Algebraic, Topological, and Geometric Driven Convolutional Neural Networks for Ultrasound Imaging Cancer Diagnosis

Jehan Sherwan Ziad Ghafuri

A Thesis Submitted for the Degree of Doctor of Philosophy
in The University Of Buckingham

June 2023

Abstract

Despite the astonishing successes of Convolutional Neural Networks (CNN) as a powerful deep learning tool for a variety of computer vision tasks, their deployments for ultrasound (US) tumour image analysis within clinical settings is challenging due to the difficulty of interpreting CNN decisions compounded by lack of availability of class labelled "good quality" US tumour image datasets that represent an i.i.d random sample of the unknown population. The use of CNN models pretrained on natural images in transfer learning (TL) mode for US image analysis are perceived to suffer from a lack of robustness to small changes and inability to generalisation to unseen data.

This thesis aims to develop a strategy for designing efficient CNN architectures customised for US images that overcome or significantly reduce the above challenges while learning discriminating features resulting in highly accurate diagnostic predictions. We first uncover the significant differences in the statistical contents and spatial distribution of image texture landmarks (e.g. Local Binary Patterns) between US images and natural images. Therefore, we investigate the effects of convolution with random Gaussian filters (RGF) on US image content in terms of spatial and an innovative texture-based entropy, and the spatial distribution of texture landmarks. These effects are determined for US scan images of malignant and benign masses for breast, bladder, and liver tissues.

We demonstrate that several pretrained CNN models retrained on US tumour scan images in TL mode achieve high diagnostic accuracy but suffer greatly from a lack of robustness against natural data perturbation and significantly low generalisation rates due to highly ill-conditioned convolutional layer filters. Thus, we investigate the behaviour of the CNN models during the training process in terms of three mathematically linked characterisation of the filters point clouds: (1) the distribution of their condition numbers, (2) their spatial distribution using persistent homology (PH) tools, and (3) their effects on tumour discriminating power of texture landmark PH scheme in convolved images. These results pave the way for a credible strategy to develop high-performing customised CNN architectures that are robust and generalise well to unseen US scans.

We further develop a new approach to ensure equal condition numbers across the different channel-wise filters at initialisation, and we highlight their impact on the PH profiles as point clouds. However, the condition number of filters continues to be unstable during training, therefore we introduce a simple novel matrix surgery procedure depending on singular value decomposition as a spectral regularisation. We illustrate that the PH of different point clouds of RGFs and their inverses are distinct (in terms of their birth/death of connected components and holes in dimensions 0 and 1) depending on variation in their condition number distributions. This behaviour changes as a result of applying SVD-surgery, so that the PH of point cloud of a filter set post SVD-surgery approaches the same shape and connectivity of a point cloud of orthogonal RGFs.

To my lovely family,
who gave me strength and unconditional love.

Acknowledgement

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Sabah Jassim, for his valuable insights, constructive feedback and unwavering support throughout my research journey. His constant encouragement, dedication, patience and belief in my abilities have been an immense source of motivation and inspiration. I am truly blessed and fortunate for his guidance and mentoring, which has not only nurtured my academic growth, strive for excellence, and overcome challenges, but also instilled in me a lifelong passion for diving into research and of course topology.

In an ideal world, my PhD journey would not have been extremely challenging, but with the COVID-19 pandemic, global/local chaos, and ongoing injustice against Kurds, the obstacles I faced have been overwhelming. Indeed, this journey would not have been possible without the support and unconditional love from my wonderful parents - Begard and Sherwan - and my brothers - Marwan, Miran, and Hemin. Your belief in me and your constant reminders of my potential have shaped me into the person I am today. You are simply my regulariser during ill-conditioned situations and gave me strength and persistence to keep going. I am incredibly grateful for the joy and inspiration you bring to my life.

Most importantly, I would like to wholeheartedly express my gratitude to everyone in the School of Computing. Their invaluable assistance, prompt responsiveness, and exceptional organisational efforts have undoubtedly fostered a highly conducive environment for my research. For that, I am thankful to Prof. Harin Sellahewa, Prof. Hongbo Du, and Dr. Hisham Al-Assam. Special thanks to Dr. Aras Asaad for introducing me to the topological data analysis for fake and genuine face images. Jayne Kelly and Alison Wood, I want to express my appreciation for your kindness, unwavering support, and prompt responses throughout my journey towards completing this thesis. Furthermore, I would like to extend my appreciation to Ten-D AI Medical Technologies Ltd for their sponsorship of this research. I consider myself incredibly fortunate to have had the opportunity to engage in a collaborative academic/industry research experience.

During my PhD journey, I consider myself fortunate to have crossed paths with remarkable individuals like Dr. Gillian Hill and Shabina Begum. Your commitment to advocating for the rights and empowerment of girls and women is truly inspiring. The opportunity to be a part of the *Soapbox Science* and *Women of the Future programme* has undeniably been the highlight of my academic experience, leaving an indelible mark on my journey.

Last but not least, I extend my sincere appreciation to my friends and colleagues who have been a source of inspiration, support, (dis)encouragement and occasional humour provided much-needed flavour that has made the challenging moments more bearable. Simply put, I could not possibly mention everyone's name, but this realisation has only magnified how truly fortunate I am to have such an incredible support network.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The field of data science and representation learning has emerged in fields across multidisciplinary research communities driven by challenging real-world scenarios including environmental, financial, and medical problems. Although, most of the challenging tasks exist prior to the artificial intelligence (AI) and advanced computing power era. The profound impact of AI, machine learning (ML), deep learning (DL), and neural networks (NNs) in computer vision and natural language processing has been evident in terms of their broad use from auto-driving cars, smart houses, to automated medical diagnoses. In particular, these technologies have already helped make significant progress in dealing with complex medical problems related to life-threatening diseases such as cancer and COVID19. There is an increased recognition of the need to exploit the great potentials of these technologies for integration into primary digital healthcare systems with all the benefits for the good of humanity. A key benefit of medical image analysis via machine and deep learning tools is that it can help healthcare professionals make more accurate and consistent diagnoses. By automating the process of analysing medical images, machine and deep learning algorithms can help reduce the risk of human error and improve accuracy and consistency of medical diagnoses besides easing the shortage of medical experts in most national health surfaces. It can lead to earlier and more effective treatment of medical conditions with ultimately improved patient outcomes.

Despite the remarkable achievements and progress achieved by deep learning models in computer vision tasks, the practical implementation of artificial intelligence across various domains faces significant barriers concerning the development of deep neural networks that possess both robustness and the ability to generalise effectively beyond the training data [1]. In clinical settings and medical image diagnostics, scarcity of data samples, black-box style of decision-making with little/no interpretability, trustworthiness and reliability are major concerns compounded by the ease with which DL systems can be accessed and tampered with, [2]. Even though most of the theoretical mathematical foundations in DL are well-known yet computing a stable model is somewhat impractical [1, 3]. Inherited numerical instability is a major issue due to the need for extremely high computational processing on complex and high-dimensional datasets. How can we trust AI and prevent it from making erroneous decisions? Even a small amount of noise or perturbations in data or models parameters can change the outcomes of the system making DL models vulnerable and more susceptible to such failure.

The inquisitiveness of how DL models learn and what kind of features are extracted from the training data through the hidden layers unfolded fascinating research outcomes while unveiling significant challenges. One of the most important aspects of medical image analysis is the development of algorithms that can accurately classify, detect and segment medical images. This involves developing algorithms that can identify and isolate the structures of interest in medical images, such as tumours or blood vessels, [4, 5]. This thesis is devoted to investigate CNN architectural factors and parameters that may underlie the observed deficiencies in robustness to minor perturbations and the limited generalisation capacity to previously unseen data, particularly in the context of analysing ultrasound tumour scan images. Additionally, the research aims to propose viable remedies to address these identified issues and enhance the overall performance and reliability of the CNN-based analysis of tumour scan images.

The next two sections of this chapter are concerned with the problem statements of research conducted in this thesis. Section 1.1, describes the general area of medical image analysis using deep learning and topological data analysis. In Section 1.2, we state the main obstacles to the deployment of deep learning for medical imaging and the inherent problems in computational approaches. In the rest of the chapter, we state the aim and objectives of the research project, describe main challenges and existing approaches, our strategy and contributions, and the overall structure of the thesis.

1.1 Medical image analysis

Medical imaging is an essential part of clinical practices whereby a variety of imaging instruments are employed to examine the interior of human body parts hidden by skin and bone structures for a variety of purposes including disease status assessment of some organs/tissue or monitoring their functions for abnormality as well as guiding medical intervention and treatment procedures. There are many different types and modalities of medical imaging that can be used for different diagnostic purposes of human body, e.g. Magnetic Resonance Imaging (MRI), Computed Tomography (CT) scans and ultrasound (US) images, [6,7]. Each of these images contain a wealth of complex information and require highly experienced expert clinicians to interpret manually, while the scanning equipments are becoming more sophisticated and rely on operators with high skills in digital technologies.

Medical image analysis has become a research-intensive field of computer vision that involves the use of advanced computational techniques to extract meaningful disease-relevant information from medical scan images of tissues/organs. By using computer algorithms to analyse medical images, healthcare professionals can speed up the process of their clinical examinations and improve reliability of detecting and diagnosing a wide range of medical conditions, including tumours, infections, and abnormal growths. With the increasing availability of medical images and the growing demand for more accurate and efficient diagnostic methods, medical image analysis has become a critical area of study in the medical field. It requires a strong understanding of advanced computational techniques, including machine learning, image processing, data analysis, and domain experts, in order to effectively analyse and interpret medical images [6]. In addition, medical image analysis has the potential to improve patient care by enabling the identification of early warning signs of disease, monitoring of treatment effectiveness and support regular screening efforts. But, such potentials rely heavily on automating the image analysis algorithms. Early research on automating medical image analysis followed the common pattern recognition approach by developing handcrafted image texture features analysis leading to noticeable success. Exploiting recent significantly increased computational power and the revolutionary advances in the field of AI for medical image analysis, provides significantly more opportunities to diagnose and treat conditions more quickly, accurately, and effectively.

In this thesis, we focus on ultrasound images of urinary bladder, liver and breast. But many of the issues raised during our investigation are by no mean confined to US imaging modality. Ultrasound images can capture information about tissues of abdominal organs and measure the blood flow in vessels. Despite its usefulness, there are several shortcoming of USI such as quality of the images, interpreting them, type of machine, air and movement inside body [7]. In general, US images have lower resolution compared to other modalities such as CT or MRI due to the fact that they are affected by the frequency of the sound waves emitted by the transducer prob, with higher frequencies resulting in better resolution but less insight to the deep layers of the tissues. The resolution can be affected by factors such as the distance between the transducer and the tissue being imaged, the presence of artefacts such as speckle noise and lateral shadows as well as the deployed image processing techniques. Figure 1.1 presents the ultrasound samples of benign and malignant tumours of three different organs such as breast, liver, and bladder, and the yellow boxes are highlighting the region of interest (RoI) of the tumour. The considerable variation in terms of the RoI size and shape is a major challenge as resizing the image, in preparation of input to CNN models, may lead to additional computational instabilities. Mitigating some/all these issues help achieve more accurate tumour classification needed for early detection and treatment purposes, but are not the only challenging factor that influences the success of deploying AL algorithms.

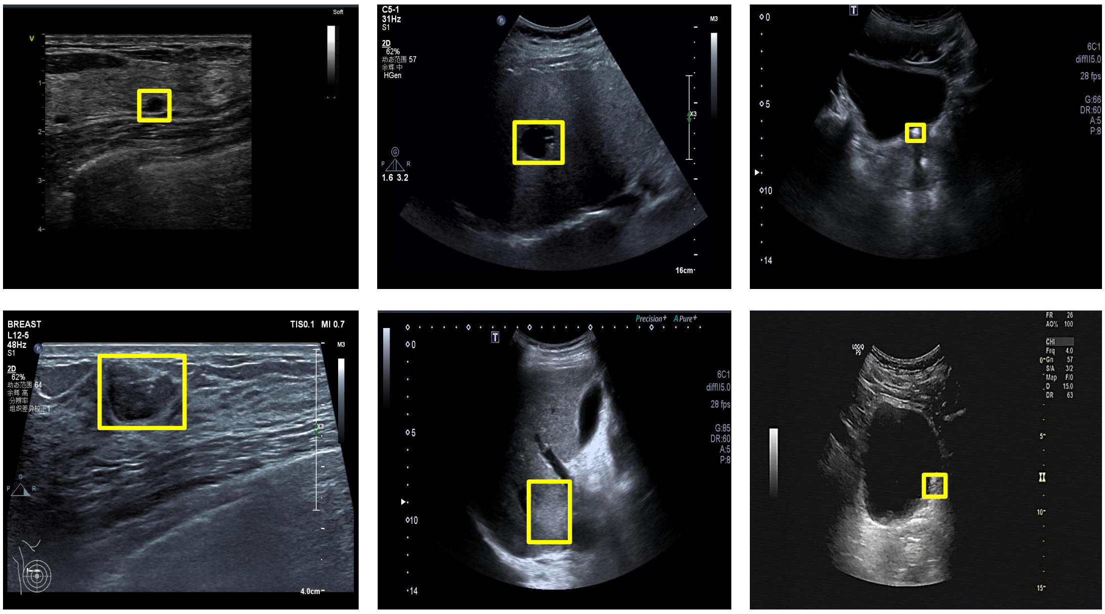


Figure 1.1: Samples of benign (top) and malignant (bottom) tumours from breast, liver, and bladder Ten-D ultrasound datasets.

1.2 Deep learning challenges

Convolutional neural network models have shown extraordinary success in diverse areas of image analysis and classification applications due to their significant learning capacities leading to uncovering huge amounts of hidden image feature patterns far beyond handcrafted features based machine learning models or human experts capacity. Moreover, during training, CNN models learn massive amounts of architectural parameters including the large sets of convolution filters that are used to extract the learnt image patterns during testing. However, this success is not without significant drawbacks, especially in critical applications like medical image analysis. CNN decisions leverage computational power to extract a large number of learned hidden image feature patterns and sift through them. The challenge lies in associating these learned patterns with the causes of medical diseases or abnormalities, making the ***interpretability of CNN decisions*** a formidable task. Despite the careful initialisation of numerous convolution filters, the learned versions of these parameters frequently change during the elaborate iterative training process, enabling the model to learn image feature patterns that align with the training set and yield high performance on similar data samples. This phenomenon is commonly known as the ***overfitting challenge*** of CNN, which often manifests as a lack of generalisation to unseen data. The third challenge of CNN models that could occur in other image modalities is the lack of ***robustness against data perturbation*** and noise which in turn becomes a source of ***adversarial attacks***. A potential source of this challenge relates to the ***instability*** of the convolution filters during training as a result of differences between successive training sample patches as a results of variable level of artefacts and noise. The third challenge faced by CNN models, which may be encountered in other image modalities, is the lack of ***robustness against data perturbation*** and noise, consequently leading to vulnerabilities to ***adversarial attacks***. This challenge is potentially rooted in the ***instability*** of convolution filters during training, arising from variations between successive training sample patches due to variable levels of artefacts and noise.

This thesis is primarily concerned with the last two operational challenges, and the first challenge of decision interpretability will be differed to future work. To understand the targeted challenges and adopt/develop mitigating solutions, we need to investigate Computer Aided Diagnostics in the clinical setting with focus on factors that can cause instabilities in deep learning. In Section 1.2.1, we describe specific scenarios that lead to possible perturbations in ultrasound images that in turn can cause misdiagnosis. In Section 1.2.2, we describe the general sources in computational instability of CNN models in relation to the two challenges for US images. We close this section by briefly describe existing approaches to deal with these two operational challenges in comparison to our alternative approach.

1.2.1 Data perturbation in medical images

The main concern with AI is its trustworthiness and reliability of the predicted decisions. Adversarial attacks on images in CNNs refer to the intentional manipulation of images with the goal of causing misclassification by the CNN model. This problem occurs even when CNN models are trained with large datasets of natural images. Even a small amount of noise or perturbations can change the outcome predictions of the system. A popular example is the image of a panda [8], where even a small amount of noise can change the classification result from a panda to another object. In medical and clinical applications, These attacks can have serious implications where incorrect diagnoses could lead to patient harm. Some examples of adversarial attacks on medical images include adding perturbations to the input image, modifying the intensity or contrast of the image, or adding small patterns that are not easily noticeable to the human eye. These attacks can be carried out through various statistical, machine learning and deep learning techniques, [9,10]. For a deeper understanding of the adversarial attacks we refer the reader to [11–15]

Adversarial attacks are not the only source of lack of robustness challenge, but medical image data perturbation and artefacts are more inherent to the process of recording medical images. For instance, in the case of a skin lesion [16], a small amount of noise from the scanning device, variation in lighting, or inter-personal variation of capturing medical images image can undermine the confidence in the output diagnostic prediction. In our case, introducing a small amount of noise that is invisible to human visuals to the testing images led to misclassifying the benign and malignant tumours as shown in Figure 1.2. This is a critical and major issue, as a misdiagnosis can have severe consequences for patients and may become a source of litigation.

In this thesis we assume two natural perturbation scenarios[1] in terms of the US image modality and cancerous tissues. The first assumption is the ultrasound scan quality can be affected by various factors such as the machine settings, radiologists, the tissue being imaged, the presence of artefacts, and the image processing techniques. Therefore, we exploit natural perturbations as means of adversarial scenarios in our empirical investigations to test the robustness of the CNN models and examine the generalisability against an external dataset.

The second source of concern regarding the presence of perturbations in medical images is associated with the spatial distribution of tissues and textures, where

[1] Natural perturbation scenarios, such as Gaussian and speckle noise, are discussed in detail in Section 3.4.2.

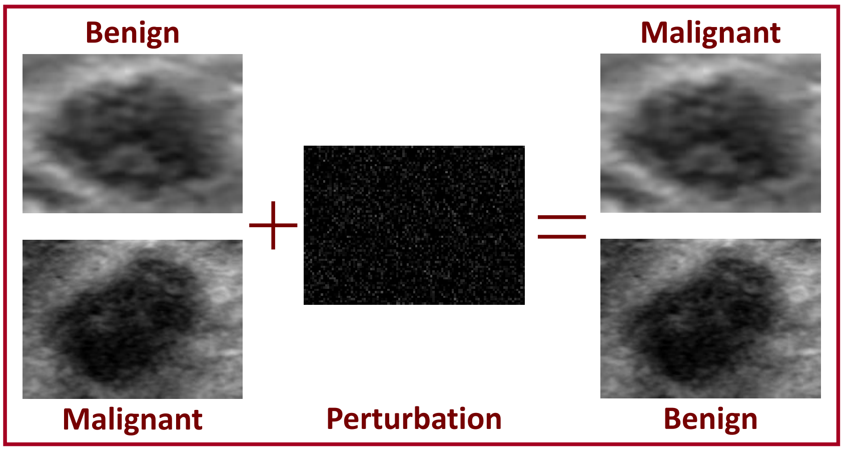


Figure 1.2: Adversarial example on ultrasound images.

cells exhibit distinct patterns. The tumour takes its shape from the changes in the cells and deformation in their structure, as illustrated in Figure 1.3. In fact, the impact of heterogeneous nature of noise interference varies across different regions of an image. In areas with intricate textures, noise can disrupt fine details, potentially obscuring critical information. Whereas, in smoother regions, noise may manifest as a subtler interference, affecting overall image clarity or introducing faint background disturbances.

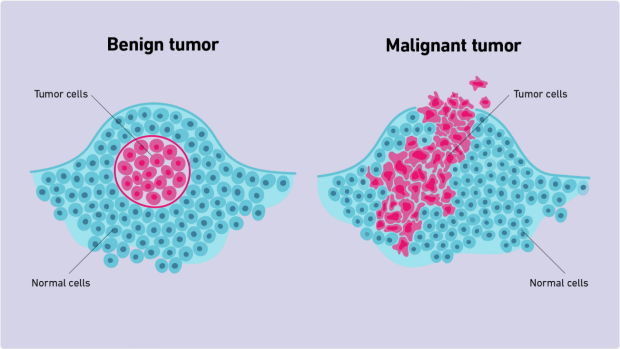


Figure 1.3: Illustration of benign and malignant tumours, [17].

1.2.2 Instability of convolution filters in DL models

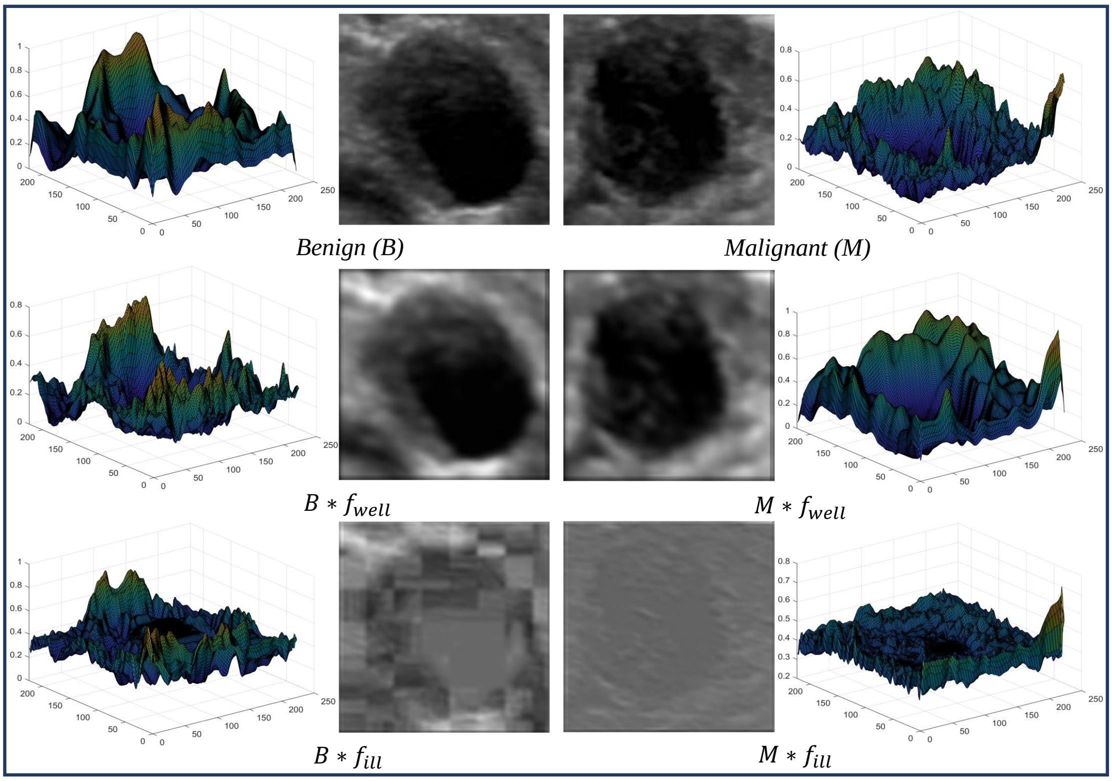
The complexity of deep learning models involve a variety of parameters, which are dominated by the sets of trainable weights and computations deployed at various stages and layers. Therefore, the source of numerical instability in DL methods are partially due to the use of a large number of parameters, hyperparameters, and data that result in floating-point errors and inaccurate results. In particular, In general, accurate and stable numerical algorithms play a significant role to computing a robust and reliable computational models [18].

Training any CNN model to learn class-discriminating image feature patterns is a complex procedure during which training set image batches are iteratively subjected to the various operations of the convolution layers as well as the fully connected layers, and post each iteration the backpropagation algorithm is used to optimise the loss function and achieve optimal learning rates by changing all the sets of weights. The corresponding changes of convolution filters entries results in possible changes to their condition numbers to enable learning feature patterns from the current batch, and this is likely to change again after each epoch or as a result of the input of a new batch. Despite the careful initialisation of convolution filters and the incorporation of regularisation techniques, the convolution filters in most pretrained CNN models tend to exhibit a high degree of ill-conditioning[2]. Retraining such pretrained CNN models on US datasets in transfer learning mode were shown to exhibit significant instability of filters condition numbers, [19]. The importance of maintaining stability in the condition numbers of convolution filters arises from the fact that convolving images with filters possessing distinct condition numbers can lead to the learning of significantly different feature patterns. Figure 1.4, below, illustrates how convolving an image with two filters of markedly different condition numbers produces distinct convolved images.

Therefore, the instability in the condition numbers of convolution filters resulting from pre-training CNN models on US images contributes to the challenge of overfitting in CNN models. It is noteworthy that within the domain of machine learning and deep learning, complications like vanishing or exploding gradients and suboptimal convergence frequently arise due to issues of ill-conditioning. To tackle these challenges, commonly employed strategies encompass regularisation, data augmentation, normalisation, re-parameterisation, standardisation, and the implementation of random dropouts.

Moreover, convolving images with highly ill-conditioned filters can result in learning significantly different feature patterns form different image patches for which the original pixels only differ by tiny amounts as a result of the way noise effect patches that differ in texture contents. This is true for any image modality, but in the case of US images this could have particular impact on the stability of CNN models used for US image analysis. Figure 1.5, below, illustrates the signif-

[2] In the context of deep learning, ill-conditioned filters, indicate an exceptional sensitivity to minor variations in input data. This sensitivity can compromise the stability and reliability of numerical computations, posing challenges in the training of DL models and potentially leading to numerical instability, thereby impacting overall robustness.



Benign (B)

Malignant (M)

$B * f_{well}$

$M * f_{well}$

$B * f_{ill}$

$M * f_{ill}$

Figure 1.4: Convolving a US image with two filters of different conditioning.

icant difference between texture features in a US image and a noisy version of it, pre- and post-convolution with two filters one reasonably well-conditioned and another highly ill-conditioned. Accordingly, the instability in the conditioning of convolution filters also contributes to a lack of robustness against data perturbation.

1.2.3 Research approaches for addressing challenges in robustness and generalisation

The two operational challenges mentioned above are not exclusive to the application of CNNs for the analysis of US images. Furthermore, the performance of CNN models in analysing natural/medical images is not solely dependent on the stability of the condition numbers of convolution filters. Indeed, the effectiveness of these CNN models relies on several critical factors, such as the chosen CNN architecture, the employed optimisation algorithm, dataset size and range of diverse examples, as well as parameter and hyperparameter choices. It is essential to avoid even the slightest oversight or incorrect selection, as they can result in suboptimal outcomes without providing any indication of the problem. Notably,

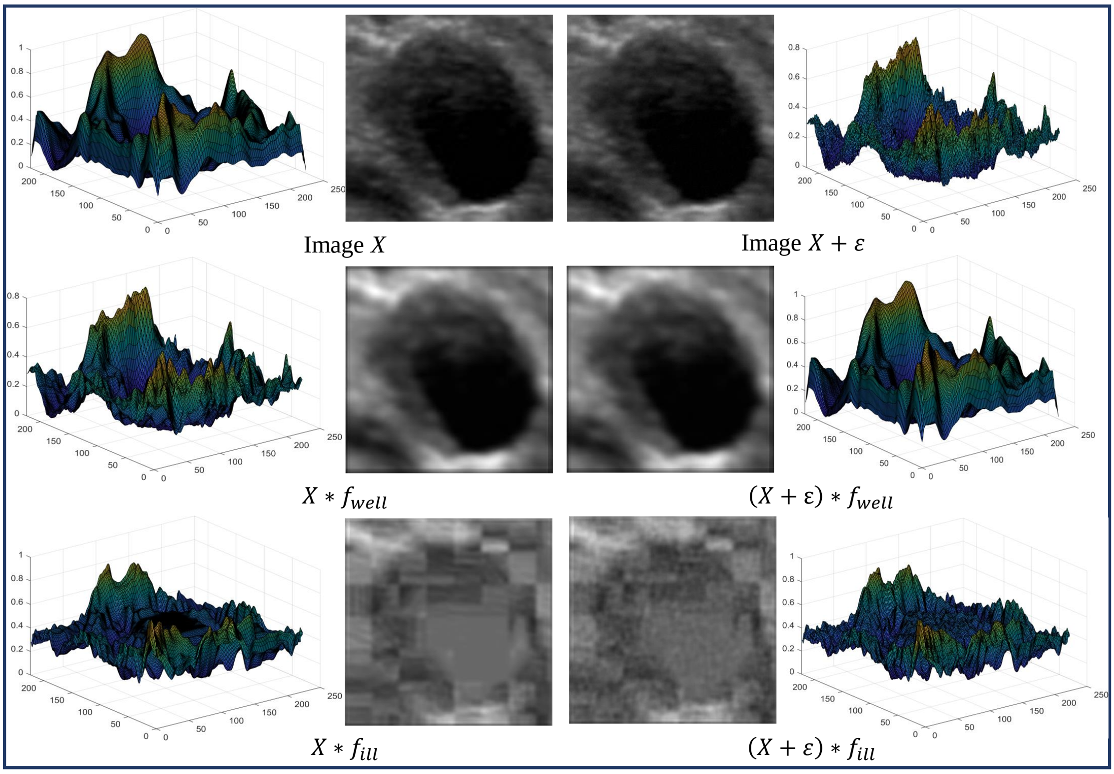


Image X

Image $X + \varepsilon$

$X * f_{well}$

$(X + \varepsilon) * f_{well}$

$X * f_{ill}$

$(X + \varepsilon) * f_{ill}$

Figure 1.5: Image texture post convolution with two different conditioned filters.

the initialisation of weights in both the convolutional layers and fully connected layers holds significant importance in achieving desirable outcomes during CNN model training. These limitations highlight the need for continued research and development in deep learning algorithms for medical imaging in clinical settings.

