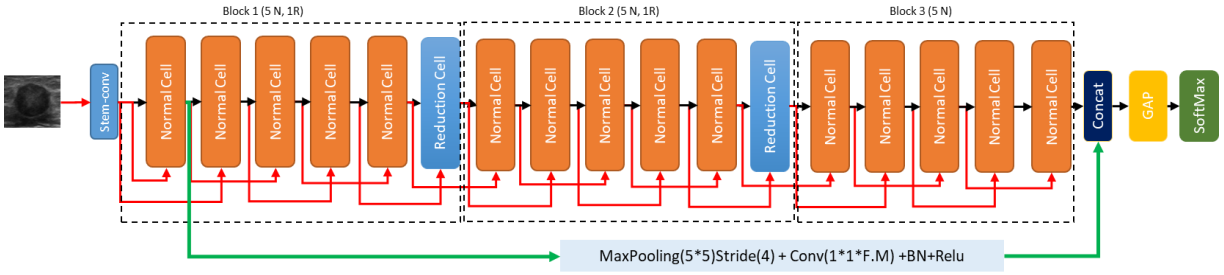


Figure 5-3:Modified ENAS architecture by adding Long Highway

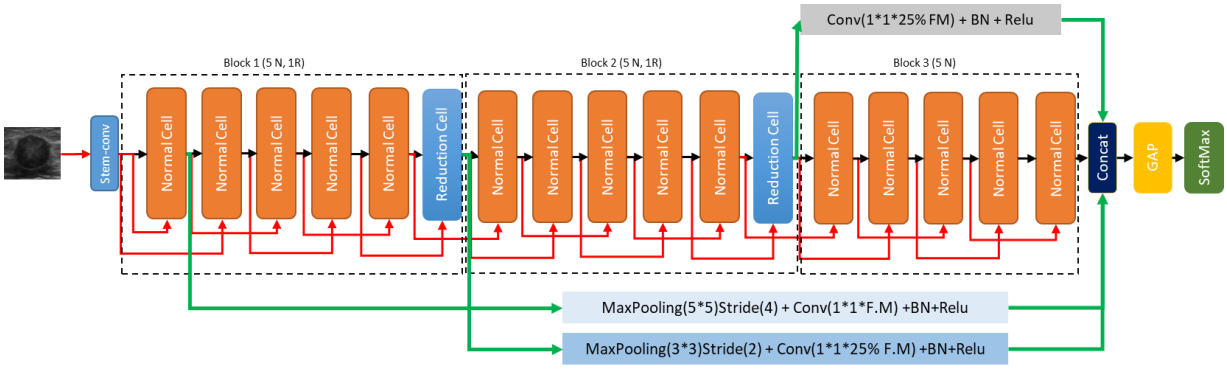


Figure 5-4:Combined Highway connection schemes for ENAS17 CNN architecture.

5.3. Evaluation of Performances of ENAS Modifications

In this section, we present the experimental results of evaluating the modified structure of the ENAS CNN architecture search. We will start by first showing performances of the expanded search space of operations, followed by results of different highway connections proposed to enhance ENAS CNN architectures.

5.3.1. Evaluation on Effects of Expanded ENAS Search Spaces of Operations

In the searching phase, we first use a balanced dataset BModelling with data augmentations as described in Section 4.1.3, and all hyper-parameter settings are the same as discussed in Section 4.2. Since there are three settings for the proposed operation expansion (see Table 5.1), the optimized normal cell and reduction cell for each of the three settings are respectively shown in Figures 5.5, 5.6 and 5.7, where the selection of the proposed operations are highlighted in red rectangles. The searched cells indicate that all the proposed operations have been used to obtain optimal CNN cells determined by the controller. The separable convolution with larger filter size 7×7 is used most by the controller.

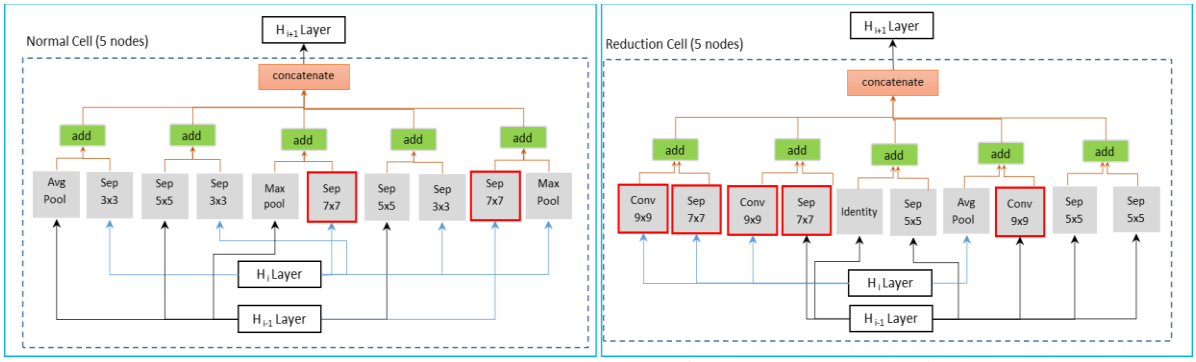


Figure 5-5: Optimal Cells Generated by ENAS-A for Breast Cancer Classification

Each pair of the optimized cells are used to design two CNN backbone architectures: ENAS7 and ENAS17 to test the performances of the optimized cells in shallower and deeper networks. Therefore, there are 6 CNN models to be trained. For model training, we use the

unbalanced dataset (Modelling, see Section 4.1.1) together with the augmentations and hyper-parameter settings as described in Section 4.2. To test the generalization of each of these CNN models, we use two external datasets (External_A and External_B) as described in Section 4.1. Performance measurement metrics is TPR, TNR and Accuracy as defined and discussed in Section 4.4. Table 5.2. shows the performance results of the models. The first row shows the performance of the ENAS7 models based on the cells optimized from the original ENAS set of operations (Section 4.2). The rest rows list ENAS models built on cells obtained from expanded search spaces. We list the operations in each search space for convenience.

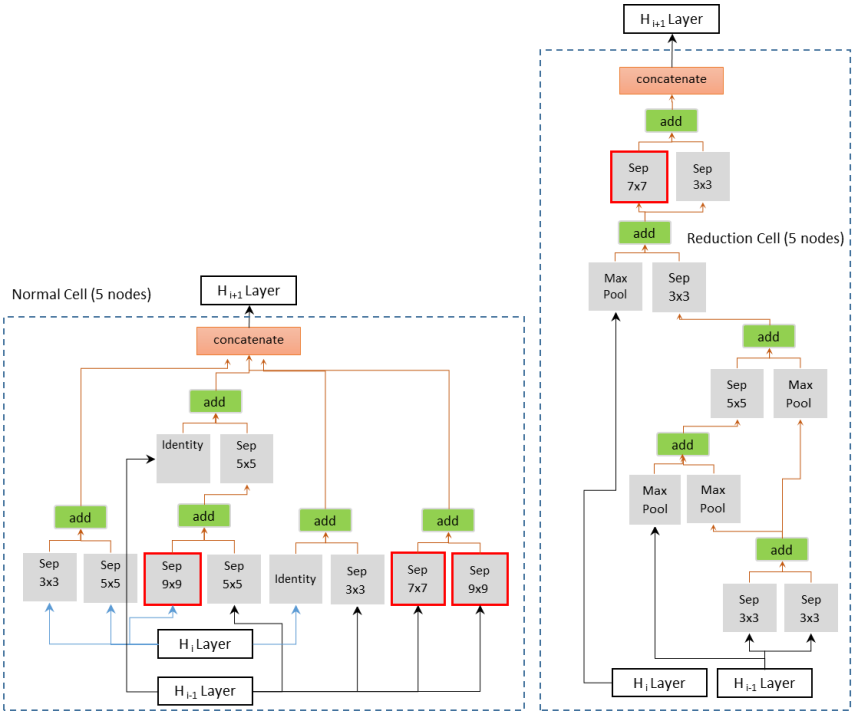


Figure 5-6: Optimal Cells Generated By ENAS-Set-B for Breast Cancer Classification

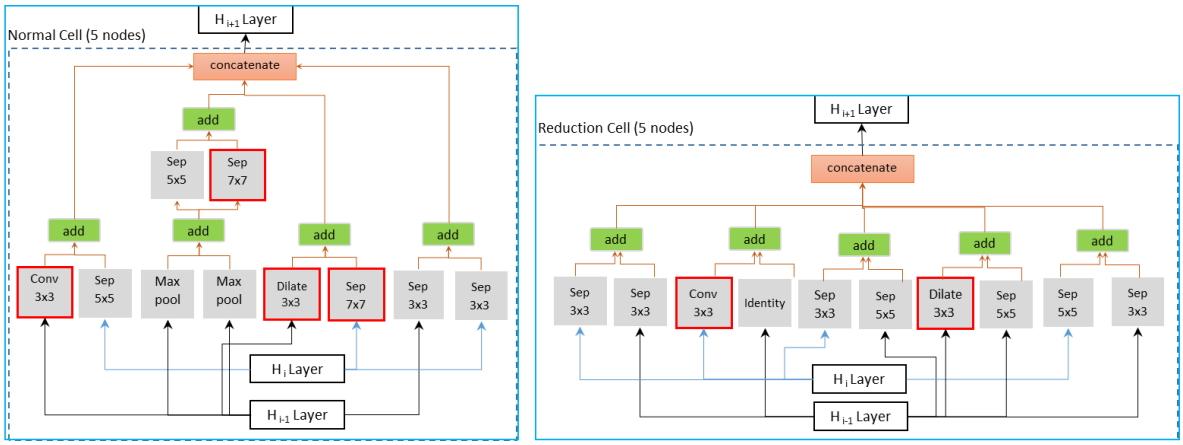


Figure 5-7: Optimal Cells Generated By ENAS-Set-C for Breast Cancer classification

Table 5-2: Performance of ENAS Models with Cells Optimized from Modified Search Space of Operations (DSC: Depthwise Separable Convolution; AP: Average Pooling; MP: Max Pooling; ID: Identity; C: Normal Convolution; DC: Dilated Convolution)

Models	Test Sets	ENAS7			ENAS17		
		TNR	**TPR**	**Acc**	**TNR**	**TPR**	**Acc**
ENAS Operation Set (3×3 DSC, 5×5 DSC, 3×3 AP, 3×3 MP, ID)	Internal	*92.0*	*69.2*	*80.6*	***89.5***	*72.6*	*81.1*
	External_A	*83.8*	*75.4*	*79.6*	*80.5*	*84.1*	*82.3*
	External_B	*80.8*	*94.6*	*87.7*	*79.8*	*94.8*	*87.3*
Operation Set A (3×3 DSC, 5×5 DSC, 3×3 AP, 3×3 MP, ID, 7×7 DSC, 9×9 C)	Internal	89.1	**76.1**	**82.6**	85.4	75.3	80.4
	External_A	80.8	**85.0**	**82.9**	78.7	**84.4**	**81.5**
	External_B	74.5	**96.5**	85.5	72.5	**95.7**	84.1
Operation Set B (3×3 DSC, 5×5 DSC, 3×3 AP, 3×3 MP, ID, 7×7 DSC, 9×9 DSC)	Internal	**90.2**	**72.9**	**81.6**	89.1	71.1	80.1
	External_A	83.6	**78.3**	**81.0**	79.8	83.2	81.5
	External_B	76.9	94.6	85.8	**81.0**	94.5	**87.8**
Operation Set C (3×3 DSC, 5×5 DSC, 3×3 AP, 3×3 MP, ID, 7×7 DSC, 3×3 C, 3×3 DC)	Internal	88.8	**73.7**	**81.3**	85.8	**78.0**	**81.9**
	External_A	81.0	**80.1**	**80.5**	77.6	**88.1**	**82.9**
	External_B	76.3	**95.0**	85.7	74.9	**97.2**	86.1

For ENAS7 models, the results in Table 5.2 show that there are some marginal improvements on overall accuracy for either the internal or some of the external data sets. Most of the improvements come from the improvements over TPR. For ENAS17 models, there are some shifts in performance between TPR and TNR, but on overall accuracy, there is no apparent improvement of performance. Overall, the original ENAS search space for operations can still produce cells that are either better than or at least equivalent to those from the expanded search space. This is not quite as we expected; the uses of alternative convolutional operators do not bring apparent added benefits.

However, in particular, there are some consistent improvements of performance on TPR for all ENAS7 models at the cost of reduced performance of TNR across the different datasets. Similar trends also exist for ENAS17 models but only for Operation sets A and C. The findings of this investigation may indicate that for breast lesion classification from ultrasound images convolutions with smaller local filters work as good as convolutions with larger local filters in the ENAS search scheme; and hence the focus of attention should be on other aspects of CNN architecture optimization. Appendix B presents more details of ENAS7 model with Set A.

5.3.2. Evaluation on Effects of Highways on ENAS Backbone Architectures

This section reports the evaluations on the effects of highways added on the ENAS17 backbone architecture. We are interested in finding if and how the added skip connections affect the accuracy of the classification models built on the modified architecture. The setting up of

training, testing, and hyper-parameters are all similar to the details set Section 4.2 where the dataset used for model development is the same unbalanced dataset as introduced in Section 4.1.1.

Table 5-3: Performances of ENAS 17 Models built on Architectures with Various Highways

Models	Test sets	TNR	TPR	Acc
ENAS17 architecture without Highway	Internal	*89.5*	*72.6*	*81.1*
	External_A	*80.5*	*84.1*	*82.3*
	External_B	*79.8*	*94.8*	*87.3*
ENAS17-SHIC architecture with short highway (Identity + Concatenation)	Internal	90.1	63.9	77.0
	External_A	82.9	73.1	78.0
	External_B	80.1	91.2	85.6
ENAS17-SHIA architecture with short highway (Identity + Addition)	Internal	88.3	73.2	80.7
	External_A	79.9	81.4	80.7
	External_B	78.3	96.4	87.3
ENAS17-SH25 architecture with short highway (25% feature maps)	Internal	86.0	79.1	**82.5**
	External_A	75.6	89.5	**82.6**
	External_B	71.8	97.6	84.7
ENAS17-MHW architecture with medium highway	Internal	91.2	73.2	**82.2**
	External_A	85.1	66.4	75.8
	External_B	80.6	93.5	87.1
ENAS17-LHW architecture with long highway	Internal	92.7	67.6	80.2
	External_A	85.7	72.2	79.0
	External_B	**86.1**	**91.8**	**88.9**
ENAS17-CHW architecture with combined highway	Internal	91.0	71.6	**81.3**
	External_A	80.8	78.0	79.4
	External_B	77.4	94.2	85.8

Table 5.3 summarizes the performances of the ENAS17 models built on the different highway architectures. The test results show a reduction in overall accuracy and TPR comparing to ENAS17 models without any highway. The combined highway architecture is also not improving the performance of ENAS17 models either. In particular, the ENAS17 models with the SH25 architecture performed slightly better than the ENAS17 models without highways, but the improvements are marginal (Appendix B for more details of the result). For instance, using long highway boosts the performance by nearly 2% with quite a balanced TPR and TNR on the External_B dataset. This finding indicates that our handcraft approach of directly modifying the ENAS architecture can be further improved by developing optimization method of stacking the ENAS cells.

5.4. Discussions

In this chapter, we investigated and evaluated the effects of modifying the ENAS cells and network structures by expanding the search spaces for operations and adding skip connections to the network layers. Although each approach has some impacts on the ENAS models, the effects are not as obvious and significant. The remaining research question is what other modifications can be carried out to the ENAS backbone architectures and what will be the effects of such further changes. Therefore, two additional modification methods have been investigated in this research to answer the current research questions concerning creating the final CNN architecture by stacking generated cells using ENAS technique.

One potential weak point of the ENAS Micro search scheme is that the same optimal cells (both normal and reduction cells) are repeatedly used to design the entire CNN architecture [90] by stacking the cells. Some researchers proposed an approach for modifying the DART method by stacking different optimal cells for designing the CNN architecture [107]. We did a pilot study by designing the final CNN architecture with three different optimal cells to attempt answering the research question. Since the ENAS architecture consists of three blocks based on two reduction cells as described in Section 4.2.2, After selecting three optimal pairs of cells from the generated cells by ENAS-set-A, we designed ENAS7 by repeating each optimal cell in one block. Then ENAS7-Mixed was trained on unbalanced dataset from scratch, and the results reported in Figure 5.8.