The overfitting problem in deep learning is expected to be more prevalent in medical image analysis due to the insufficient number of samples available for training a well-performing CNN architecture from scratch. Using data augmentation prior retraining pretrained CNN models in transfer learning mode, that have optimal performance on natural image analysis, have been widely proposed for many applications to overcome model overfitting, but in many cases only suboptimal performance are reported. The scarcity of reliably cropped and labelled US tumour images is somewhat very severe not only in terms of the number of available training images but also of restricted diversity within the rather unknown population. In general, pretrained CNN models for medical images analysis are retrained with a specific dataset collected at a single/multiple clinical centres that practice common scan procedure may still suffer from overfitting when tested on data obtained in other clinical centres that use different devices, practice different recording conditions due to interpersonal variation in the level of expertise of radiologists.

For other possible causes of the overfitting problem, several methods have been extensively explored in the literature with varying degrees of success. These methods aim to mitigate complexities and overfitting issues without compromising the inherent accuracy of the model. Notable techniques include random dropout, regularisation, compression, clustering, filter and/or layer pruning, batch normalisation, impose strict initialisation strategies for filter weights including orthogonality/orthonormality, as well as the utilisation of diverse and sophisticated optimisation algorithms [20–22].

Our approach will be focused on addressing the factors contributing to the high level of instability in convolution filters during the retraining of pretrained CNN models on US images. We will investigate various solutions aimed at mitigating this instability throughout the training process, particularly those that help control the condition numbers of filters. We search for solutions that have additionally desired properties about their spatial distributions in their domains manifested by Topological Data Analysis tool of Persistent Homology. We shall also investigate the development of customised Slim efficient CNN models that can be trained with US images from scratch that incorporate these solutions and perform well in terms of generalisation to unseen data as well as being robust against tolerable levels of data perturbation.

1.3 Aim and Objectives

This PhD research project aims to develop highly efficient convolutional neural network models that can effectively learn distinctive features from ultrasound or radiology images, supporting clinical diagnostic decisions. These models should exhibit robustness against natural data perturbations within acceptable limits and minimise the risk of overfitting when tested on new, unseen data. In a nutshell, the overall aim of the research investigation is to understand, explain, and identify potential source of overfitting in CNN models. The main objectives that guided our research are stated below:

1. Investigate and determine the effects of pretrained CNN models convolution layers on pixel and textural data contents in US images, and how these effects vary in relation to data perturbation by noise and the algebraic properties of convolution filters conditioning. These investigations require good statistical knowledge of US tumour images pixel/texture contents and how do they relate to those of natural images. Among other things, the outcome of these investigations are expected to determine if the convolutions layers

extraction of feature maps could contribute to potential lack of robustness and/or overfitting.

2. Investigate, the impact of the pretrained CNN model's convolution layers components, post retraining, on the spatial distribution of textural landmarks in US tumour images prior to feeding into the FCL component for classification. The outcome from this is expected to help determine the discriminating power of these distributions in the learnt feature maps. For these investigations, we deploy the TDA tools and image processing methods for texture feature extraction.

3. Determine the performance of retraining a few state-of-the-art pretrained CNN models on US breast tumour images in transfer learning (fine-tuning) mode, and determine the actual extent to which these models suffer from lack of robustness against tolerable data perturbation and overfitting in terms of ability to generalise to unseen data. The experimental works for these investigations should monitor the stability of the conditioning of pretrained convolution filters during the iterative retraining procedure.

4. The remaining objectives are concerned with building on the outcomes from the previous tasks to explore methods for mitigating the challenges of overfitting and lack of robustness in CNN analysis of US tumour image datasets. These investigations can benefit from (1) studying the topological profiles of point clouds of convolution filters and their inverses in relation to their conditioning instability, and (2) leveraging existing research on controlling condition numbers of matrices.

1.4 Contributions

In this thesis, we investigate the way DCNNs work. The main demonstrated contributions in this research include:

1. The first contribution of this thesis is analysing ultrasound texture features in terms of their nature, statistics, and spatial distribution for original (raw) and convolved images (feature maps) in general as well as per convolutional layer in CNN model settings i.e. at the convolution stage, after applying the activation function, after local response normalisation, and down-sampling (if any). This investigation is described in Chapters 3 and 4. These results highlight the importance of knowing the extent to which a dataset of images

are distinct from natural images, before adopting CNN models pretrained with large datasets of natural images, for their analysis.

2. Uncovering the link between the condition number of matrices and persistent homology of filter (matrix) point clouds. To understand the algebraic properties of CNN filters at initialisation, during and post-training from scratch and pretrained models through Chapters 4 and 5.

3. Developing a strategy for constructing efficient slim CNN architectures customised for analysis of ultrasound tumour scan images, that are capable of learning discriminating features for reliable diagnostic predictions while robust against tolerable data perturbation and less prone to overfitting effects. This strategy have been successful in designing customised CNN models for classifying malignant and benign tumours from ultrasound images based on the given knowledge in Chapters 3 and 4. In Chapter 5, we showcase several customised CNN models, we designed, and test the models against various clean and noisy datasets. In addition, we propose innovative techniques to initialise well-conditioned filters (weights).

4. Comprehensive empirical investigation of the instability of filters' condition number during the training process and its relation to factors such as the nature of the training dataset, the initial conditioning of the filters, from scratch as well as from pretrained models throughout Chapters 4, 5, and 6.

5. Developing a flexible matrix conditioning based singular value decomposition (SVD), filter surgery, as a regularisation technique. We demonstrate the impact of SVD-Surgery on extensive sets of filter point clouds, including (1) the distribution of condition numbers, (2) the distribution of eigenvalues, and (3) the topological profiles before and after applying the reconditioning.

6. Applying SVD-Surgery and replacement on convolutional layer filters at initialisation and/ or during training either per epoch or mini-batches for training CNN models to control well-conditioning and the stability of filters from scratch and pretrained.

7. Integrating the tools of TDA as a research tool for understanding the performance of DL models of US image analysis, providing a more comprehensive understanding of the underlying structure in images feature maps. The link between these two fields lies in their complementary strengths. The gained

knowledge can also contribute to the current active research into the interpretability of CNN models for the analysis of medical image analysis.

1.5 Publications and preprints:

The following publications arose directly from the work on this thesis:

Publications

- J. Ghafuri, H. Du, and S. Jassim, "**Impact of Convolutional Layer Filters' Instability on Robustness of Classification Decisions for Tumour Diagnosis from Ultrasound Images**", 26 UK conference on Medical Image Understanding and Analysis, Cambridge, UK, 2022. `https://www.miua2022.com/`
- J. Ghafuri, H. Du, and S. Jassim,"**Sensitivity and stability of pretrained CNN filters**". In Multimodal Image Exploitation and Learning 2021 (Vol. 11734, p. 117340B). International Society for Optics and Photonics. `https://doi.org/10.1117/12.2589521`
- J. Ghafuri, H. Du, and S. Jassim, "**Topological aspects of CNN convolution layers for medical image analysis**", ECCV 2020, Women in Computer Vision Workshop. `https://sites.google.com/view/wicvworkshop-eccv2020/home`
- J. Ghafuri, H. Du, and S. Jassim, "**Topological aspects of CNN convolution layers for medical image analysis**". In Mobile Multimedia/Image Processing, Security, and Applications 2020 (Vol. 11399, p. 113990X). International Society for Optics and Photonics. `https://doi.org/10.1117/12.2567476`

Preprints

- J. Ghafuri, and S. Jassim, "**Singular value decomposition based matrix surgery**", arXiv:2302.11446. `https://arxiv.org/abs/2302.11446`
- J. Ghafuri and S. Jassim, "Filter surgery and replacement on the state-of-the-art DCNN for small datasets", (under preparation)

Technical report

J. Ghafuri, H. Du, and S. Jassim, "Extremely slim robust CNN for ultrasound tumour scan image classification", Feb 2022. Reported to Ten-D AI Medical Technologies Ltd.

1.6 Thesis Outline

The rest of the thesis is organised as follow:

- **Chapter 2: Background**, covers preliminaries, general concepts, and mathematical background of Convolutional Neural Networks (CNN) and Topological Data analysis (TDA) that are essential for the rest of the chapters.

- **Chapter 3: Effects of Convolutions on US Images**, we show our investigation on convolutions/kernels/filters on ultrasound images bladder, liver, and breast scans in terms of the amount of spatial and textural information content and their distribution, the spatial distribution of texture features using TDA, and the algebraic properties of filters and their impact on feature maps through the TDA and spectral analysis lenses.

- **Chapter 4: CNN Models for US Image Analysis**, we present the performance of various state-of-the- art CNNs in transfer learning mode for classifying malignant and benign lesions from ultrasound scans and their ability to the tolerable data perturbation and generalisation to unseen data. We then investigate the convolutional layers to understand the impact of pre-trained CL filters in the CNN settings on ultrasound image information content as well as on the algebraic and topological properties of the convolution filters pre and post training, based on [19]. In addition, we explore the possibility to reduce the complexity of CNN models and keeping the relatively well-conditioned CL filters via filter pruning.

- **Chapter 5: Towards Slim, Robust and Generalisable CNNs for US Scans**, based on the work of the knowledge gained in Chapters 3 and 4, we present our customised CNN models in terms of specific requirements such as the number of convolutional layers, the number of filters, and weight initialisation techniques. We introduce two new approaches to initialise the wights guided by their condition number and PH spatial distribution. We show our investigation and findings on benchmark datasets such as Digits, MNIST, and CIFAR-10 as well as breast ultrasound scan datasets, based on [23,24]. In addition, we monitor the instability of the condition numbers during the training, and test CNN models in terms of their robustness against tolerable data perturbations and ability to generalise to unseen data.

- **Chapter 6: Stabilising Filters Conditioning by Matrix Surgery**, we introduce an innovative approach based on Singular Value Decomposition called

SVD matrix surgery to control and/or reduce the condition number of convolutional filters at initialisation/pretrained as well as during the training process. We then present our findings on the effect of the SVD-Surgery in large filter sets in terms of their distribution of condition numbers, distribution of singular values, distribution of eigenvalues, and their topological profiles, based on [25]. We showcase the application of matrix surgery in the context of CNNs before the training as well as throughout the training process.

- **Chapter 7: Conclusion and Future Directions**, we conclude this thesis, provide a summery of key findings, and reporting on possible directions of follow-up research.

Chapter 2

Background

This thesis objectives involve a variety of different mathematical and computational concepts that are fundamental to different areas of deep learning, topological data analysis, numerical linear algebra, and spectral analysis that are needed to deal with the peculiarity of ultrasound image analysis in comparison to other image modalities in relation nature of textural/structural content distribution.

We introduce the deep learning concepts in computer vision for medical image classification using convolutional neural networks in Section 2.1. In Section 2.2, we describe the algebraic topology preliminaries and its application in data analysis along with point cloud settings for images and convolutional layer filters. We then discuss aspects of the spectral analysis of matrices in Section 2.3. The depth to which these sections delve into their topics is by no mean extensive, but sufficient for readers of diverse level of knowledge in each of these area to follow the rest of the thesis with reasonable ease. Readers familiar with the background to any/all these topics may skip parts/all this chapter.

2.1 Deep learning for computer vision

Traditionally, computer vision tasks relied on quantifying structural measurements of objects of interest in images and/or extracting texture features, that are deemed relevant to the domain/requirements investigated tasks, to design computer aided tool. These approaches eventually developed into what is referred to, in the literature, by handcrafted features based Machine Learning (ML) that work by following traditional procedures adopted in pattern recognition and forensic analysis. All the handcrafted schemes involve the choice of a classifier applied on the extracted feature. Significant progress was made in a variety of computer vision tasks by automating the components of such ML algorithms. However, success relied on several factors including the choice of textures assumed to be characterising the sought after image analysis task and invariant to certain changes in image recording conditions. It is also important to note that, for the most part, classifiers make decisions according to the rather stringent mathematical concept of ***Equivalent Classes***, although attempts are often made to introduce some flexibility or associate a confidence level with the predicted decisions.

Neural networks emerged at the early stages of computer science as a strong and reliable contender as a classifier in such tasks. Their success and capabilities, however, only emerged as a result of incorporating backpropagation more recently. Together, with the success of the Neocognitron, in the early 1980s by K Fukushima [26], energised research in AI with focus on mimicking the way the human brain learn and work. In fact, the structure of the Neocognitron was inspired by that of the visual nervous system of vertebrates.

Towards the end of 1980s, a period of pioneering work began with the work of Yann LeCun [27], who was the first to coin the term Convolutional neural network, that culminated before the turn of the 21^{st} century in developing LeNet as the first CNN model used for character recognition within a real-time document recognition system. The model exploit the advantages of training multilayer Neural Networks with backpropagation algorithm for automatic ***gradient decent learning***. Since then a series of CNN models developed for a variety of computer vision applications, including AlexNet, and GoogleNet.

The idea behind building blocks of convolutional neural networks is inspired by the hierarchical organisation of the visual cortex in the human brain [32]. Hypothetically, CNNs mimic the brain's processing of visual information through multiple layers, including convolutional, down-sampling, and fully connected layers. The convolutional layer performs operations similar to receptive fields in the visual cortex, highlighting specific patterns, [28]. The aim is to capture key principles of visual processing for efficient pattern recognition in tasks such as image classification and object detection. Figure 2.1 illustrates the levels of identifying features recognised by human visual cortex in four stages starting by detecting the edges and lines, shapes, objects, and faces i.e. starting by broadly looking at the object then narrowing down the focus on details. Considering the work of typical CNN models of image analysis, one notice that the filters in the early convolution layers are typically small and designed to detect simple patterns such as edges, corners, and blobs. But the filters in deeper layers become more complex with ability to detect higher-level features such as objects structure and texture.

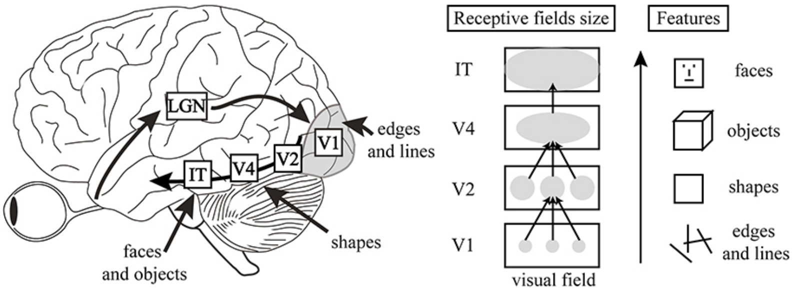


Figure 2.1: Illustration of human visual path, [28]

Though CNNs are becoming staggeringly powerful and successfully solving extremely complex computer vision challenges, they are still considered to be simplified models and far from acting as exact replicas of the brain. In the rest of this section, we introduce the main concepts of CNN architectures in Subsection 2.1.1 followed by detailed description of AlexNet in Subsection 2.1.2 as one of the typical state-of-the-art models. Subsection 2.1.3 describe the common training procedure. We close this section by discussing relevance to using CNN to this thesis objectives.

2.1.1 Convolutional neural network

Convolutional neural networks are designed to automatically learn features from images in an end-to-end process. It consists of two major parts namely feature extraction/learning and classification. The feature extraction part typically consists of multiple convolutional layers (CL) that may be complimented with pooling layers, while the classification part includes Fully Connected layers (FCL), see Figure 2.2 below.

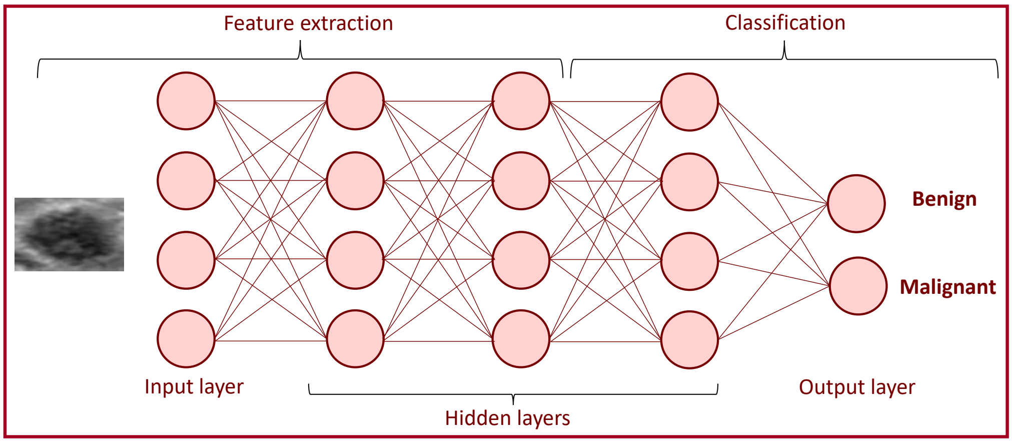


Figure 2.2: Convolutional neural network structure.

The convolutional layers extract features from input images using sets of multi-channel convolution filters (also called Kernels),that are initialised prior to training but updated during the training. The filters compute features from image patches, all over the image, formed weighted averaging patch pixels using the entries of the filter. The convolved images are often normalised and passed on to an ***activation function*** as a mean of endowing non-linearity and finally in many cases outputs are passed onto a ***pool*** Layer . Activation functions include Rectified Linear Unit (ReLU), sigmoid, and tanh, (for an extended activation function list see [21]). Pooling layers, subsamples the activated feature maps prior to passing to the next layer. The most widely used pooling operations include max-pooling and average-pooling. Max-pooling replaces data in windows of size 2×2 (or 3×3) with their maximum value.

The FCL connect neurons in one layer to neurons in the subsequent layer. These layers aggregate information from previous layers and perform classification or regression tasks. The weights of the FCL and convolution filters are learnt through the training process, the goal being to minimise the difference between the predicted output and the actual output (label). This is done by using a loss function such as cross-entropy loss. All weights are updated during the back-propagation step using an optimiser such as stochastic gradient descent (SGD) to minimise the loss. These layers are either followed by random dropout or transformation functions such as softmax. The FCL component could be used for a variety of tasks such as image classification, object detection, and image segmentation.

CNN feature learning layers serve dual purposes reducing spatial resolution of input data while deepening the learnt hidden pattern in features by producing multiple feature maps to be passed on to the next layer or to the FCL component. As a consequence of the multiple convolution layers, input images are represented by several low-dimensional maps each encapsulating different hidden patterns of features that are difficult to align with handcraft features. This provides significant advantages for use of CNN in computer vision tasks over handcrafted features. However, this comes at the expense of complexity of interpretation of CNN predictions.

Besides the plethora of image operations applied on input images, several parameters and hyperparameters are selected and deploy prior to training. The hyperparameters that characterise each CNN architecture and its complexity consist of:

1. ***Depth*** - Number of convolutional layers and fully connected layers involving

a large proportion of learnable parameters and affects the capacity to learn complex features.

2. ***Width*** - Number of filters (with multiple channels) of chosen size and overlapping and which determines the number of features or patterns that the CNN can learn. Filters size determines the receptive field of the CNN i.e. smaller filters capture local features, while larger filters capture more global features.

3. ***Pooling window*** size and operation that affects the spatial resolution and invariance of the model's representations.

Factors determining the expected application dependent outcome of CNN models include:

- ***Learning rate:*** controls the step size of the optimisation algorithm during training.

- ***Batch size and the number of epochs:*** determines the number of samples processed before updating the model's parameters, and how many times the entire training dataset is passed through the network. The right balance is application dependent, otherwise, the model might underfit or overfit to the training dataset.

- ***Regularisation:*** appropriate regularisation technique with its corresponding hyperparameters need to be selected based on the complexity of the task to reduce overfitting. The most widely used methods are L1 or L2 regularisation, dropout, or batch normalisation.

- ***Weight and bias initialisation*** techniques as well as the type of activation function.

These criteria are usually user selected based on the characteristics of the dataset, computational resources, and specific task requirements. Experimentation and iterative tuning are often necessary to find the optimal combination of parameters and hyperparameters for a given CNN model.

2.1.2 AlexNet architecture

AlexNet is the most successful deep convolutional neural network architecture representing a significant milestone leap after LeNet[1], and was developed by

[1] LeNet: the first model that successfully classified handwritten digits using CNNs, [27].

Krizhevsky et. al. in 2012, [29]. It was designed to compete in the ImageNet Large Scale Visual Recognition Challenge, which involved classifying images into one of 1000 object categories. The AlexNet architecture achieved a significant improvement in image classification performance, winning the competition with a top-5 error rate of 15.3%. The insights gained from AlexNet have also informed the development of other deep learning applications, such as natural language processing and reinforcement learning. The success of AlexNet has opened the door for the development of even more complex and powerful deep learning architectures, which continue to push the boundaries of artificial intelligence.

AlexNet was one of the first CNN architectures to incorporate several important features that are now commonly used in deep learning, including the use of ReLU activation functions, dropout regularisation, and data augmentation techniques. ReLU activation functions are used to introduce non-linearity into the network, which allows the model to learn more complex features. Dropout regularisation is used to prevent overfitting of the model by randomly dropping out neurons during training. Data augmentation techniques are used to artificially increase the size of the training dataset by applying random transformations to the input images to improve the generalisation performance of the model.

AlexNet consists of five convolutional layers and three fully connected layers as shown in 2.3. The first layer performs local contrast normalisation to normalise the input images and reduce the effects of lighting variations. The next five layers are convolutional layers, each followed by a max pooling layer, that reduces the dimensionality of the feature maps. Filter sizes for the first two convolutional layers are 11×11 and 5×5 and for other layers are 3×3. Bias values at first and third layers are initialised with zero and one for the rest. The fully connected layers perform the classification task.

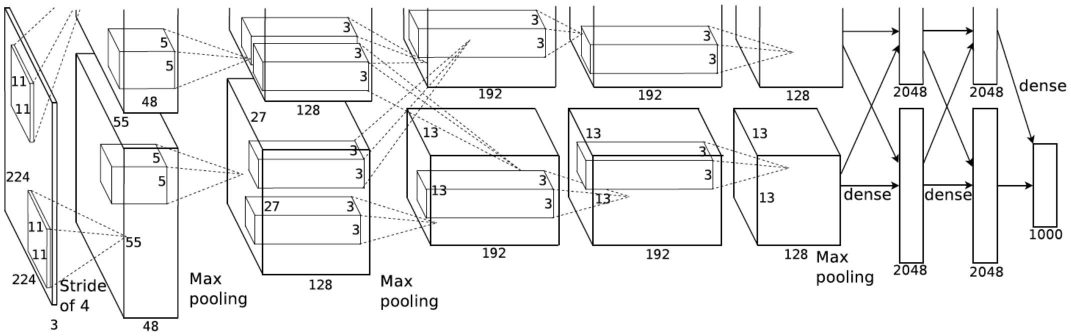


Figure 2.3: AlexNet architecture, [29].

2.1.3 Training and testing settings

The training setting of convolutional neural networks is far from a straight forward process as it depends on several factors, as mentioned in Subsection 2.1.1. There are several important training settings and/or rules to follow for effective and successful training based on theories, best practices, and empirical observations as follows:

- ***Image pre-processing and augmentation:*** Suitable pre-process technique of the input data before training depending on the image modalities including normalisation.

- ***Training-Validation-Testing CNNs:*** The application relevant dataset of available image samples is split into three subsets: a training set, a validation set, and a test set. The training set is used to update the model parameters, the validation set is used for hyperparameter tuning and model selection, while the test set is used for final evaluation. The test set should be kept separate and only used once to avoid biasing the evaluation.

- ***Early stopping criteria:*** A mechanism used to halt training to prevent overfitting, It is based on monitoring the performance on the validation set during training, and used if performance starts to deteriorate before reaching the designated number of epochs.

The number of layers in a Convolutional Neural Network (CNN) can have a significant impact on its performance in terms of accuracy, efficiency, and robustness. Deeper networks tend to have more capacity to learn complex features and patterns in the data, particularly with skip connections circumnavigate singularity problems to retain information from earlier layers, which can lead to improved accuracy on the training data and generalisation, [30, 31]. However, deeper networks can also be more prone to overfitting, especially when the dataset is small, leading to poor generalisation performance on new data. In addition, deeper networks may require more computational resources to train and run, i.e. less efficient [32]. On the other hand, shallower networks may have less capacity to learn complex patterns, yielding lower accuracy on the training data. However, shallower networks may generalise better to unseen data.

A trade-off between accuracy, robustness and efficiency, [33], is often based on employing techniques such as skip connections, batch normalisation, and residual connections to facilitate the flow of information across different layers and reduce the vanishing gradient problem and avoid overfitting, [30, 31]. In addition, model compression techniques such as pruning, quantisation, and low-rank factorisation [34–36] can be used to reduce the number of parameters and operations in the network, while maintaining or improving its performance. The number of convolutional layers in a CNN can impact the representational power of the network, which refers to the ability of the network to learn complex features and patterns in the data, [37]. A deeper network with more convolutional layers can have more capacity to learn complex representations, but it can also be more prone to overfitting.

The need of large datasets is mainly considered that the more training samples would make the CNN models recognise the shapes in the same way when humans are exposed to an unlimited amount of data and start recognising objects at early stages of life. Understanding and handling natural images is not as difficult as medical imaging as it requires the specialists and years of training and dedication in the domain. Though, collecting and processing medical data to train CNN models is a challenging task due to the accessibility, quality, prior medical history, the nature of the problem such as the difficulty of identifying the internal abnormalities when occur at early stages, lack of data inclusiveness such age, ethnicity, gender, geographical area.

One of the outstanding properties on CNN models is its transferability to image tasks/problems different from the original purpose of their development. Transfer learning modes (or fine-tuning) of deep learning models is based on retraining a pretrained CNN model on a small size dataset of images that differs in modality and/or in content from the original training dataset. The aim is to adapt an existing pretrained model to enable using it for the analysis of the new dataset. The pretrained parameters of existing CNN models, usually determined by training on large datasets of natural images, are used as initialising parameters for retraining on the new task. In fact, it has become a common approach for developing CNN models for the analysis of ultrasound or other radiology image datasets.

2.1.4 Relevance of CNN to thesis objectives

Having reviewed various aspects of the recent advances in theory and practices of CNN models for image analysis and computer vision applications, we realise the challenges in our work include scarcity of US tumour scan images and the differences between US image tumour-relevant features and those learnt by state-of-the-art CNN models pretrained on natural images. Hence, the starting point in investigating the use of deep learning for analysis of US tumour images, must rely on the concept of transfer learning[2]. In this respect, we realise that there are several state-of-the-art (SOTA) CNN architectures that have achieved excellent performance on a variety of computer vision tasks. These architectures include, but are not limited to, AlexNet [29], VGGNet [38], Residual Network [39], the Inception architecture [37], EfficientNet [40], DenseNet [41], . . . , etc. Beside the manual design of CNN architectures, the authors in [42–44] developed Efficient Neural Architecture Search (ENAS) to automatically design CNN architecture for natural images and for breast ultrasound image in [45].

In order to conduct the research project of this thesis, we need to gain a reasonably comprehensive investigations into the main factors that contribute to learning by a chosen state-of-the-art pretrained CNN model during retraining on US datasets. It is crucial to conduct a thorough analysis of the individual convolutional layers within the network and this encompasses a range of elements within the layer, including convolution filters/kernels, convolved images or feature maps, the presence or absence of bias, the impact of the activation function, normalisation, and down-sampling. In particular, our analysis of the convolution filters must link their mathematical properties to their effect during the training. Our focus is on the robustness and generalisation properties of the CNNs for classifying malignant and benign tumours from ultrasound images rather than limiting ourself to gaining the optimal classification performance on the provided dataset.

2.2 Topological Data Analysis

Underlying any data and image analysis task is a set of data/image samples from a larger population and a notion of similarity considered as a distance function. The spatial distribution of such a dataset, in its domain, endows it with a notion of a shape (topological space) the neighbourhood system of which is determined by the similarity metric. To understand/analyse certain behaviour and properties of data in terms of their topological shape, it is essential to determine the spatial distribution data records and the topological invariants of their shapes, [46]. Traditional data ***clustering*** algorithms builds on the knowledge of data shape and similarity/distance function to identify and study the properties of significant

[2] Transfer learning in deep learning repurposes pretrained models, leveraging knowledge from one task to enhance performance on related tasks with limited data. Strategies include feature extraction and fine-tuning. It is beneficial in scenarios where training models from scratch is computationally expensive or lacks sufficient labeled data, offering efficiency and improved performance in various applications, such as computer vision and natural language processing.

subsets. Topological data analysis (TDA) is an innovative paradigm for analysis of structured/unstructured data, which goes well beyond basic clustering, by associating with the data records a nested sequence of shapes in terms of similarity/proximity information. Unlike TDA, clustering algorithms concentrate on identifying dataset connected components but do not go beyond basic properties of clusters size and density without considering the components full topological profile.

TDA emerged at the turn of this century as natural advancements in the fields of computational geometry and topology. It builds on exploits the tools and concepts of algebraic topology that were rigorously formulated and developed over a long period of time since the middle of the 19^{th} century. Algebraic topology emerged as a mean of characterising topological spaces in terms of different connectivity parameters using tools of linear algebra and group theory. These parameters are referred to as topological invariants, due to being invariant under continuous deformation of the topology, and form the terms of what is known as the ***Euler Characteristics*** of the space. TDA is suitable for analysis of high dimensional and noisy datasets, and is able to uncover shape/object characteristics that may change by intended/unintended distortions.

The most common relevant form of topological spaces are polyhedral shapes that are constructed by gluing together basic building blocks, of well-known shapes such as simplices/cells/spheres, according to certain conditions, see Figure 2.4.

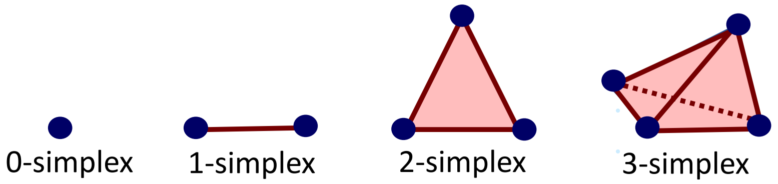


Figure 2.4: Simplices of different dimensions.