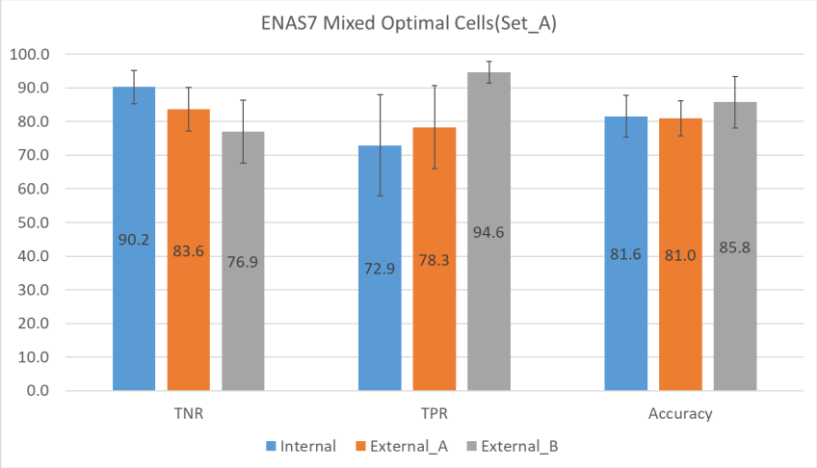


Figure 5-8: Result of ENAS7 Designed by Using Three different Optimal Cells

The result showed that ENAS7 model that designed by repeating one optimal cell, slightly outperforms ENAS7 by using three different optimal pair of cells. The original ENAS approach uses the same cells to train the Supernet in searching stage. Therefore, further research to use

multiple optimal cells during the search stage is needed in the future. However, this investigation is outside the scope of this thesis.

As described in Chapter 2, CNN architecture consists of two main parts, feature extraction and classification part. The modification applied to ENAS17 was on the feature extraction part, which was adding a new skip connection. Besides, the classification part also directly impacts CNN architecture's performance. Moreover, most State-of-the-art CNN architectures, including ENAS-based CNN methods, use GAP instead of a fully connecting layer. Although using GAP reduces the number of CNN architecture parameters [26], using GAP may result in loss of information, especially in medical images [108]. Therefore, to explore the effect of fully-connected layer on ENAS-based CNN architecture, we modified ENAS17 architecture by adding a fully connecting layer (ENAS17-FC) with (1152) nodes which is a double number of output nodes of the global Average Pooling layer. Then we trained ENAS17-FC from scratch with the same setting and dataset mentioned before.

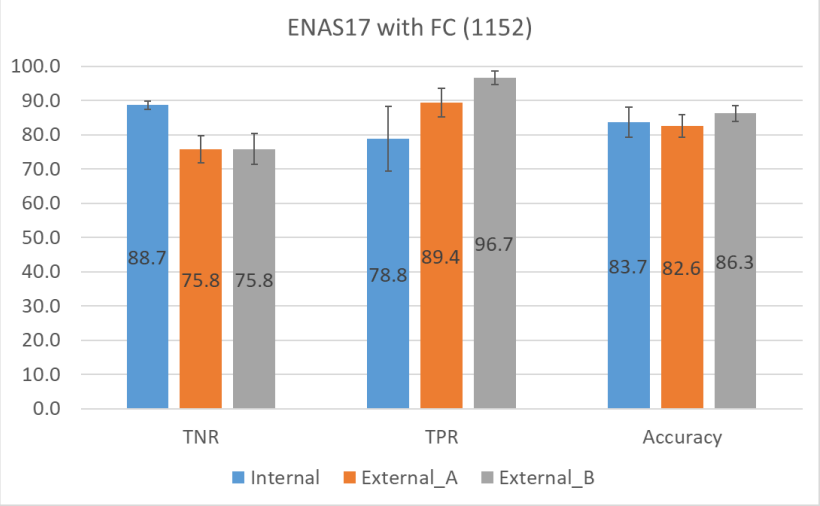


Figure 5-9: Results of ENAS17 with New Fully-connected Layer

The result in Figure 5.9 shows that adding a fully-connected layer for ENAS17 affected model performance. The overall accuracy of the internal test increased by (2.6%), 0.3% increase in External_A and a drop of 1% for External_B. Hence, we can say that we have not gained a generalisation improvement using a fully-connected layer and GAP remains a good choice for medical US images as well as natural image classification. Although, designing a proper fully-connected layer for CNN architecture requires more investigation regarding the number of fully-connected layers and the number of nodes per FC layer.

5.5. Summary

This chapter presented several approaches to improve ENAS-based CNN models for breast lesion classification. The first part of this chapter focused on enriching ENAS search space of operations by providing a new set of extra operations for the controller to generate optimal cells. The analysis showed that ENAS7 with the proposed operations perform marginally better than the original operation set on internal and external datasets especially balancing TPR and TNR while boosting overall accuracy.

Secondly, we investigated the effect of different highway connections on the ENAS17 model. Different highway strategies were proposed, and its effect presented using different internal and external datasets with unbalanced data. In conclusion, there is no significant improvement by adding extra skip connection into ENAS17 regardless of the dataset used in training and testing phases.

Furthermore, we also investigated the use of multiple different optimal cells instead of repeating one optimal cell in the pipeline of optimal CNN architecture engineering. Although there is not improvement in the overall performance on the three datasets we utilised, but we must state that this experiment is not enough to prove that repeating one optimal pair of cells is the best approach for designing the final CNN in the Micro search space. Several hyper-parameters will impact the CNN's design by using more than one optimal cell. For instance, the number of different optimal cells and the order of optimal cells according to the blocks (i.e. which optimal cell will use in block 1, block 2 and block 3) are important factors in designing CNN models. Investigating all of these possibilities is a complex task and time-consuming but is on the list of future work to design an automatic framework to address this research question

The final set of experiments focused on the use of fully-connected layer after of GAP operation to see if it improves the performance of ENAS17 CNN architecture. There is a small performance boost for internal and External_A datasets but if we consider the complexity increase and the number of parameters increase, then the increase of performance is insignificant.

The investigations in this chapter answered some of the research questions regarding the design of automatic CNN architecture search in general and ENAS Micro search space in special such as the impact of a fully-connect layer on overfitting, the effect of diversity of operation set that used as search space and the mixing of optimal cells for designing final CNN architecture. In addition, this exploration can be used as guidelines for researchers that ought to modify ENAS search space and design skip-connection for other medical image modalities.

These investigations in Chapters 4 and 5 lead us to propose a new automatic scheme in designing CNN method that searches for other effective hyper-parameters that the ENAS method cannot generate, such as trainable hyper-parameters and number of cells per CNN architecture. In the next chapter, we will describe our proposed automatic design method to address this research question.

Chapter 6. Bayesian Optimization of Hyper-Parameters for ENAS-based CNN Architectures

In Chapters 4 and 5, various aspects of adapting the ENAS approach for designing robust CNN architectures for breast lesion classification have been attempted. The research outcomes from the two chapters revealed the following facts. The ENAS approach for automatic architecture design can be effective for breast lesion classification from ultrasound images. This is demonstrated by higher levels of accuracy of ENAS models than those models built on existing CNN architectures with transfer learning and those models on purposely but manually designed CNN architectures. Although model overfitting can be an issue for the ENAS approach, the issue can be resolved with the methods developed in Chapter 4.

We acknowledged the potential limitations of the original ENAS approach and investigated two different schemes of modifying ENAS by adding more convolutional operations in the search space and building highway connections on top of the stacked layers of cells. However, these manual design attempts of modifying ENAS lead to the following findings. First, the cell structures generated from the original ENAS search space of operations can still be optimal for the intended purposes despite that the expanded search space can influence the cell structure design. In other words, the added operations into the search space do not have significant advantages. Secondly, the ENAS simplified method of stacking cells in layers may not be the optimal configuration of the network graph of cells. Building highway connections discovers that true positive or true negative rates can be significantly improved although the overall accuracies only marginally change, indicating that changing the configuration of the layer structures does have an effect on model performance. A better way of searching for the optimal configuration is needed. A further issue of interest is that ENAS fixes the CNN trainable hyper parameters. This raised the following question: *Can the trainable hyper parameters be optimized?* Finally, the manual modification of ENAS operations, number of cells, and training hyper-parameters for an optimal design is time consuming.

This chapter addresses the ENAS limitations for not being able to optimise the depth of CNN architectures and trainable hyper-parameters. We adopt the Bayesian Optimisation method to search for (a) the optimal number of CNN layers based on ENAS optimal cell structures and (b) the optimal setting of the trainable hyper-parameters. This approach is the

first piece of research work to automatically optimize the depth and modelling hyper-parameters for ENAS.

The rest of this chapter is organised as follows. Section 6.1 presents the general framework for optimizing CNN layer structures and trainable hyper-parameters by Bayesian Optimisation and the details for each stage of the process. Section 6.2 evaluates the effectiveness of the proposed approach through a series of experiments using different datasets of breast lesion US images. Section 6.3 compares the performances of the models from the proposed approach with those developed using the state-of-the-art methods of alternative designs. Section 6.4 discusses further issues arise from the earlier sections before Section 6.5 summarizes the main findings of the chapter.

6.1. Bayesian Optimization for ENAS CNN Architecture Design (ENAS-B)

We propose a novel framework for developing DCNN classification models as shown in Figure 6.1. The main approach of the proposed solution is to automatically optimize DCNN architectures and their architectural hyper-parameters, including trainable hyper-parameters, by adapting and utilizing the Bayesian Optimization algorithm. The innovation of the proposed solution is in the *"Automatic Search and Optimization"* part of the framework in Figure 6.1. As illustrated in Figure 6.1, the framework consists of three phases: Image Preparation, Automatic Search and Optimization, and Model Training and Refinement. In particular, the proposed framework can be briefly outlined as follows. Phase 0 is a general preparation of US images which includes cropping the regions of interest (RoI) or the lesion, image resizing, and selecting the number of training examples of both classes (benign and malignant). This phase is similar to the US image preparation presented in Chapters 4 and 5. The purpose of Phase I is to obtain an optimized backbone DCNN architecture and a set of optimized trainable hyper-parameters. Phase II uses the optimized architecture and the optimized trainable hyper-parameters to build a DCNN model for breast lesion classification. Bayesian Optimiser requires a definition of initial normal and reduction cells. Therefore, ENAS method with the same setting in Chapter 4 is used to search for the optimal internal structures of normal and reduction cells (see Section 6.1.1). Then, these two cells are used as input for the proposed Bayesian Optimization method. In particular, two methods (ENAS-B-I and ENAS-B-II) are proposed to search for optimal DCNN architecture for breast lesion classification. In ENAS-B-I, Bayesian optimiser searches for optimal number of layers (Block structure) and trainable hyper-parameters all at once (see Section 6.1.3). On the other hand, ENAS-B-II consists of

two search stages (Stage 1 and 2 in Figure 6.1) to identify the optimal architecture. In ENAS-B-II Stage 1, the optimized normal and reduction cells by ENAS are stacked on top of each other in a process controlled by Bayesian optimization algorithm, creating a sequential layer structure of the cells (see Section 6.1.4) and the whole architecture. After designing the whole DCNN architecture in Stage 1, ENAS-B-II Stage 2 employs the Bayesian optimization principle to search for optimal trainable hyper-parameters within the optimized DCNN structure of cells, delivering eventually an optimized backbone DCNN architecture for later modelling (see Section 6.1.4). Finally, Phase III takes on the optimized DCNN architecture created from either ENAS-B-I or ENAS-B-II to train the final DCNN model. Sections 6.1.1 – 6.1.3 further describe the details of each phase and stage.

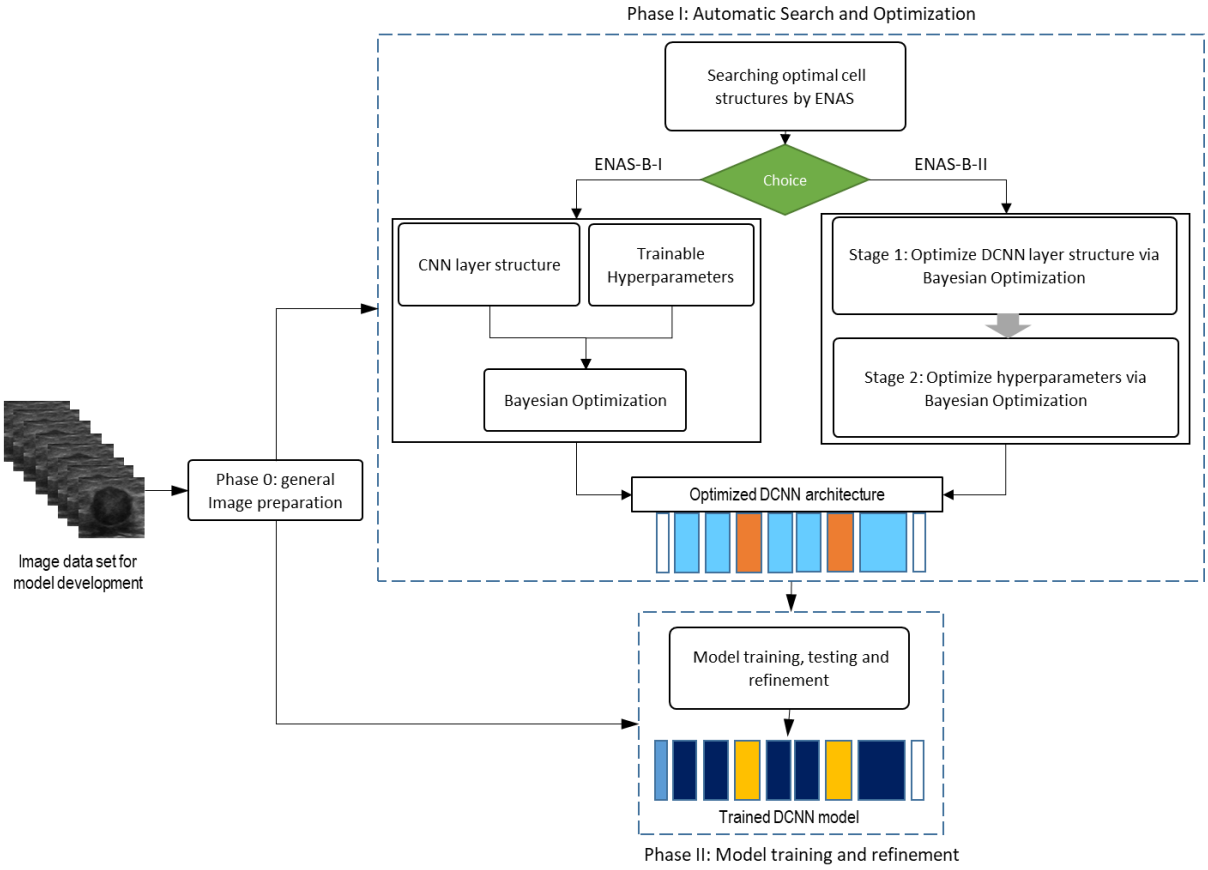


Figure 6-1: The Proposed Framework for automatic CNN Model designs for breast cancer classification from US images.

6.1.1. Normal and Reduction Cells Optimization

Automatic neural architecture search can be seen as searching for a subgraph within a directed acyclic graph of nodes (operations) and edges (information flow as input and output). Based on our initial investigation and evaluation, as described in Chapter 4, we follow the ENAS

micro approach for searching optimal cells (a constrained subgraph with a limited number of nodes) instead of the whole architecture search due to efficiency and accuracy.

The ENAS method with default search space set of operation and the same adaptation of ENAS (as described in Section 4.2) are applied here for generating optimal cells (Normal and Reduction cells). However, unlike the setup in Section 4.2, the unbalanced dataset was used (Section 6.2) for the search stage of normal and reduction cells. The rationale of using unbalance dataset is two folds: first, the investigations and results in Chapter 4 demonstrated that the modelling of ENAS using unbalance dataset produces accurate and robust DCNN models; and second, the Bayesian optimization methods (Sections 6.1.2 and 6.1.3) rely on unbalance dataset. In addition, the proposed method generates the final CNN architecture for breast cancer classification, thus using the same dataset for searching and modelling tasks will provide better models. Figure 6.2 shows an example of optimal cell structure (Normal and Reduction cells) generated from the data set we used. These two optimal cells are used to develop ENAS-B-I and ENAS-B-II.