TDA approach to the analysis of datasets (referred to as point clouds) is based on gradually constructing (via ***triangulation***) nested sequence of ***simplicial complexes*** by connecting pairs of data points according to an increasing distance/similarity sequence of thresholds. This procedure determines the topological profile of the data in terms of the well-researched algebraic topology tool of ***Homology*** for each of the constructed nested shapes from which one can extract persistency information. ***Persistent homology*** is an algebraically computable features of topological invariants of shapes/functions at multiple distances or similarity resolutions.

In this section, we briefly introduce homology preliminaries in Subsection

2.2.1 then persistent homology in Subsection 2.2.2. We then describe the point cloud settings to compute PH in Subsection 2.2.3. In Subsection 2.2.4, we briefly describe the Mapper algorithm to visualise high dimensional data in low dimensions.

2.2.1 Homology preliminaries

Homology is a *functor* from the category of topological spaces into that of exact sequences of Abelian groups. It helps turning difficult analytical problems in topology into algebraic problems on Abelian groups that more susceptible to numerical solution by computers. For topological space X, this functor defines an ***exact sequence*** $\{H_k(X)\}$ of finitely generated abelian groups and for any continuous function $f : X \to Y$ the functor associates a sequence $H_k(f)$ of homomorphism:

$$H_k(f) : H_k(X) \to H_k(Y)$$

satisfying some conditions on homomorphism composition. The rank of the abelian group $H_k(X)$, i.e. the number of its generators, is known as the k-th Betti counting the number of k-dimensional holes (connectivity descriptors) in X.

This functor is well understood and is easier to compute for simplicial complexes, and is referred to ***singular/simplicial homology***, which is suitable for use in TDA analysis of point clouds, [47–49]. For simplicity, we adopt the ***Vietoris–Rips*** approach to constructing simplicial complex from point clouds instead of ***alpha-complexes***, [50] and Delaunay-Čech complexes, [51]. Below is a formal definition and illustration of the Vietoris–Rips complex for point clouds, with an example shown in Figure 2.5.

Definition (Vietoris-Rips complex): Given a collection of points $X = \{x_\alpha\}$ in the Euclidean space $\mathbb{E}^n$. For each distance threshold t, let $\mathbb{S}_t$ be the Rips simplicial complex whose k-simplices correspond to unordered $(k+1)$-tuples of points $\{x_\alpha\}_0^k$ in X that are pairwise within distance $\leq 2\epsilon$ of each other.

For a point cloud X, and let $\mathbb{S}_t$ be the Rips simplicial complex constructed at distance threshold t. For each integer $k \geq 0$, the k-th chain complex $\mathcal{C}_k(\mathbb{S}_t)$ is defined as the finite dimensional vector space, over the Boolean field $\mathbb{Z}_2$, freely generated by the k-simplices $\sigma = [v_0, v_1, \ldots, v_k]$ in $\mathbb{S}_t$. Vectors in $\mathcal{C}_k(\mathbb{S}_t)$, called k-chains, are linear combinations of k-simplices with binary coefficients, i.e.

$$\sum a_i\sigma_i \quad \text{where} \quad a_i \in \{0, 1\} \quad \text{and} \quad \dim a_i = k$$

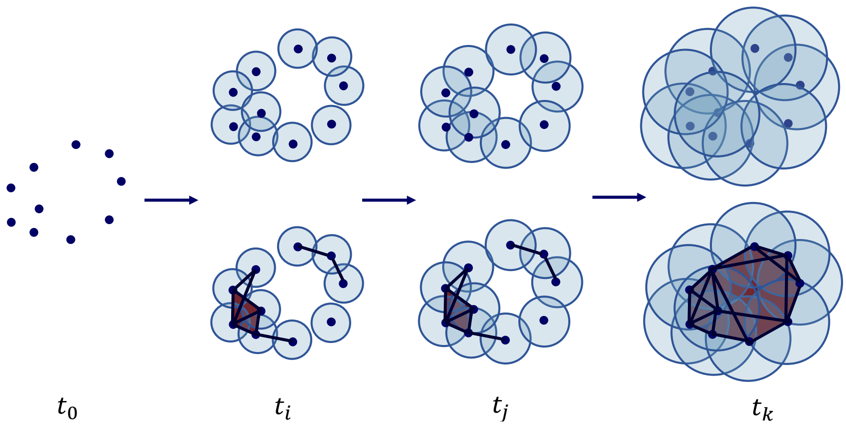


Figure 2.5: The construction of the Ribs simplicial complex of a point cloud.

For each integer $k \geq 0$, define the boundary operator

$$\partial_k : C_k(\mathbb{S}_t) \to C_{k-1}(\mathbb{S}_t)$$

$$\partial_k(\sigma = [v_0, v_1, \ldots, v_k]) = \sum_{i=0}^{k} [v_0, \ldots, \hat{v}_i, \ldots, v_k] \tag{2.1}$$

where the 'hat' symbol ˆ over v_i indicates that the vertex together with all sub-simplices of σ that intersect at v_i are removed. In other words, boundary operator ∂_k maps each k-simplex into its bounding chain. For each k, one can establish the composition relation $\partial_k \circ \partial_{k+1} = 0$. In homology terms, this means that we have an ***exact sequence*** of chain complexes:

$$\cdots \to C_{k+1}(\mathbb{S}_t) \xrightarrow{\partial_{k+1}} C_k(\mathbb{S}_t) \xrightarrow{\partial_k} C_{k-1}(\mathbb{S}_t) \xrightarrow{\partial_{k-1}} \ldots \xrightarrow{\partial_2} C_1(\mathbb{S}_t) \xrightarrow{\partial_1} C_0(\mathbb{S}_t) \tag{2.2}$$

Clearly, image of ∂_{k+1} is a subspace of kernel of ∂_k, and hence we define the k-th homology group of $\mathbb{S}_t$ by the quotient vector space:

$$H_k(\mathbb{S}_t) = ker(\partial_k)/\mathrm{im}(\partial_{k+1}) \tag{2.3}$$

This means that $H_k(\mathbb{S}_t)$ quantifies the number of k-dimensional subspaces of $\mathbb{S}_t$ that have no boundary in it and do not bound any $(k+1)$-dimensional subspace in $\mathbb{S}_t$. Note that, the length of the above chain complex sequence is $\leq dim(\mathbb{S}_t)$. The nth dimensional Betti number β_k is

$$\beta_k = \dim(H_k(\mathbb{S}_t)) = \dim(\ker \partial_k) - \dim(\mathrm{im}\, \partial_{k+1}) \tag{2.4}$$

It is worth noting that the Betti numbers of low dimensions are referred to as holes: β_0 = # connected component, β_0 = # 1-dim holes, and β_2 = # voids (2-dim holes). In Figure 2.6, we display manifolds of dimensions 0, 1 and 2 with their Betti numbers profile.

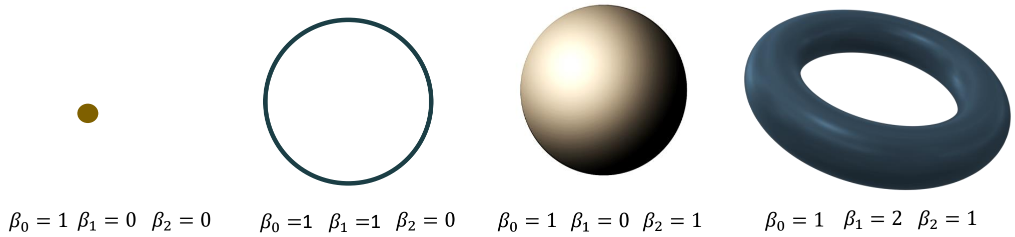

$\beta_0 = 1$ $\beta_1 = 0$ $\beta_2 = 0$
$\beta_0 =1$ $\beta_1 =1$ $\beta_2 = 0$
$\beta_0 = 1$ $\beta_1 = 0$ $\beta_2 = 1$
$\beta_0 = 1$ $\beta_1 = 2$ $\beta_2 = 1$

Figure 2.6: Betti numbers Profile of manifolds of different dimensions.

2.2.2 Persistent homology and filtration

Persistent homology (PH) is a computational tool of TDA that encapsulates the spatial distribution of point clouds of data records, sampled from metric spaces, by recording the topological features of the nested triangulated shapes, described above, by connecting pairs of data points according to an increasing distance/similarity sequence of thresholds. The main idea behind persistent homology is to capture the topological features (connected components, loops, voids, etc) of the shape of a point cloud at multiple scales, rather than at a single scale analysis, [46–49]. It extends the concept of singular homology features by considering the nested sequence of Ribs-complexes formed for an increasing sequence of distance thresholds, called a ***filtration*** of the final simplicial complex of the point cloud. The persistent homology of a point cloud is then defined as the sequence of topological invariants (such as connected components, loops, and voids) of the nested Ribs-simplicial complexes formed over the filtration.

In general, *a filtration of any simplicial complex X is a sequence $\{X_i\}$ of nested simplicial complexes each embedded as a subcomplex of the next one and together they cover X*, i.e.

$$\emptyset = X_0 \subseteq X_1 \subseteq \cdots \subseteq X_{end} = X \tag{2.5}$$

For each i, the PH tool traces the evolution of the i-dim homology groups Betti numbers over the subsequent nested Ribs-complexes in the filtration. It tracks the birth (forming) and death (merging) of the topological features as the filtration progresses and records its persistency as the length of its life span over the filtration. For each topological dimension, it is customary to visualise the corresponding persistency information by a stack of barcodes, called the Persistent Barcode (PB), or a multiset of pints in the first quadrant of the real plane $\mathbb{R}^2$ on/above the diagonal line $y = x$, called the ***Persistence Diagram*** (PD), of the point Cloud, [47,48]. The length of each barcode in PB(X) equals $(death - birth)$ of the corresponding topological feature, while each point in PD(X) represents the pair $(x = birth, y = death)$ of a topological feature. Both PB and PD provide ways of comparing the topological profile of different point clouds.

2.2.2.1 Persistent homology of a torus with/without noise

To illustrate the visual representations of PH representation and the computing process, we first create a point cloud of 1500 points sampled randomly on the surface of the Torus:

$$T = \{(x, y, z) \in \mathbb{R}^3 : (\sqrt{x^2 + y^2} - a)^2 + z^2 = b^2\} \tag{2.6}$$

then we add noise to approximately 15% of the sampled points on the torus. Figure 2.7, below, displays both point cloud.

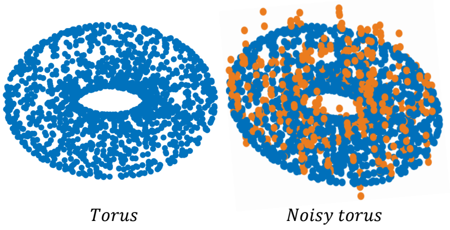


Figure 2.7: Clean and noisy point clouds on tori.

Steps of the Rips-complex construction from the above point clouds on tori is shown in Figure 2.8, below, then connecting nearby points with distance thresholds $\epsilon_i = 0.1, 0.2, \ldots, 0.6$.

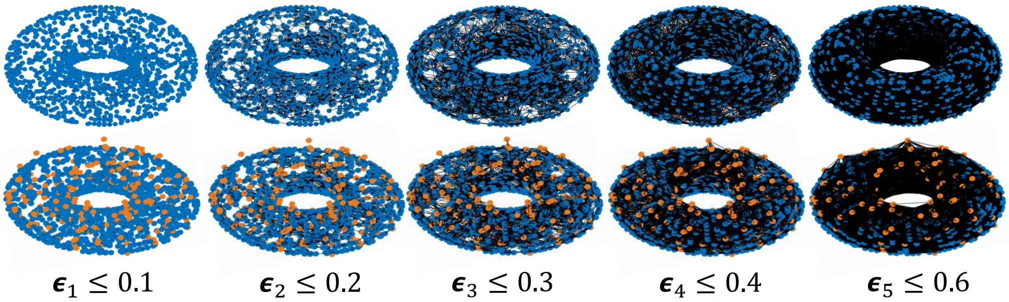
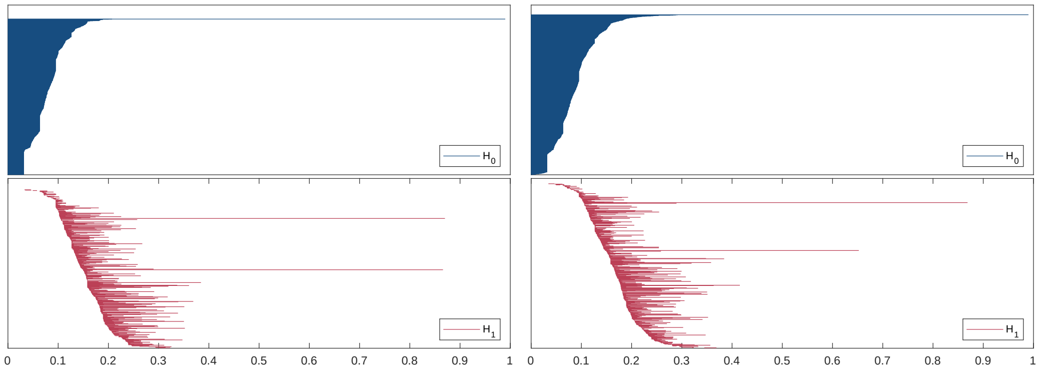


Figure 2.8: Steps of Rips-complex of point cloud on tori.

Figures 2.9 shows the topological representation PBs of the point clouds for both dimensions 0 and 1. In 0-dim, little/no effect can be visible between the two PBs. In 1-dim, the two long persisting barcodes represent the two empty discs (holes) whose cartesian product generates the torus. The persistency lengths of these two holes depend on the radii $(a - b, b)$ of the generating circles. In this case, $a = 2b$ and the two longer persistent holes (dim 1) are nearly equal. Noisy sampling will mostly have visible effects on the shorter barcodes by slightly increasing their persistent lengths further.

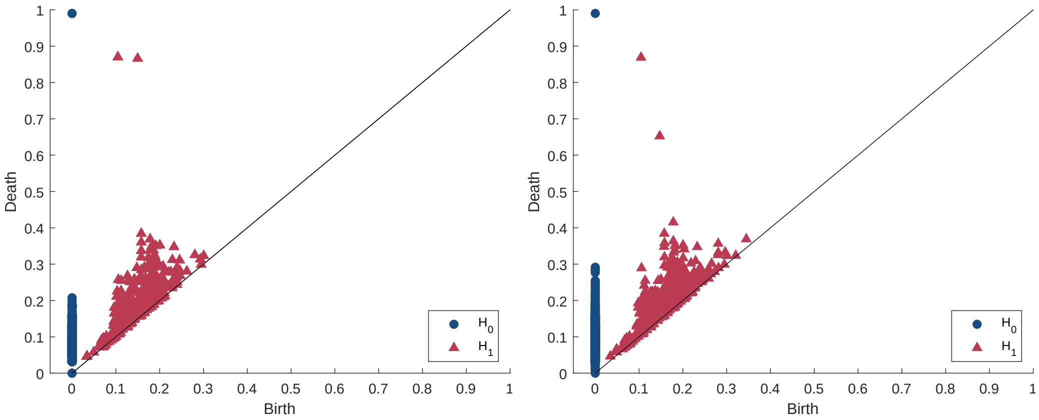


(a) Torus

(b) Noisy torus

Figure 2.9: PBs, in H_0 and H_1, for the clean (a) and the noisy (b) point clouds.

Figures 2.10, below, shows the topological representation PDs of both point clouds for both dimensions 0 and 1. There are visible differences between the PDs in both dimensions, as a result of the noise, indicating effects of noise on topological profiles of point clouds. Specifically, the presence of noise appears to shorten the time spans of topological features such as holes while simultaneously impeding the process of merging connected components.



(a) Torus

(b) Noisy torus

Figure 2.10: Topological representation of a torus using PD in H_0 and H_1.

2.2.3 Setting point clouds for PH computation

Setting the point clouds is essential to compute their topological representations, so far assumed the points are vectors in $\mathbb{R}^n$. In this thesis, we deal with different datasets of matrices representing convolution filters and their inverses. We also deal with datasets of images mostly, but not entirely, US. Point clouds of graphs and networks are another type of data structures, for which PH computation are relevant to Deep Learning and AI research.

2.2.3.1 Point clouds setting from datasets of matrices

Let A be a set of randomly generated real $n \times n$ matrices based on a Gaussian distribution, where each entry A in defines a point in $\mathbb{R}^{n \times n}$. To maintain uniformity, we normalise all matrices $A/\|A\|$ to reside on the $(N-1)$-dimensional spheres, denoted as $\mathbb{S}^{N-1}$, where $N = n \times n$. We construct a second point cloud, B, comprising the inverse matrices normalised as $A^{-1}/\|A^{-1}\| \in \mathbb{S}^{N-1}$. For each of these point clouds, we compute the pairwise distance matrix to construct the Vietoris–Rips simplicial complex. Throughout this thesis, our computations were carried out using the L_2-norm , and we explore various representations of their topological profiles.

2.2.3.2 A cubical set approach for image analysis

This thesis being concerned with image analysis, we have to deal with considering their representation that provides natural input to PH computation procedures. Although, in computers, images are represented by matrices of their pixel values, the above setting is of any interest. On the other hand, vector representation of images by flattening their rows or columns are occasionally used to compute PH of such point clouds. But this approach ignores local image features.

The sublevel set of a function, defined on a space, is a concept used in persistent homology [49, 52, 53]. A grayscale image can be abstractly represented as a real-valued function $f : \mathbb{R}^2 \to \mathbb{R}$, where each point in the 2D plane corresponds to a pixel, and the function value at that point signifies the pixel's intensity or brightness. The sublevel set pertaining to a given threshold t is defined as the set of all points in the image for which the function values are less than or equal to t. Formally, this sublevel set, denoted as S_t, is expressed as:

$$S_t = \{(x, y) \in \mathbb{R}^2 : f(x, y) \leq t\} \tag{2.7}$$

In the context of image analysis, the sublevel set represents the regions of the image that are darker than or equal to the threshold value. Considering the image as a 3D space, S_t represents the subset of the image bounded above by the intersection of the image with the $z = t$ plane. An increasing sequence of thresholds $\{0 = t_0 < t_1 < t_2 < \ldots < t_k = 256\}$ defines a filtration of the image spaces, are used to define PH representation of images. By varying the threshold value, we obtain a hierarchy of sublevel sets, each capturing different levels of topological profiles of the image. In Chapter 3, we shall describe, the LBP texture Landmark based PH of images developed at the University of Buckingham, [54]. This approach provides multiple topological representations of image local information defined in terms of different sets of landmarks.

2.2.4 Mapper algorithm

Mapper algorithm is a standard algorithm and software for implementing and visualising TDA introduced by Singh et al. [55] in 2007. Mapper differs from the tools of visualising PH information provided by persistence diagrams or barcodes. It is used to summarise vital information about datasets in high-dimensions, by partially clustering and visualising the outcome in low dimension using multidimensional scaling (MDS), using different types of lenses. In other words, it is a TDA data visualisation tool that reflects similarity, while reducing dimensionality.

The most highlighted characteristics of Mapper, include being coordinate free, invariant under "small" deformations, and its compressed representations of shapes. The overall aim of this algorithm was to have a tool that can be used for simplification, qualitative analysis, and visualise high dimensional data sets.

The main inputs of Mapper algorithm are point cloud samples, filter functions (also known as lenses), covering of a metric space, clustering algorithm, and other parameters. While the output of Mapper tool is simplicial complex (or graph) that represents the topological aspects of the point cloud. There are many different parameter, hyper-parameter, distance function, clustering algorithm and lenses in Mapper algorithm to be chosen, [56].

Given a high dimensional point cloud $\mathcal{X}$, the steps behind Mapper algorithm [55–57] to construct the graphical representation of $\mathcal{X}$ are as follows:

- *Filter function* f: also known as *lens* to map $\mathcal{X}$ to a lower dimension i.e. $f : \mathcal{X} \to \mathbb{R}, d \geq 1$.

- *Cover U*: to construct a cover $(U_i)_{i \in I}$ of the projected space typically in the form of a set of overlapping intervals which are also knows as bins.

- *Clustering algorithm:* to cluster the points for each interval U_i in the preimage $f^{-1}(U_i)$ into sets $\{V_j^i\}_{j=1}^k$, where V_j is the number of vertices in each bin.

- *Construct the graph:* cluster sets are connected if there are some points in common.

There are many choices to select the lens, covers, and clustering algorithm as these could be problem/data dependent. However, the common choices for the lenses are projection onto one or two dimensions via one method or combined together such as Principal Component Analysis (PCA), isolation forest, L_2-norm, see [55, 58] for further detail. Figure 2.11 illustrates a sampled data from a noisy circle in $\mathbb{R}^2$, with 5 intervals for each set with length 1. There are common points shared between the clusters (nodes) due to the 20% overlapping, therefore the nearby nodes are connected.

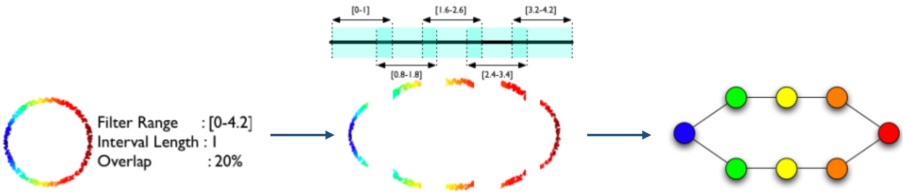


Figure 2.11: Graphical illustration of point cloud using Mapper algorithm, [55].

2.3 Spectral analysis of matrices

CNN models use large numbers of convolution filters that are mostly initialised as Random Gaussian matrices of relatively small sizes, but their entries are updated frequently during training in order to fit the model performance to that of the training dataset. In such situation, it is essential to maintain certain algebraic as well as properties that avoid them getting them nearer to be non-invertible. The condition number[3] of a matrix is a measure of how sensitive it is to changes in its input, and CNN models with high conditioning convolution filters are more susceptible to instabilities during training. In particular, highly ill-conditioned filters can lead to vanishing or exploding gradients, which can cause the model to converge slowly or not at all.

[3] The condition number of a matrix was first introduced by A. Turing in [59].

The condition number $\kappa(A)$ of a square $n \times n$ matrix A, considered as a linear transformation $\mathbb{R}^{n\times n} \rightarrow \mathbb{R}$, measures the sensitivity of computing its action to perturbations to input data in their domains of action and round-off errors. For an arbitrary vector norm $\|\cdot\|$, it is defined as $sup\|Ax\|/\|x\|$ over the set of nonzero x. It depends on how much the calculation of its inverse suffers from underflow (i.e. how far $det(A)$ is from 0).

A stable action of a matrix/filter A implies that small changes in the input data are not expected to result in significant changes in the output. The extent of these changes is bounded by the reciprocal of the condition number. Therefore, the higher the condition number of A, the more unstable its action becomes to small data perturbations, and matrices with high condition numbers are referred to as ill-conditioned.

CNN layers consist of large numbers of multi-channel convolution filters, and it is necessary to control the growth of the condition numbers to achieve reasonable stability in model performance and robustness against data perturbation and adversarial attacks. Indeed, the distribution of condition numbers of a random matrix simply describes the loss in precision, in terms of the number of digits, as well as the speed of convergence due to ill-conditioning when solving linear systems of equations iteratively, [60]. The condition number of matrices and numerical problems was comprehensively investigated in [59–66]
Definition (Condition number): Suppose matrix $A \in \mathbb{R}^{n\times n}$ is a non-singular square and the condition number, κ of A is defined as:

$$\kappa(A) = \|A\|\|A^{-1}\| \tag{2.8}$$

Where $\|\cdot\|$ is the norm of the matrix. Note that for the Euclidean norm (L_2-norm) where $\|\cdot\| = \|\cdot\|_2$, then $\|A\|_2 = \sigma_1$ and $\|A^{-1}\|_2 = 1/\sigma_n$. Thus:

$$\kappa(A) = \sigma_1/\sigma_n \tag{2.9}$$

where σ_1 and σ_n are the largest and smallest singular values of the singular value decomposition of A, respectively.

A matrix is said to be ill-conditioned if any small change in the input results in big changes in the output, and it is said to be well-conditioned if any small change in the input results in a relatively small change in the output [18]. Alternatively, a matrix with a low condition number (close to one) is said to be well-conditioned, while a matrix with a high condition number is said to be ill-conditioned and the ideal condition number of an orthogonal matrix is one[4].

The most common efficient and stable way of computing $\kappa(A)$, is by computing its Singular Value decomposition (SVD) from which $\kappa(A)$ is calculated as the ratio of A's largest singular value to its smallest non-zero one, [67].

J. W. Demmel, in [64], investigated the upper and lower bounds of the probability distribution of condition numbers of random matrices and showed that the sets of ill-posed problems including matrix inversion, eigenproblems, and polynomial zero finding all have a common algebraic and geometric structure. In particular, Demmel showed that in the case of matrix inversion, the further away a matrix is from the set of non-=invertible matrices, the smaller is its condition number. Accordingly, the spatial distribution of random matrices, in their domains, are indicators of the distribution of their condition numbers.

These results provide clear evidence of the viability of our approach to exploit the tools of topological data analysis (TDA) to investigate the condition number stability of point clouds of random matrices and convolution filters in particular.

In this thesis, we shall first determine the level of instability and fluctuations in the conditioning numbers of convolution filters in different layers and investigate the means of controlling their instabilities. Many approaches have been proposed to impose orthogonality conditions, but this may result in low performance on the training data as a result of underfitting and stifling learning. We shall discuss several approaches, including an SVD based matrix manipulation, that controls the instability of condition numbers in a reasonable manner without imposing rigid orthogonality criteria.

In this chapter, we presented brief introductions to both tops to provide the necessary background for the investigations conducted in the rest of the thesis for designing CNN models for the analysis of US tumour scan images. There are three main challenges: (1) interpreting CNN decisions, (2) maintaining robustness against data perturbation, and (3) ability to generalise to unseen image datasets. We shall use TDA together with spectral analysis to investigate properties of point clouds of convolution filters as a potential source of overfitting and lack of robustness. We also use the combined knowledge of topology and spectral analysis of matrices to develop regularisation schemes to reduces these two challenges.

[4] The acceptable range for the condition number depends on the specific application. For instance, in the context of CNN convolution filters, a matrix is considered well-conditioned when κ is small, typically in the range of $1 \leq \kappa < 10^2$, and *ill-conditioned* if κ is large (e.g. $\kappa \geq 10^2$).

Chapter 3

Effects of Convolutions on US Images

Convolution of an image in the spatial domain refers to the process of combining the image patch entries with a kernel of the same size by calculating their inner product (i.e. summing up the entry-wise multiplications) and sliding the kernel over the image. The architecture of Deep Learning CNN models consists of sufficiently large sets of convolution filters that form the main part of extracting feature maps of input data samples prior to feeding into the Neural network layers for decision making. The entries of these filters are modified through an elaborate iterative process of training on a sufficiently large samples of images (in batches) for training, related to the investigated application, using the backpropagation procedure that also modifies the neural network parameters by controlling the growth of the gradient descent while optimally fitting the training image set. The convolution operation is designed to highlight/suppress certain features in an image such as edges, textures, noise, and patterns depending on the type and properties of the adopted kernel. It is therefore essential for this thesis research to have an understanding of the effects of the various types of convolutions on as many as possible image descriptors and attempt to link these effects with computationally known descriptors of the deployed kernels.

In this chapter, we investigate the impact of random Gaussian filters/kernels on extracted features in the spatial domain from bladder, liver, and breast ultrasound scans in terms of the amount of information, texture features and their spatial distribution. We use entropy to measure the amount of information before and after applying Gaussian kernels on ultrasound images to their related texture features. For extracting texture features, we apply Local Binary Pattern (LBP) to evaluate the local geometrical behaviour of the landmarks. We investigate the spatial distribution of these LBP landmarks by using persistent homology. We shall demonstrate the effect of well or ill-conditioned filters on the output of convolved ultrasound images.

3.1 Introduction

For the sake of self-containment, we give the definition of the Gaussian kernels, describe their generation, and illustrate the effects of different filters on an ultrasound image. We shall then describe the datasets of US scan images of different human tissues/organs that we deploy in some or all the experimental work in this chapter and thesis.

3.1.1 Gaussian filters in the spatial domain

Weights in neural networks, especially in convolutional neural networks (CNNs), are crucial parameters influencing the network's ability to capture intricate data patterns. Initialisation, typically achieved through zero-mean random Gaussian distributions, involves choices between layer-dependent/independent variances. Common methods include *Xavier/Glorot* [68], *He* [69] or less commonly used constant standard deviation of 0.01 to initialise weights in each layer [29]. The aim is to break symmetry, promote effective learning, and improve the convergence as well as the performance of the network during training, [70]. Due to exponentially vanishing/growing gradient of loss function and for compatibility with activation functions, *Glorot* and *He* weight initialisation techniques select variances per layer, that may depend on the number of in/out neurons or only the number of input neurons, respectively. In all these initialisation strategies, no explicit consideration is given to the conditioning of filters or their stability during training. Weight initialisation in medical imaging needs rethinking due to limited data availability, complex and varied image structures, class imbalance, and rare abnormalities. Furthermore, in these cases using CNN models pretrained with natural images in transfer learning modes, there is a need to take into account sensitivity and interpretability, [71] and to consider model convergence conditions during the retraining, [72]. In the next chapter, we shall develop/adapt specific weight initialisation for improved model performance, robustness, and interpretability in US image analysis. Here, we shall first describe the construction of Gaussian Kernels.

Gaussian filters in the spatial domain are one of the most well-known methods for detecting edges, blurring, smoothing, and extracting texture features from images, and it is widely used in image processing and computer vision tasks. It is a two-dimensional filter that is based on the Gaussian distribution that convolves the image with a Gaussian function i.e. it is the product of two such Gaussian functions at (x, y) distances from the origin :

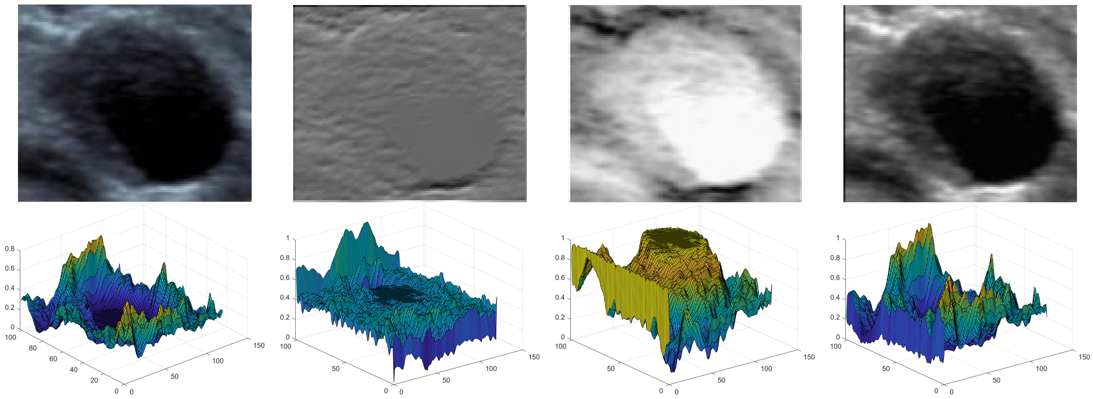
$$G(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x^2+y^2)/2\sigma^2} \tag{3.1}$$

where σ is the standard deviation of the Gaussian distribution, and it determines the spread of the filter. The choice of σ in He and Glorot weight initialisation techniques is dependent on the structure of building blocks of the layers and the number of neurons.

The result of the convolution operation is a new image with modified pixel values that represent texture features that the kernel is designed to extract. The effect of the convolution includes reducing the high frequency content and/or small variations i.e reducing noise, thus smoothing out the overall appearance of the image. The level of noise removal or blurring effect is partially bound to the choice of the standard deviation σ. A larger standard deviation results in a stronger blurring effect, while a smaller standard deviation results in a weaker blurring effect. Gaussian filters may cause artefacts such as ringing around edges due to the convolution operation or enhance less important features in the image.

Often a set of random filters are applied in deep learning architectures for extracting texture features. These sets are a variation of individual filters, where the values of the filter coefficients are generated randomly based on i.i.d. Gaussian or uniform distribution. The aim of using random Gaussian filters is to add a degree of randomness to the smoothing process and produce a set of various texture features which can help provide relevant information about the source input image. The settings as well as the effectiveness of such filters are depended on the specific applications and the characteristics of the image being processed.

The behaviour and smoothing properties of Gaussian kernels are well understood for natural images, hence our focus is on their effect on various ultrasound images. In Figure 3.1, we show examples of applying three different 3×3 Gaussian kernels w_i (i=1,2,3) on an ultrasound image and their effect on the texture features from the original image. The convolved image texture features are shown in two and three-dimensional landscapes to illustrate the changes as a result of convolutions. All three kernels are producing new convolved images that either preserve the general characteristics of the original image or preserve a certain level of the image characteristics. The standard deviation of the selected kernels is 0.01, however, their coefficients are different from each other. The convolved images obtained with w_1 and w_2 kernels seem to create a smoother version of the original images as shown in 2D and 3D visualisations, while w_3 may be seen as an enhanced version of the image. Furthermore, w_2 filter seems to produce a version is similar to the negative of the original image in terms of brightness, w_3 filter seems to act in a manner similar to histogram equalisation, while w_1 filter seems to compress the pixel values into a narrow Gray-level range.



(a) Image G (b) $G * w_1$ (c) $G * w_2$ (d) $G * w_3$

Figure 3.1: Example of Gaussian kernel convolutions on an ultrasound image.

The above example, shows that different kernels have different impacts on the input image descriptors and these differences must reflect the variation in their entries and algebraic properties. Deep learning schemes, use large sets of convolution filters in a number of layers that apply several processing steps on the convolved images arriving from their predecessor layers and try to learn distinguishable feature maps from the combined different actions of all these convolution filters on the training set of images. The above example illustrates the impracticality of estimating the cumulative effect of numerous convolution filters on a single image, let alone on a large set of training images.In the rest of the chapter, we try to investigate the effect of sets of convolution filters on known image descriptors to determine if there is any linking of algebraically computable parameters of Gaussian filters with their effect on the selected image descriptors.

Ultrasound scan images of different tissues/organs are usually produced by different frequency ultrasound signals due to differences in the textural/structural features within different organ tissues, for detail see [73]. Accordingly, the effects of convolution filters applied to US image descriptors may vary between different human tissues/organs. In order to investigate this variation, we shall next introduce and describe our experimental ultrasound image datasets of different human tissues as we extend our investigations on the effect of random Gaussian kernels.

3.1.2 Ultrasound scan image datasets

Since undertaking this thesis, our primary investigations have been based on ultrasound data sets provided by Ten-D AI Medical Technologies Ltd. These data sets include the bladder, liver, and breast scans which we use throughout the thesis unless stated otherwise. These data sets were collected and sorted from Shanghai Pudong People's Hospital by experienced radiologists. The ground truth of each scan was based on the patient's pathological tests. For classification purposes, these ultrasound scans are cropped around the tumour by radiologists which are called the region of interest (RoI) and labelled as malignant or benign as a few samples are shown in Figure 1.1. The following are descriptions of each data set:

- **Bladder ultrasound dataset**: consists of 176 images which 100 of the cases are benign with image sizes ranging between 9×22×3 to 254×254×3, and 76 of the cases are malignant with image sizes ranging between 25×41×3 to 480×640×3.

- **Liver ultrasound dataset**: consists of 193 images which 106 of the cases are benign with image sizes ranging between 26×18×3 to 157×157×3, and 87 of the cases are malignant with image sizes ranging between 49×48×3 to 454×440×3.

- **Breast ultrasound dataset**: consists of 524 images which 262 of the cases are benign with image sizes ranging between 32×54×3 to 275×373×3, and 262 of the cases are malignant with image sizes ranging between 62×82×3 to 468×846×3.

To ensure consistency and meet the requirements of CNN models, we resize all images to the dimensions of 224×224×3, unless otherwise specified. Figure 3.2 shows different samples of benign and malignant tumours per data set after resizing the regions of interest.

There are several image descriptors used in image analysis to (1) measure the quality such as Entropy, Mean Absolute Error (MAE), Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM);

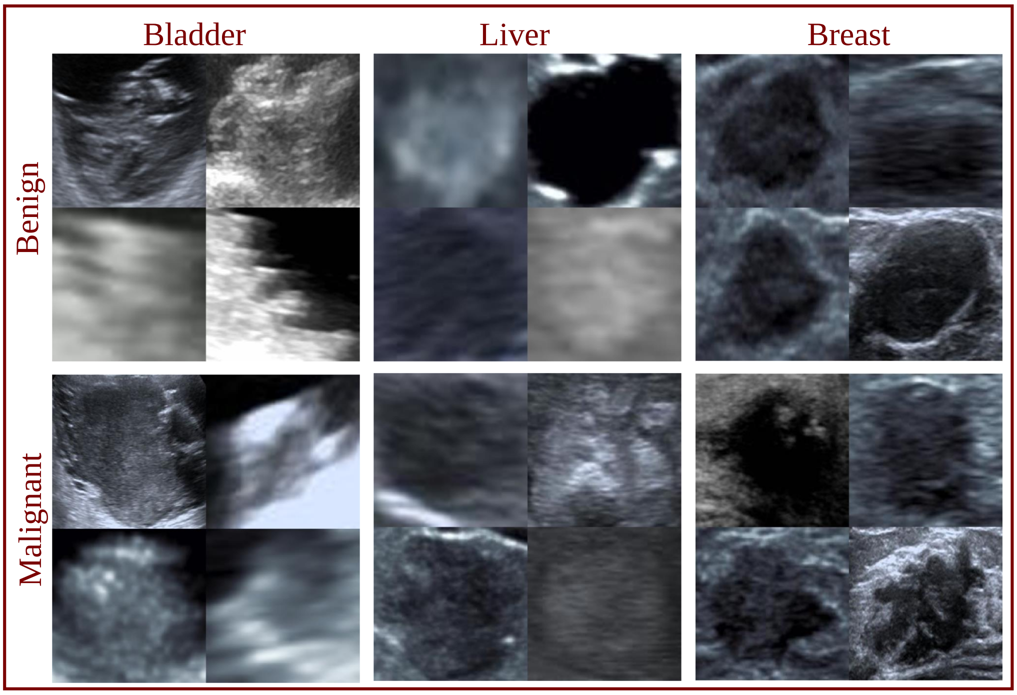


Figure 3.2: Samples of US images with cropped RoI's for benign (B) and malignant (M).

(2) extract texture features such as Gray Level Co-occurrence Matrix (GLCM), Local Binary Patterns (LBP), and histogram of oriented gradients (HOG). In this thesis, we are particularly interested in entropy, texture, and noise image descriptors. We use Shannon entropy to evaluate the amount of information of convolved images. For image texture, we use local binary patterns to extract LBP-based landmarks from which we can compute statistical and entropy descriptors of landmark groups as well as spatial distribution descriptors of these texture landmarks. For impacts of noise on convolved US image descriptors will focus on Gaussian and Speckle noise of different strengths.

3.2 Effect of convolutions on entropy descriptor

The fact that Gaussian filters are image smoothing then it is natural to expect loss of image information as a result of convolutions by such kernels. In this section, we utilise the information theory[1] concept of image entropy as an effective and simple metric to evaluate image content from original and convolved images with random Gaussian kernels. It is a well-known image descriptor, often referred to

[1] Information theory is a field of mathematics that concerns about how information is quantified, stored, and communicated. It focuses on measuring information, finding efficient ways to encode and transmit it, and understanding the limits of information processing systems.

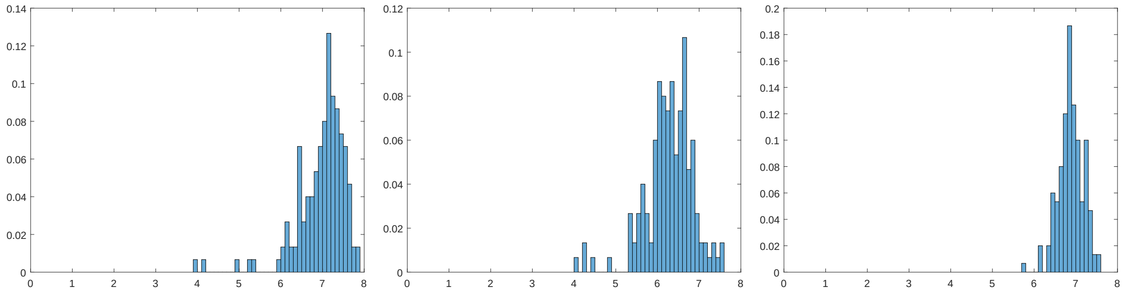
as Shannon Entropy, that measures the amount of uncertainty or randomness in an image, [74]. Shannon Entropy measures the disorder of a signal/image content in terms of the proportion of different symbols/gray-values present in the signal/image. High entropy indicates more randomness or disorder in the signal/image, while low entropy means more regularity or structure. For image entropy, the probability, $p(i)$, represents the histogram of each pixel or bin values occurring, $0 \leq i \leq 255$ s.t. for 8-bit grayscale images the entropy value can be computed as follows:

$$H(X) = -\sum_{i=0}^{255} p(i) \log_2 p(i) \tag{3.2}$$

Entropy values are in the range 0-8 corresponding to 256 possible numerical pixel values. A low or 0 entropy value means no information and uniform pixel value distribution (i.e. mostly redundant information), while a medium to high entropy value indicates the presence of reasonable information with increased certainty. Alternatively, the above equation can be re-written for computing feature maps as follows:

$$H(X) = -\sum_{i=1}^{m}\sum_{j=1}^{n} p(x_{ij}) \log p(x_{ij}) \tag{3.3}$$

where X is an $m \times n$ grayscale image with x_{ij} pixel values of index (i, j), and $p(x_{ij})$ is the probability of histogram of pixel values. Image entropy is widely used in image analysis for tasks such as image segmentation, feature extraction, and compression evaluation. Here, we compare the distribution of entropy of a sufficient number of images convolved pre- and post-convolution by a sufficient number of Gaussian filters. Figure 3.3, below, displays the distribution of image entropy for randomly selected 150 images from the US (Bladder, Liver and Breast tissue) scan data sets, All images are resized to 224×224.



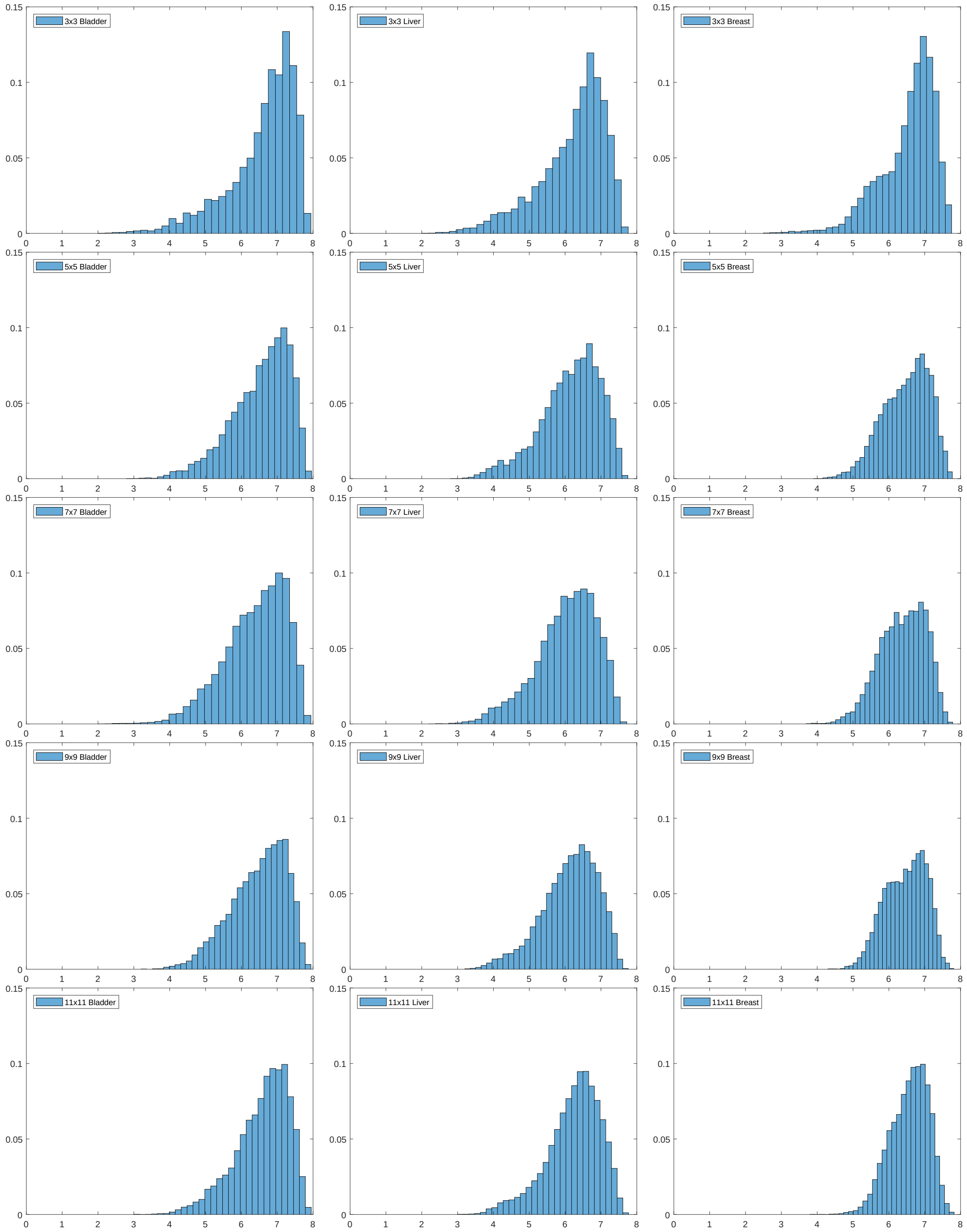
(a) Bladder USI (b) Liver USI (c) Breast USI

Figure 3.3: Distribution of entropy from original ultrasound images.

The distribution of entropy values seem to differ on the position of the mean and (min-max) range in different tissues. Low entropy values ranging from 3.8 to 5.5 in bladder and liver histograms are representing the small sizes of RoI images with less texture variation in the image compared to RoI images with entropy values between 5.5 to 8. The breast US scan dataset entropy values are in the tighter and higher range (5.7-7.2) and there are less variation in terms of tumour sizes measured by the cropped RoI.

To test the impact of convolution of these image sets with a reasonable number of different Gaussian kernels, we generated five different sets of 100 random Gaussian filters of size $n \times n$ for $(n$=3,5,7,9,11$)$ of mean $\mu = 0$ and standard deviation σ = 0.01. Note that, these kernel sizes and parameters are most common kernel sizes in constructing convolutional neural network architectures. The corresponding entropy distribution histograms are displayed in figure 3.4 for all three ultrasound data sets. Interestingly, the differences in the shape/range of entropy distributions between the different tissues disappeared as a result of convolutions with Gaussian kernels of any size. In all cases, the entropy values post convolution have a mean of around 7 with a negative skewness distribution. Furthermore, the lower bound values increase, in all cases, as the kernel sizes increase. Similar entropy distributions, post convolution, were observed when σ = 0.1 and 0.001 for all three datasets.

These results confirm that the observed similar impact of convolving images from these different tissues on their entropy do result in changing image contents, but the fact each image corresponds to a single entropy value makes it difficult to gain useful insight into the effect of different convolution filters in terms of the spatial distribution of changes or in terms of enhancements/elimination. Therefore, it is crucial to complement the gained knowledge in this section by investigating the impact of convolution on image texture features. The rest of this chapter will deal with this issue, but first, we attempt to identify existing descriptors associated with image texture features used in machine learning and propose a new image descriptor that quantifies image texture information content.



(a) Bladder USI (b) Liver USI (c) Breast USI

Figure 3.4: Distribution of entropy values of convolved US images with $n \times n$ kernels.

3.3 US image texture features and descriptors

The three-dimensional landscape illustration in Figure 3.1, confirm the observation made at the end of last section about the necessity of using landmark based texture feature analysis descriptors such as LBP, HoG, and GLCM. For HOG texture descriptor, it is cumbersome to link magnitude and orientations to local landmark positions, because it computes the distribution of the gradient map values, in blocks, post binning the range of the slope values of the gradients. The same is true about GLCM and many other texture features. Accordingly, only LBP and other similar texture descriptors can be presented in a manner that are associated with image positions and pixels (see next section for an explicit analysis).

In this section, we explore a statistical analysis of US scan texture features before and after convolutions using LBP Texture descriptors. This help considering the LBP image transform as a random space defined on the different LBP code symbol, and consequently we introduce and propose a simple measurement based on the statistical distributions of LBP codes called ***Entropy LBP*** landmarks.

3.3.1 The local binary pattern image texture feature

Texture features are associated with changes in spatial/transform image representations. Over the years a variety of image texture analysis methods have been developed and applied in computer vision applications. These methods often exploit the statistical distribution of changes, structural-based changes, model-based changes, and/or spectral-based changes. See [75,76] for a recent review on texture analysis. This chapter is not designed to determine the effect of convolutions on all kinds of texture features. In image processing and computer vision, local binary patterns is one of the powerful texture descriptors that is used to extract texture information from images for a variety of automatic image analysis tasks such as facial recognition, object recognition, and image segmentation. LBP is known to be robust to noise and is particularly useful because it is invariant to monotonic grayscale transformations, (i.e. it is not affected by changes in lighting or other changes that only affect the overall intensity of the image). The LBP has been extended in various ways depending on the radius of neighbouring pixels and the relation between the targeted pixel to its neighbouring regarding additional conditioning. These variants include rotation-invariant LBP, uniform LBP (ULBP), and multi-scale LBP that are deemed to improve performance/robustness of the descriptor in various tasks.

LBP was first introduced by Ojala et al. [77] to extract image texture features based on the comparison of the intensity of the centre pixel with the intensity of the pixels in its neighbourhood pixels and to assign a binary code to each pixel depending. The LBP transformation is a composition process of image pixel values based on their link to the surrounding pixels $\Psi : \mathbb{R} \to \mathbb{R}$ according to the following formula:

$$\Psi(x_c, y_c) = \sum_{i=0}^{7} \psi(k_i - k_c)2^i \tag{3.4}$$

where k_i is the neighbouring pixel, k_c is the centre pixels at (x_c, y_c) location, and the function $\psi : \mathbb{R} \to [0, 1]$ is defined as:

$$\psi(x) = \begin{cases} 1 & if \quad x \geq 0 \\ 0 & if \quad x < 0 \end{cases} \tag{3.5}$$

If the intensity of the centre pixel is greater than the intensity of its surrounding pixels, a binary code of 1 is assigned, otherwise, a binary code of 0 is assigned (e.g. see Figure 3.5). The binary codes of all the pixels in an image I are combined to form a LBP image, which is a representation of the texture of the original image. For 8-bit grayscale images, there are 256 LBP patterns that are rotational invariant. When considered as circular strings, the set of all LBP codes can be grouped[2] according to the number of transitions between 0 and 1. In general, the LBP patterns come in five different transitions namely 0,2,4,6,8-transition, each with various number of groups per transition, and each group with different subgroup of rotations.

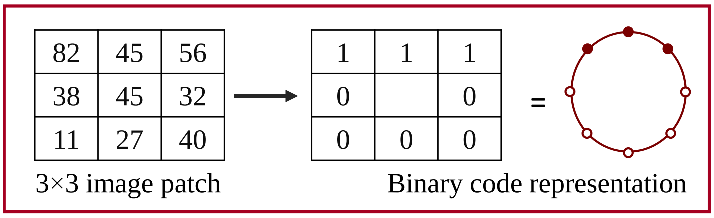


Figure 3.5: An example of LBP code of the centre of a 3×3 image patch.

A 0-transition indicates the values either 00000000 or 11111111, and 8-transitions are 01010101 or 10101010. The 2-transition groups, also the well-known Uniform Local Binary Pattern (ULBP) consisting of seven groups, each with 8 different rotations. The 4-transition and 6-transition groups consist of five and three groups, respectively, each with a different number of subgroup rotations. Each transition

[2] The use of notation for groups and subgroups is solely for the purpose of distinguishing between different LBP operator landmarks.

group are known to indicate the presence of different structural features such as corners, edges, as well as smooth locality. For simplicity, we show all transitions and groups in Figure 3.6 with including subgroup rotations as ($\#transition$, $\#rotations$). The empty and filled circles are indicating 0s and 1s, respectively.

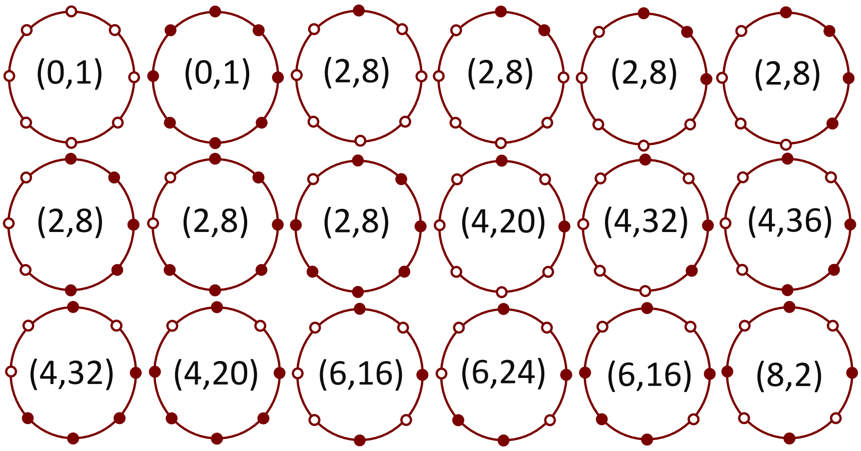
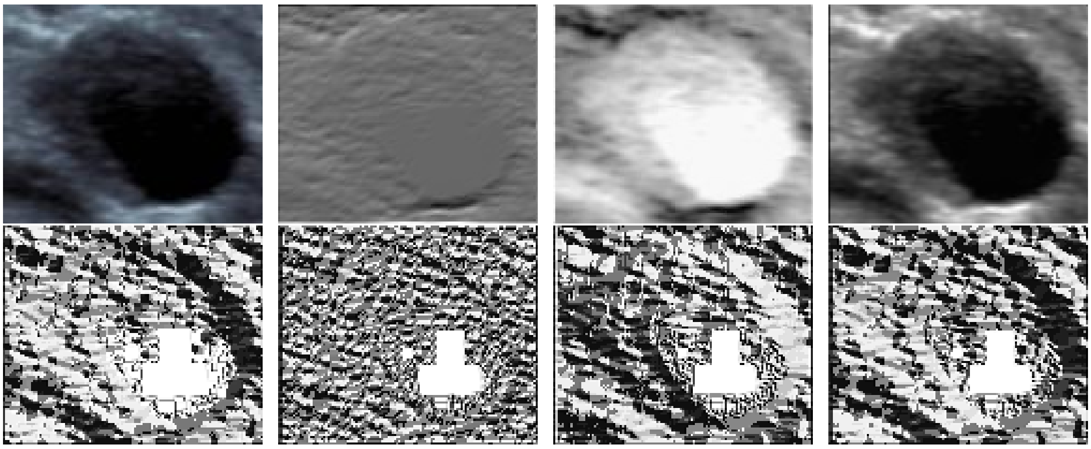


Figure 3.6: Illustration showing the 0 and 1 binary values of LBP landmark groups represented by empty and filled circles, respectively. The count of transitions between 0 and 1, along with the number of rotations per transition, is presented as $(\#transition, \#rotation)$.

The commonly used statistical representation of LBP codes in image analysis tasks is a histogram with 58 bins of 0- and 2-transitions with/without a 59th bin for the rest of the transitions. However, the 256-bins have also been used in some applications to represent the LBP transformed image after computing image intensity post LBP transformation.

In Figure 3.7, below, we show an example of transferring the original and convolved ultrasound images into LBP representation using the same Gaussian kernels from Figure 3.1. The LBP transformed images highlights the local information about the texture features present in the original image. The white-patch areas in all four LBP transformations represent the 0-transitions around this area, while the dark black patches represent 1-transitions. The LBP representations from convolved images with kernels w_1 and w_2 are showing fine details from the original images. Whereas, w_3 is showing most of the texture features from the original image around.

We shall next determine the probability distributions of these different LBP transitional codes in US images, but most of our focus in later chapters will be on the ULBP groups for texture analysis and spatial distributions. We extend the statistical computation of these LBP groups for all three ultrasound datasets and their convolved images from the previous section.



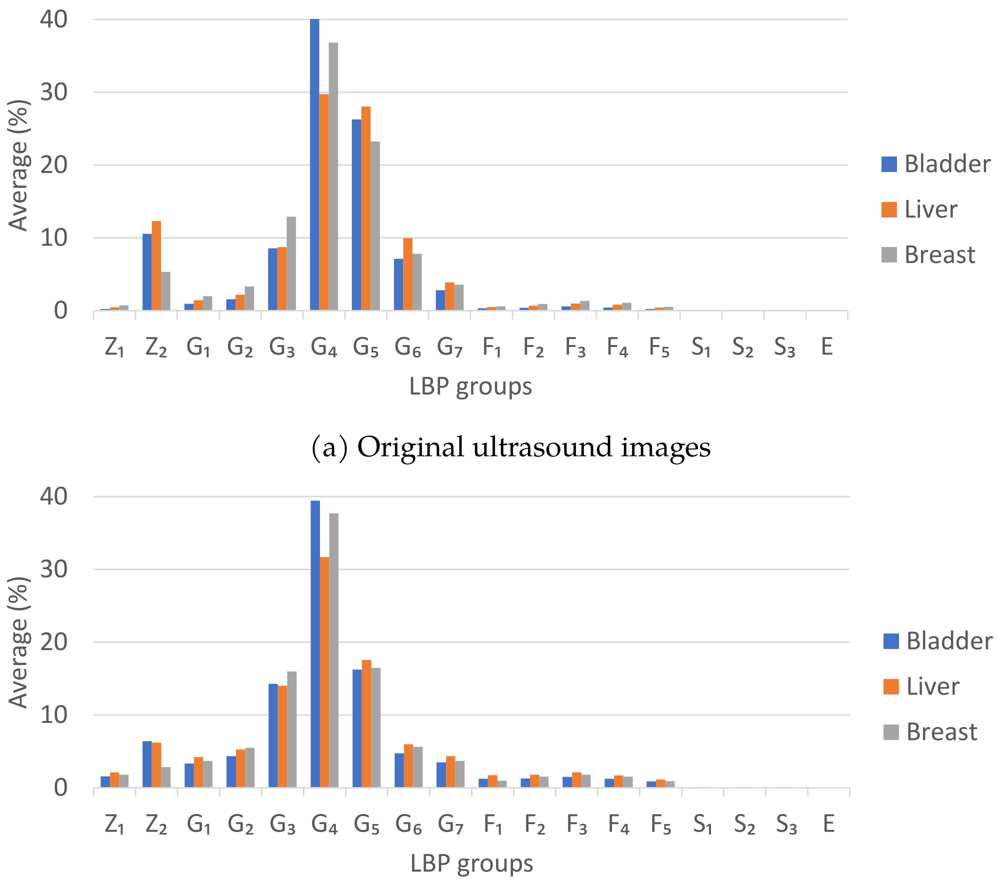
(a) Image G (b) $G * w_1$ (c) $G * w_2$ (d) $G * w_3$

Figure 3.7: Example of LBP operator performed on original and convolved ultrasound images.