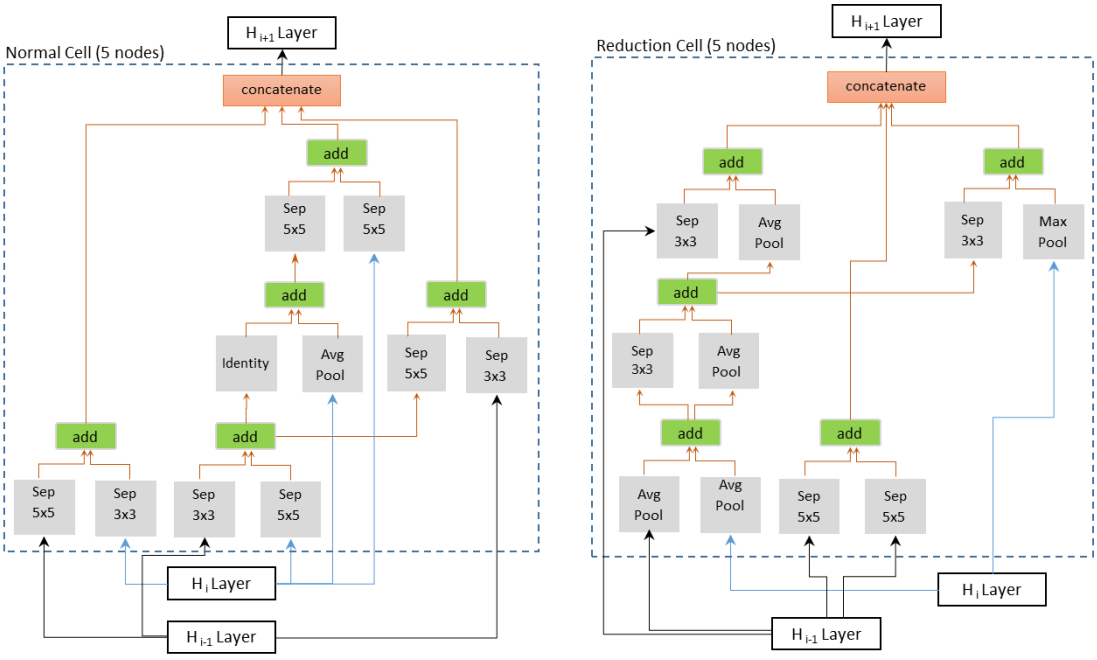


Figure 6-2: Optimal Cells (Normal and Reduction) optimized by ENAS method for Breast cancer classification

6.1.2. Search Space and Strategy

The ENAS architecture is a combination of normal and reduction cells (known as blocks) in a sequential pattern of normal, reduction, normal, reduction and finally, normal cells. The original ENAS method manually defines such structure by a pre-defined formula. The optimal model is finally determined based on test results at the modelling stage as described in Chapter

2. In addition, as mentioned in Section 2.5.2, ENAS forces all child models to use the trained weights of the *supernet*. This can be seen as one of the limitations of ENAS; it limits the ENAS's ability to search for optimal network depth and restricted ENAS to having one fixed backbone structure.

To overcome this limitation, we consider in our solution that the number of blocks and the arrangement of cells and tuning the trainable hyper-parameters is also a matter for optimization, and hence we propose to use an optimizer to search for the best block structure of CNN architecture with optimal trainable hyper-parameters. In particular, we propose an approach for searching optimal number of cells and trainable hyper-parameters for the final CNN architecture by using the normal/reduction cells generated by ENAS as the search space and Bayesian optimization based on a Gaussian Process as a searching strategy. We call this new network search strategy ENAS-B which links ENAS and Bayesian optimisation CNN architecture search. Next, we will describe the main components (*search space, backbone architecture and search strategy)* for building ENAS-B.

Search Space

In preparation for the Bayesian optimisation, search space needs to be defined. Suitable search space is a key element for successful architecture and model building. The search space in this study consists of deep learning architectural components and training parameters that can be used by the search strategy to design CNN models (see Chapter 2 for details). In particular, the search space is organized into two groups of searchable hyper-parameters: 1) Structural hyper-parameters (number of layers (or normal cells) per block); and 2) Modelling hyper-parameters.

- *Structural Hyper-parameters Search Space*: Section 6.1.1 shows the method of generating optimal normal and reduction cells (Figure 6.2). This section aims at using these two cells and identifying the optimal number of normal cells to build the whole optimal architecture for breast lesion classification in ultrasound images. In particular, a backbone architecture and the minimum/maximum number of cells are provided as an input to the Bayesian optimization to find the optimal CNN architecture (or ENAS-B). Figure 6.3 shows the proposed backbone architecture with input, stem convolution layer, three blocks, GAP, fully-connected layer, and output layer. The input size set as the 100×100, stem convolution layer is the normal convolutional layer with filter size (3×3) stride one and number filter is 108 followed by ReLU and batch normalisation. Each block consists of normal cells optimized by Bayesian and one fixed reduction cell. The output of the final layer (final normal cell) is followed by Global Average Pooling (GAP) for reducing the feature map dimensionality, the

last layer called fully-connected layer which consists of two nodes (number of classes), and finally SoftMax used for classification purposes. Since the Reduction cells are used as a pooling layer in an ENAS-based CNN architecture, which reduces the feature map size to half, then for controlling the output size, the backbone architecture includes two Reduction cells. Consequently, based on the number of Reduction cells and for avoiding input vanishing, we only search for number of Normal cells and divided the search area into 3 blocks and Bayesian Optimizer searches for the optimal number of Normal cells in each block. The search range of block is defined as [Min=1, Max= 5 and step=1) as shown in Figure 6.3. Therefore, the step is the smallest distance between to values that select by Bayesian optimiser during the searching stage for determining optimal number of layers per block. Given this setting, the maximum depth of the architecture can be 15 normal cells and 2 reduction cells. The shallowest architecture can be 3 normal cells and 2 reduction cells. In other words, Bayesian optimisation searches for CNN layers between 5 to 17 layers in total.

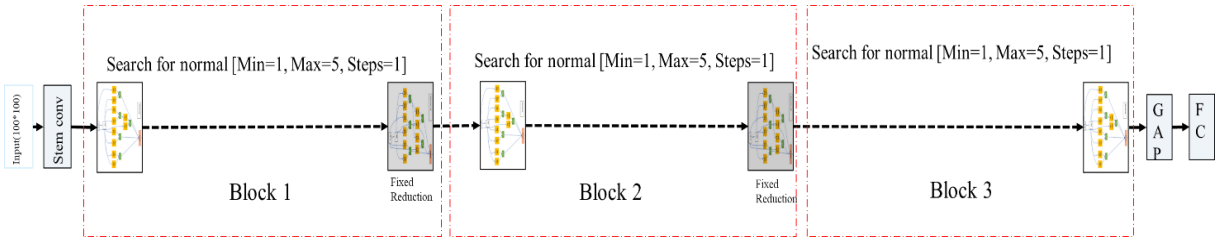


Figure 6-3: Backbone Architecture for Bayesian Optimization search.

- *Trainable hyper-parameters Search Space*: The Structural hyper-parameters Search Space section concerned with the depth of the architecture (5 to 17 layers). On the other hand, this section defines suitable model training hyper-parameters sets for Bayesian optimizer to build optimal architecture (ENAS-B) for breast lesion classification in ultrasound images. In particular, a set of trainable hyper-parameters (Table 6.1) have been defined and used as an input for Bayesian optimizer. Several criteria have been used to select the suitable searchable hyper-parameters with their ranges including: the knowledge we obtained from the investigation in Chapters 4 and 5, literature [107, 108, 109], , and the hyper-parameters used in original ENAS method. As illustrated in Table 6.1, our search spaces consist of Learning rate: [min=0.0001, Max=0.01]; Optimization: [Adam, SGD, RMSprop]; Loss function: [Sparse categorical cross-entropy SCCE, Binary cross-entropy BCE] ; Weight Initialization: [He_normal, Glorat_normal]; Dropout Rate: [min=0%, Max=90%]; Layer Normalization: [Batch Normalization, Group normalization(4)]; and L2 regularize: [min=0.00001, max= 0.001].

Table 6-1: List of Trainable Hyper-parameters used as search space

Hyper-parameters	Search Space	Rational of Selection
Optimization	[Adam, SGD, RMSprop]	Optimization function is one of the most important hyper-parameters thus, suitable optimizer provides better training process of CNN [55]. SGD is used in default ENAS and Adam and RMSprop are commonly used for designing CNN for medical image analysis such as [78] and [77].
Learning rate	[Min=0.0001, Max=0.01]	Learning rate controls the amount of weight changing based on the estimated error during the backpropagation process. Large learning rate results in unstable training, and small rates cause training failure. Therefore, searching for an optimal learning rate is necessary for better training. The default learning rate of (Adam, SGD, RMSprop) used a source for defining searching rage.
Loss function	[Sparse categorical cross-entropy SCCE, Binary cross-entropy BCE]	Loss function is an important element of back propagation process, then searching for optimal loss function is required. Thus, SCCE used in default ENAS method and BCE selected as a searchable candidate based on number of classes in our dataset which is two classes.
Weight Initialization	[He normal, Glorat normal]	Defining suitable weight initialization method for CNN architecture can generate a valuable filters and avoids gradient vanishing issue [24]. He initializer is used by ENAS and our investigations in Section 4.4.3 explored that Glorat Normal initializer is useful for designing CNN architecture for breast cancer classification from ultrasound images.
Dropout Rate	[Min=0%, Max=90%]	Dropout used for reducing overfitting, and in Chapter 4 Section 4.5 experimentally investigated that different CNN architecture requires different dropout rate. Since the proposed method searches for CNN depth and trainable hyper-parameters, thus in the search space defined from minimum dropout rate 0% to highest possible value 90%.
Layer Normalization	[Batch Normalization, Group Normalization(4)]	In default ENAS method each convolutional layer followed by Batch Normalization. In addition, Group Normalization added to the search space because it performs well with small batch size, and in this method small batch size used for training the CNN models.
L2 regularize	[Min=0.00001, Max= 0.001]	L2 regularization used for reducing model overfitting, and each specific CNN model requires specific L2 rate. For defining the searching rage, the ENAS default rate (0.00001) used as Min value and (0.001) used as Max values.

Despite there are many trainable hyper-parameters, we selected the most effective parameters for Bayesian optimisation search space. As described in 6.1, the search space of all hyper-

parameters was designed carefully. For example, the learning rate is directly related to the optimisation function; thus, the learning rate of Adam used as the min value and the default learning rate of SGD used as the max value. Furthermore, default ENAS used SCCE loss function, and we added the Binary cross-entropy function to the searching list, because our dataset includes two classes. Furter, the rationale of adding the Group Normalisation operation for the search space is due to its effectiveness dealing with small mini-batch size (this study uses a batch size of 8).

Search Strategy

The search strategy method is used to explore the search space and construct candidate convolutional neural network architectures according to a set of given constraints. Given the aforementioned normal/reduction cells, backbone architecture, Structural and Trainable hyper-parameters search spaces, the Bayesian optimisation algorithm is used to find the optimal architecture for breast lesion classification. As discussed in Section 2.5, Bayesian optimisation has shown capability to successfully solve computationally expensive functions in order to find the extrema [110]. The method can be used to solve functions without closed-form expressions, as well as for calculating expensive functions[111]. When there is difficulty in evaluating the derivatives, or the function is non-convex, the optimisation aims to determine the sampling point's maximum value for an unknown function which is called the objective function [55].

In general, Bayesian optimization for hyper-parameter tuning is working as follows: due to the high cost of calculating the objective function *f* for configuration **x**, it approximates *f* using a probabilistic surrogate model M: p(*f* |D) that is significantly more affordable to assess. Bayesian optimization iterates the following three phases in the number of iterations:

First, maximizing the acquisition function $xn = arg\,maxx \in X\, a(x;\, M)$, by using the surrogate model *M* to select a configuration.

Second, evaluate the configuration $\boldsymbol{xn}$ to get its performance yn;

Third, combine this measurement (x_n, y_n) with the observed measurements D = (x_1, y_1),..., (x_{n1}, y_{n1}), and re-fit the surrogate M to the enhanced D.

In this study, Gaussian Process (GP) has been used as a surrogate model, the objective function is maximising validation accuracy, and the number of initial points of the search is 3.

In this method of designing CNN architecture Bayesian optimiser for will tune 10 hyper-parameters (3 values for determining number of Normal layers, 7 trainable hyper-parameters) for designing CNN architecture for breast cancer classification from ultrasound images. In the first stage surrogate model (GP) randomly initialise three points for each of the searchable

hyper-parameter for determining the optimal combination between hyper-parameters. Then in the second stage the generated CNN architecture train on the provided dataset and the validation accuracy use for updating the surrogate model for determining next selection of hyper-parameters, in order to maximize the objective function. This process will continue based on the initialised number of iterations. The algorithm pseudo code is presented below:

Algorithm 1 Searching Strategy for ENAS-B

Input: [Search space S,
Objective Function $f(s)$,
Max No of Iteration E_{max},
Initial seeds (points) p.]
Output: [Optimal Setting of hyper-parameter s'
Validation accuracy of Generated Model m']
Select: initial setting of hyper-parameter $s_0 \in s$ for p No. of points
Evaluate: The initial Validation accuracy $m_0 = f(S_0)$
Set $S' = S_0$ and $m' = m_0$
for E =1 to E_{max} do
Select: a new set of hyper-parameters $S_n \in S$ by maximising
acquisition function $D(S_n)$
S_n = argmax(D(S_n))
Evaluate: f for S_n to obtain the Validation accuracy
Mn = $f(S_n)$ for setting of selected hyper-parameters S_n
Update: the surrogate model
if $m_n < m'$ then $S' = S_n$ and $m' = m_n$
end if
end for
return S' and m'

6.1.3. ENAS-B

In the previous Sections (6.1.1 and 6.1.2), all components of the automatic design of CNN architectures using Bayesian optimisation have been described, including search strategy and search space. Thus, this section describes the two proposed methods (ENAS-B-I and ENAS-B-II) for optimising final DCNN model for breast tumour classification from ultrasound images. In order to reducing the search space complexity, some of the hyper-parameters were fixed for both ENAS-B-I and ENAS-B-II, such as the number of filters (36), filter size (3×3 and 5×5) in operations of the cells, batch size (8), the number of epochs (50), and the number

of Reduction cells. Figures 6.4 and 6.6 present the optimal architectures designed by ENAS-B-I and ENAS-B-II, respectively.

Method-I (ENAS-B-I):

ENAS-B-I method combines both structural and trainable hyper-parameter into one set for Bayesian optimisation. In particular, 10 hyper-parameters (3 values for structural and 7 trainable hyper-parameters) are used by Bayesian optimiser to find the optimal CNN architecture depth and training hyper-parameters for breast cancer classification from ultrasound images. In other word, the optimiser searches for optimal number of normal layers in each block as well as the trainable hyper-parameters (Table 6.1). Moreover, in this searching phase of this method, the Bayesian optimiser generates 40 CNN architectures and the optimal one will be trained from scratch on the provided dataset (see Section 6.2). The rational of combining the two search spaces and performing the search in one task is to consider the effect of one search space into the other.

Method-II (ENAS-B-II):

The one-stage search strategy approach of the ENAS-B (or ENAS-B-I) raises a further question as to whether the optimal number of layers and trainable hyper parameters should be searched separately. ENAS-B-II method performs the search of optimal CNN network in two stages. The first stage searches for the structural hyper-parameters (or the number of normal cells in each block) to find an intermediate optimal architecture. Then, the architecture from the first stage is used with the trainable hyper-parameters (Table 6.1) to find the final optimal network (or ENAS-B-II). Similar to or ENAS-B-I, Bayesian optimization based on Gaussian process is used for maximising the objective function by selecting the optimal hyper-parameters from provided search space as described in 6.1.2.

Generally, this method consists of two search stages, *first stage* the optimiser (see Section 6.1.2 (search strategy)) searching for optimal number of Normal cells per block. Moreover, the searchable hyper-parameters is only 3 in this stage and the Bayesian optimisation generates three values for determine optimal depth of the CNN architecture. Therefore, in this stage to reduce the complexity of the searching stage, we fixed some of the hyper-parameters including 1) the number of filters (36) and filter size (3×3 and 5×5) in operations of the cells; 2) batch size (8) ; 3) the number of epochs (50); 4) the number of Reduction cells; 5) and all other the trainable hyper-parameters are defined as original ENAS.

Therefore, after optimising the CNN architecture depth and structure using Bayesian optimization, in the *second stage*, we propose another searching environment to find optimal trainable hyper-parameters for ENAS-B. For that purpose, we select a set of trainable hyper-parameters as shown in Table 6.1(see Section 6.1.2), to be used by Bayesian optimizer for the architecture that automatically generated in the first stage. The final CNN model named as ENAS-B-II.

6.2. Experiments and Results

This section reports the evaluation results of the optimal CNN model from ENAS-B-I and ENAS-B-II. Five different datasets of breast lesion US images collected from different sources are used to search for the optimal architecture and building CNN models. The Modelling dataset used in Section 4.1.1 (1,102 cases (726 Benign and 376 Malignant)) was expanded by including a new set of 420 images (278 benign and 142 malignant) collected from 6th Hospital in Shanghai by TenD AI Medical Technologies to perform the search and modelling of ENAS, ENAS-B-I and ENAS-B-II networks. In particular, the combined dataset consists of 1522 images (1004 benign and 518 malignant) in total, which is used to search for the optimal cells using ENAS, and then optimal number of layers and trainable hyper-parameters using Bayesian Optimisation, and finally to train the ENAS-B-I and ENAS-B-II model from scratch. Similar to the evaluation protocols in Chapters 4 and 5, 5-fold cross-validation was used. One split of the 5-folds was used to perform the architectures search and all 5 folds for modelling the networks. The same imbalance ratio of benign to malignant (1.92:1) as described in Section 4.3.1 was used in the modelling and searching stages. For expanding the training set we used all data augmentation methods mentioned in Section 4.1.3.

The external (unseen) datasets used in Section 4.1.1 (External_A (310 Benign and 210 Malignant) and External_B (300 Benign and 200 Malignant) was enlarged by including a new set of collected from Renji Hospital in Shanghai and consists of 168 images (72 Benign and 96 Malignant), provided by TenD AI Medical Technologies. This dataset is named as the External_C. Other internal and external test datasets are as described in Chapter 4.

Optimising ENAS-B-I Architecture: ENAS-B-I method (Section 6.1.3) that combines both structural and trainable hyper-parameter with the new modelling dataset were used to evaluate our first method of breast lesion classification in ultrasound images. As was previously mentioned, Bayesian optimisation is capable of finding the optimal set of hyper-parameters for CNN architecture through a limited number of iterations. In addition, the computation power

and time constraints prevented us from producing a large number of samples by Bayesian optimisation for breast cancer classification from ultrasound images. Therefore, in order to find the best CNN design, the Bayesian optimiser was instructed to sample 40 CNN architectures. in other word, in this experiment, Bayesian optimisation generated 40 architectures during the search process. During the searching stage, each generated model (a model with *Structural and Trainable* hyper-parameters) was trained on the mentioned dataset for 50 epochs and the validation accuracy used for updating the optimiser with input image size is 100×100. As a result, the architecture with the highest validation accuracy was selected as the optimal architecture. As a result, the optimal observed architecture consists of 12 cells (10 normals and 2 reductions) with the sequence of layers that showed in Figure 6.4 with the following trainable hyper-parameters:

Normalization Layer = Batch Normalization; L2_Regularization =0.00042; Learning Rate =0.0001; Weight initialization = He-Normal; Loss function = BCE; Optimization function = Adam; and Dropout rate = 0. Given the optimal architecture in Figure 6.4, the ENAS-B-I architecture was evaluated on the modelling dataset. Figure 6.5 shows the results of the 5-fold cross-validation and the performance on the external datasets. The overall accuracy achieved by the ENAS-B-I model on internal tests is 76.6%, and the average accuracy of external test sets is 80%. The ENAS-B-I model.

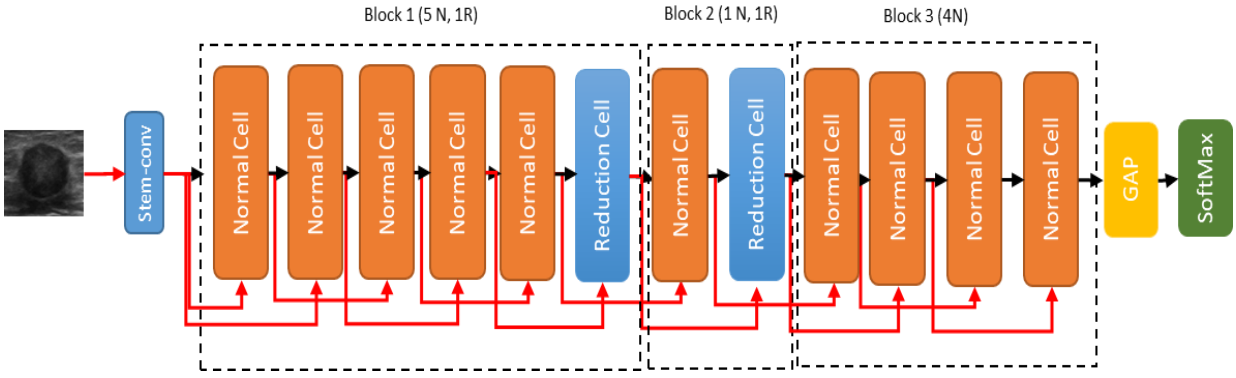


Figure 6-4: The Optimal CNN architecture Designed by ENAS-B-I Method

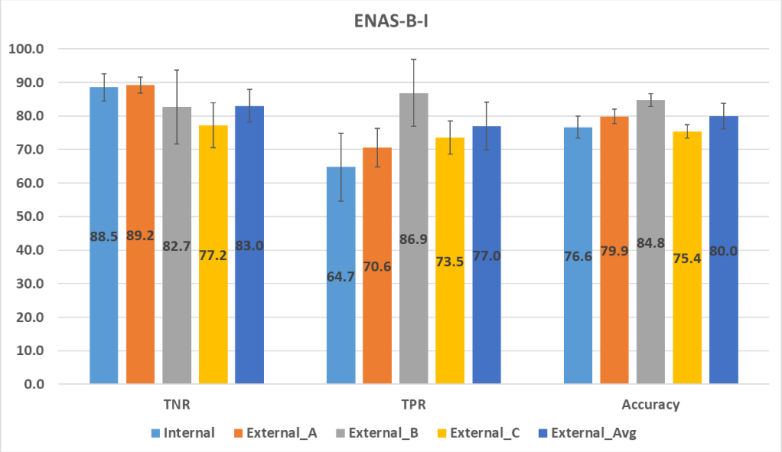


Figure 6-5: Performance of ENAS-B-I Model of average of 5 folds on Internal and External datasets

In Table 6.2, the details result of the 5-fold ENAS-B-I model is presented. The result showed that Model5 is the best model among the five models of ENAS-B-I in terms of accuracy on internal and external test sets and the balance between TNR and TPR.

Table 6-2:shows the detail of 5-fold-cross-validation of ENAS-B-I models

Models	Datasets	TNR	TPR	Acc
Model1	Internal	89.1	56.7	72.9
	External_A	88.5	68.6	78.5
	External_B	87.0	88.0	87.5
	External_C	79.2	76.0	77.6
	External_Avg	84.9	77.5	81.2
Model2	Internal	88.1	70.2	79.1
	External_A	90.4	73.8	82.1
	External_B	67.0	97.5	82.3
	External_C	76.4	68.8	72.6
	External_Avg	77.9	80.0	79.0
Model3	Internal	94.0	53.8	73.9
	External_A	91.5	61.0	76.3
	External_B	92.0	80.5	86.3
	External_C	76.4	74.0	75.2
	External_Avg	86.6	71.8	79.2
Model4	Internal	90.0	61.2	75.6
	External_A	90.7	71.4	81.1
	External_B	95.0	71.5	83.3
	External_C	87.5	67.7	77.6
	External_Avg	91.1	70.2	80.6
Model5	Internal	**81.5**	**81.7**	**81.6**
	External_A	84.8	78.1	81.4
	External_B	72.3	97.0	84.7
	External_C	66.7	81.3	74.0
	External_Avg	74.6	85.4	80.0

Optimising ENAS-B-II Architecture: ENAS-B-II method consists of two stages. In the first stage, the Bayesian optimisation algorithm searched for 30 networks. Since this method consists of two searching stages, and the number of searching hyper-parameters smaller than ENAS-b-I's search space, thus number of iterations is reduced to 30 in each searching stage in method II. Where each generated network was trained from scratch on the Modelling dataset with 50 epochs during the search. The criteria for selecting the optimal CNN architecture among the generated child architectures include the validation accuracy and the architecture complexity in terms of the number of weight parameters within the model. As a result, the optimal observed architecture consists of three normal cells and two reduction cells (Normal, Reduction, Normal, Reduction, Normal), as shown in Figure 6.4. The resulting CNN model is named ENAS-B-II.

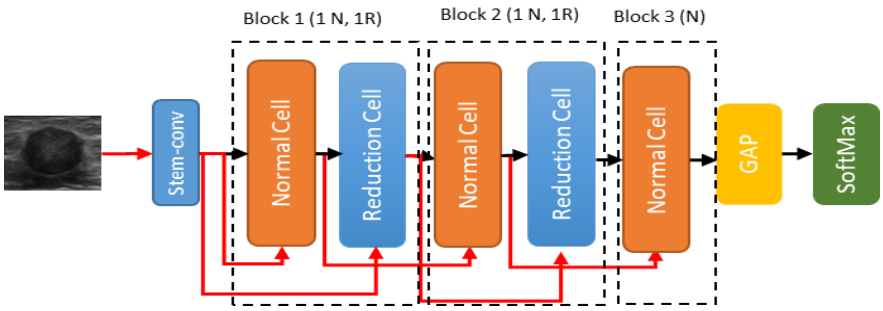


Figure 6-6: ENAS-B-II architecture based on ENAS cells and Bayesian depth search.

Then, the second stage searches for trainable hyper-parameters of the architecture in Figure 6.6 with the following setting: input image size is 100×100, the number of trails (sample model) is 30, and each model is trained on the unbalanced dataset for 50 epochs with batch size 8, and the objective function used the validation accuracy to identify the optimal model. Bayesian optimisation of this stage resulted in two optimal with the same validation accuracy and different hyper-parameter values. For selecting optimal model, we trained both optimal CNN models and evaluated on internal and external test sets. Then, the model with highest accuracy and lowest generalisation error selected as our optimal CNN model for breast cancer classification. The best observed model has the following hyper-parameters: Normalization Layer = Group Normalization; L2_Regularization =0.00036; Learning Rate =0.0001; Weight initialization = He-Normal; Loss function = SCCE; Optimization function = SGD; and Dropout rate = 0.3. The optimal architecture depth in Figure 6.6 and the aforementioned hyper-parameters represent the optimized network ENAS-B-II.

Figure 6.7 shows the performance of ENAS-B-II using the internal and the three external datasets. ENAS-B-II model achieved an average accuracy of 82.7% with TNR 82.9% and TPR=82.5%, when tested on the external test set (1233 breast ultrasound images with 727 Benign 506 Malignant). Table 6.3 shows the details of individual folds.

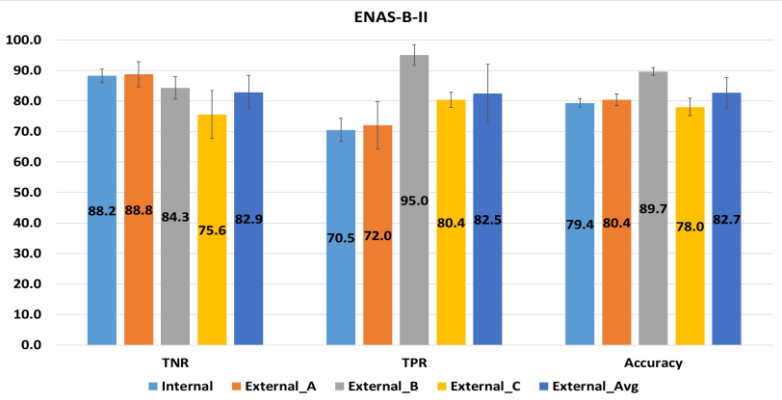


Figure 6-7: Performance of ENAS-B-II Model On internal and External datasets

Table 6-3:shows the detail of 5-fold-cross-validation of ENAS-B-II models

Models	Datasets	TNR	TPR	Acc
Model1	Internal	90.0	68.3	79.2
	External_A	90.7	66.7	78.7
	External_B	84.0	97.0	90.5
	External_C	68.1	83.3	75.7
	External_Avg	80.9	82.3	81.6
Model2	Internal	89.6	65.4	77.5
	External_A	82.8	83.3	83.1
	External_B	86.3	97.0	91.7
	External_C	73.6	79.2	76.4
	External_Avg	80.9	86.5	83.7
Model3	Internal	85.5	71.2	78.3
	External_A	85.4	77.6	81.5
	External_B	78.0	98.0	88.0
	External_C	66.7	83.3	75.0
	External_Avg	76.7	86.3	81.5
Model4	Internal	90.5	70.9	80.7
	External_A	94.1	61.4	77.8
	External_B	89.3	89.0	89.2
	External_C	86.1	77.1	81.6
	External_Avg	89.8	75.8	82.8
Model5	Internal	85.5	76.9	**81.2**
	External_A	90.8	71.0	80.9
	External_B	84.0	94.0	89.0
	External_C	83.3	79.2	81.3
	External_Avg	86.0	81.4	**83.7**

The results in Table 6.3 shows that the ENAS-B-II model generalise well on unseen dataset with balance between TNR and TPR. In addition, the standard division between models' performance is too small, which shows the stability of the ENAS-B-II model. Furthermore, Model5 outperforms other models by achieving 81.2% on internal test and 83.7% on External average.

All trainable hyper parameters of ENAS-B-I, with the exception of weight initialization and learning rate, are different from ENAS-B-I besides the number of layers. Note that the probability of the search space is increased in ENAS-B-I method, because the optimiser will tuning trainable hyper-parameters beside optimising the CNN depth, where we generate 40 child architecture instead of 30 in ENAS-B-II method.

ENAS-B-I and ENAS-B-II produced two different networks with different depths and training hyper-parameters. This observation is very interesting given both approaches used the same datasets for finding the optimal parameters. Table 6.4 shows a comparison between ENAS-B-I and ENAS-B-II.

Table 6-4: Comparison between CNN generated by ENAS-B-I and ENAS-B-II method

Models	Test sets	TNR	TPR	Acc	#Parameters
ENAS-B-I	Internal	88.5 ±4	64.7 ±10	76.6 ±3	2,651222
	External_A	89.2 ±2	70.6 ±6	79.9 ±2	
	External_B	82.7 ±11	86.9 ±10	84.8 ±2	
	External_C	77.2 ±7	73.5 ±5	75.4 ±2	
	External_Avg	83.0 ±5	77.0 ±7	80.0 ±4	
ENAS-B-II	Internal	88.2 ±2	70.5 ±4	**79.4 ±1**	1,053398
	External_A	88.8 ±4	72.0 ±8	**80.4 ±2**	
	External_B	84.3 ±4	95.0 ±3	**89.7 ±1**	
	External_C	75.6 ±8	80.4 ±3	**78.0 ±3**	
	External_Avg	82.9 ±5	82.5 ±10	**82.7 ±5**	

The results demonstrate that ENAS-B-II outperformed the ENAS-B-I CNN model consistently across all test datasets in terms of average overall accuracy. Especially, the average TPR of external testing datasets of ENAS-B-II outperforms the ENAS-B-I CNN model by 5.5% while obtaining almost the same result in terms of average TNR of external testing images. In addition, the average TNR and TPR of external test images are balanced with 82.9% and 82.5 respectively. Finally, the performance of ENAS-B-II on internal dataset is higher than the

ENAS-B-I in terms of average TPR by nearly 6%. Another interesting finding is that the resulting ENAS-B-II architecture is less than half of the size of ENAS-B-I, a simpler and more robust architecture.

Based on the experimental evidence, The ENAS-B-II method outperformed ENAS-B-I as the search for optimal network was performed in two stages with 30 configurations each. In addition, due to the limited hardware resources, 40 search iteration with the combined hyper-parameters might not be sufficient. Moreover, the searching time of ENAS-B-I was 19 days, while both stages of ENAS-B-II took 25 days.

6.3. Comparison with Existing CNN models

In this section, we compare the proposed ENAS-B models with the existing state-of-art models that are designed for breast cancer classification using US images. Similar to Chapter 4, three of the existing architectures (CNN3 [80], CNN4 [78], and Fus2Net [79]) were selected for comparison. Each of these CNNs was trained on the new modelling dataset with the same training protocol that was used for training the ENAS-B models. Furthermore, the ENAS17_V1 architecture that is automatically designed for breast cancer classification in Section (4.2), is also compared with ENAS-B models. For a fair comparison, we trained ENAS17_V1 on the expanded dataset (new modelling) that was used for training the ENAS-B models. Table 6.5 shows the comparison results. For further comparison the ENAS17_V2 model designed by manually stack Optimal ENAS cells obtained in the first stage of ENAS-B and trained from scratch on the dataset mentioned in Section 6.2. Note that the cell structure of ENAS17_V1 is different from ENAS17_V2 (see Figure 4.4 and Figure 6.2). In Table 6.5, we report the performance of ENAS-B models, ENAS17_V1, and ENAS17_V2 together with CNN3, CNN4 and Fus2Net.

Firstly, the ENAS-B-II model outperforms all state-of-the-art models and ENAS17_V1 and ENAS17_V2 in the average TPR and TNR on the three external datasets (Table 6.5). The ENAS-B-II model outperforms the CNN3, CNN4 and Fus2Net, by a large margin, on internal test set by 4.5%, 18.8% and 13.3% in overall accuracy, respectively. Manually designed CNN architectures do not generalise well on our datasets, especially CNN4 and Fus2Net, whereas the ENAS-B-II model generalises better than CNN3, CNN4 and Fus2Net; it outperforms the other three networks by 6.4%, 20.5% and 17.7% in an average accuracy of external test sets respectively. In addition, the ENAS-B-II network has more balanced TPR and TNR, while

CNN4 and Fus2Net have large gaps between these two-performance metrics when tested on the three external datasets.

Table 6-5: Comparison Result of state-of-the-art CNNs and ENAS-B to classify US images of breast tumors.