3.3.2 Statistical distribution of LBP US datasets

Statistical analysis of the LBP landmarks can provide insight into the nature of the image or dataset properties in terms of the dominant, existence, or absence of certain geometric patterns. Therefore, we compute all five different transitions with their associated groups regardless of subgroup rotations mainly to understand the most and/or less frequent transitions for bladder, liver, and breast ultrasound datasets. The statistical analysis includes the minimum, maximum, range, average, standard deviation and mode for each LBP transition/group for 150 images per dataset with 75 images per class. We present the average percentage of all three data sets in Figure 3.8-(a). The groups Z_1 and Z_2 are 0-transitions, G_1-G_7 are 2-transitions, F_1-F_5 are 4-transitions, S_1-S_3 are 6-transition groups, and E is 8-transition. Dominant texture features in all three datasets are G_4 and G_5 with [24,40] % , whereas no image landmarks are found in 6 and 8-transitions. All three datasets have the 0, 2, and 4-transition LBP landmarks with a noticeably similar pattern among each group. After the statistical analysis of the original ultrasound scans, we compute the LBP statistics after convolution with 100 random Gaussian filters from the previous section. In Figure 3.8-(b), average percentages of LBP landmark groups from 15000 convolved images show a similar statistical pattern to the original image with a notable increase in 9 of the groups $(Z_1, G_1, G_2, G_3, F_1, F_2, F_3, F_4, F_5)$ and a very modest increase in the %s of the S and E groups that were almost absent in the original images. These increases, post convolution, occurred at the expense of the other dominant groups.

The results in Figure 3.8 reveal mostly modest differences between the various



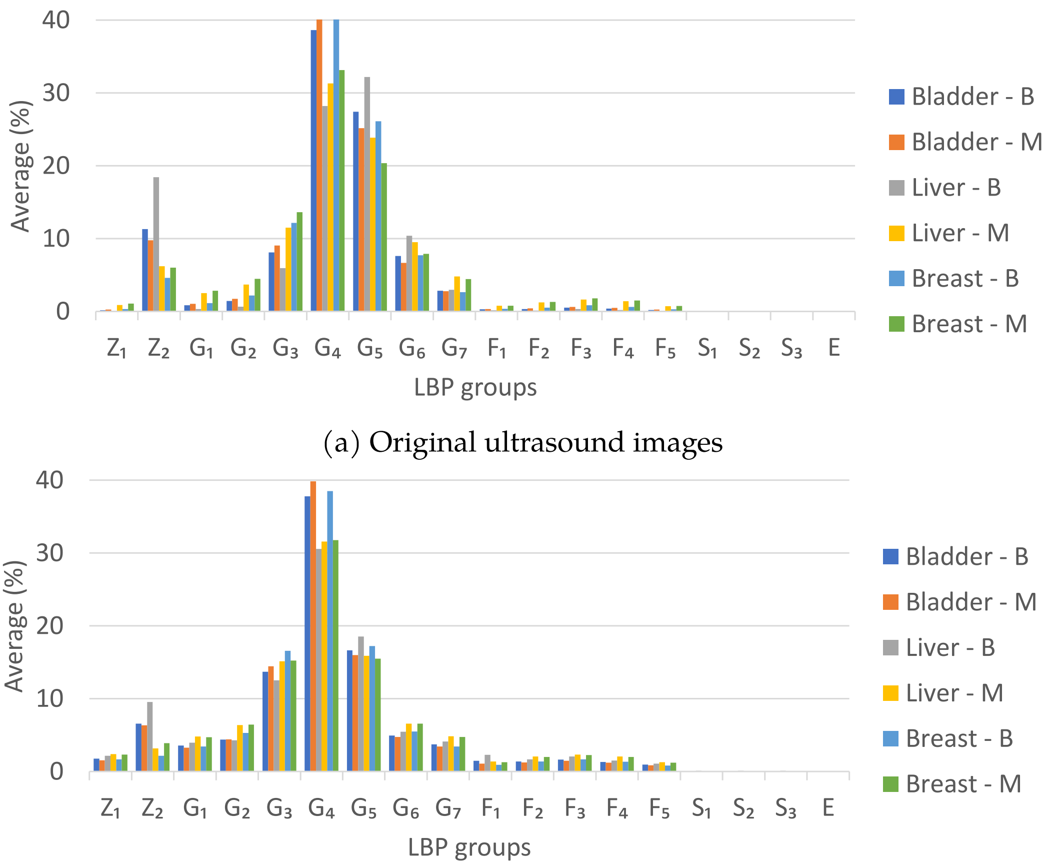
(a) Original ultrasound images

(b) Convolved ultrasound images

Figure 3.8: Average % of LBP transitions for bladder, liver, and breast US datasets.

group statistics between different tissue types in both the original images and in the convolved images. Our investigation extended to examining whether these statistics exhibit class-dependent patterns in terms of the nature of the tumour. Our results indicate that the statistical patterns of individual LBP groups for malignant and benign classes are nearly identical, with only minor, imperceptible differences observed in some LBP groups 3.9).

Remark 1. (Texture contents of ultrasound vs atural images) The statistical distribution of the LBP different groups in Figure 3.8-(a), provides a clear significant distinction between Natural images and US medical images in terms of texture content. In [54], A. Asaad computed statistical distribution of the ULBP groups (i.e. Z_1,Z_2, and $G_1 - G_7$) in passport face images in two known publicly available databases (Utrecht and London DBs). While the Z_2 (11111111) landmark group in ultrasound images form in the low range of (5-11)%, in face images it was estimated to form 87.67% and 72.96% in the London DB and Utrecht DB, respectively. Furthermore, on average each of the G_i groups form 1.4% (3.06%) in the face im-



(a) Original ultrasound images

(b) Convolved ultrasound images

Figure 3.9: Average % of LBP transitions per class for the 3 US datasets.

ages of the London DB (Utrecht DB), whereas the G_4 alone form (30%-40%) of US images. Thus, unlike natural face images, texture features are predominant in US images. This is a motivation to propose the use of texture information content as an important image descriptor.

3.3.3 A new texture entropy descriptor of LBP landmark groups

The statistical analysis of LBP landmark groups provides an additional motivation for introducing a new image descriptor of texture information content by a straightforward entropy based formula. Here, we propose evaluating the amount of information conveyed by LBP landmarks in general and per transitions and/or groups. Integrating entropy with LBP descriptor has shown significant advantages for image processing tasks for example the authors, in [78], used entropy based local binary pattern (ELBP) for a biometric database. The ELBP is computed based on replacing each image pixel value with its entropy value for extracting texture features and performed better than the conventional rotation invariant LBP methods. We have become aware of a very recent similar approach to that of ELBP but based on a block-wise consideration, [79]. Our proposed texture entropy descriptor, denoted by TE(LBP), differs in terms of the utility and computability. It can be used to identify subtle changes in texture features and enhance the ability to estimate information content of texture landmarks in images. In addition, it can facilitate further analysis along with, and in comparison to, the traditional image entropy. Computing Entropy of LBP landmarks, TE(LBP), is based on the statistical analysis of the 256 LBP landmarks substituted in the conventional entropy equation 3.2:

$$H(X) = -\sum_{i=0}^{4} p(x_i) \log_2 p(x_i) \tag{3.6}$$

Where $p(x_i)$'s are the probability of the presence of the five LBP landmark transition groups $i = 0, \ldots, 4$. Alternatively, transform the image into LBP representation and compute the histogram of each bin from 0 to 255 and the probability of nonzero bins. The resulting entropy values will be between 0 and 8, as explained in Section 3.2 for 8-bit images.

The intuition of using this approach is to provide distinct general insight into the image content information in terms of various texture features. To illustrate the difference between our proposed texture entropy descriptor from that of traditional image entropy, consider an ideal 16×16 array/image (Figure 3.10) of 256 sorted pixel values from 0 to 255 where the each bin occurs once exactly and have the probability of 1/256 resulting in traditional image entropy value of 8. Afterwards, we encode the pixels into their correspondence LBP descriptors and compute the occurrence of each bin. shows the ideal image pixel from black, 0, to white, 255 and their LBP encoding where the changes are noticeable at the top/bottom left/right corners. In this ideal image, the dominant LBP rotation is 00111100 (2-transition) with high probability i.e. there are four neighbours less than the targeted pixel value, and four neighbours greater than the targeted pixel value. The TE(LBP) of the image was found to be 1.0059 while the individual entropy values are 0.0624 and 0.9433 for 0 and 2-transitions, respectively.

Depending on the application, the texture-based entropy type can be customised to compute the entropy of a selected set of landmarks or for conditional occurrences. We assessed the Texture Entropy (TE) derived from Local Binary Patterns (LBP) in ultrasound images of bladder, liver, and breast tissues. The results, presented in Figure 3.11, are compared to the corresponding distributions

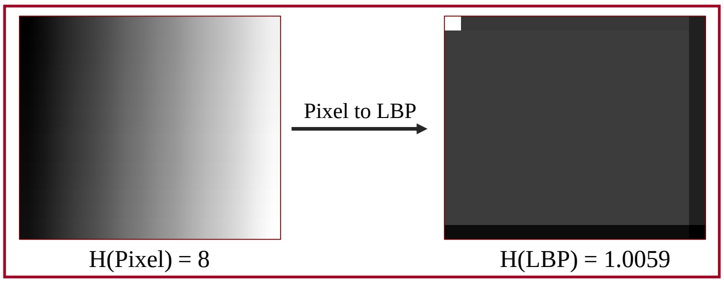
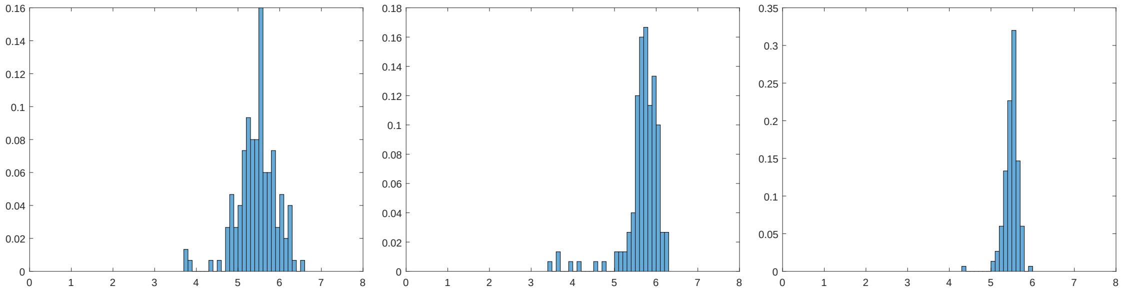


Figure 3.10: An illustration of high variation pixels and its LBP landmarks.

of traditional entropy in Figure 3.3. The TE(LBP) distributions exhibit similar shapes but are consistently shifted to the left by approximately 1.5 values across all tissue types. This shift indicates a higher level of certainty regarding texture information compared to pixel value information.



(a) Bladder USI (b) Liver USI (c) Breast USI

Figure 3.11: Distribution of Texture-based LBP entropy from original ultrasound images (USI).

Due to the fact that CNN have strict requirements important for identifying changes in the tissue texture features due to resizing, adding or removing noise. In general, ultrasound scans can have varying levels of tumour RoI resolution and noise depending on several factors, including the US devices, the imaging technique, and the specific application. The effect of these and other factors will be discussed in Section 3.4.

The success of the topological data analysis paradigm, that is based on gradual construction of simplicial complexes for point clods of data records encapsulating the spatial distribution of the point cloud, is a strong motivation for studying the spatial distribution of LBP landmarks. Clearly, neither the LBP landmark entropy nor their statistical distribution provides information about the spatial distribution of image pixels or textures. Next, we utilise topological data analysis tools to gain insight into the underlying patterns and relationships in image data.

3.3.4 The spatial distribution descriptor of LBP groups

Studying the spatial distribution of LBP landmarks is expected to provide additional tools for automatic image analysis, and benefits from using emerging paradigm of Topological Data Analysis (TDA) and its tools. The main idea is that the LBP landmark groups from point clouds of image pixel positions (i.e. points in the Euclidean plane $\mathbb{R}^2$) that can be represented as a topological space, the structure of which can be used to gain insight into the underlying patterns and relationships in the corresponding image datasets. Recall that the spatial distribution of a point cloud in $\mathbb{R}^n$ is determined by the TDA tool of persistent homology, defined in Section 2.2, records the topological invariants (number of connected components, loops, holes, and voids) in the gradually constructed sequence of simplicial complexes formed by the point cloud as 0-simplices at an increasing sequence of distance thresholds. In recent years, topological data analysis has emerged as a promising approach for studying the spatial distribution of LBP landmarks for genuine and tampered face images [80]. Here, we follow this approach of investigating LBP landmarks persistent homology to model the landmarks spatial distribution. Figure 3.12, illustrates the process of generating the PH representation of LBP group landmarks in ultrasound scan images. We selected rotation R_1 in G_5 group of 2-transitions constructed the sequence of simplicial complexes using different distance thresholds d (0,10,20,25,30, and 35). Clearly, the topological invariants (*#connected components*) in dim 0, and (*#holes* bounded by >3 sides) in dim 1 change at each level of increased distances from 0 to 35.

In this thesis, we exploit this process to analyse the spatial distribution of LBP landmark groups in US images to analyse the nature of malignant and benign texture tissues before and after convolutions and other texture distortion factors.

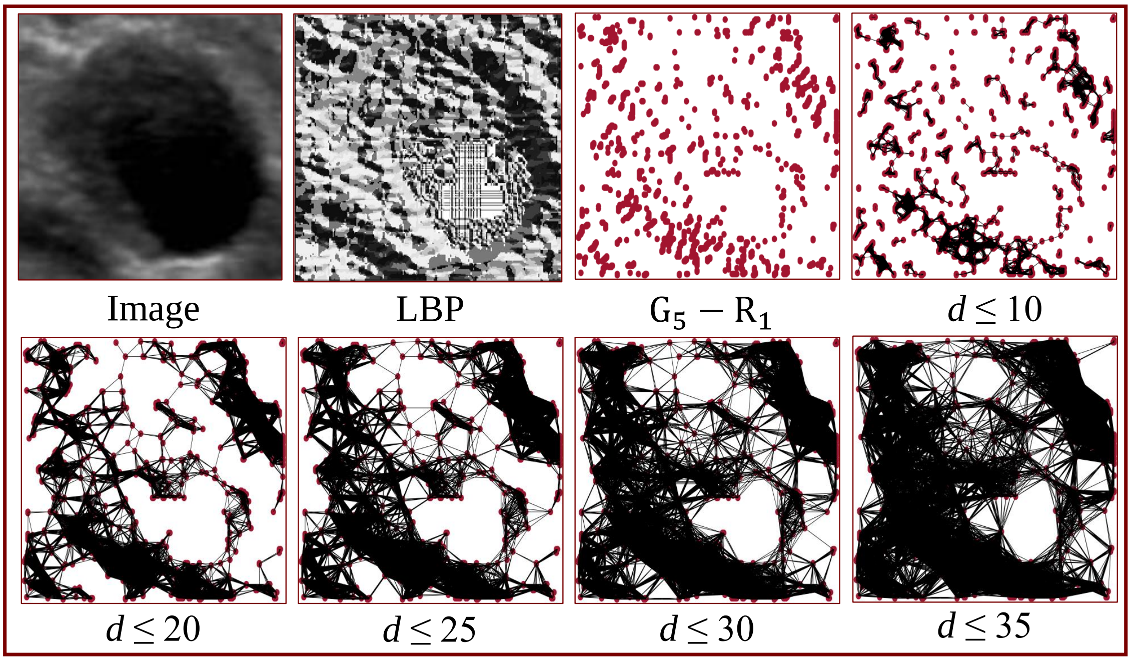


Figure 3.12: Illustration of the determining the PH process of (G_5, R_1) landmarks.

3.4 Factors influencing US texture descriptors

In this section, we highlight the potential factors that may result in changes in the texture features and influence the effect of convolution kernels. These factors include image resizing, presence of noise, and Algebraic properties of convolution kernels.

3.4.1 Impact of image resizing

Resizing images is a challenging task in computer vision and particularly for low resolution region of interest (RoI) in medical images. Existing state-of-the-art CNN models require the input images to be of fixed size, while the RoIs of US tumour scan image datasets include significant variation in size. CNN required US RoI resizing is expected to have size-dependent effects on the quality of resized image. Image resizing techniques can impact input image texture features as well as the performance of automated or AI-based computer-aided systems. Resizing and rescaling ultrasound images are important to standardise image sizes for analysis, usability with DL models and visualisation purposes. Various methods are available for resizing and rescaling images, each with its own strengths and limitations. These methods include nearest neighbour, linear, cubic, spline, and wavelet-based interpolation techniques. The simplest method is the nearest-neighbour interpolation, where the nearest pixel to the new location is used when resizing an image. Bilinear and bicubic interpolations are straightforward methods, where the weighted average of the four and sixteen nearest pixels are taken into account for the new pixel value and location, respectively. Spline and B-Spline interpolations are more sophisticated techniques to estimate the new pixel values and locations by fitting smooth curves. Wavelet-based interpolation is another method that can produce high-quality ultrasound images by decomposing the image into a series of wavelets and using these wavelets to estimate the value at the new pixel location. This method is useful for ultrasound images with a high level of detail, such as in vascular imaging or fetal ultrasound.

There are significant advancements and successes in terms of enhancing image resolutions for natural and some medical images, [81,82]. The super-resolution process in ultrasound images typically involves applying various algorithms and techniques to the original low-resolution or relevantly small-size image to create an upscaled version with a higher resolution, which can reveal fine details and structures that were previously hidden or could be lost during the traditional resizing method. The challenge in choosing the ultimate method for US image resizing and rescaling stems from the need to take into account simplicity, impact on image quality, as well as the significant variation in tumour RoI sizes. For more details, see [83, 84]. In this research, we deploy the default MATLAB image resizing function as exploring image resizing and cropping is not within the scope of our research.

Ultrasound image sizes in our research project is reflecting the cropped areas around the tumour (i.e. the RoI) and all the RoI's in the US datasets differ significantly in their size. In deep learning models, it is required to have a unified image size to proceed with the end-to-end process. The ratio between the size of the original image and the targeted size can influence the quality of the new resized image. Unlike natural images, there is no agreed standard of how to quantify the quality of US images. If the ratio is large, then resizing is expected to result in loss of texture features manifested by blurring effects. For example, when the original tumour image size is as small as less than 40×40 then resizing it into to 224×224 is expected to be worse than doing the same for tumour image size of 600×600. Figure 3.13 shows the difference of a gradual resizing of an ultrasound image from 102×121 to 224×224. It is difficult to determine the amount of information change with visual inspection, whereas LBP transformed images seem to show some noted changes of local information, in certain regions, after resizing the image. The reader is advised to examine the nature of the vertical lines pattern in marked regions as the resizing increases. This example indicates possible changes to the TE(LBP) as well as changes to the spatial distribution of some LBP landmark groups.

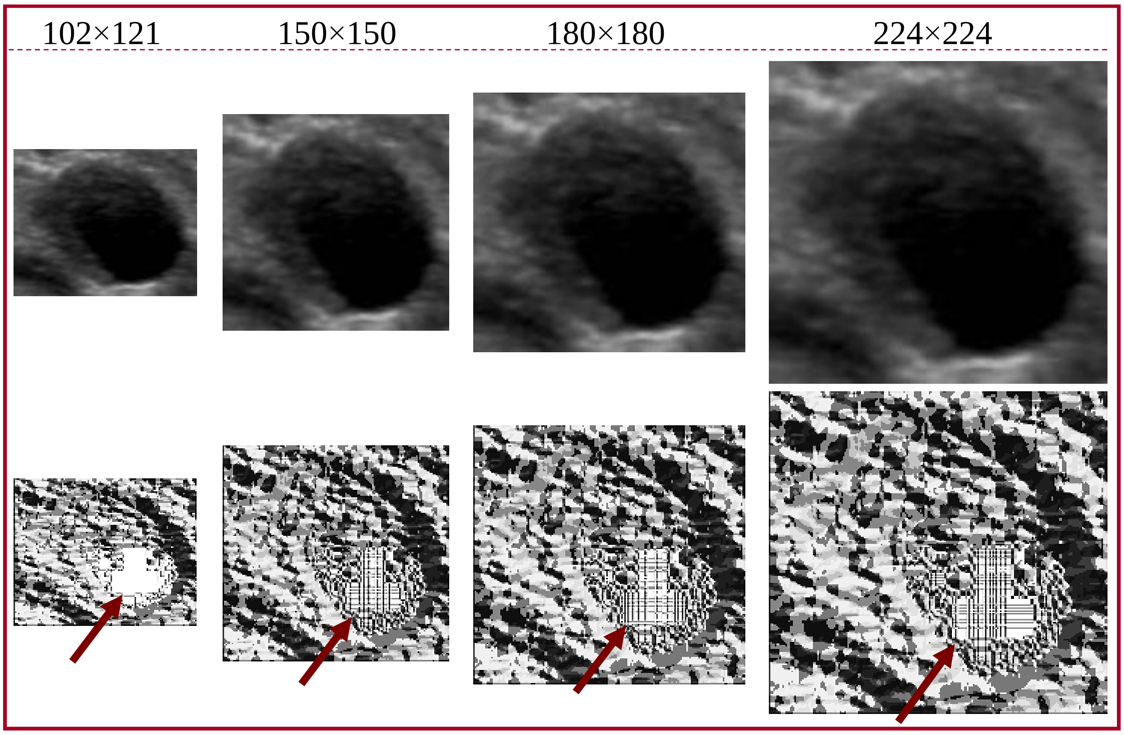
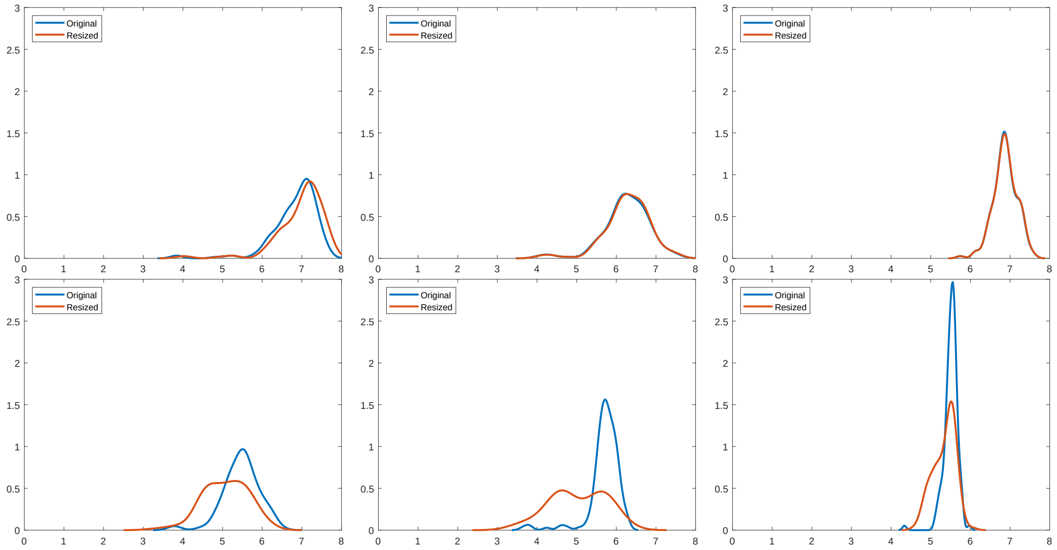


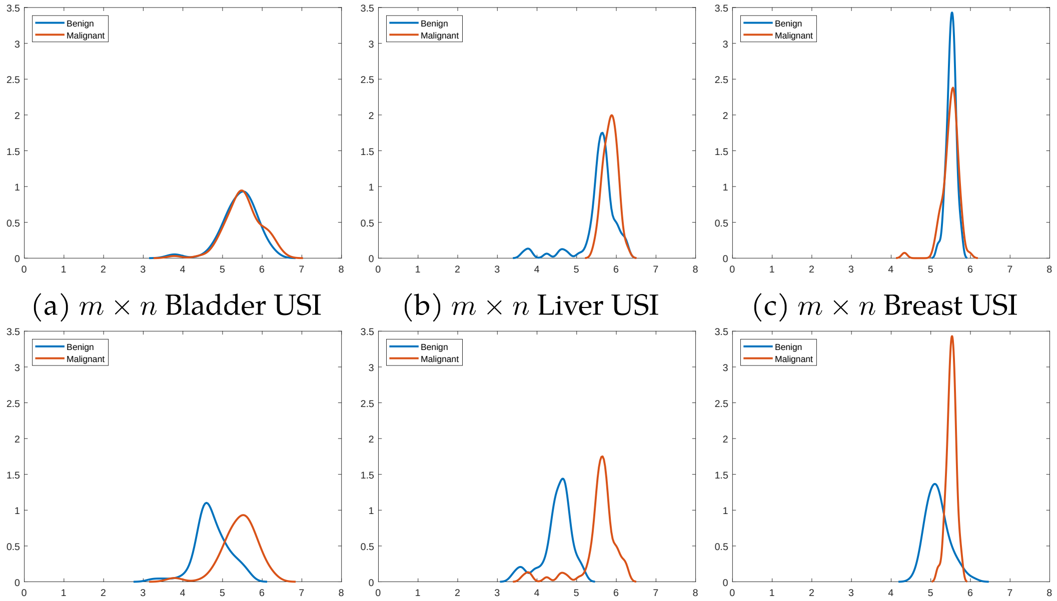
Figure 3.13: Illustration of an ultrasound image resizing and LBP representations.

To demonstrate the above observation, we experimented with $m \times n$ ultrasound scans images in the trio of US datasets, by computing the entropy of the original image sizes and that of resized version of the 224×224, as well as their texture-based entropy TE(LBP) before and after resizing. The distribution of entropy values of 150 per dataset are presented in Figure 3.14. There is hardly any noticeable differences between entropy of the original and resized images of all tissue scans, whereas the differences between the texture-based entropy of the original and resized images for all types of tissues are not negligible. In addition, we compute the texture-based entropy per class before and after resizing to examine which class is particularly affected by the image resizing. There are significant differences in TE(LBP) distribution between malignant and benign classes after resizing for the three datasets as shown in Figure 3.15. In fact, the separation of TE(LEP) between the classes of bladder and liver show good discriminating power due to the nature of the tissue and tumour sizes.



(a) Bladder USI (b) Liver USI (c) Breast USI

Figure 3.14: Distributions of entropy values (top), and LPB texture entropy (bottom).



(a) $m \times n$ Bladder USI (b) $m \times n$ Liver USI (c) $m \times n$ Breast USI

(d) 224×224 Bladder USI (e) 224×224 Liver USI (f) 224×224 Breast USI

Figure 3.15: Class-distribution of TE(LPB): Original sizes (Top) and the resized (Bottom).

3.4.2 Impact of noise on image descriptors

Ultrasound images can be affected by noise of different types arising from several sources such as electronic interference, signal processing, and physical phenomena such as acoustic speckle. Noise are manifest as random fluctuations in pixel values, making it difficult to distinguish between structures of interest and noise. However, advances in ultrasound technology and image processing have led to improved noise reduction techniques, which can help improve image quality [85]. It is worth noting, that addition of noise to images is a well-established source of adversarial attacks on Deep learning CNN models.

LBP texture features are robust against illumination, contrast and some uniform distortion in image pixel values. However, their robustness against non-uniform structural/textural changes, in US images is not well understood. Speckle and random Gaussian noises are excellent candidates to generate such changes. Random Gaussian noises may make changes to LBP landmark statistics and spatial distributions as a result of random fluctuation in pixel values surrounding the landmarks. To determine the effect of such random pixel value fluctuation around LBP landmarks, we recomputed the statistical distribution of LBP landmark groups for the 3 datasets post noise addition to the images. Figure 3.16, below, illustrates a significant change in these distributions across the 3 datasets when Gaussian noise, $mean(\mu)$ =0 and $variance = 10^{-3}$ were added in comparison to those of clean images.

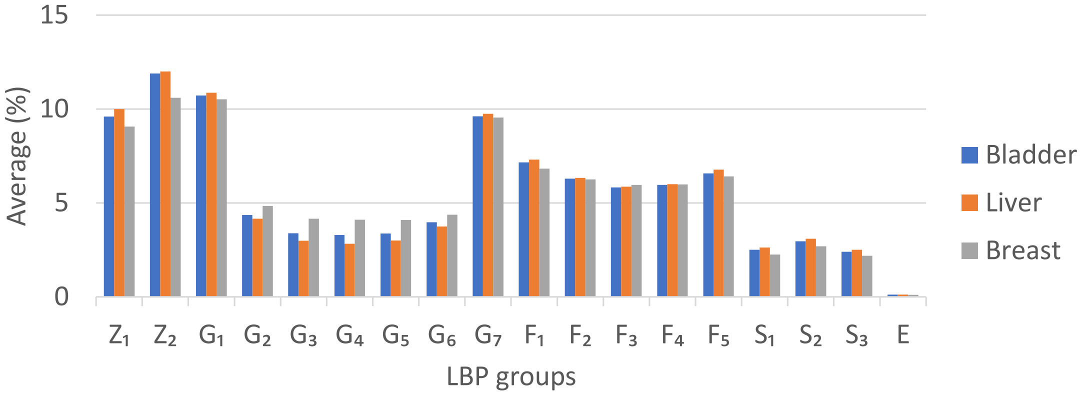


Figure 3.16: LBP landmarks distribution in the 3 US datasets post Gaussian noise.