Models	External test	TNR	TPR	Accuracy	#Parameters
CNN3	Internal	88.7 ±6	61.1 ±13	74.9 ±5	619,202
	External_A	78.8 ±4	68.1 ±12	73.5 ±4	
	External_B	78.9 ±11	86.4 ±7	82.6 ±3	
	External_C	73.6 ±10	71.0 ±12	72.3 ±1	
	External_Avg	77.1 ±7	75.2 ±10	76.1 ±8	
CNN4	Internal	91.3 ±13	29.9 ±34	60.6 ±11	628,418
	External_A	88.2 ±11	39.8 ±33	64.0 ±11	
	External_B	89.3 ±18	39.4 ±34	64.4 ±9	
	External_C	79.7 ±29	36.9 ±31	58.3 ±6	
	External_Avg	85.7 ±8	38.7 ±2	62.2 ±5	
Fus2Net	Internal	83.0 ±15	49.2 ±38	66.1 ±12	889,714
	External_A	63.1 ±34	56.2 ±40	59.7 ±10	
	External_B	84.9 ±14	64.3 ±42	74.6 ±15	
	External_C	68.9 ±27	52.7 ±41	60.8 ±9	
	External_Avg	72.3 ±11	57.7 ±4	65.0 ±7	
ENAS17_V1	Internal	89.1 ±2	62.8 ±8	76.0 ±4	4,251,780
	External_A	89.3 ±4	61.5 ±11	75.4 ±4	
	External_B	87.7 ±3	86.1 ±8	86.9 ±3	
	External_C	74.7 ±8	73.5 ±8	74.1 ±1	
	External_Avg	83.9 ±18	73.7 ±9	78.8 ±9	
ENAS17_V2 (Unbalance search)	Internal	86.4 ±1.1	81.1 ±3.4	83.8 ±1.4	3,927,636
	External_A	90.0 ±2.8	63.0 ±3.5	76.5 ±0.5	
	External_B	84.9 ±4.3	89.3 ±3.4	87.1 ±1.4	
	External_C	74.4 ±4	73.8 ±4.7	74.1 ±1.7	
	External_Avg	83.1±14.5	75.4 ±8.4	79.2 ±8.2	
ENAS-B-I	Internal	88.5 ±4	64.7 ±10	76.6 ±3	2,651,222
	External_A	89.2 ±2	70.6 ±6	79.9 ±2	
	External_B	82.7 ±11	86.9 ±10	84.8 ±2	
	External_C	77.2 ±7	73.5 ±5	75.4 ±2	
	External_Avg	83.0 ±5	77.0 ±7	80.0 ±4	
ENAS-B-II	Internal	**88.2** ±2	**70.5** ±4	**79.4** ±1	1,053,398
	External_A	88.8 ±4	72.0 ±8	80.4 ±2	
	External_B	84.3 ±4	95.0 ±3	89.7 ±1	
	External_C	75.6 ±8	80.4 ±3	78.0 ±3	
	External_Avg	**82.9** ±5	**82.5** ±10	**82.7** ±5	

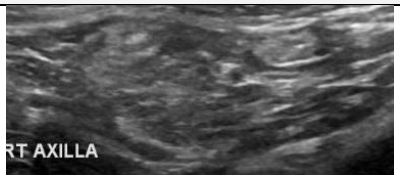
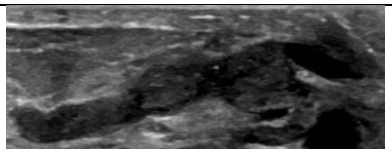
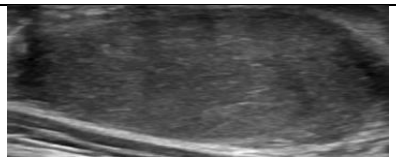
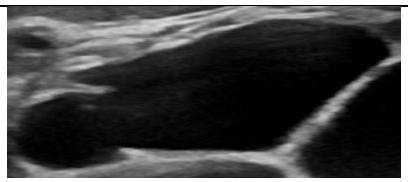
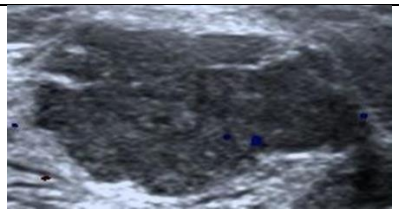
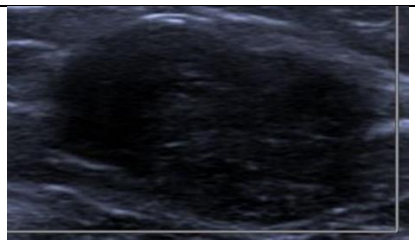
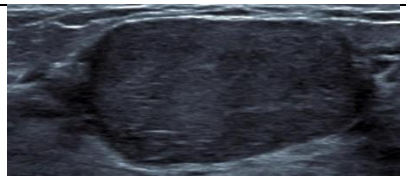
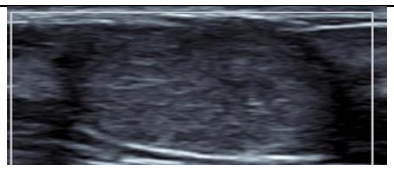
The main difference between CNN4 and ENAS-B is that CNN4 is a chain-based model and there is no skip connection in the architecture together with large filter sizes of 11×11 and 7×7.

The Fus2Net architecture, on the other hand, lacks skip connection between the block models, which may result in input vanishing and hence not generalise well on external datasets. CNN3 is performing better than CNN4 and Fus2Net mainly because the input size of this model is close to ENAS-B-II input size, which is 150×150. Also, CNN3 uses GAP instead of flattening layer with reduces the number of parameters which avoids over-parameterisation. It also uses small filer sizes of 3×3 which is close to the filter sizes used in ENAS-B-II. It is worth mentioning that the number of weight parameters for CNN3, CNN4 and Fus2Net is smaller than that of the ENAS-B-II model; the number of weight parameters is about 200K to 400K fewer. However, this gain in architecture complexity cannot justify the lower levels of accuracy.

On the other hand, the number of weight parameters in ENAS-B-II is smaller than ENAS17_V1 and V2 with higher TPR and TNR on the three averaged performance external datasets. The high performance of ENAS-B-II demonstrates the effectiveness of our approach to optimising the number of layers and trainable hyper-parameters for accurate and robust network for breast lesion classification in ultrasound images.

Moreover, Table 6.6 demonstrates a group of Benign cases from External_A and External_B datasets, all of which have been misclassified by the ENAS17_V2 model. In contrast, only half were misclassified by the ENAS-B-II model. Furthermore, some of the cases have text artefacts and lines which may contribute negatively to the classification decision, such as row one in External_A and row one, two and four of External_B. However, the case in row three of External_B includes a line and is misclassified by ENAS17_V2 but correctly classified by ENAS-B-II. Moreover, other samples, like row three and four of External_A and External_B, have regular shapes, but still, ENAS17-V2 failed to classify them correctly, while ENAS-B-II correctly classified them. The examples presented in Table 6.6 is a representative to other cases where ENAS-B-II performs better than ENAS17-V2. However, there is no apparent justification for correctly classifying the cases by ENAS-B-II models, which ENAS17_V2 misclassified. This showcases an example of the general limitations of deep learning models, which is called the interpretability issue. Therefore, this will be the next step of our research journey in the future.

Table 6-6: Sample of Misclassified Benign cases of Breast lesion by ENAS17_V2 and ENAS-B-II

Test Sets	Benign Cases	Models Prediction	
		ENAS17_V2	**ENAS-B-II**
External_A		Misclassified	Misclassified
		Misclassified	Misclassified
		Misclassified	**Correctly Classified**
		Misclassified	**Correctly Classified**
External_B		Misclassified	Misclassified
		Misclassified	Misclassified
		Misclassified	**Correctly Classified**
		Misclassified	**Correctly Classified**

6.4. Discussions

This chapter presented and evaluated our methods ENAS-B-I and ENAS-B-II for accurate and robust breast lesion classification in ultrasound images. This section discusses three issues. First, Bayesian Optimiser may produce multiple optima to solve the same task. In other words, many networks with the same lowest validation error may be produced during the Bayesian Optimiser search. This section further discusses the similarity and differences of these equally optimal architectures. The second issue is the potential impact of the stage two of the search of ENAS-II on the overall performance. Finally, the third issue to discuss is whether the ENAS-B can be used for other types of lesion classification from ultrasound images, such as thyroid nodules.

Multiple Optimal Solutions:

To study and understand the issue, two optimal networks with same (or very similar) validation error rate were selected and their architectures compared. Table 6.7 shows the details of two optimal CNN architectures optimised by Bayesian in both ENAS-B-I and ENAS-B-II. The two optimal architectures designed by ENAS-B-II search strategy achieved the same validation accuracy and only three hyper-parameters are different from each other which are L2-Regularization rate, Optimisation Function, and dropout rate. The difference between the L2-Regularization rate and the dropout rate is small, while the optimisation function is different. This shows different optimisation functions (SGD and RMSProp) performs similarly and Bayesian approximation is very accurate. On the other hand, ENAS-B-I search strategy produced two different optimal architectures (optimal 1 and optimal 2) as described in Table 6.7. The two architectures have different structure/depth with significant variations in the dropout rate and L2-Regularization rate. Given ENAS-B-I approach perform the search on the combined search spaces (structure and training hyper-parameters), the chance of producing multiple optima with high variation is highly likely.

Table 6-7:The Details of the Two Optimal CNN models Optimized By each of the Proposed Bayesian Optimization based search Strategy

Hyper-parameters	**ENAS-B-I Searching Strategy**		**ENAS-B-II Searching Strategy**	
	Optimal 1	**Optimal 2**	**Optimal 1**	**Optimal 2**
Normalisation Layer	BN	BN	GN	GN
L2_Regularization rate	**0.00042**	**0.001**	**0.00036**	**0.00015**
Learning Rate	0.0001	0.0001	0.0001	0.0001
Weight Initialisation	He	He	He	He
Loss Function	BCE	BCE	SCCE	SCCE
Optimisation Function	Adam	Adam	**SGD**	**RMSprop**
dropout Rate	**0**	**0.9**	**0.3**	**0.4**
Architecture Structure	**5N, R, 1N, R, 4N**	**1N, R, 5N, R, 1N**	N, R, N, R, N	N, R, N, R, N
Searching Validation	86.46 %	86.13 %	87.78 %	87.78 %

Our interest goes beyond studying the differences between the CNN structures and training hyper-parameters in relation to the performance of each optimal network. Figures 6.8 and 6.9 show classification performance of the two other optimal networks (the second optimal model determined by Bayesian optimiser) from ENAS-B-I and ENAS-B-II tested on both internal and external datasets. The results in Figures 6.8 and 6.9 show that the CNN model designed by the ENAS-B-II method outperforms the model generated by ENAS-B-I. These results align with our findings in Sections 6.2 and 6.3 where ENAS-B-II method outperforms ENAS-B-I. However, ENAS-B-I (Optimal 2) slightly perform better than ENAS-B-II (Optimal 2) in term of TNR on internal test and External_Avg test about 1% and 3%, respectively. In opposite ENAS-B-II (Optimal 2) performed better compared with ENAS-B-I (Optimal 2) in term of TPR and the gap between TNR and TPR in ENAS-B-II (Optimal 2) model smaller than ENAS-B-I (Optimal 2) models.

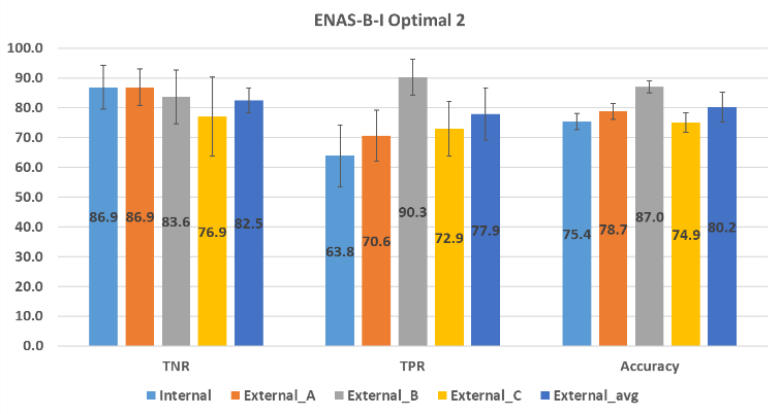


Figure 6-8: Results of Second Optimal CNN architecture Designed by ENAS-B-I Search Strategy

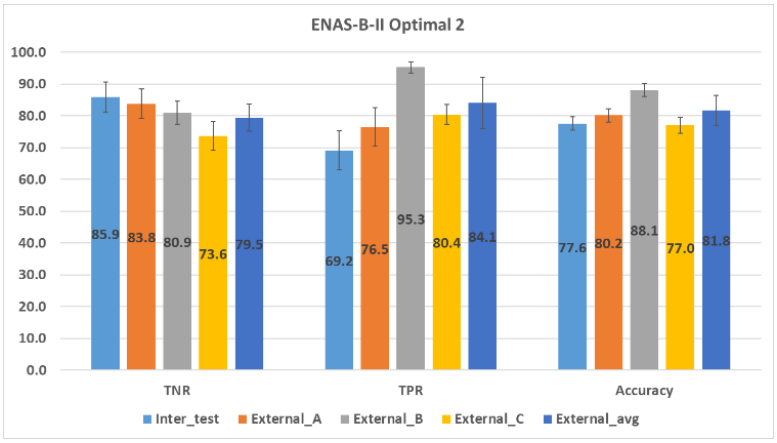


Figure 6-9:Results of Second Optimal CNN architecture Designed by ENAS-B-II Search Strategy.

ENAS-B-II: First Search Stage vs the Second:

The analysis demonstrated that ENAS-B-II has a higher overall classification performance in classifying breast lesions in ultrasound images. This approach has two search stages (structure and training hyper-parameters). This raised another research question ‘what is the impact of the first optimization stage on the overall performance of ENAS-B-II’?. To answer this question and investigate the effect of trainable hyper-parameters on CNN architectures performance, the optimal architecture that generated in the first stage (optimizing number of layers) was trained from scratch with the default ENAS method setting of trainable hyper-parameters. In other words, dismissing the second stage of training hyper-parameters. Figure 6.10 shows the performance of the ENAS-B-II (first stage) model.

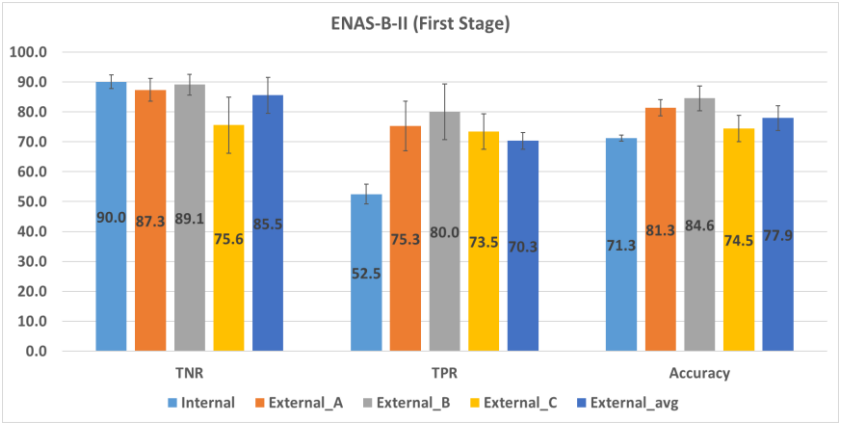


Figure 6-10: Performance of ENAS-B-II (First Stage) Model on internal and External Test sets

The results show that the performance of ENAS-B-II (First stage) dropped on internal test and External average test by approximately 9% and 4.5%, respectively compared to the ENAS-B-II two stages model. Moreover, ENAS-B-II (two stages) model more stable with smallest gap between TNR and TPR. As a result, this experiment shows that

there is direct relation between structure of CNN architecture and the hyper-parameters that related to training process. In other word, although, optimising CNN structure is important process for designing CNN architecture, but tuning trainable hyper-parameters is essential.

ENAS-B for Thyroid Nodule Classification in Ultrasound Images:

Finally, all the experiment results revealed that the model automatically designed by the ENAS-B-II method outperforms the other CNN architecture that manually designed for breast cancer classification from ultrasound images. To investigate the transferability of the ENAS-B-II architecture on other types of cancer, ENAS-B-II was evaluated on thyroid ultrasound images. Thyroid cancer was selected due to the similarity with the breast cancer as described in the literature [67]. Due to the time constraint, a pilot study was performed by searching for an optimal ENAS-B-II architecture using breast lesion US images, and then train an ENAS-B-II model for thyroid nodule classification from ultrasound images. For this experiment two different scenarios used for training ENAS-B-II architecture, in both scenarios all data augmentation methods that mentioned in Section (4.1) used for expanding training set and the same setting and training protocol used for training ENAS-B-II as described in Section (6.2).

In first scenario balance thyroid dataset was used which consists of 500 ultrasound images (250 Benign and 250 Malignant). For expanding training set all the data augmentation methods described in (Section 4.1.3) were used and 5-fold cross validation was used for evaluating the model. Figure 6.11 shows the classification performance of ENAS-B-II for thyroid nodule classification. The result demonstrates that ENAS-B-II architecture that originally designed for breast cancer achieved 73.6% overall accuracy for classifying thyroid nodules. Moreover, the result also shows that the models achieved 54% on TNR with a large standard deviation between the folds and 93.2% TPR with small standard deviation between the folds

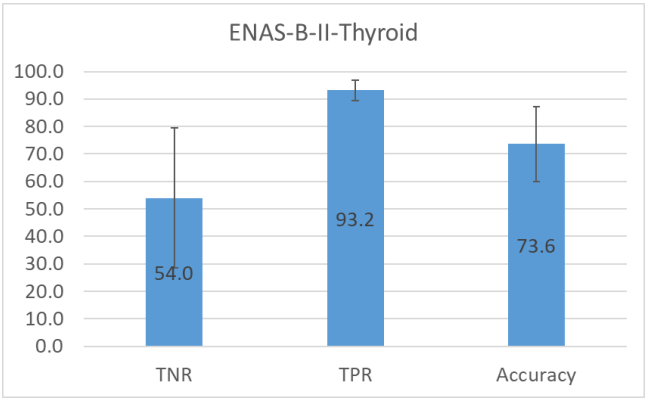


Figure 6-11: Performance of ENAS-B-II that Trained on Balance Thyroid dataset

The investigation in Chapter 4 confirmed the importance of training the model on unbalance classes to ensure the model generalization. Therefore, the ENAS-B-II model was trained again on unbalanced thyroid dataset with ratio (1.92:1) benign to malignant (see Section4.3.1) used for training ENAS-B-II architecture. Also, the 5-fold cross-validation used as training protocol with expanding training set by using mentioned data augmentation methods in Section 4.1.3. Figure 6.12 shows the result.

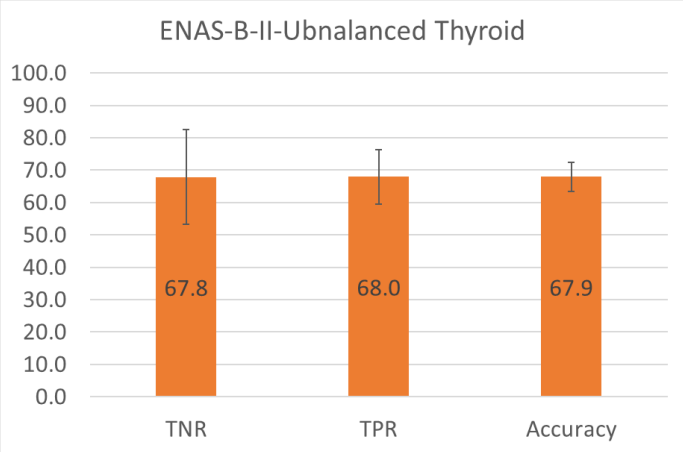


Figure 6-12: Performance of ENAS-B-II trained on unbalanced thyroid dataset

The results reveal that ENAS-B-II architecture with unbalanced dataset produced a model with a balance of TNR and TPR when compared to ENAS-B-II trained on a balanced thyroid dataset. As a result of this experiment, new understanding has been learned from this discussion. First, designing CNN architecture for both thyroid and breast cancer classification from ultrasound images suffers from the same problem which is poor TNR performance. Therefore, this experiment provides more proof of the efficacy of our proposed method (unbalanced dataset) for addressing this issue and reducing gap between TNR and TPR. Second, despite the general applicability of ENAS-B-II on thyroid nodule classification task, performing the search and modelling using ENAS-B-II on thyroid datasets might be needed for building more accurate network. To further investigation in the future, we will used ENAS-B method for searching for optimal CNN architecture for different type of cancer including thyroid cancer.

6.5. Summary

The focus of this chapter was of two-folds: automatic search of the depth of CNN architectures and automatic search of optimal trainable hyper-parameters. We adopted the Bayesian optimisation approach for the purpose of finding the optimal number of CNN layers for the

purpose of breast ultrasound lesion classification. The Bayesian optimiser used the optimal cell structures obtained from ENAS as a searchable parameter to find the best number of layers between 5 and 17 and the list of 7 trainable hyper-parameters. The process of searching is divided into two categories: ENAS-B-I searching strategy and a ENAS-B-II search strategy. In the ENAS-B-I search strategy Bayesian optimisation searches for the optimal number of layers and other trainable hyper-parameters at once such as weight initialisation, loss functions (see Section (6.1.2).

On the other hand, in the ENAS-B-II search strategy, optimal ENAS cell structures are used to search for the best CNN depth while fixing the rest of the hyper-parameters, once the Bayesian process determines the number of layers, another search is conducted via Bayesian optimiser to find the best setting for seven trainable hyper-parameters. Two new breast lesion US image datasets were used, besides three datasets used in previous chapters, to conduct experiments to expand the diversity of modelling and external testing images.

Overall, we obtained an efficient, shallow, and robust CNN model that constituted 5 CNN cells better than ENAS17 and outperformed State-of-the-art CNN models developed for breast US classification. These results provided evidence that the structure of the cell and the dept and trainable hyper-parameters are also important parameters that need to be optimised while designing CNN architectures for the problem in hand.

Another finding of this chapter is that the strategy of conducting the search of the optimal number of layers and trainable hyper-parameters is important. We demonstrated that the two-stage approach (ENAS-B-II) is better and gives a better chance to the Bayesian optimiser to narrow the search and provide a robust CNN model. Furthermore, it is also crucial to decide which hyper-parameters to fix and which ones to search for. This chapter demonstrated that the number of layers and the 7 specified hyper-parameters affect the final CNN model.

Of course, using a grid search where one might consider all possible combinations of searchable hyper-parameters could provide a better CNN model, but the complexity and computation power increases exponentially, which we do not have the facility to conduct at present. Other hyper-parameters that could influence the final optimal CNN layer we have not searched for are the number of filters, filter size and different types of ENAS cells

Chapter 7. Conclusion and Future Work

This thesis presented methods for breast lesion classification in ultrasound images using deep convolutional neural network. This chapter serves as the conclusion for the whole thesis and is set to summarize the research work mentioned in the thesis, highlights the key findings and contributions to knowledge made by the research, and outlines the possible future work following this research. Consequently, this chapter consists of three main parts each of which is designated for one of the purposes.

7.1. Summary of the Thesis

The main goal of this research is to design a robust CNN model for breast cancer classification from ultrasound images. Figure 7.1 outlines the main components of this research work. First of all, it is worth noting that to ensure the research reflects the clinical practice, through our collaborators, we first collected five different ultrasound image datasets of breast lesions from different hospitals and the different machine makes. In addition, we also collected one public domain dataset from a different continent. The datasets were divided into modelling datasets and external test datasets, where the modelling datasets were used for searching CNN architectures and modelling the eventual CNN models whereas external datasets were used to evaluate the performances of the CNN models. Therefore, this research is unique in utilizing various datasets collected from clinical practices with very limited pre-processing and no image enhancements. The findings of the thesis are based on characteristics of real-life datasets rather than laboratory-controlled datasets. The performances of the CNN models tend to be more realistic.

To achieve the main goal of the research, we started by adapting one of the most efficient automatic architecture searches, ENAS, to generate CNN for breast lesion classification. After testing the automatically designed CNN architecture for breast cancer classification on unseen datasets and facing the generalisation error issue, several approaches were proposed. As shown in the lefthand side of Figure 7.1, four different methods were proposed for overcoming the generalisation error, such as (reducing architecture complexity, the effect of data augmentation, using an unbalanced dataset and effect of dropout rate). In addition, several structural modifications were applied to the ENAS method for further investigation, in terms of

modifying the ENAS method search space by expanding the search operation set, adding a highway connection, and adding a fully-connected layer for ENAS based model.

Furthermore, to overcome the limitation of the ENAS framework, we proposed a new approach for automatic designing CNN architecture for breast cancer classification by adopting Bayesian Optimization as a search strategy and the optimal cell generated by ENAS as a search space operation. This approach is the first to use Bayesian Optimization with ENAS for ultrasound image classification. It also provides the foundation for other researchers to adapt this approach for other cancer types and modalities. Besides, the hand-crafted CNN architecture is not neglected in this thesis; a list of the existing CNN architectures manually designed for breast cancer classification from ultrasound images and the State-of-the-art architectures proposed for natural image recognition have been evaluated. Also, we investigated the effect of several hyper-parameters by manually modifying AlexNet architecture.

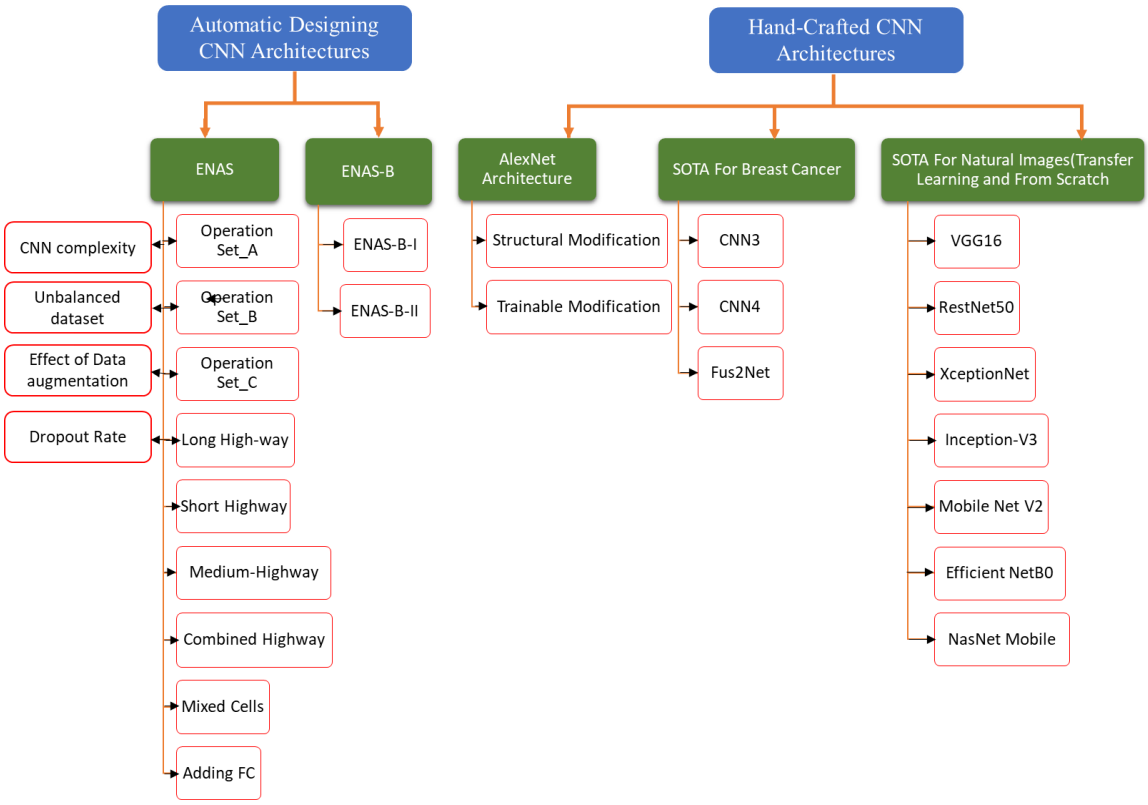


Figure 7-1: Summary of the Main Components of This Research

7.2. Main Achievements of the Thesis

The main contributions of this thesis start from our investigations in Chapter 4, which explores ENAS's effectiveness for generating specific CNN architectures for breast cancer classification using US images. We searched for optimal cells by ENAS, and then designed ENAS17 by

stacking optimal cells. The average test accuracy of ENAS17 reached 89.3%, while the evaluation accuracy of ENAS17 models on external test datasets reduced by ~10% in overall test accuracy and TNR reduced by more than 20%.

We hypothesised that benign lesions have many variations in terms of the feature than the malignant. Therefore, increasing benign images in the training set will improve the ENAS model's capability to recognise benign cases. Thus, after determining the generalisation error of the ENAS17 CNN model for breast tumour recognition, we examined several approaches for reducing the generalisation error of the ENAS-based models, such as reducing architecture complexity, the effect of different types of data augmentation, investigating optimal dropout rate and using unbalanced datasets for training ENAS. Exploring these approaches revealed that training ENAS based on an unbalanced dataset reduces generalisation error better than complexity factor, data augmentation, and dropout rate changes. The test accuracy is similar between internal and external dataset when unbalanced dataset is used for training ENAS and TNR difference between internal and external test accuracies is improved by nearly 10%.

Moreover, through the work reported in Chapter 4, we found that automatically designed ENAS17 and ENAS7 models both outperform a list of selected State-of-the-art CNN architectures such as AlexNet, VGG16, ResNet50, InceptionV3, MobileV2, DenseNet, EffecientB0, NasNet-mobile and XceptionNet. In addition, we investigated the effect of tunning of several hyper-parameters for designing CNN architecture for ultrasound images by manually modifying AlexNet architecture such as filter size, number of layers, weight initialisation method and batch normalisation layer. Comparing to these well-known architectures, ENAS17 and ENAS7 models not only deliver higher levels of accuracy for the given datasets but also have less model complexity, showing the great potentials of the ENAS approach for the intended purpose.

Based on the explorations in Chapter 4, we investigated the effect of different CNN components on designing CNN models for breast cancer classification. First, we modified ENAS search space by adding new operations for the list of default operations defined as search space for ENAS. Several new operations added to the list in three different scenarios (see Section 5.1). The results showed that the CNN architecture designed by modified ENAS with separable convolutions of size 7×7, and normal convolution of size 9×9 outperform the ENAS7 with original operation set by 2% on average accuracy. As a result, by this modification, we found that although the ENAS method was originally designed for natural image classification but can be used for designing CNN architectures for breast cancer classification. The second exploration of Chapter 5 investigates the effect of modifying ENAS17 backbone structure by

adding a new high-way connection. Although the individual model of ENAS17 with highway connection outperformed the best individual model of ENAS17, the average test accuracy showed no significant improvement, especially on the External_B dataset. Therefore, further research is still needed to fully explore adaptations of high-way connections in ENAS architectural designs. Besides, we also found that the modified ENAS17 by adding fully-connected layer after GAP achieved accuracy slightly higher on internal and External_A by 2.6% and 0.3%, respectively, while dropped on External_B by 1% compared to the original ENAS17. Furthermore, we also investigated the effect of designing CNN architectures by using three optimal cells instead of using only one to design the final CNN. In this experiment we selected three optimal cells generated by the ENAS-Set_A method to design ENAS7. The result demonstrated that stacking one optimal cell is better than using three optimal cells for the breast cancer classification problem.

The ENAS adaptations and modifications we have conducted in Chapters 4 and 5 demonstrated that manually modifying ENAS architectures will lead to a trial-and-error scenario and finding the optimal combination between them is a daunting and time-consuming task. The limitation of ENAS at this point is the fact that one cannot search for optimal CNN block structure and the number of layers, i.e. depth, and trainable hyper-parameters all at the same time. To address this issue, we focused on automatic search of stacking and designing the final CNN using ENAS optimal cells via Bayesian optimisation. The key contribution of Chapter 6 is the framework of utilizing automated neural network cell unit search followed by automatic network layer structure search followed by neural network hyperparameter search. The proposed framework is novel, and the proposed solution, to the best of our knowledge, is the first unified automatic CNN search for lesion classification from ultrasound images. It aims to automate the whole architecture and model design, entirely driven by data and controlled by the controller unit. Our test results have shown that the models produced by the proposed three-stage optimization process outperform all handcrafted CNNs and more stable than most handcrafted CNN counterparts.

Based on our extensive research and experimental work for the breast lesion test cases, our adaptation of ENAS is promising for building customized DCNN architectures and models for breast lesion classification. Despite the power of ENAS as an automated method for building CNN cell structures, the original ENAS approach still require manual design of architectures and settings of hyper-parameters. Another novelty of the proposed solution is the use of Bayesian optimization techniques in searching for the optimal depth of layer architecture based on the optimal cell structures as designed by ENAS. This optimization step overcomes the

limitation of the original ENAS and makes the architecture design even more automatic with less human operator interventions.

The third novelty of the proposed solution is our use of Bayesian Optimization techniques to find the optimal hyper-parameters in a pre-defined search space. This part of the search can be considered as a two-stage search: one for layer structure and the other for hyper-parameters in sequence. Such a two-stage approach aims to reduce the amount of time by constraining the size of the search space without sacrificing model performance. This two-stage approach for optimization seems compatible with the ENAS philosophy of two-stage design: cell design and layer design, a process of two quite autonomous steps.

The approach we proposed in this thesis fundamentally differs from the mainstream approaches of adapting existing handcraft architectures originally designed for natural images to ultrasound images of breast lesions. It aims to take CNN designers out of the architecture design loop for better efficiency and efficacy. The proposed solution is consistent with the technology development trends in automatic machine learning.

Every research study has its limitations, including our own. Therefore, the work presented in this thesis can be further extended. ENAS-B requires more computation time comparing to ENAS but achieved higher classification performance. The increase in the computation time comes from the fact that ENAS-B performs the search in two stages (optimising cell structure by ENAS method, searching for optimal CNN depth by using Bayesian optimiser, and tuning the trainable hyper-parameters by Bayesian optimiser.). Section 7.3 next includes the main limitations of the proposed solutions, and future development to improve the work presented in this thesis.