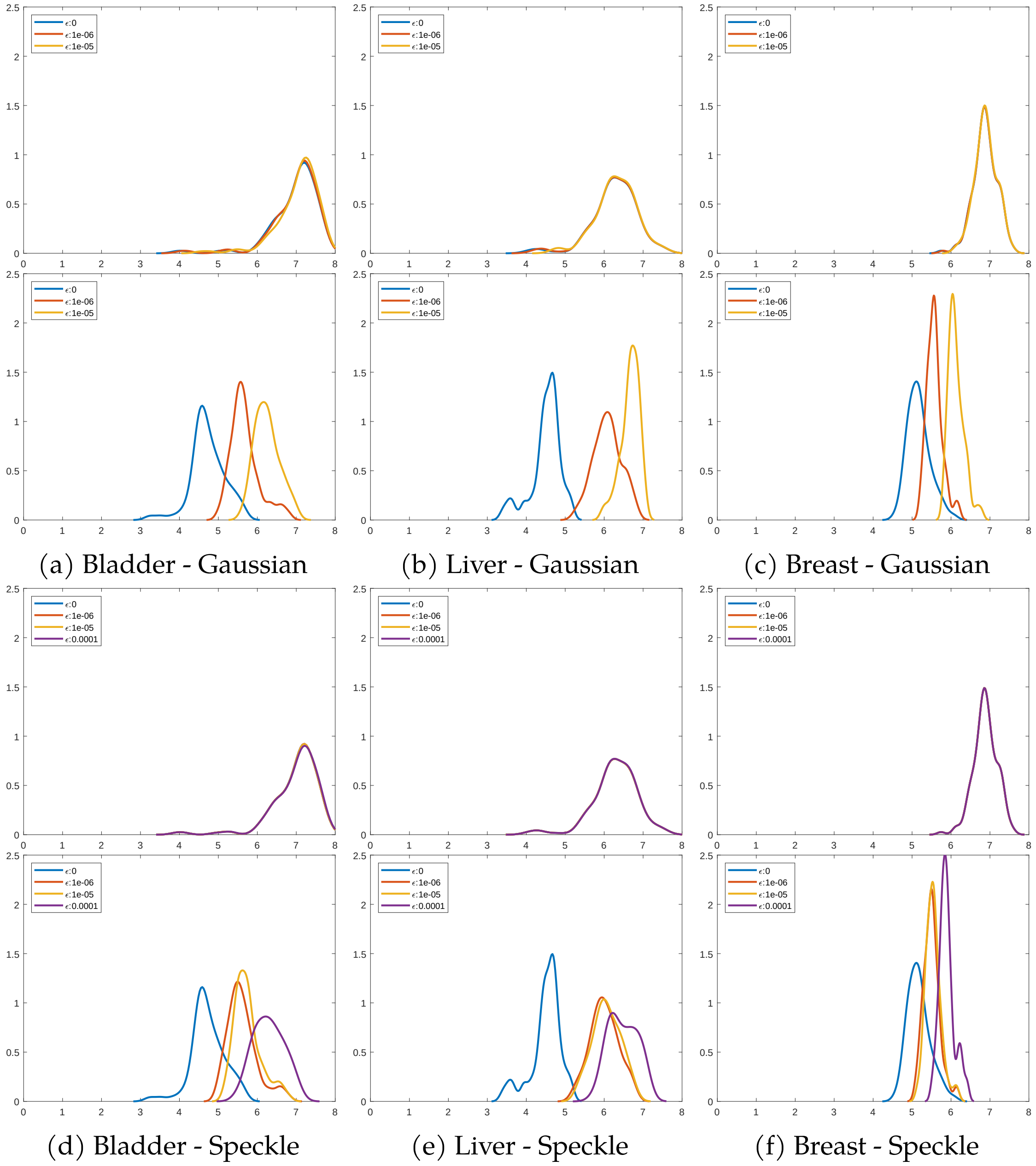
Comparing the results presented here with those in Figure 3.8-(a) reveals an intriguing pattern of change in the statistics of LBP landmarks caused by the addition of noise. The only group that maintained its statistics is the bright Z_1 landmarks (i.e., 0-transition represents all 11111111). The more dominant LBP landmark groups G_3, G_4, G_5, and G_6 have been significantly diminished proportionately. All other groups exhibit an increased presence due to the alteration and variation of 8-pixel neighbours around the centre pixel values. We anticipate that these changes will translate into improved TE(LBP) values for noisy images, and we may also expect similar, if lower, effects with the addition of speckle noise given the nature of ultrasound scans. The subsequent set of experiments was conducted to assess the effects on TE(LBP) with different levels of noise addition.

Our experiments aim to compare the distributions of traditional image entropy as well as texture-based TE(LBP) of the original clean images and their noisy version for the images in the 3 datasets. Several levels of noise addition have been tested in these experiments, namely random Gaussian noise with $\mu = 0$ and $var = 10^{-6}$, & 10^{-5} and speckle noise with $\mu = 0$ and $var = 10^{-6}$, 10^{-5}, & 10^{-4}. In Figure 3.17, we observe little or no change to the distributions of entropy values between clean and noisy images. However, the distributions of TE(LBP) changes and become more uncertain as more noise is added to each dataset. Low and medium TE(LBP) values are most affected by the noise and the shift in the distribution is caused by the fluctuations around the pixel values.

To determine the impact of convolutions on image entropy and TE(LBP) on noisy images in comparison to clean ones, we repeated the above set of experiments after applying the same set of convolution kernels on the original and noisy images. The results are shown in Figure 3.18.

The random Gaussian kernels are smoothing the effects of the added Gaussian and speckle noises, therefore we added further level of noises to test the level of noise that makes the distribution of entropy values separate from the distribution of the original images without noise. These results show that the combined effects of adding noise and convolutions on traditional entropy are of less use to these effects on TE(LBP). Figure 3.17 show that the TE(LBP) in the Bladder and Liver datasets are more affected by the increased level of added noise compared to the breast dataset. For both bladder and liver datasets the TE(LBP) distributions shift and/or change with small overlapping percentages. Figure 3.18, show that the Gaussian kernels suppress most of these changes and it takes higher noise level to get reasonable differences.

This type of investigation will contribute to our exploration and analysis in the subsequent chapters of deep learning settings, where the prevalent challenges include underperformance, robustness, and adversarial attack issues. The robustness of deep learning models is evaluated by introducing small changes to the input data or image, with the expectation that these changes should lead to minimal or no alterations in the output, especially when the changes are imperceptible to human vision.



(a) Bladder - Gaussian

(b) Liver - Gaussian

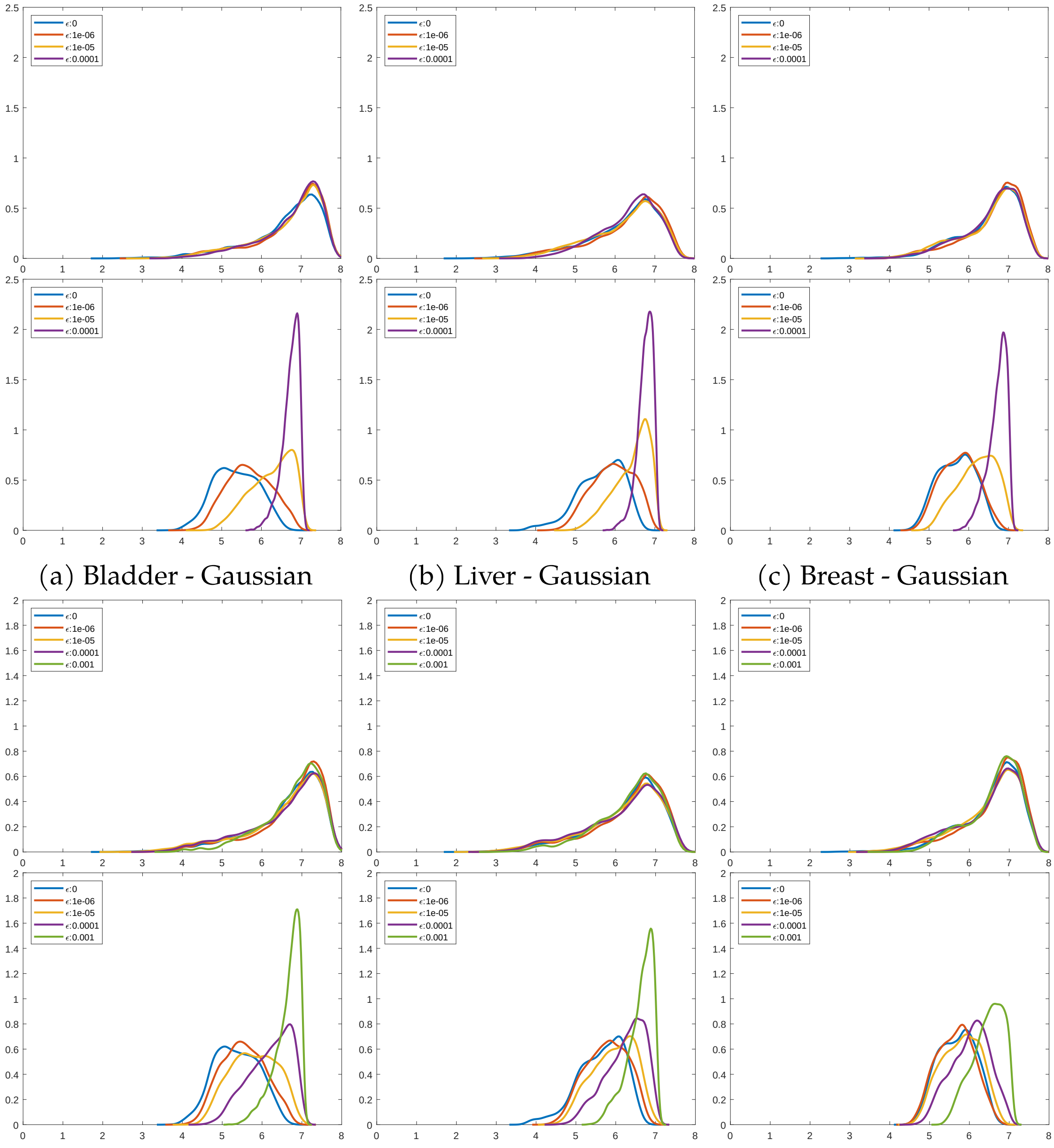
(c) Breast - Gaussian

(d) Bladder - Speckle

(e) Liver - Speckle

(f) Breast - Speckle

Figure 3.17: Impact of noise on distributions of image entropy and TE(LBP) post various Gaussian (Top) and Speckle (Bottom) noise levels.



(a) Bladder - Gaussian (b) Liver - Gaussian (c) Breast - Gaussian

(d) Bladder - Speckle (e) Liver - Speckle (f) Breast - Speckle

Figure 3.18: Effects of convolutions on entropy and TE(LBP) on noisy images – Gaussian (Top) and Speckle (Bottom).

3.4.3 Impact of kernels conditioning on image descriptors

In the previous sections, we illustrated we studied the effects of random Gaussian kernels on traditional image entropy as well as TE(LBP) in US images. We found that the addition of noise to images have an impact on the statistical distribution of LBP landmark groups, and the Gaussian kernel acts to suppress these changes to some extent. The condition number of convolution filters is a well-known algebraic descriptor that relate to the sensitivity of their action on image patches as a result of a certain level of pixel value perturbation of the sort caused by the addition of noise. Filters with low condition numbers are less sensitive to data perturbation. Hence, we are interested in the range of condition number of kernels that causes low instability of TE(LBP) in the presence of noise. Such investigations are useful for understanding the factors that influence robustness of deep learning models and absence/presence of overfitting.

The condition number, $\kappa(A)$, of a kernel is an important factor that can affect the performance of a Gaussian filter, especially when applied in the spatial domain. When a well-conditioned RGF is applied to an image, it can help to preserve the texture features by smoothing the image in a controlled and stable manner. The filter will tend to blur out noise and high-frequency information while retaining important texture features at lower spatial frequencies. In contrast, when an ill-conditioned RGF is used, it can lead to the loss of important texture features or the introduction of artefacts. An ill-conditioned filter can lead to unpredictable and unstable smoothing, which can cause texture features to become distorted or blurred beyond recognition. In some cases, an ill-conditioned filter can also introduce artificial texture features into the image that do not exist in the original data. This can be a significant problem in applications where the image contains a high level of noise or where the desired output is a high-quality, noise-free image. In practice, the condition number of a Gaussian kernel is independent of the standard deviation of the generating Gaussian function, but we expected that the use of well-conditioned kernels preserve important texture features in US images when a patch of the image changes or an additional noise is introduced, while the use of ill-conditioned kernels will introduce artificial artefacts and distortions.

Figure 3.19, illustrates these effects for an input US image. We selected two distinct 3×3 kernels, one well-conditioned with a low condition number, and the second is ill-conditioned with a high condition number. We added a small amount of Gaussian noise with $\mu = 0$ and $var = 10^{-5}$ to the images then encode them to the LBP domain. On visual inspection, both kernels, produce the matching convolved images to the original ultrasound.

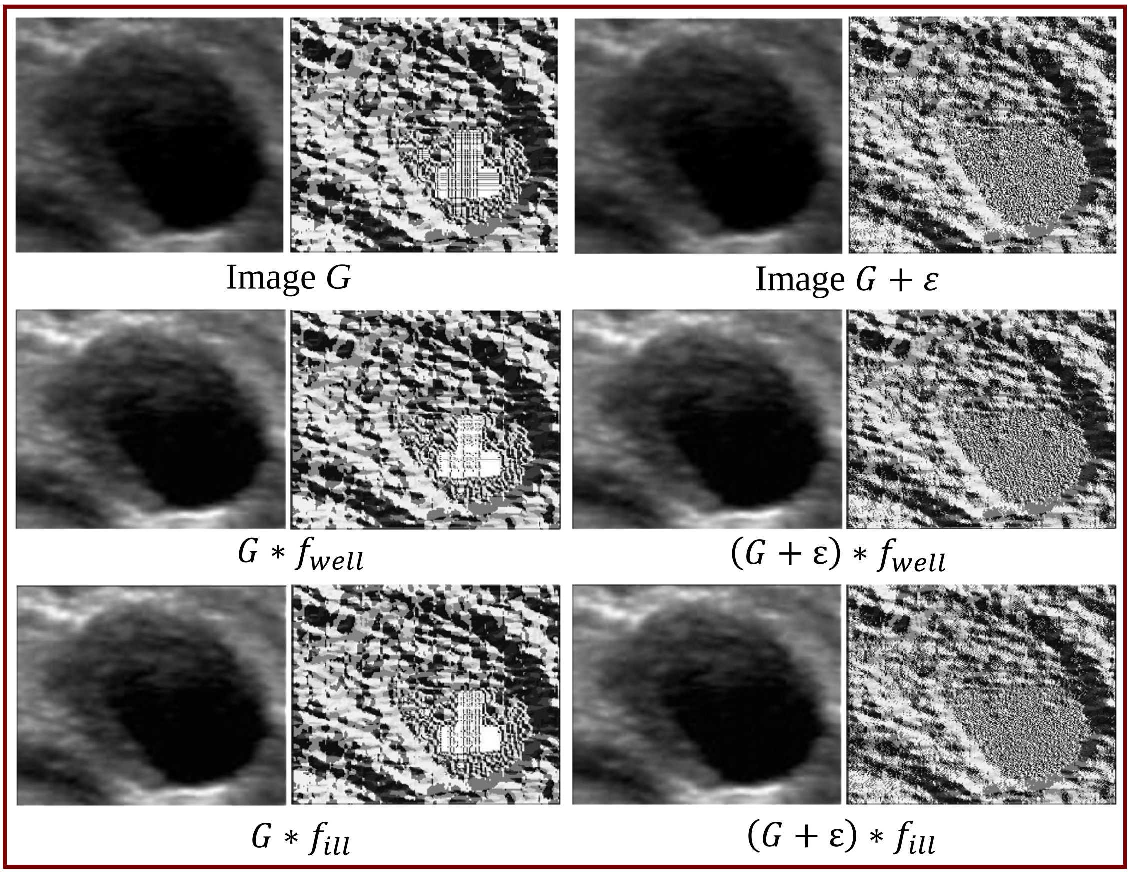


Image G

Image $G + \varepsilon$

$G * f_{well}$

$(G + \varepsilon) * f_{well}$

$G * f_{ill}$

$(G + \varepsilon) * f_{ill}$

Figure 3.19: Illustration of original and convolved images and their corresponding LBP transformation when applying well- and ill-conditioned filters.

The LBP patterns for these images with well- and ill-conditioned filters are showing similar patterns to the original unfiltered images. However, the spatial distribution of the LBP landmarks seem to undergo some hidden changes in different regions. To support this assertion, we extract R_1 and R_2 rotations in G_1 group (2-transition), then build the simplicial complex at different D distance thresholds for all six images (see Figure 3.20). The number of connected components and the number of holes (*#connected components*, *#holes*) in dimensions zero and one, are displayed below each image.

It is evident that the addition of noise, with or without convolutions, increases the *#connected components* due to the change in the LBP landmark groups as shown in Subsection 3.4.2. Therefore, there is a change in the rate at which these numbers decrease relative to the threshold distance increase with the addition of noise, with or without convolutions. In all cases, the number of holes fluctuates with different patterns, but more importantly, these changes occur in different regions of the ultrasound scans.

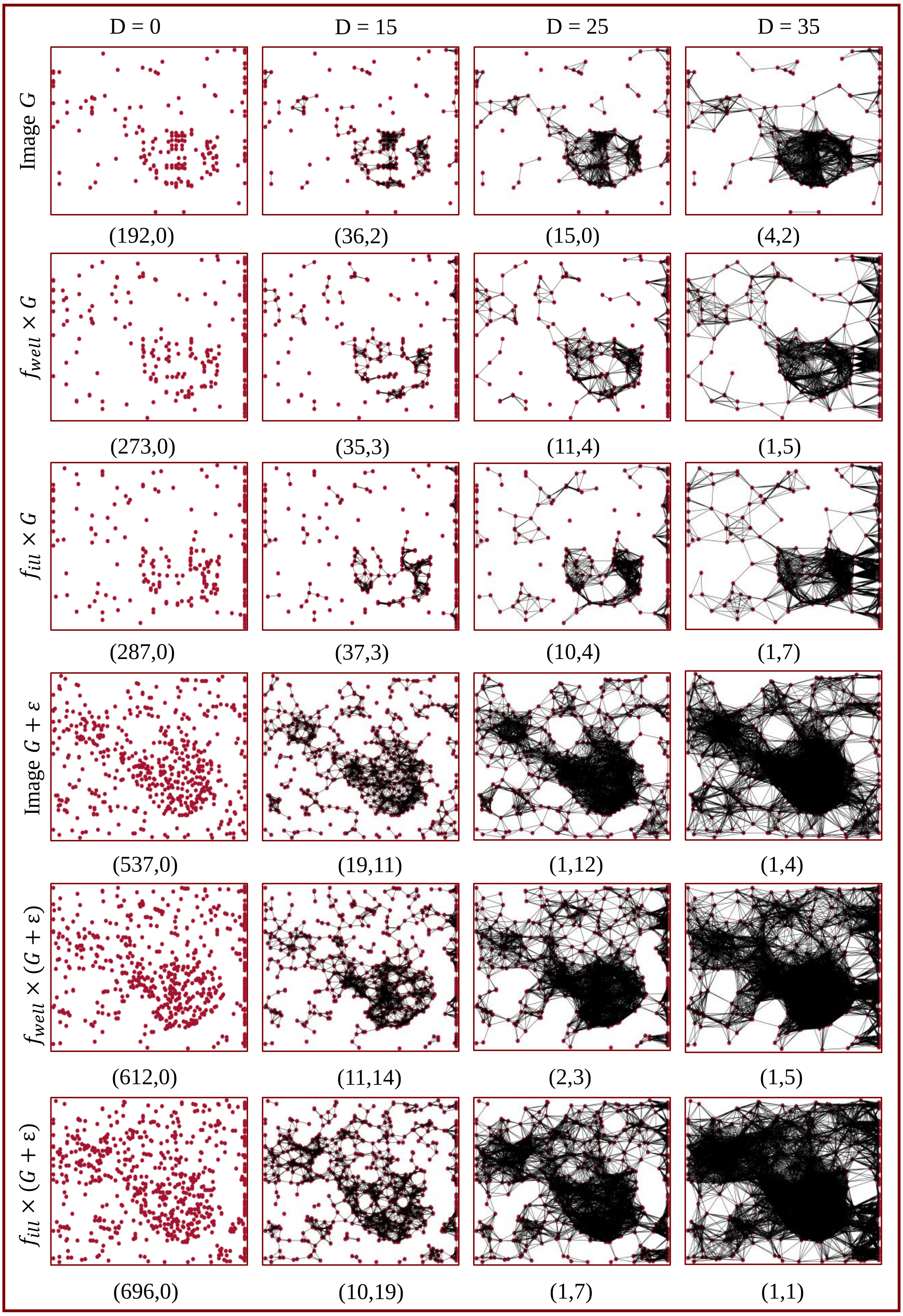


Figure 3.20: Changes to the spatial distribution of original and convolved US images with/out Gaussian noise ε, with selected well-/ill-conditioned filters.

These results support the hypothesis that the properties of the Gaussian kernels can impact the preservation or distortion of texture features in ultrasound images of various types of tissue. The characteristics of the texture features in an image can be affected by the condition number of these filters as it can impact the texture features in the resulting image. Investigating other algebraic properties of the kernels and their topological behaviour may provide more understanding of their impact on the performance of CNN models of analysing US images.

3.5 Conclusion

The investigations conducted in this chapter aimed to gain a better understanding of ultrasound image properties and the impact of random Gaussian filters on the information content of the US convolved images with presence/absence of noise. Although, different convolutions were found to have different impacts on conventional entropy image descriptor of US images of different types of tissue but exploiting these differences may be not straightforward. The statistical analysis of the LBP texture landmark groups revealed significant differences between texture contents in US images and those in natural face images. This observation is a kind of warning not to expect image analysis models, trained on natural images, to generalise with ease and high performance when deployed for US image analysis. Furthermore, it motivated the introduction of an effective LBP texture-based entropy (TE(LBP)) to quantify texture feature content. Our simple observational illustrations of the impact of image resizing, presence of noise, and condition number of convolution kernels on some image descriptors point to the importance of more research into the link between robustness against noise and the conditioning of convolution kernels. Furthermore, these studies indicate that persistent homology plays an important role in understanding the effect of convolution layers on the discriminating power of texture features. Deep learning models such as CNN involve large datasets of convolution filters, organised in different layers which involve other image processing procedures such as activation function, down-sampling, and normalisation. The performance of such models will be influenced by the combination of the above mentioned method. It is useful to determine if the impacts of the various convolution filters deteriorate or improve by these additional procedures. In the next chapter, we shall study these issues.

Chapter 4

CNN Models for US Image Analysis

The rapid advances in computer vision as a result of using Convolutional neural networks (CNN's) have demonstrated, beyond any doubt, the richness and power of CNN tools. However, interpretability of CNN decisions is far from ideal and present a serious hurdle in the much urgent critical tasks of embracing AI in medical image analysis. Researchers have long identified some shortcomings of CNN, even for natural images, are the inability of state-of-the-art CNN models trained with tens of millions of images to generalise to unseen data as well as robustness against data perturbation due to presence of noise. Moreover, various approaches have been proposed to overcome/reduce the severity of these challenges for datasets of natural images, and many highlighted the importance of TDA properties of convolution filters in this respect. In light of what we learned in the last chapter about the significant differences between nature of US image contents and those of natural images, this chapter is designed to investigates these issues and approaches for datasets of US tumour scan images the availability of which present an added constraint.

We first, in Section 4.1, review recent research relevant to identifying various approaches to designing CNN models that overcome/reduce the severity of the above challenges. In Section 4.2, we test the performance of several state-of-the-art CNN models in transfer learning mode for classification of US images, besides testing the ability of these models to generalise to an external dataset and to be robust against tolerable level of noise. Section 4.3 is dedicated to understanding the effect of CNN training procedures on US image entropy content as well as on the algebraic and topological properties of the convolution filters pre and post training. Section 4.4 is concerned with the learnt features during training, while Section 4.5 investigates the widely practised filter pruning as a mean of controlling CNN architecture complexity.

4.1 Review of existing work linking TDA with CNN

Linking TDA to understand decisions of machine learning models of natural images can be traced back to the work of G. Carlsson et al., in [86, 87], that investigated the geometry of the space of small (3×3) normalised natural image patches of ***high contrast*** and established that it is topologically equivalent to that of the 2-dimensional non-orientable ***Klein Bottle*** manifold. Originally, the interest in the space of natural image patches arose in relation to investigating the non-Gaussian structures of natural images, [88]. Moreover, CNNs for image analysis may be constructed directly using TDA of the graph structure on the grid of pixels, without any additional information. G. Carlsson and R. Gabrielsson, [87], consider this TDA approach as a powerful source of methods for constructing CNNs for any data sets that admit notions of distance between features that can be used as nodes in a graph with connections restricted by distance proximities.

The review of recent literature revealed several research contributions that aim at combining TDA and its tools with Deep learning neural networks for image analysis using a variety of approaches. Although, all these approaches target analysis of natural images they have lots of synergies with our current investigations and provide valuable guidance for future investigations for developing customised efficient CNN models for analysis of US images that are robust against data perturbation and avoid overfitting. Most attempts of combining TDA and its tools with the learning process and decisions of CNN models developed for image analysis are implicitly aim to contribute to one or more of the specific difficult challenges of (1) interpreting CNN decisions, (2) maintaining robustness against data perturbation, and (3) ability to generalise to unseen image datasets. The momentum for research on combining TDA with deep learning began to escalate since 2019, (e.g. [89–97]), and mostly develop innovative procedures to construct simplicial complex topologies with the various learning related structured components of CNN architecture, the features of which can be summarised by persistent homology tools. In terms interpreting CNN decisions these approaches are yet to be formalised but it is certainly different from existing work on explainable AI for image analysis which are focused on identifying image regions that contribute to decisions [98].

Recall that the main structured matrices/graphs ingredients of CNN architectures in both the multiple convolution layers of the ***Feature maps extractor module***, and those of the ***FCL classifier*** module are adjusted by the elaborate training procedures based on the gradient descent backpropagation to minimize the loss function. The various ingredients are admissible to topological analysis, perhaps in more than one way. In Chapter 2, we described and reviewed the various ways of constructing simplicial complex topologies to be associated with point clouds in $\mathbb{R}$, images and graphs/networks that would be amenable to PH analysis. Matrices of weights can be either represented by point clouds by flattening to 1D vectors, but mathematically large matrices can be treated as images. We noted that new ways of constructing simplicial complex topologies for graphs/networks are emerging, but a good starting point for these approaches is to represent a graph by the simplicial complex formed by the $(k+1)$-clique subgraphs as its k-simplices. Analysing the topological features of graphs/networks can benefit from the various types of *filtrations* defined for different types of directed/undirected and weighted/unweighted networks. For a detailed review and comparisons of the various filtrations, see Mehmet E Aktas et al. [99]. G. Jorgenson et al., in [96], point out that while coarse topological data summaries endow data with stability (and other desirable properties) it could inhibit learning fine scale features that have considerable discriminatory power. For image data, the authors propose convolving the original images with a collection of Random Gaussian filter to enhance the discriminatory power of TDA features computed from the sublevel filtration of simplicial complex representation of the data samples. In section 4.4, we shall present similar results of more extensive experiments we conducted during the early part my PhD project and appeared in [19] on the discriminatory power of LBP-based landmarks PH features when applied on US images pre and post convolving with the pretrained AlexNet filters through the different layers.

R. Gabrielsson et al., in [89], demonstrate that the weights of convolutional layers at depths from 1 through 13 learn simple global structures. They investigated the changes in simple structures during the training stage by analysing the spaces of spatial filters of the convolutional layers over a thousand CNN models applied to the well-known natural image datasets (MNIST, CIFAR-10, and SVHN datasets).

Algebraic topological methods can help in understanding how the connections between neurons are structured and how it affects the network's performance. Accordingly, some existing work focused on analysing deep neural networks using TDA PH-based features to analyse the directed graph structure of the FCL hidden layers and attempted to link these features to the performance of the corresponding models. S. Chowdhury et al., in [90], introduced two different types of directed persistent homology schemes (PathPH and FlagPH) to characterise feedforward deep neural networks, in general, and the FCL of existing CNN architectures in particular. These types of directed PH schemes, aim to understand the structure of the network and how information is processed and transformed as it passes through the layers. The directed flag homology of deep networks is computed by determining the simplicial homology of the underlying undirected graph, and explicitly using Euler characteristic computations. The path homology of these networks is non-trivial in higher dimensions and relies on the number and size of the layers in the network. M. Gabella, in [92] designed a feedforward neural networks to train on the MNIST dataset and track the weight evolution through TDA methods to study how structure emerges in the weights during training.

To gain insight into how DCNNs such as GoogLeNet, ResNet, and BERT achieve their optimal performances, A. Rathore et al., [91], developed a visual topological tool called TopoAct to enable exploration of topological summaries of activation vectors. TopoAct displays the shape of the activation space, the organisational principle behind neuron activations, and the relationships of these activations within a layer. Furthermore, in [95], TDA methods are used to investigate the generalisation gap, where the approach involves computing homological persistence diagrams of weighted graphs that are constructed based on neuron activation correlations observed during training, with the aim of capturing patterns that are associated with the network's generalisation ability.

T. Lacombe et al., in [93] proposed a post-training method that assesses the reliability of predictions by investigating the entire network's topological properties, rather than the common approach of restricting the investigations to the j final layer. This method assigns a Topological Uncertainty score to each new observation, which can be used for trained network selection, Out-Of-Distribution detection, and shift-detection.