7.3. Future work

The work presented in this thesis opens doors for several important future works that will complement the investigations we conducted to design an optimal CNN architecture for breast lesion classification from ultrasound images:

1. Searching time and computational power is one of the main limitations in automatic CNN design, including the proposed methods in this thesis, especially the automatic design of CNN architecture by using a Bayesian optimiser. Because, in this approach, every generated model is trained from scratch on the prepared dataset. Therefore, we have two ideas for reducing our method's searching time. First, we can adapt the weight sharing approach used by most automatic search methods within the Bayesian optimiser. For that

purpose, we will design a CNN architecture which consists of two cells, one normal cell and one reduction cell (Mother Model). Then before starting the search for the depth, the mother architecture should be trained and saved. During the search, the generated child models will inherit the weights of the mother model. Therefore, this approach will be more efficient than our current and ENAS methods. Second, we can apply early stopping criteria during the training of Bayesian optimisation to reduce the time required to select the final optimal CNN architecture. This approach can be used in two directions, either to stop Bayesian optimiser from generating new architecture or stopping training of the generated model when there is no improvement in the validation accuracy.

2. Expand the searchable hyper-parameters for our proposed search space. Filter size and the number of filters are the most effective hyper-parameters of a CNN architecture. In ENAS and our proposed approach they are fixed. Therefore. optimizing them may provide better CNN architectures for cancer classification using US images.
3. Chapter 5 presented the result of manually designing CNN architecture by using more than one optimal cell for breast cancer classification from US images. We plan to extend the work to include searching environment for optimising the number of optimal cells per block and the order of blocks of optimal cells.
4. In CNN architectures, all input images must be of one specific size and determining the optimal image size is one of the important hyper-parameters for medical images. Especially for US images, the size of ROI is different from patient to patient. Hence, image size as one of the searchable hyper-parameters can be another important future work.
5. ENAS optimises the connection between nodes inside cells, while the connection between layers(cells) is fixed. The skip connection and connectivity between CNN layers is one of the most important hyper-parameters in designing a robust CNN. The majority of the existing CNN defined the connection between layers manually (including our own approaches reported in Chapter 5). Therefore, automatically optimising the connection between layers will be one of the most important hyper-parameters for our automatic CNN design in the future.
6. Many types of cancer such as thyroid cancer, lymphoma, etc. share common ultrasound image characteristics. One of the future works for this research is to expand the proposed framework for automatically CNN architecture design to other types of lesions such as thyroid and lymph nodules. Similarly, the success of the proposed framework for US images of breast lesions may also apply to other medical image modalities such as CTScan,

Mammography and MRI, and hence applying the proposed framework to these image modalities is worth attempting.

7. One potential future work may include applying our methods for automatically optimizing the depth and trainable hyper-parameters of some existing CNNs such as ResNets, GoogleNet and Mobile Net by using their blocks as the search space.

Deep learning is a vast field of research. Automatic search of CNN architectures for building effective cancer diagnosis models is a bright future. With technical advances of deep learning and machine learning in general, as well as more and more accumulated image data, it is just a matter of time when machine-based CNN models will outperform not only inexperienced junior doctors but also senior and experienced medical consultants. Of course, such a statement is not intended to remove human doctors from a medical diagnosis process, but to enhance their decision-making.

References

[1] D. Ponraj and M. Jenifer, “A Survey on the Preprocessing Techniques of Mammogram for the Detection of Breast Cancer,” *J. Emerg. …*, vol. 2, no. 12, pp. 656–664, 2011.

[2] H. Sung *et al.*, “Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries,” *CA. Cancer J. Clin.*, vol. 71, no. 3, pp. 209–249, 2021, doi: 10.3322/caac.21660.

[3] J. Ferlay *et al.*, “Cancer statistics for the year 2020: An overview,” *Int. J. Cancer*, vol. 149, no. 4, pp. 778–789, 2021, doi: 10.1002/ijc.33588.

[4] E. M. Livstone, “Colorectal cancer (Colon Cancer) - Merck Manual, Professional Version,” *Merck Manual, Prof. Version*, vol. 67, no. 1, pp. 7–30, 2019, doi: 10.3322/caac.21387.

[5] T. A. Stavros, D. Thickman, L. Rapp, M. A. Dennis, S. H. Parker, and G. A. Sisney, “Solid Breast Nodules : Use of Sonography to Distinguish Lesions,” *Radiology*, vol. 196, pp. 123–134, 1995.

[6] J. H. Yoon, M. H. Kim, E. K. Kim, H. J. Moon, J. Y. Kwak, and M. J. Kim, “Interobserver variability of ultrasound elastography: How it affects the diagnosis of breast lesions,” *Am. J. Roentgenol.*, vol. 196, no. 3, pp. 730–736, 2011, doi: 10.2214/AJR.10.4654.

[7] J. Ker, L. Wang, J. Rao, and T. Lim, “Deep Learning Applications in Medical Image Analysis,” *IEEE Access*, vol. 6, pp. 9375–9379, 2017, doi: 10.1109/ACCESS.2017.2788044.

[8] G. Litjens *et al.*, “A survey on deep learning in medical image analysis.,” *Med. Image Anal.*, vol. 42, no. December 2012, pp. 60–88, 2017, doi: 10.1016/j.media.2017.07.005.

[9] T. Elsken, J. H. Metzen, and F. Hutter, “Neural Architecture Search: A Survey,” vol. 20, pp. 1–21, 2018, [Online]. Available: http://arxiv.org/abs/1808.05377.

[10] B. Zoph and Q. V. Le, “Neural Architecture Search with Reinforcement Learning,” pp. 1–16, 2017, [Online]. Available: http://arxiv.org/abs/1611.01578.

[11] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean, “Efficient Neural Architecture

Search via Parameter Sharing," 2018, [Online]. Available: http://arxiv.org/abs/1802.03268.

[12] S. K. Sharma GN, Dave R, Sanadya J, Sharma P, "VARIOUS TYPES AND MANAGEMENT OF BREAST CANCER: AN OVERVIEW," vol. 1, no. 2, pp. 109–126, 2010.

[13] D. Kashyap, V. K. Garg, E. N. Sandberg, N. Goel, and A. Bishayee, "Oncogenic and tumor suppressive components of the cell cycle in breast cancer progression and prognosis," *Pharmaceutics*, vol. 13, no. 4, pp. 1–28, 2021, doi: 10.3390/pharmaceutics13040569.

[14] "How cancer start and spreads." https://www.cancer.ca/en/cancer-information/cancer-101/what-is-cancer/how-cancer-starts-grows-and-spreads/?region=on.

[15] "Diagram of Breast cancer vector image." file:///C:/Users/Alastair/Desktop/Diagram of breast cancer Royalty Free Vector Image.html.

[16] R. J. Stafford and G. J. Whitman, "Ultrasound physics and technology in breast imaging," *Ultrasound Clin.*, vol. 6, no. 3, pp. 299–312, 2011, doi: 10.1016/j.cult.2011.02.001.

[17] "Ultrasound scan," *NHS*, 2018. https://www.nhs.uk/conditions/ultrasound-scan/.

[18] A. Türkkol, "FOTOAKUSTİK TOMOGRAFİ," *tiptamuhendislik.wordpress*, 2018. .

[19] M. A. Spinelli Varella, J. Teixeira da Cruz, A. Rauber, I. S. Varella, J. F. Fleck, and L. F. Moreira, "Role of BI-RADS Ultrasound Subcategories 4A to 4C in Predicting Breast Cancer," *Clin. Breast Cancer*, vol. 18, no. 4, pp. e507–e511, 2018, doi: 10.1016/j.clbc.2017.09.002.

[20] Zeimarani, "Breast Tumor Classification in Ultrasound Images, Using Deep Convolutional Neural Network," in *Bashir Zeimarani*, 2019, p. thises.

[21] H.-T. Thai, "Machine learning for structural engineering: A state-of-the-art review," in *Structures*, 2022, vol. 38, pp. 448–491.

[22] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, "A Guide to Convolutional Neural Networks for Computer Vision," *Synth. Lect. Comput. Vis.*, vol. 8, no. 1, pp. 1–207, 2018, doi: 10.2200/s00822ed1v01y201712cov015.

[23] M. L. Giger, “Machine Learning in Medical Imaging,” *J. Am. Coll. Radiol.*, vol. 15, no. 3, pp. 512–520, 2018, doi: 10.1016/j.jacr.2017.12.028.

[24] Y. . Goodfellow, I.; Bengio, “deep learning.” MIT Press,USA,2016.

[25] L. SIfre and S. Mallat, “Rigid-Motion Scattering for Texture Classification,” 2014, [Online]. Available: http://arxiv.org/abs/1403.1687.

[26] M. Lin, Q. Chen, and S. Yan, “Network In Network,” pp. 1–10, 2013, [Online]. Available: http://arxiv.org/abs/1312.4400.

[27] S. Koturwar and S. N. Merchant, “Weight Initialization of Deep Neural Networks(DNNs) using Data Statistics.”

[28] Q. Zhang, M. Zhang, T. Chen, Z. Sun, Y. Ma, and B. Yu, “Recent advances in convolutional neural network acceleration,” *Neurocomputing*, vol. 323, pp. 37–51, 2019, doi: 10.1016/j.neucom.2018.09.038.

[29] B. Monien, R. Preis, and S. Schamberger, “ImageNet Classification with Deep Convolutional Neural Networks,” *Handb. Approx. Algorithms Metaheuristics*, pp. 60-1-60–16, 2012, doi: 10.1201/9781420010749.

[30] Y. B. Xavier Glorot, “Understanding the difficulty of training deep feedforward neural networks,” *Proc. Int. Jt. Conf. Neural Networks*, vol. 2, pp. 1701–1704, 2010, doi: 10.1109/ijcnn.1993.716981.

[31] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, pp. 1026–1034, 2015, doi: 10.1109/ICCV.2015.123.

[32] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” pp. 1–18, 2012, [Online]. Available: http://arxiv.org/abs/1207.0580.

[33] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

[34] C. S. S. Ioffe, “Batch Normalization: Accelerating Deep Network Training b y Reducing Internal Covariate Shift.” Journal of Machine Learning Research (JMLR, pp. 448–456,

2015.

[35] Y. Wu and K. He, “Group Normalization,” *Int. J. Comput. Vis.*, vol. 128, no. 3, pp. 742–755, 2020, doi: 10.1007/s11263-019-01198-w.

[36] S. Ruder, “An overview of gradient descent optimization algorithms,” 2017.

[37] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, “A survey of the recent architectures of deep convolutional neural networks,” *Artif. Intell. Rev.*, pp. 1–70, 2020, doi: 10.1007/s10462-020-09825-6.

[38] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient_Based Learning Applied to Document Recognition,” *proc. IEEE*, 1998, [Online]. Available: http://ieeexplore.ieee.org/document/726791/#full-text-section.

[39] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” pp. 1–14, 2014, [Online]. Available: http://arxiv.org/abs/1409.1556.

[40] C. Szegedy *et al.*, “Going deeper with convolutions,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 1–9, 2014, doi: 10.1109/CVPR.2015.7298594.

[41] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, “A Survey of the Recent Architectures of Deep Convolutional Neural Networks,” pp. 1–67, 2019, [Online]. Available: http://arxiv.org/abs/1901.06032.

[42] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2015, doi: 10.1109/CVPR.2016.90.

[43] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2261–2269, 2017, doi: 10.1109/CVPR.2017.243.

[44] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 2818–2826, 2016, doi: 10.1109/CVPR.2016.308.

[45] M. Längkvist, L. Karlsson, and A. Loutfi, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,” *Pattern Recognit. Lett.*, vol. 42, no. 1, pp. 11–24, 2014, doi: 10.1016/j.patrec.2014.01.008.

[46] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1800–1807, 2017, doi: 10.1109/CVPR.2017.195.

[47] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.

[48] M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” *36th Int. Conf. Mach. Learn. ICML 2019*, vol. 2019-June, pp. 10691–10700, 2019.

[49] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for hyper-parameter optimization,” *Adv. Neural Inf. Process. Syst. 24 25th Annu. Conf. Neural Inf. Process. Syst. 2011, NIPS 2011*, pp. 1–9, 2011.

[50] D. C. James Bergstra, Daniel Yamins, “Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures,” *Med. Anthropol. Cross Cult. Stud. Heal. Illn.*, vol. 34, no. 5, pp. 407–424, 2015, doi: 10.1080/01459740.2015.1058375.

[51] K. Li and J. Malik, “Learning to optimize,” *5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc.*, 2017.

[52] A. A. Radhakrishnan Rahulan, “Vehicle Pair Activity Classification Using QTC and Long Short Term Memory Neural Network,” 2018.

[53] Y. Jaafra, J. Luc Laurent, A. Deruyver, and M. Saber Naceur, “Reinforcement learning for neural architecture search: A review,” *Image Vis. Comput.*, vol. 89, pp. 57–66, 2019, doi: 10.1016/j.imavis.2019.06.005.

[54] B. Zoph and Q. V. Le, “Improving Neural Architecture Search with Reinforcement Learning,” *5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc.*, 2019.

[55] J. Wu, X. Y. Chen, H. Zhang, L. D. Xiong, H. Lei, and S. H. Deng, “Hyperparameter optimization for machine learning models based on Bayesian optimization,” *J. Electron.*

Sci. Technol., vol. 17, no. 1, pp. 26–40, 2019, doi: 10.11989/JEST.1674-862X.80904120.

[56] E. Brochu, V. M. Cora, and N. de Freitas, "A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning," 2010, [Online]. Available: http://arxiv.org/abs/1012.2599.

[57] C. Jeffery, "Practical Bayesian Optimization of Machine Learning Algorithms," *Relig. Arts*, vol. 17, no. 1–2, pp. 57–73, 2013, doi: 10.1163/15685292-12341254.

[58] B. Zoph and Q. Le, "Learning transferable architectures for scalable image recognition," *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, pp. 8697–8710, 2018.

[59] H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable Architecture Search," pp. 1–13, 2018, [Online]. Available: http://arxiv.org/abs/1806.09055.

[60] S. D. De, M. G. F. Costa, W. C. De, and C. F. F. C. Filho, "Breast tumor classification in ultrasound images using neural networks with improved generalization methods," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, vol. 2015-Novem, pp. 6321–6325, 2015, doi: 10.1109/EMBC.2015.7319838.

[61] H. Brody, "A Comprehensive Survey on Deep-Learning-Based Breast Cancer Diagnosis," *Nature*, vol. 579, no. 7800, p. S1, 2020, doi: 10.1038/d41586-020-00840-9.

[62] N. K. Ragesh, a R. Anil, and R. Rajesh, "Digital Image Denoising in Medical Ultrasound Images : A Survey," *ICGST Int. Conf. Comput. Sci. Eng. AIML-11*, no. April, pp. 12–14, 2011.

[63] R. Liao, T. Wan, and Z. Qin, "Classification of benign and malignant breast tumors in ultrasound images based on multiple sonographic and textural features," *Proc. - 2011 3rd Int. Conf. Intell. Human-Machine Syst. Cybern. IHMSC 2011*, vol. 1, pp. 71–74, 2011, doi: 10.1109/IHMSC.2011.127.

[64] S. Khazendar *et al.*, "Automated characterisation of ultrasound images of ovarian tumours: the diagnostic accuracy of a support vector machine and image processing with a local binary pattern operator.," *Facts, views Vis. ObGyn*, vol. 7, no. 1, pp. 7–15, 2015, [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/25897367%0Ahttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4402446.

[65] D. Martínez-Más, J., Bueno-Crespo, A., Khazendar, S., Remezal-Solano, M., Martínez-Cendán, J.P., Jassim, S., Du, H., Al Assam, H., Bourne, T. and Timmerman, "Evaluation of machine learning methods with Fourier Transform features for classifying ovarian tumors based on ultrasound images," *PLoS One*, vol. 14(7), 2019.

[66] J. S. Al-Karawi D, Al-Assam H, Du H, Sayasneh A, Landolfo C, Timmerman D, Bourne T, "An Evaluation of the Effectiveness of Image-based Texture Features Extracted from Static B-mode Ultrasound Images in Distinguishing between Benign and Malignant Ovarian Masses," vol. May;43(3), pp. 124–138, 2021, doi: 10.1177/0161734621998091.

[67] Y. Mao, H. Lim, M. Ni, W. Yan, and D. W. Wong, "Breast Tumour Classification Using Ultrasound Elastography with Machine Learning : A Systematic Scoping Review," pp. 1–18, 2022.

[68] A. Nahid and Y. Kong, "Involvement of Machine Learning for Breast Cancer Image Classification : A Survey," vol. 2017, no. i, 2018.

[69] M. Masud, A. E. Eldin Rashed, and M. S. Hossain, "Convolutional neural network-based models for diagnosis of breast cancer," *Neural Comput. Appl.*, vol. 5, 2020, doi: 10.1007/s00521-020-05394-5.

[70] Y. Jim, "applied sciences Deep-Learning-Based Computer-Aided Systems for Breast Cancer Imaging : A Critical Review," no. Figure 1.

[71] M. Muhammad, D. Zeebaree, A. M. A. Brifcani, J. Saeed, and D. A. Zebari, "A Review on Region of Interest Segmentation Based on Clustering Techniques for Breast Cancer Ultrasound Images," *J. Appl. Sci. Technol. Trends*, vol. 1, no. 3, pp. 78–91, 2020, doi: 10.38094/jastt1328.

[72] Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," pp. 248–255, 2009, doi: 10.1109/cvprw.2009.5206848.

[73] N. Tajbakhsh *et al.*, "Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?," *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1299–1312, 2016, doi: 10.1109/TMI.2016.2535302.

[74] J.-Y. J. Seokmin Han, Ho-Kyung Kang, "A deep learning framework for supporting the

classification of breast lesions in ultrasound images," *Phys. Med. Biol.*, pp. 11–14, 2017.

[75] S. M. Badawy, A. E. N. A. Mohamed, A. A. Hefnawy, H. E. Zidan, M. T. Gadallah, and G. M. El-Banby, "Classification of Breast Ultrasound Images Based on Convolutional Neural Networks - A Comparative Study," *2021 Int. Telecommun. Conf. ITC-Egypt 2021 - Proc.*, 2021, doi: 10.1109/ITC-Egypt52936.2021.9513972.

[76] Y. C. Zhu *et al.*, "A generic deep learning framework to classify thyroid and breast lesions in ultrasound images," *Ultrasonics*, vol. 110, p. 106300, 2021, doi: 10.1016/j.ultras.2020.106300.

[77] M. Byra, H. Piotrzkowska-Wroblewska, K. Dobruch-Sobczak, and A. Nowicki, "Combining Nakagami imaging and convolutional neural network for breast lesion classification," *IEEE Int. Ultrason. Symp. IUS*, pp. 5–8, 2017, doi: 10.1109/ULTSYM.2017.8092154.

[78] B. Zeimarani, M. G. F. Costa, N. Z. Nurani, S. R. Bianco, W. C. De Albuquerque Pereira, and C. F. F. C. Filho, "Breast Lesion Classification in Ultrasound Images Using Deep Convolutional Neural Network," *IEEE Access*, vol. 8, pp. 133349–133359, 2020, doi: 10.1109/ACCESS.2020.3010863.

[79] H. Ma *et al.*, "Fus2Net: a novel Convolutional Neural Network for classification of benign and malignant breast tumor in ultrasound images," *Biomed. Eng. Online*, vol. 20, no. 1, pp. 1–15, 2021, doi: 10.1186/s12938-021-00950-z.

[80] T. Xiao, L. Liu, K. Li, W. Qin, S. Yu, and Z. Li, "Comparison of Transferred Deep Neural Networks in Ultrasonic Breast Masses Discrimination," *Biomed Res. Int.*, vol. 2018, pp. 1–9, 2018, doi: 10.1155/2018/4605191.

[81] Y. Zhu, P. Jin, J. Bao, Q. Jiang, and X. Wang, "Thyroid ultrasound image classification using a convolutional neural network," vol. 9, no. 20, 2021, doi: 10.21037/atm-21-4328.

[82] M. Yamakawa, "Computer aided diagnosis system developed for ultrasound diagnosis of liver lesions using deep learning," pp. 2330–2333, 2019.

[83] U. A. H. Khan *et al.*, "Improving Prostate Cancer Detection with Breast Histopathology Images," pp. 1–9, 2019, [Online]. Available: http://arxiv.org/abs/1903.05769.

[84] H. Li *et al.*, "An improved deep learning approach for detection of thyroid papillary cancer in ultrasound images," *Sci. Rep.*, vol. 8, no. 1, pp. 1–12, 2018, doi:

10.1038/s41598-018-25005-7.

[85] C. Liu *et al.*, “Progressive neural architecture search,” *Eccv*, 2018.

[86] L. Jiang, Y., Zhao, C. and Pang, “Neural Architecture Refinement A Practical Way for Avoiding Overfitting in NAS.” .

[87] S. Kim *et al.*, “Scalable Neural Architecture Search for 3D Medical Image Segmentation,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11766 LNCS, pp. 220–228, 2019, doi: 10.1007/978-3-030-32248-9_25.

[88] L. Faes *et al.*, “Automated deep learning design for medical image classification by health-care professionals with no coding experience: a feasibility study,” *Lancet Digit. Heal.*, vol. 1, no. 5, pp. e232–e242, 2019, doi: 10.1016/S2589-7500(19)30108-6.

[89] Y. Weng, T. Zhou, Y. Li, and X. Qiu, “NAS-Unet: Neural architecture search for medical image segmentation,” *IEEE Access*, vol. 7, pp. 44247–44257, 2019, doi: 10.1109/ACCESS.2019.2908991.

[90] X. He *et al.*, “Automated Model Design and Benchmarking of 3D Deep Learning Models for COVID-19 Detection with Chest CT Scans,” 2021, [Online]. Available: http://arxiv.org/abs/2101.05442.

[91] A. Kwasigroch, M. Grochowski, and A. Mikolajczyk, “Neural architecture search for skin lesion classification,” *IEEE Access*, vol. 8, pp. 9061–9071, 2020, doi: 10.1109/ACCESS.2020.2964424.

[92] T. Elsken, J. H. Metzen, and F. Hutter, “Simple and efficient architecture search for convolutional neural networks,” *6th Int. Conf. Learn. Represent. ICLR 2018 - Work. Track Proc.*, pp. 1–14, 2018.

[93] M. E. Billah and F. Javed, “Bayesian Convolutional Neural Network-based Models for Diagnosis of Blood Cancer,” *Appl. Artif. Intell.*, vol. 00, no. 00, pp. 1–22, 2021, doi: 10.1080/08839514.2021.2011688.

[94] W. Al-dhabyani, M. Gomaa, H. Khaled, and A. Fahmy, “Dataset of breast ultrasound images,” *Data Br.*, vol. 28, p. 104863, 2020, doi: 10.1016/j.dib.2019.104863.

[95] K. S. Sudeep and K. K. Pal, “Preprocessing for image classification by convolutional

neural networks," *2016 IEEE Int. Conf. Recent Trends Electron. Inf. Commun. Technol. RTEICT 2016 - Proc.*, pp. 1778–1781, 2017, doi: 10.1109/RTEICT.2016.7808140.

[96] R. G. Keys, "Cubic Convolution Interpolation for Digital Image Processing," *IEEE Trans. Acoust.*, vol. 29, n, no. I, pp. 1153–1160, 1981, doi: 10.1109/TASSP.1981.1163711.

[97] M. Jogin, Mohana, M. S. Madhulika, G. D. Divya, R. K. Meghana, and S. Apoorva, "Feature extraction using convolution neural networks (CNN) and deep learning," *2018 3rd IEEE Int. Conf. Recent Trends Electron. Inf. Commun. Technol. RTEICT 2018 - Proc.*, pp. 2319–2323, 2018, doi: 10.1109/RTEICT42901.2018.9012507.

[98] P. Dutta, P. Upadhyay, M. De, and R. G. Khalkar, "Medical Image Analysis using Deep Convolutional Neural Networks: CNN Architectures and Transfer Learning," *Proc. 5th Int. Conf. Inven. Comput. Technol. ICICT 2020*, pp. 175–180, 2020, doi: 10.1109/ICICT48043.2020.9112469.

[99] U. Imaging, "Transfer Learning in Breast Cancer Diagnoses via," pp. 1–15, 2021.

[100] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do ImageNet classifiers generalize to ImageNet?," *36th Int. Conf. Mach. Learn. ICML 2019*, vol. 2019-June, pp. 9413–9424, 2019.

[101] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018, doi: 10.1016/j.neunet.2018.07.011.

[102] C. Shorten, T. M. Khoshgoftaar, and B. Furht, "Deep Learning applications for COVID-19," *J. Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-020-00392-9.

[103] Z. Li, W. Yang, S. Peng, and F. Liu, "A Survey of Convolutional Neural Networks : Analysis , Applications , and Prospects."

[104] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway Networks," 2015, [Online]. Available: http://arxiv.org/abs/1505.00387.

[105] J. Wu, "Introduction to Convolutional Neural Networks," *Introd. to Convolutional Neural Networks*, pp. 1–31, 2017, [Online]. Available: https://web.archive.org/web/20180928011532/https://cs.nju.edu.cn/wujx/teaching/15_CNN.pdf.

[106] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?,” *Adv. Neural Inf. Process. Syst.*, vol. 4, no. January, pp. 3320–3328, 2014.

[107] L. Liu *et al.*, “MixSearch: Searching for Domain Generalized Medical Image Segmentation Architectures,” vol. XX, no. Xx, pp. 1–10, 2020.

[108] M. Rahimzadeh, S. Parvin, E. Safi, and M. R. Mohammadi, “Wise-SrNet: A Novel Architecture for Enhancing Image Classification by Learning Spatial Resolution of Feature Maps,” pp. 1–24, 2021, [Online]. Available: http://arxiv.org/abs/2104.12294.

[109] L. Cai, J. Gao, and D. Zhao, “A review of the application of deep learning in medical image classification and segmentation,” *Ann. Transl. Med.*, vol. 8, no. 11, pp. 713–713, 2020, doi: 10.21037/atm.2020.02.44.

[110] M. Loey, S. El-Sappagh, and S. Mirjalili, “Bayesian-based optimized deep learning model to detect COVID-19 patients using chest X-ray image data,” *Comput. Biol. Med.*, vol. 142, no. October 2021, p. 105213, 2022, doi: 10.1016/j.compbiomed.2022.105213.

[111] M. Fatih, K. Sabanci, A. Durdu, and M. Fahri, “COVID-19 diagnosis using state-of-the-art CNN architecture features and Bayesian Optimization,” no. January, 2020.

Appendix A: Detailed experimental results of modified ENAS and AlexNet

This appendix includes the detailed experimental results of the ENAS17 model that was trained on the unbalanced dataset and achieved the highest result compared to other modified ENAS17. In addition, another part of this appendix presents the detailed results of the structural modification of AlexNet by examining different filter sizes and reducing the convolutional layer for breast cancer classification.

Table A-1: Detail Results of ENAS17 Models Trained on Unbalance Breast Cancer Dataset.

Models	Datasets	TNR	TPR	Acc
Model1	Internal	92.4	52.0	72.2
	External_A	82.8	84.3	83.6
	External_B	87.3	88.0	87.7
	External_Avg	87.5	74.8	81.1
Model2	Internal	86.3	81.6	83.9
	External_A	72.1	96.7	84.4
	External_B	70.3	97.0	83.7
	External_Avg	76.2	91.7	84.0
Model3	Internal	92.4	65.3	78.8
	External_A	84.2	84.8	84.5
	External_B	82.7	97.5	90.1
	External_Avg	86.4	82.5	84.5
Model4	Internal	88.9	82.7	85.8
	External_A	82.0	83.3	82.7
	External_B	80.7	95.5	88.1
	External_Avg	83.8	87.2	85.5
Model5	Internal	87.7	81.6	84.6
	External_A	81.4	71.4	76.4
	External_B	78.0	96.0	87.0
	External_Avg	82.4	83.0	82.7

Table A-2: Detail Results of the Structural Modification of AlexNet for Breast Cancer Classification

Modified AlexNet	Filter size	TNR	TPR	Acc
AlexNet	Alex_F.S	51.5	49.1	50.3
	F.S(3*3)	52.4	50.6	51.5
	F.S(5*5)	51.0	49.8	50.4
	F.S(7*7)	51.9	51.8	51.9
	F.S(9*9)	59.2	60.2	59.7
	F.S(11*11)	69.7	69.9	69.8
	F.S(13*13)	75.0	75.3	75.1
	F.S(15*15)	76.3	79.3	77.8
Alex_RL2	Alex_F.S	52.8	52.0	52.4
	F.S(3*3)	50.8	50.3	50.6
	F.S(5*5)	53.0	52.2	52.6
	F.S(7*7)	63.9	64.9	64.4
	F.S(9*9)	73.6	78.3	76.0
	F.S(11*11)	81.7	80.6	81.2
	F.S(13*13)	79.2	82.5	80.8
	F.S(15*15)	80.2	78.8	79.5
AlexRemL2_notMax	AlexRemL3_notMax	50.6	50.2	50.4
	F.S(3*3)	48.9	49.3	49.1
	F.S(5*5)	53.7	53.5	53.6
	F.S(7*7)	59.9	60.2	60.1
	F.S(9*9)	71.0	72.0	71.5
	F.S(11*11)	76.3	78.4	77.3
	F.S(13*13)	77.2	79.3	78.3
	F.S(15*15)	77.7	81.2	79.4
Alex_RL3	Alex_F.S	50.6	48.4	49.5
	F.S(3*3)	48.0	48.4	48.2
	F.S(5*5)	50.4	52.3	51.4
	F.S(7*7)	55.5	54.1	54.8
	F.S(9*9)	57.0	60.4	58.7
	F.S(11*11)	64.0	67.4	65.7
	F.S(13*13)	70.5	71.9	71.2
	F.S(15*15)	73.1	73.3	73.2
Alex_RL4	Alex_F.S	50.5	49.0	49.7
	F.S(3*3)	49.3	49.8	49.5
	F.S(5*5)	51.6	51.3	51.5
	F.S(7*7)	52.3	55.7	54.0
	F.S(9*9)	57.5	59.9	58.7
	F.S(11*11)	68.1	66.0	67.1
	F.S(13*13)	69.3	72.1	70.7
	F.S(15*15)	73.1	75.2	74.2

Table A-2 continue: Detail Results of the Structural Modification of AlexNet for Breast Cancer Classification

Modified AlexNet	Filter size	TNR	TPR	Acc
Alex_RL5	Alex_F.S	49.6	52.6	51.1
	F.S(3*3)	51.7	47.9	49.7
	F.S(5*5)	51.1	52.3	51.7
	F.S(7*7)	55.5	56.7	56.1
	F.S(9*9)	58.8	60.5	59.7
	F.S(11*11)	66.4	67.9	67.2
	F.S(13*13)	72.9	74.0	73.4
	F.S(15*15)	73.4	74.2	73.8
Alex_RL3 and 5	Alex_F.S	52.6	55.5	54.1
	F.S(3*3)	51.7	52.7	52.2
	F.S(5*5)	48.0	54.5	51.3
	F.S(7*7)	56.2	60.1	58.2
	F.S(9*9)	64.0	66.7	65.3
	F.S(11*11)	72.6	74.2	73.4
	F.S(13*13)	73.9	77.7	75.8
	F.S(15*15)	76.7	76.8	76.7
Alex_RL2 and 4	Alex_F.S	53.9	54.2	54.1
	F.S(3*3)	50.3	51.1	50.7
	F.S(5*5)	53.6	53.3	53.4
	F.S(7*7)	63.8	62.7	63.2
	F.S(9*9)	74.3	72.8	73.5
	F.S(11*11)	77.4	79.2	78.3
	F.S(13*13)	78.3	77.0	77.6
	F.S(15*15)	78.6	79.2	78.9

Appendix B: Detailed results of structural modification of ENAS method

This appendix presents the detailed experimental results of the structural modification of the ENAS method, which consists of two strategies: expanding the search space operation set and designing highway connections.

Table B-1: Detail Results of the ENAS7 Designed by Expanded Search Space Set-A

Models	Datasets	TNR	TPR	Acc
Model1	Internal	86.2	77.3	81.8
	External_A	75.2	95.2	85.2
	External_B	72.6	97.5	85.1
	External_Avg	73.9	96.4	85.1
Model2	Internal	95.2	61.8	78.5
	External_A	86.5	78.1	82.3
	External_B	82.7	93.5	88.1
	External_Avg	84.6	85.8	85.2
Model3	Internal	88.9	72.0	80.4
	External_A	76.9	84.3	80.6
	External_B	69.0	97.0	83.0
	External_Avg	73.0	90.6	81.8
Model4	Internal	91.0	84.0	87.5
	External_A	83.4	80.5	81.9
	External_B	81.0	96.0	88.5
	External_Avg	82.2	88.2	85.2
Model5	Internal	84.2	85.5	84.9
	External_A	82.3	86.7	84.5
	External_B	67.0	98.5	82.8
	External_Avg	74.6	92.6	83.6

Table B-2: Detail Results of Modified ENAS17 with Short Highway (25) on Unbalanced Breast dataset

Models	Datasets	TNR	TPR	Acc
Model1	Internal	86.9	65.3	76.1
	External_A	75.2	84.8	80.0
	External_B	75.7	96.0	85.8
	External_Avg	75.4	90.4	82.9
Model2	Internal	89.0	72.4	80.7
	External_A	74.4	93.3	83.8
	External_B	73.0	96.0	84.5
	External_Avg	73.7	94.7	84.2
Model3	Internal	87.5	78.7	83.1
	External_A	79.2	91.0	85.1
	External_B	71.0	98.0	84.5
	External_Avg	75.1	94.5	84.8
Model4	Internal	83.3	89.5	86.4
	External_A	75.8	89.0	82.4
	External_B	72.7	98.5	85.6
	External_Avg	74.2	93.8	84.0
Model5	Internal	83.3	89.5	86.4
	External_A	73.5	89.5	81.5
	External_B	66.7	99.5	83.1
	External_Avg	70.1	94.5	82.3