Around the same time Yang Zhao & Hao Zhang, in [94], proposed topological-based entropy to quantify the information content of CNN unit as a mean of quantifying the status of the Unit. A unit is defined by the highly activated positions in the feature map of an image output by a CNN model, modelled as a weighted graph, and its topological features are obtained from the simplicial complex formed by its k-clique subgraphs, (Aktas et al. [99]). The topological feature entropy is defined in terms of the distribution of birth times of the topological features per class in the corresponding simplicial complex. It is expected to accurately indicate status for units in different networks. They showed that feature entropy shares trends with loss during training and decreases the deeper the layer is, and investigating their values only on training data could distinguish between networks of different abilities for generalisation. In Chapter 3, of this thesis, we introduced the concept of image texture entropy that complements the concept of topological feature entropy. In subsection 4.3.1, below we shall investigate the effects of CNN training in transfer learning for US tumour classification on both conventional as well as texture entropy of the training images of different tissue datasets.

Inspired by the work in G. Carlsson [100] on the geometry of high contrast natural image patches being that of the Klein bottle E. R. Love, in [97] introduce a new topological convolution neural network (TCNN) architecture where layer structures exploit the prior knowledge on natural image data with topological data analysis (TDA). They define the space of image ***Klein*** filters, closely related to a subfamily of the Gabor filters, based on a graph discretisation of the Klein bottle. The Klein filters form a template for new layers and help produce sparse feature maps. The experiments on image/video data yield significantly improved performance of TCNNs compared to conventional CNNs.

The research project in this thesis was designed to investigate the use of CNN models for the analysis of US tumour scan images for different tissue/organ types. Ultimately, we should aim to develop CNN models that (1) can reliably classify benign and malignant masses from the ultrasound images, (2) is robust against tolerable data perturbation, and (3) able to generalise to unseen images. Although, interpreting the sought after model decisions is even more essential for the critical medical image analysis, that challenge was not deemed as urgent as the above ones. Notwithstanding the significant differences, we uncovered in the last chapter, between contents of US image and natural images, there are obvious synergies between our objectives and the motivating objectives of the above reviewed publications. Recognising the significant role played by the convolution filters in determining the extracted feature maps to be learnt by CNN models for image analysis, it was essential to link the algebraic and topological properties to their effects on the content of US images through the different convolution layers. The fact that the weights of the convolution filters are updated by the backpropagation procedure through an elaborate iterative training with image batched, monitoring the changes in the filters properties is equally essential.

The current common practice in designing CNN models for US (or medical) image analysis, is to use existing CNN models, trained on natural images, in transfer learning mode which involves adding a new layer to train the pretrained model on a dataset of US images. Accordingly, we need to determine the effect of this additional training on the extracted US feature maps as well as on the algebraic and topological properties of the output transfer learnt filters. In the last chapter, we investigated the various effects of convolution filters on US image features including spatial and textural entropy. The feature maps obtained during CNN architecture elaborate training schemes (originally or in transfer learning mode) deploy large sets of multi-channel Gaussian convolution filters in different number of layers each layer of which involves different normalisation, activation functions, and possible down-sampling applied on the convolved images before passing lower-resolution image maps into the next layer. In this chapter, we investigate how these added processing steps impact the effect of each convolution layer on conventional/textural entropy and the spatial distribution of texture features (using LBP landmarks). We shall also investigate the algebraic and topological properties of convolutional filters in terms of sensitivity to small changes using condition numbers and their spatial distribution by TDA tools.

4.2 Performance of pretrained CNN for US images

Due to lack of availability of sufficiently large datasets of US images, the common practice in designing CNN models for US (or medical) image analysis is to use existing CNN models, trained on natural images, in transfer learning mode which involves adding a new layer to train the pretrained model on a dataset of US images. This section is designed to determine the commonly perceived level of overfitting and lack of robustness against image data perturbation of CNN models of US image analysis. For that, we first implement several commonly used state-of-the-art CNN architectures pretrained on natural images in transfer learning mode for analysis of US breast scan images. These SOTA architectures selected are AlexNet [29], VGG16 [38], ResNet18 [39], and EfficientNet [40]. These models were trained and tested exclusively on Tend-D breast ultrasound images, as mentioned in section 4.2.1, to ensure consistency and equitable comparison. We shall test the performance of these transfer learnt models with regards to: (1) discriminating Benign against Malignant masses, (2) ability to generalise its performance to an external dataset, and (3) robustness against certain level of noise-caused image data perturbation.

4.2.1 Classification performance

In our experiments, we shall use the internal Ten-D recorded database of US breast tumour dataset (a class balanced set of 524 images) for training and testing the selected CNN models in TL[1] mode. To achieve models of improved performances, we adopted different image augmentation schemes to increase the number of training and testing dataset. The Ten-D/Buckingham research team adopted known data augmentation methods, and developed new schemes, for this purpose and each cropped RoI ultrasound image was used to generate 7 additional image versions. These methods include geometric methods such as mirroring and rotation with degrees 90, 180, and 270, respectively, and singular value decomposition with 45%, 35%, and 25% ratios of the selected top singular values. As a result, our training and testing dataset consists of 4192 class-balanced US RoI images.

We followed the common practice in determining the models performance in terms of classification accuracy, sensitivity, and specificity computed as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.1}$$

$$Sensitivity = \frac{TP}{TP + FN} \qquad Specificity = \frac{TN}{TN + FP} \tag{4.2}$$

where
True positive (TP): Malignant case correctly classified,
False positive (FP): Benign cases incorrectly classified as malignant,
True negative (TN): Benign cases correctly classified, and
False negative (FN): Malignant cases incorrectly classified as benign.

In these and other experiments, we follow the training and testing protocol that randomly select 80% of the samples (i.e. 3352 images) for training plus validation and the remaining 20% (840 images) are used for testing the performance of the trained models. We tried other training/testing ratios, but this ratio achieved a more desirable performance during training (i.e. resulted in the lowest difference between classification and validation accuracy) as a result of having wider range of data samples in the database.

We used MATLAB[2]deep learning toolbox to implement pretrained CNN models and test the performance of the selected CNN models. During the training we

[1] Replacing the last fully connected layer and re-train the model on ultrasound datasets, it is also known as fine-tuning.

[2] Deep learning toolbox – MATLAB (version R2019a – R2022a), MathWorks Inc.

used the default parameters, but we repeated the experiments for three different batch sizes: 32, 64, and 128. We also tried different numbers of epochs. Below, in Table 4.1, we only present the results for the 64 batch sizes with 20 epochs.

Table 4.1: CNN models Performance in transfer learning mode for US breast scans.

TL models	Validation Acc	Test Acc	Sensitivity	Specificity
AlexNet TL	97.16	96.53	95.95	97.14
VGG16 TL	96.87	96.79	97.86	95.71
ResNet18 TL	97.31	94.17	95.00	93.33
EfficientNet TL	88.06	89.4	91.19	87.62

All CNN models achieved a high test accuracy ($> 89\%$) with reasonable 2-3% difference between sensitivity and specificity rates. AlexNet, VGG16, and ResNet18 models attained notably higher performance compared to the EfficientNet model. Notably these performances are suboptimal to that achieved on natural images.

Using batch size 32 causes unstable training i.e. the classification accuracy and validation rates suffer from sudden change/shift during the training and the model seems to be fitted specifically well for the training set. In contrast in the case of batch size 64 the classification accuracy and validation are more balanced and stable. In the other case of using batch size 128, the model is underfitted compared to batch size 64. These assertions can be deduced from the training output curves of performance and loss functions (for instance as shown in Figure A.1).

To test if similar performances hold for US scan images of other than breast tissue, we repeated the experiment for training and testing the pretrained AlexNet in TL mode using a dataset of US bladder tumour scan images. However, the dataset for bladder tissue was considerably smaller compared to that of breast tissue, containing only around 200 images for both classes, without augmentation. We followed the same training/testing protocol and found that the Training accuracy (95.15%), Validation accuracy (86.67%), while Test Accuracy = 76% with a big gap between Specificity = 93.33% and Sensitivity = 60%. Note the behaviour of the training performance and loss functions displayed in the following Figure 4.1. Obviously, lower validation and test performances are due the rather much smaller size training and testing samples compared to the breast dataset (with augmentation) case.

Similar patterns of results were achieved, in terms of fluctuation of learning

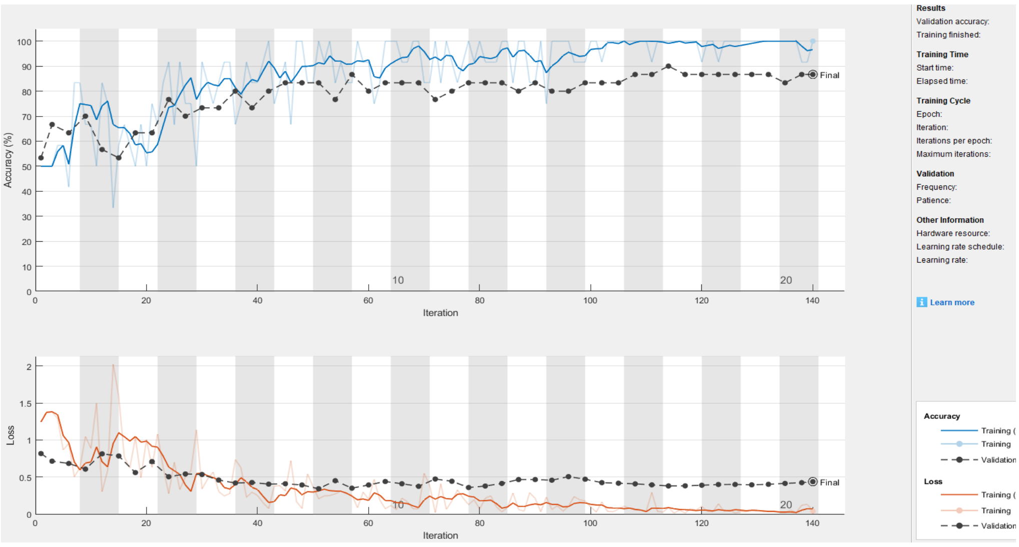


Figure 4.1: Classification accuracy on Ten-D dataset, with applying two levels of Gaussian (GN) and Speckle (SN) noises.

and loss rates, when we used Ten-D Liver ultrasound dataset. Even though the liver dataset is slightly larger than the Bladder dataset.

Achieving high test accuracy in the upper 90%s, though very impressive, is not enough to adopt as a credible CNN model for US tumour image classification. Recall that the dataset, prior to augmentation, were recorded in a single clinical centre and by no mean constitutes a random sample of the global population of US images recorded using different US devices and labelled by differently trained radiologist who may follow different clinical practices. In the rest of this section, we investigate robustness of the almost best performing transfer Learning CNN model (AlexNet) against tolerable perturbations followed by their ability to generalise to an external dataset.

4.2.2 Robustness against natural perturbation

To evaluate the robustness of each CNN model, we deployed a simple adversarial setting where we tested the classification performance on degraded versions of the testing dataset. Similar to the noise distortion factors in Chapter 3, we used Gaussian and Speckle noise with varying levels of degradation to assess the models' performance at stages where the CNN models were expected to be highly or poorly sensitive to small perturbations [3]. Specifically, we applied Gaussian noise with μ=0 and var = 0.001 to 0.01 referred to as GN_1 and GN_2, respectively. We

[3] The level of noise perturbation affects both the pixel value distribution and their descriptors, as established in Subsection 3.4.2

also added speckle noise to the testing set with μ=0 and var=0.01 to 0.1 which will be referred to as SN_1 and SN_2, respectively. Figure 4.2, below, shows the significant drop in accuracy indicating lack of robustness against the perturbed data with both types of noise. AlexNet and VGG16 responded better than ResNet18 and EfficentNet-b0 to the image distortion and the accuracy dropped by 4-6% for AlexNet and VGG16, whereas for ResNet18 and EfficentNet-b0 the accuracy dropped by 10-30% for the minimum noise levels.

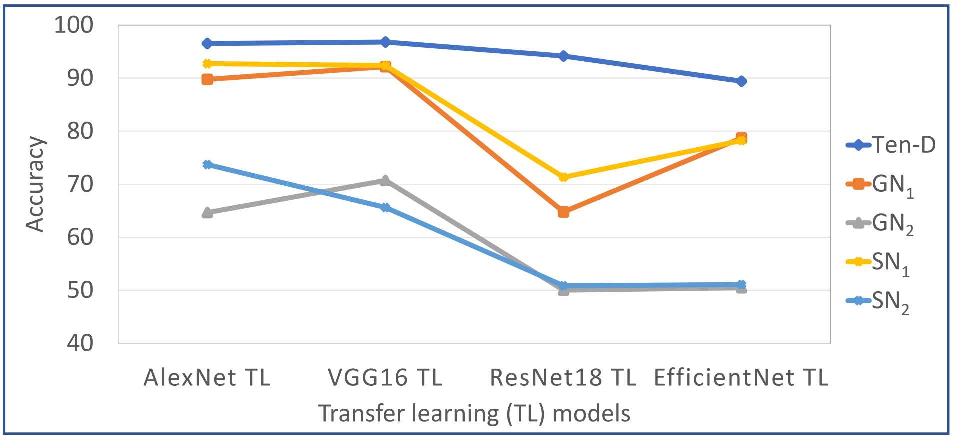


Figure 4.2: Classification accuracy performance of CNN models on Ten-D dataset and robustness testing against noise levels (GN_1, GN_2, SN_1 and SN_2).

4.2.3 Generalisability performance

To check the inherent overfitting problem to the training dataset in the TL models, we utilise a publicly available external breast ultrasound dataset called BUSI dataset [101]. BUSI dataset served as a benchmark to estimate the generalisability and performance of the CNN models beyond the training dataset in our experiments throughout the thesis. Figure 4.3 illustrates the significant decrease in accuracy by 8-20% on 160 cropped RoI, clean, and balanced class images. The accuracy drops further when it is evaluated on unbalanced classes.

However, it is worth remembering that the performance of any deep learning model can be affected by many factors such as the (depth, width) architecture parameter and/or hyperparameters selection; distribution of quality training images; size of training and validation datasets; and distribution of the testing population (see Subsection 2.1.3). The purpose of the above experiments was beyond finding the optimal performance for the selected CNN models. In fact the results have confirmed that adopting CNN models pretrained on natural images, for analysis of US images, suffer from lack of robustness against noise and

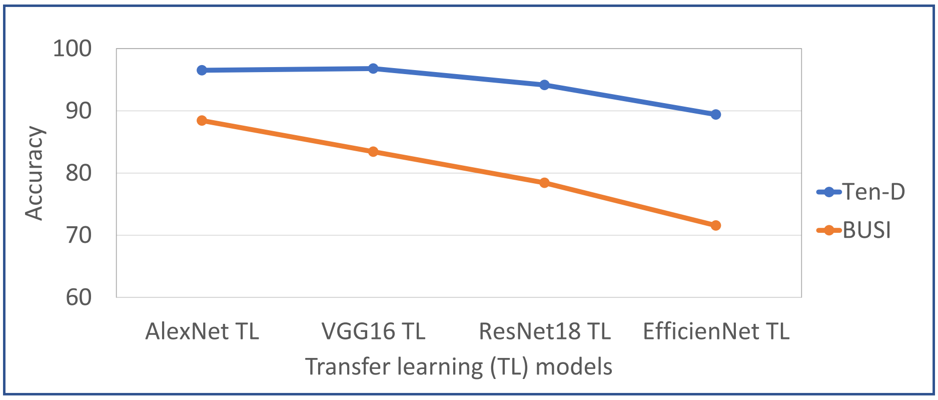


Figure 4.3: Classification accuracy performance on Ten-D and BUSI datasets.

inability to generalise to unseen data. Recall that in Chapter 3, we have shown that texture landmarks distributions of US images are notably distinct from those of natural images. This together with the scarcity of US images of standardised image "quality", may partially explain the above observed behaviour of the investigated CNN models in the presence of noise or when tested on unseen datasets. For wider explanations, however, we need to identify CNN model's training components/parameters that contribute to their lack of robustness and poor generalisation. We focus on the effects of the convolutional layers processes and parameters on the output feature maps of the training dataset images. These effects need to be expressed in terms of the dynamic behaviour of algebraic and topological properties of the convolution filters during training.

4.3 Training-caused effects of convolutional layers

This section, is concerned with the CNN model training processes/parameters that have an impact on the feature maps as they input into the fully connected Layers. we study the effect of convolution layer pretrained filters on grayscale entropy and texture-based LBP entropy on ultrasound images. Furthermore, we investigate the algebraic and topological properties of these filters as well as post retraining on US images. These investigations are expected to reveal the challenging complexity of interpreting the CNN models decisions. The related investigations and experiments have been done for the various CNN models used in Section 4.2, but here we shall only present the results for the AlexNet not only for being well performing architecture but for having only 5 convolution layers with filters of different sizes.

4.3.1 Effect of pretrained filters on entropy

In the previous chapter, sections 3.2 and 3.3, we evaluated the impact of randomly selected Gaussian kernels on original and convolved images in terms of grayscale entropy and texture-based LBP entropy. However, when using Alexnet, as well as other CNN architectures, in transfer learning mode the convolution filters used to initiate retraining of US images are the ImageNet pretrained filters and the retraining may result in filters that differ significantly in terms of their effects on image contents as well as in terms of their algebraic and topological properties. In order to establish these effects in relation to the results in the last section, we confine our experiments to tumour class dependency effects.

Note that the first convolution layer of AlexNet deploys 96 3-channels set of pretrained filters in the first layer of size 11×11 with stride 4 (i.e. overlap of 7), see section 2.1.2 for more detail. Together with the other steps, this layer reduces the size of any input image to 55×55.

We shall first demonstrate, in Figure 4.4, the visual effects on a sample image during the convolution process with an example of a pretrained filter w from AlexNet, using the adapted bias b, and the ReLU activation function σ. For the visualisation purpose, all images appear with the same size.

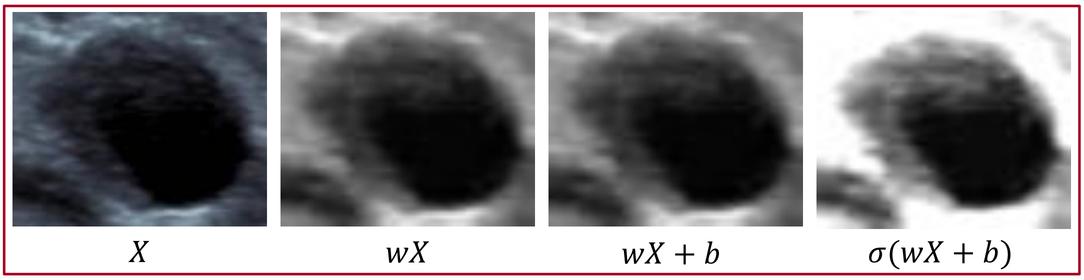
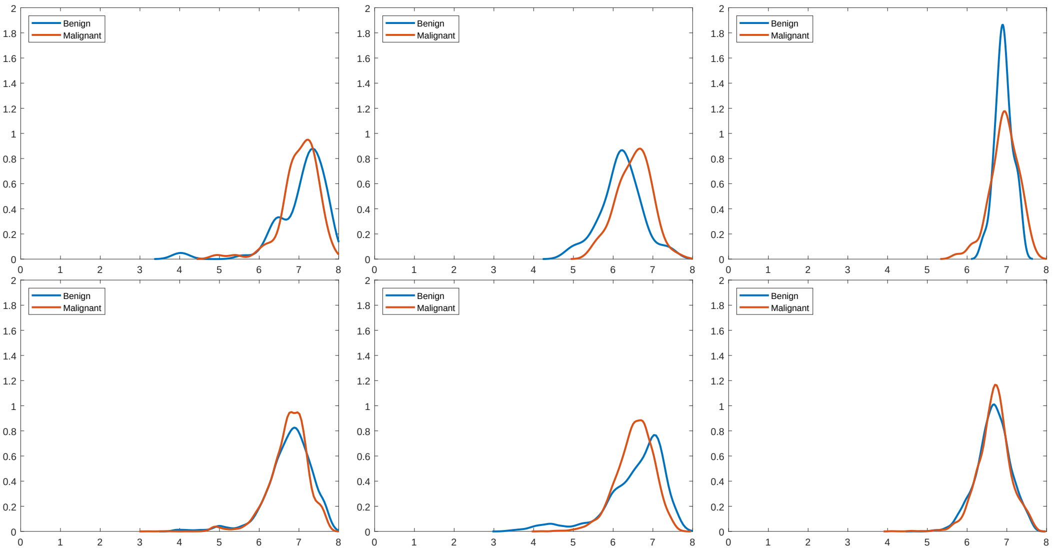


Figure 4.4: US image convolved with pretrained filter w, bias b, ReLU activation σ, prior to max-pooling.

Next, we examine the effect of these pretrained filters in the convolutional layer settings by compute the entropy values of 75 original images per class with a different sampling selection than the samples used in chapter 3 experiments. In these experiments, we use conventional and texture-based LBP entropy measurements as a metric to quantify the information content of each convolutional layer of AlexNet architecture. We observed a similar pattern in terms of entropy values across layers taking into account that size of feature maps are getting smaller due to convolution overlapping and max-pooling. For simplicity, we discuss the first convolutional layer after operations within the layer such as the convolution and adding bias, ReLU activation function, local response normalisation, and max pooling. For the convolutions and adding biases, we use the 96 pretrained filters on the US images that result in producing 7200 convolved images per class. Distribution of entropy values of original and convolved images, are displayed in Figures 4.5 and 4.6, using both entropy metrics for bladder, liver, and breast ultrasound scans.

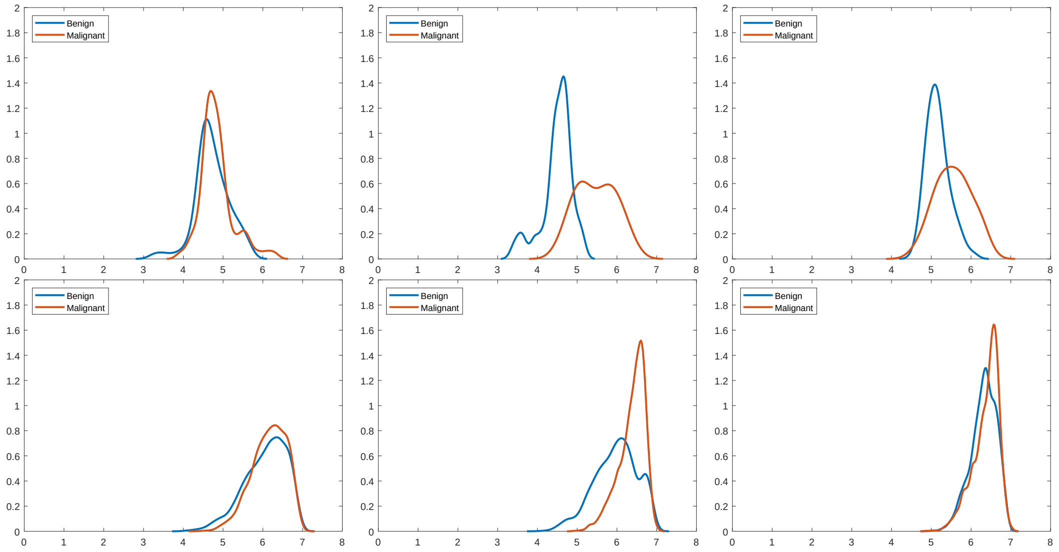


(a) Bladder USI (b) Liver USI (c) Breast USI

Figure 4.5: Distribution of conventional entropy 1^{st} layer: original images (top) and convolved images (bottom).

For all tissue types, the minimum entropy values in all original US image classes are shifted by just above 1 unit after applying convolution. The application of the local response normalisation did not result in any significant loss of information as it normalises the local contrast among every 5 feature map channels i.e. brightness normalisation. The ReLU activation replaces all negative values with 0 and keeps the positive values and consequently most of the feature maps become redundant as the entropy values become zero or below 0.5. A small percentage of non-zero entropy value feature maps are passing the texture features to the next stage for down-sampling. The cause of negative values mainly is due to the fact that biases are mostly negative in the first convolutional layer. After applying max-pooling, the entropy values are increasing and/or decrease as a result of selecting the maximum value out of four values for each window.

The results in Figure 4.6, show an understandable reduction in texture entropy by almost similar amounts for the 3 different tissue type in comparison to traditional entropy distributions. But the pattern of change in the distributions of texture entropy output by the first convolution layer, is almost similar to those



(a) Bladder USI (b) Liver USI (c) Breast USI

Figure 4.6: Distribution of LBP texture entropy pre- and post- CNN convolution.

corresponding to conventional entropy results in Figure 3.4 for both tissue classes.

Incidentally, both experiments show the futility of attempting to classify and discriminate tumour classes using either type of entropy as the sole criteria. Therefore, we need to investigate the effect of the convolution layers on the spatial distribution of texture features to understand the challenges uncovered in Section 4.2 with regards to the performance of the CNN models for distinguishing between malignant and benign tumours. This will be done in Section 4.4.

4.3.2 Algebraic properties of various CL filters

Prior to training, CNN convolution filters are initialised using random Gaussian matrices, with mean zero and small standard deviations, with a range of condition numbers. Fitting the model performance to the training set could contribute to more learnt filters to become ill conditioned while achieving high classification accuracy on the training, validation, and some testing sets that are drawn from the same distribution. Ill-conditioned filters may cause instability of model performance and perform poorly on unseen data or as result of slight changes in the input data. A. Sinha et al., [102], point out that ill-conditioned learnt weight matrix contributes to neural network's susceptibility to adversarial attacks. They proposed an orthogonal regularisation that is meant to keep the learnt weight matrix's condition number sufficiently low, and demonstrated its increased robustness to adversarial attacks when tested on the natural image datasets of MNIST and F-MNIST.

Accordingly, the algebraic properties of filters relevant to the objectives of this thesis must cover both the distribution of their condition numbers and their spatial distributions within their domains. W. Demmel, in [64], investigated the upper and lower bounds of the probability distribution of condition numbers of random matrices and showed that the sets of ill-posed problems including matrix inversion, eigenproblems, and polynomial zero finding all have a common algebraic and geometric structure.

Investigating the effects of convolution layers on texture features and their conventional entropy as well as spatial distributions are motivated by: (1) most DL analysis of US images use CNN models, developed for natural image analysis, are trained in transfer learning and (2) yet in chapter 3 we have already shown that the distribution of texture features in US images differ significantly than their counterparts in natural images. To clarify these issues we start with an example.

4.3.2.1 Illustrating example

To further explain our emphasis on filters condition numbers, we illustrate the impact of a relatively well- and ill-conditioned pretrained filters, selected randomly from the AlexNet setting, on a given US image before and after adding noise. This is motivated by the fact that ill-conditioned $n \times n$ matrices, considered as linear transformations of $\mathbb{R}^n$, are highly likely to map nearby vectors in $\mathbb{R}^n$ onto far apart vectors. The two pretrained first layer filters of size 11×11×3 were chosen to have on average (over the 3 depth-wise channels) condition numbers: w_1 <200 and w_2 $>$ $2{\times}10^4$. Both multi-channel filters were applied with stride 4 on the input US image and after adding Gaussian noise μ=0 and var=0.001, besides adding biases (b) and applying the ReLU activation function (σ). Visual examination of the resulting feature maps and their LBP descriptors with-/without noise, show obvious differences between the case of using the relatively well-conditioned filter and the case of the highly ill-conditioned one. Despite the obvious visible differences in the LBP texture contents between the original image and the noisy version, convolution with w_1 results in significant reduction of differences in texture contents. In contrast, the convolution with w_2 results in big differences between the corresponding texture contents both in quantity and in spatial distribution. These differences in the effects of noise can be explained by the known facts about the link between conditioning of filters and their actions as linear transformations on nearby vectors. In turn this helps illustrate sensitivity to noise by the conditioning of the filters.

This illustrating example may lead to predicting that (1) feature maps ob-

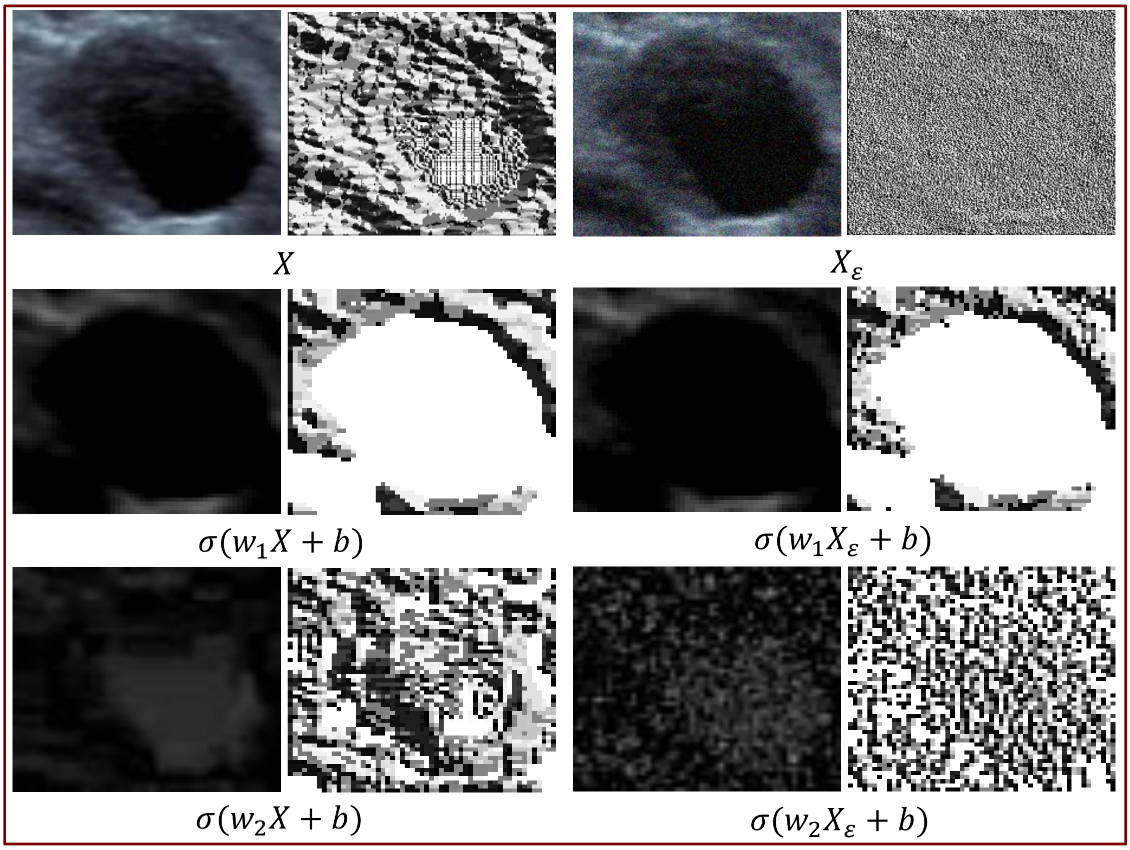
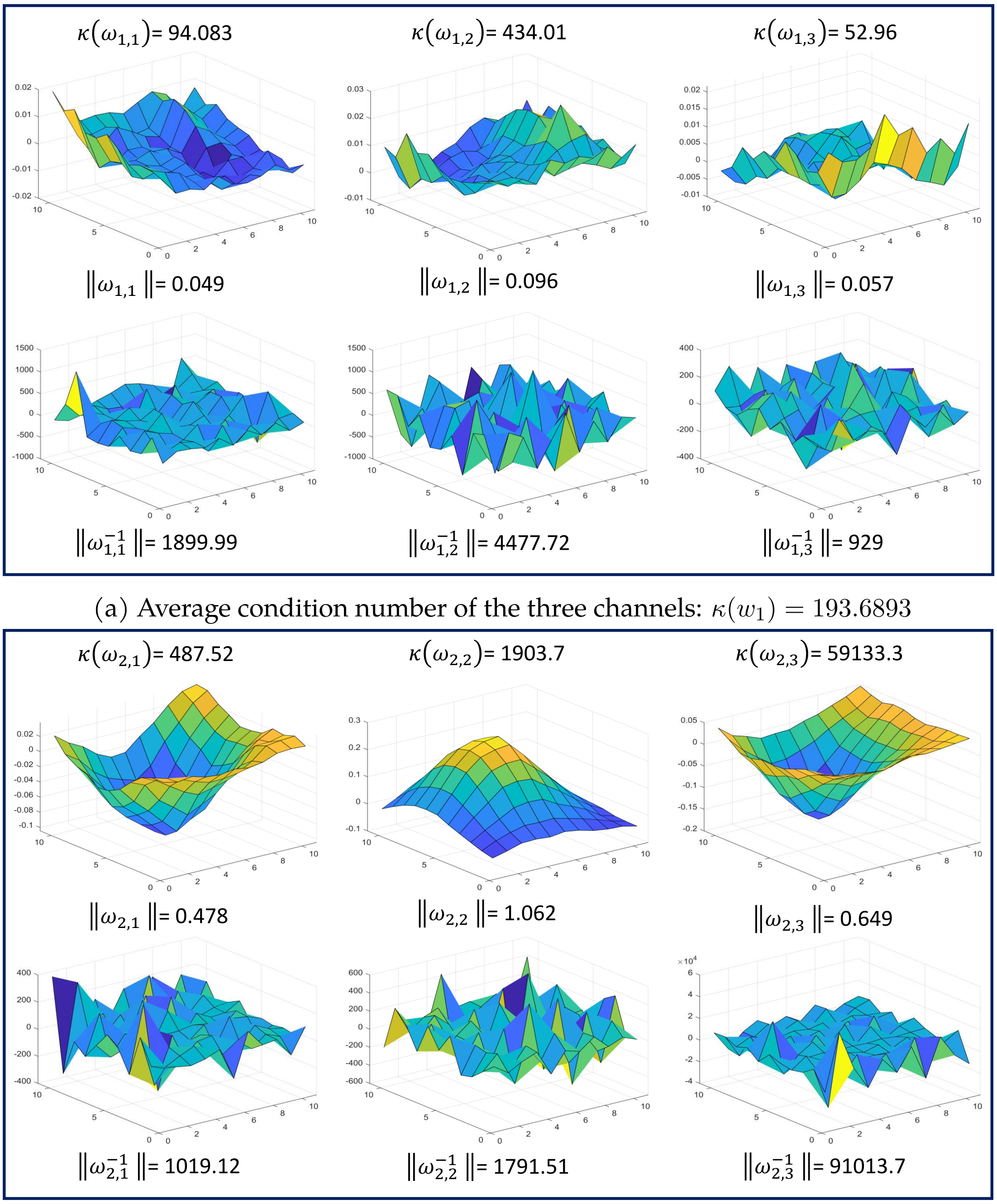

X
X_ε
$\sigma(w_1X + b)$
$\sigma(w_1X_\varepsilon + b)$
$\sigma(w_2X + b)$
$\sigma(w_2X_\varepsilon + b)$

Figure 4.7: Impact of well- and ill-conditioned pretrained filters, w_1 and w_2, on a US image with and without random Gaussian noise ε.

tained with reasonably lower condition number filters preserve the texture contents of the original US images, and preserve the shape of the tumour RoI, with or without noise, and (2) feature maps obtained with ill-conditioned filters exhibits instability and sensitivity to small changes. It may also illustrates a potential link between the distribution of filters condition numbers across the various convolution layers and the overfitting problem in CNN models. But the action of filters on images are influenced by many properties of the filters such as variation in their entries, their norms and the norms of their inverses. For the 11×11 pretrained filters w_1 and w_2, Figure 4.8 displays their 3-channel versions and inverses besides providing information on their norms and condition number values. For both filters: (1) there is a significant variation in the condition numbers of their channel versions, (2) increased condition numbers do not yield larger norms, (3) norms of inverses are significantly higher than the original filters, and intriguingly the ill conditioned filters have smooth entry shapes with one global maximum/minimum.



(a) Average condition number of the three channels: $\kappa(w_1) = 193.6893$

(b) Average condition number of the three channels: $\kappa(w_2) = 20508.1842$

Figure 4.8: Visualisation of the pretrained 3-channel first layer filters w_1 and w_2.

Although, the range of entries of the different channel filters are relatively small but their local maximum/minimum entries appear in different positions, indicating that convolving an image with the different channel filters result in learning different patterns that may not be reflected by their average condition numbers. Accordingly, the visual effects noted in Figure 4.7 may change if we use a different channels than the one used. Consequently, the above stated predictions are not valid and allowing different channel filters (as is the case for the pretrained AlexNet filters) to have significantly different conditioning numbers adds to complexity of interpreting CNN decisions. Recall that, the number of depth-wise channels beyond the first convolutional layer are significantly greater than 3 to be compatible with the number of filters in the previous layer.

4.3.2.2 Distribution of condition number of filter sets

To get the wider and more informative picture beyond that of the above illustrating example, we shall now investigate the algebraic and topological properties of three different sets of the CNN convolution filters: (1) initialised sets of filters, (2) filters pretrained on natural images, and (3) filters post-retraining on ultrasound images in transfer learning mode. We generated the initialisation sets, in a similar way to the original AlexNet[4], consist of Gaussian filters with μ=0 and σ=0.01. We imported the pretrained filters from [103], while the transfer learning filters were output when AlexNet were retrained on breast ultrasound images. First, we shall determine the distribution of condition numbers of these different sets of 96 3-channel filters. Due to the extremely wide ranging condition numbers, we show the distributions with different size binning in Figure 4.9 below.

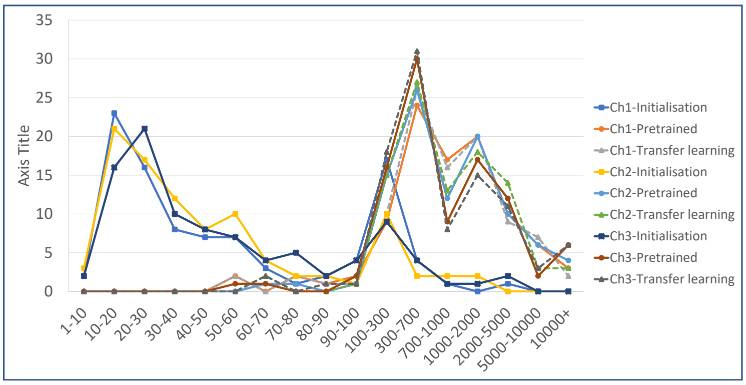


Figure 4.9: Distribution of condition numbers of the sets of 96×3 filters.

[4] Notably, the MATLAB pretrained model filters were initialised using Glorot initialisation [68] instead of the original setting ($\mu = 0$ and $\sigma = 0.01$) detailed in [29].

This chart show that various training has led to creating more ill-conditioned filters which inevitably could undermine the sensitivity and stability of CNN models. Majority of the initialised filters are among the lowest range of the condition numbers (<50) and no filters in the other two sets are in that range. While majority of the pretrained and transfer learnt filters have their condition numbers in the upper range (>700) in which only very few initialised filters belong. Table 4.2 below, show the level of variation in the condition numbers between the different channels of the 3 sets, as illustrated by the maximum condition numbers. Within each set, variation within the different channels get more significant: (*Initialised*: 3323, 1190, 4076), (*Pretrained*: 108612, 46049, 59133), and (*Transfer Learnt*: 49223, 46491, 226323).
Based on the final observations from the illustrating example, we assert that:

> *Due to the significant variations of condition numbers, not controlling filters condition numbers within the channels compounds the complexity of interpreting CNN decisions.*

In Figure 4.10, below, we present the actual changes in pretrained filters' condition number (in Log_{10} values) during the retraining of AlexNet on US images over 20 epochs. Clearly this reveals significant instability during training, perhaps as a result of variation in the training US image batches, and the final destination may not entirely reflect its behaviour. Only few filters are somewhat stable.

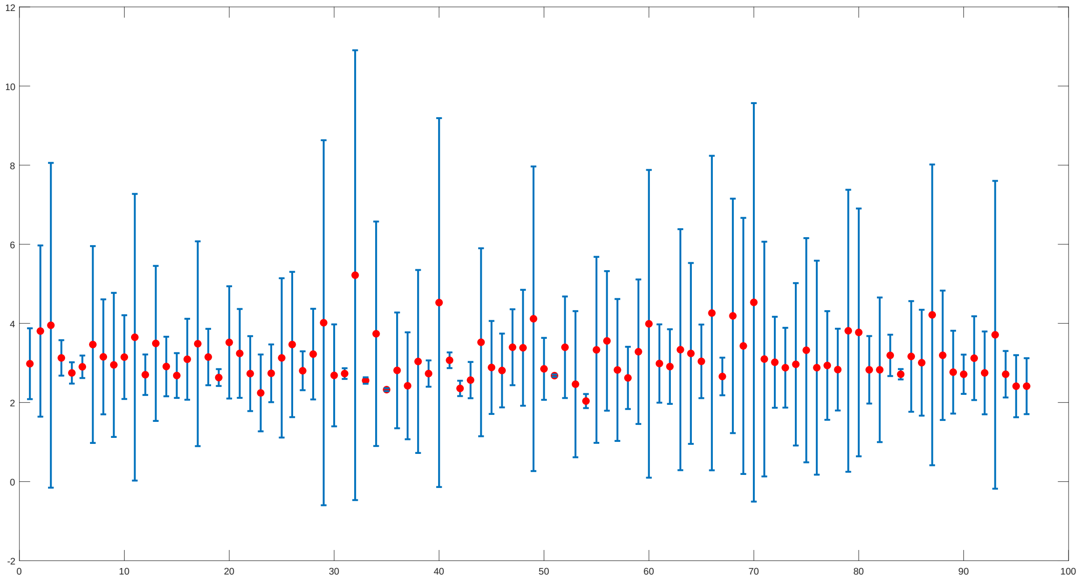


Figure 4.10: Condition Number fluctuation during retraining AlexNet.

We conducted an examination of the singular value distribution of filters to attain a deeper understanding of distinctions among initialised, pretrained, and transfer learning filter sets across various convolutional layers. The findings presented in Table 4.2 include the minimum and maximum values for each of the three filter sets. Notably, the table emphasises a significant difference in the smallest singular values across all three filter sets, whereas revealing a proximity between the largest singular values of the pretrained and transfer learning filter sets.

Table 4.2: Minimum and Maximum singular values of filters in the 3 sets.

Filters	Initialised	Pretrained	Transfer learnt
min	1.49×10^{-5}	2.39×10^{-6}	8.55×10^{-7}
max	0.08273462	1.27163079	1.27102534

We conclude this subsection to remark that CNN decision interpretation can benefit from controlling the range of filters condition numbers and/or singular values over the different channels, which in turn can improve performance stability, improve generalisation and robustness against data perturbation. We shall elaborate on these remarks in the next two chapters.

4.3.3 Topological properties of convolution filters

In the last subsection, we discussed the work of W. Demmel, in [64], who investigated the upper and lower bounds of the probability distribution of condition numbers of random matrices for ill-posed problems and showed that the further away a matrix is from the set of noninvertible matrices, the smaller is its condition number. Accordingly, the spatial distribution of random matrices, in their domains, are indicators of distribution of their condition numbers. These results provide a clear evidence of the viability of our approach to exploit the tools of topological data analysis (TDA) in our investigations of the stability of condition numbers of point clouds of convolution filters.

To visualise the spatial distributions of point clouds of filters (post flattening), we use the two TDA established tools: Mapper and the PD's. We apply these tools on the initialised, pretrained, and transfer learning AlexNet filters. In particular, we focus on the 3 set of filters in the first convolutional layer to check whether there is a link between the performance of filters in terms of entropy or the state of their condition numbers. Readers, however are reminded that the performance of filters in the CNN settings is dependent on other convolution layers steps such as normalisation, activation function, and down-sampling.

4.3.3.1 The Mapper Visualisation

The Mapper algorithm is dependent on selecting the many parameters as described in section 2.2.4. We first flatten each $k \times k$ filter to create a point cloud in $\mathbb{R}^{k\times k}$, then we select different lens projection techniques to visualise the data in lower dimension such as isolation forest and L_2-norm, or principal component analysis (PCA) lenses. Overlapping parameters, covering space, and clustering algorithms are playing an important role for the Mapper layout and connections between the point clusters and neighbouring ones. In figure 4.11, we show a simple case of all three types of filters whereby all parameters are fixed except the lenses. The first lens is isolation forest and L_2-norm (top row), and the second lens is PCA 1 and 2. In both cases, the clustering and the number of connected components for initialised filters are similar. This indicates the state of those filters involving less anomalous property and/or behaviour to be effected by the choice of parameters. For the pretrained and transfer learning filters, both lenses are capturing slightly different information manifested by differences in the connected components, number of nodes, edges and samples.

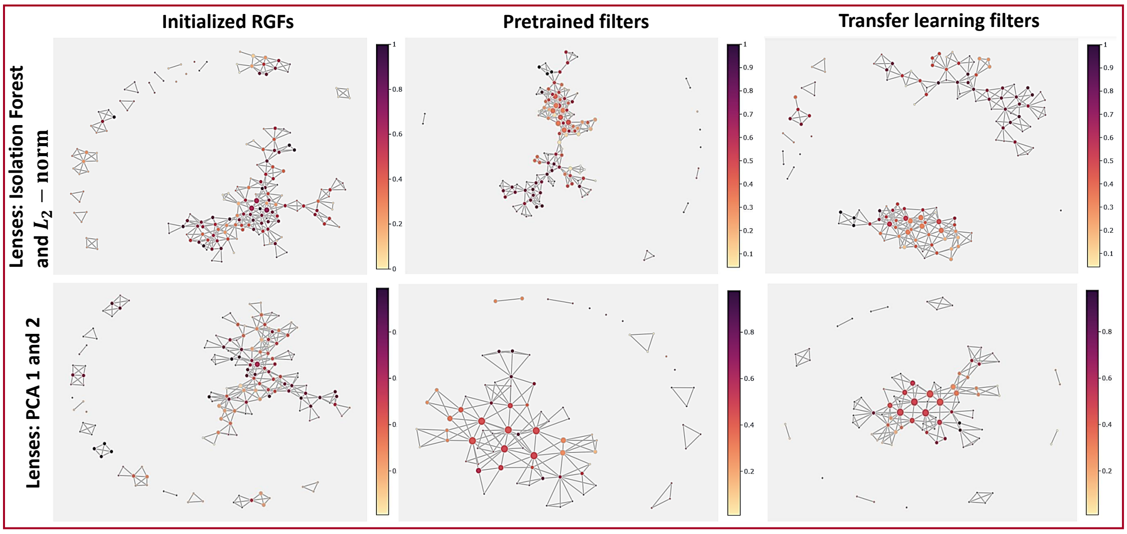

Initialized RGFs
Pretrained filters
Transfer learning filters
Lenses: Isolation Forest and L_2 − norm
Lenses: PCA 1 and 2

Figure 4.11: Visualising AlexNet filters via Mapper algorithm.

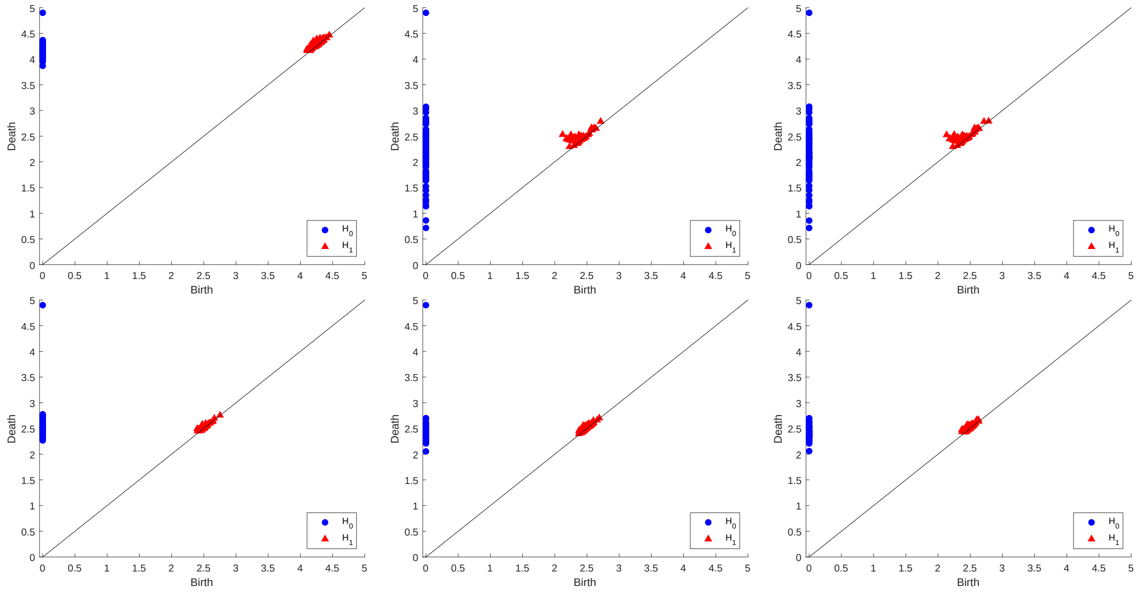
The overall behaviour of all weights are difficult to analyse through Mapper as it requires parameter selection and in some cases a priori knowledge of the data. For example, there are nodes that contain multiple filter point when their L_2-norm are close to each other however the norm of their inverses are significantly far from each other i.e. this type of spatial distribution and condition number may not be linked directly. In fact, repeating the same steps on the inverses of the filters for visualisation require a larger covering space for pretrained and transfer learning filters which indicated the larger distance between the points i.e. considering the norm of the filters and its inverses may provide the condition number states. We will focus on the topological behaviour of filters exclusively in the next chapters.

4.3.3.2 The PD visualisation

The TDA persistent homology tool and its persistence diagrams (PDs) provide a more informative visualisation of the spatial distribution of the various filter sets. Taking into account the discussion on variation of condition numbers of filters in different channels, we can either construct the PDs of the 3 sets by concatenating the flattened filters of the 3 channels and create 96 vectors of 3×(11×11)-dimensional vectors or construct the PD of 288(=3×96) 121-dimensional vectors. In both cases, besides constructing the PDs, in both dimensions 0 and 1, of the filters as well as their inverses.

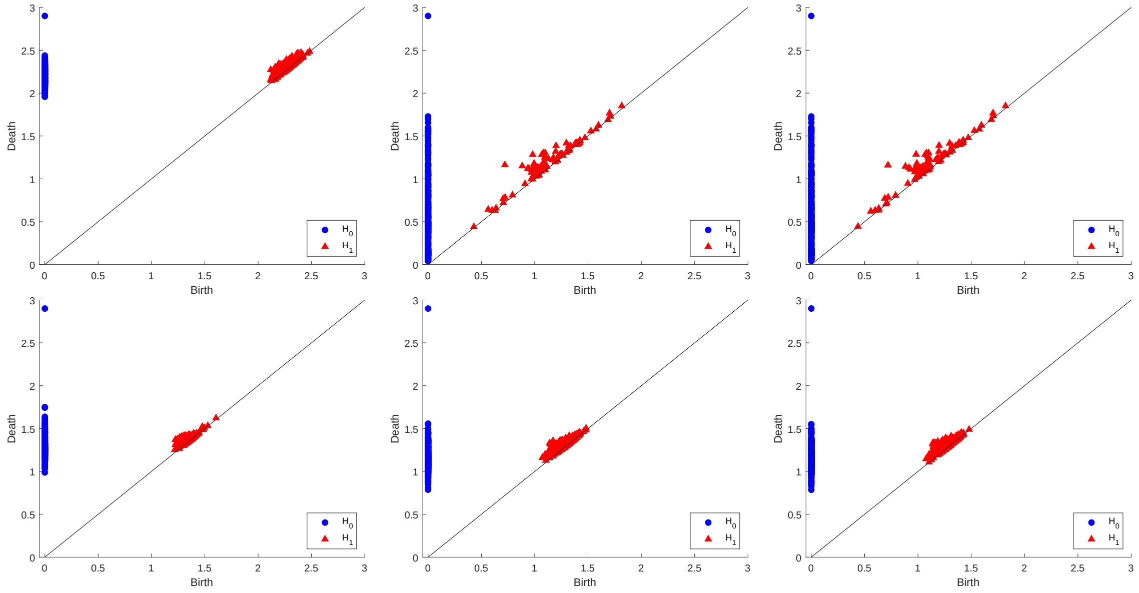
Examining the PDs in Figure 4.12, that are computed from the concatenation of the flattened filters in the 3-channels, reveal that the initialised filters are spread out widely and their connected components start to die, by merging, at a later stage ($death > 3.8$) with only one component that continues to live beyond (4.4). The holes in that set start to be born (at $birth = 4$) when significant numbers of connected components die, and very few holes have long life spans. On the other hand, the PD's of the pretrained and transfer learnt sets have more or less similar patterns but differ from that of the initialised set. These two sets of filters are less spread out than the initialised one and they start to merge much sooner (at around $death = 0.75$), and again only one connected component stay alive. Again, the holes in both these sets start to appear when most connected components merge, and rarely have noticeable life spans. The inverse filters in all the three sets have very few differences in their topological profile. The connected components start to merge much earlier than that of the original filters indicating that the inverse filters are more packed.

Figure 4.13, displays the PDs when each of 3-channel filters are considered singularly. The topological profile of these filter sets, only differ marginally from those in 4.12 as a result of having larger point clouds in lower dimensional domains. This exploration can reveal essential features and connections that might be obscured in higher-dimensional representations.



(a) Initialisation (b) Pretrained (c) Transfer learning

Figure 4.12: PDs of the first layer convolutional filters $96 \times (11 \times 11 \times 3)$ (top) and their inverses (bottom).



(a) Initialisation (b) Pretrained (c) Transfer learning

Figure 4.13: PDs of 1^{st} layer convolutional filters as $288(11 \times 11)$-dim vectors (top) and their inverses (bottom).

4.4 Convolution layer effects on LBP landmarks PH

Feature Learning by CNN models does not only depend on the properties of the convolutional filters. It is also influenced by the training datasets, and the training procedures. Based on the differences of the texture contents in US images compared to those in natural images, it is necessary to determine the effect of convolution layers on the spatial distribution of texture in US images. In this section, we investigate the impact of convolutional layers on the discriminatory power of persistent homology features of feature maps throughout the CL operations on malignant and tumour lesions.

4.4.1 Topological representation of texture landmarks

To this extent, we have seen the impact of the convolution filters on the discriminatory power using the entropy values on the original and convolved images. We extend the in-depth investigation of the CL effects on the spatial distribution of texture features in images and feature maps. In particular, we extract LBP features, as described in Section 3.3.4, from original US images and feature maps after each operation of the convolution layers. After selecting the LBP landmarks, the persistent homology features are computed in dimensions zero and one. The dimension of topological features for each geometric group is:

$$TFV = k \times \tau \times n \tag{4.3}$$

where k is the number of rotations per LBP group, τ is the maximum distance thresholds for computing PH in both dimensions, and n is the number of filters or feature maps at each convolutional layer. Simple classification methods such as K-Nearest Neighbour (KNN) or Support Vector Machine (SVM) can be used to classify the TVF of malignant and benign ultrasound images.

4.4.2 Experimental setting and results

For compatibility with AlexNet architecture, the ultrasound images are resized to 227×227. To prepare the topological feature vector, we only extract the 2-transitions LBP landmarks for k=8 rotations per geometric groups G_i (i=1, …, 7) of landmarks. These choices are based on the distribution of ultrasound LBP descriptors and the statistical analysis shown in Chapter 3. The maximum distance thresholds τ for computing PH, is set to 30 for ultrasound images whereby the number of connected component in dimension zero is one and all holes in dimension one are closed - using "Ripser" software [104] to construct the Rips Simplicial Complexes. This parameter selection is dependent on the image modality, e.g. natural and face images require thresholds beyond 30. The number of n filters or feature maps are n=1, 96, 256, 384, 384, 256 representing the input image and number of filters per layer as the grouped convolutions are concatenated for the evaluation purposes. We feed the TVF to SVM classifier at to evaluate the separation of the spatial distribution from malignant and benign tumours. For the training and testing selection strategy, four single-split schemes with dataset ratios of 30-70%, 50-50%, 60-40%, and 70-30% were compared after repeating each experiment 100 times. Due to the high variation of ultrasound images as explained in the previous chapter, the 70% of training and 30% of testing scheme is chosen.

4.4.2.1 Experimental results

Figure 4.14, below, illustrates the performance of the ULBP landmark groups based PH schemes (in dimensions 0 and 1) on the original ultrasound scan images of three tissue types: liver, bladder, and breast. The classification performance, based on the spatial distribution of each class and type, reveals that the bladder dataset is underperforming, with accuracy for each landmark ranging between 52-60%. In contrast, the breast and liver datasets exhibit significantly higher accuracy, ranging between 76-82%.

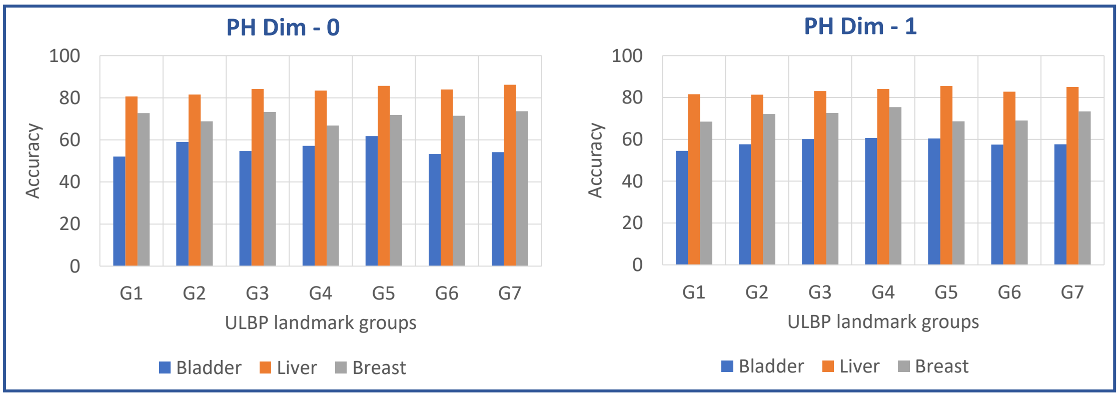


Figure 4.14: ULBP landmark based PH for All US Original images.

The following intriguingly settled patterns can be noted during the first convolution layer: (1) except for ULBP landmark G_4 in dimension 0 and G_3 in dimension 1, each of the first convolution three operations (Convolution, ReLU, and Normalisation) improve accuracy on their predecessors, and (2) except for

ULBP landmark G_1 in both dimensions and G_4 and G_3 in dimension zero and one, the max-pooling operation results in reduced performance. The improved accuracy may be attributed to observations that many filters eliminate landmark sets, especially in benign cases. This reduction in entropy does not necessarily have a negative impact, (see Tables A.1, A.2, and A.3 for detailed results on specificity, sensitivity, and classification accuracy). This raises the possibility of using some convolution filters to improve the performance of PH schemes in conventional machine learning paradigms. Additionally, it provides empirical guidance for CNN architectures in terms of the depth and width of the convolutional layers.

As a case study, we will initially present and discuss the results up to and including the first convolutional layer. Figures 4.15 and 4.16 display the accuracy of original images for comparison with the post-ReLU feature maps of the first convolutional layer. The differences in sensitivity and specificity of both persistent homology dimensions are relatively small at the ReLU layer compared to the original images.

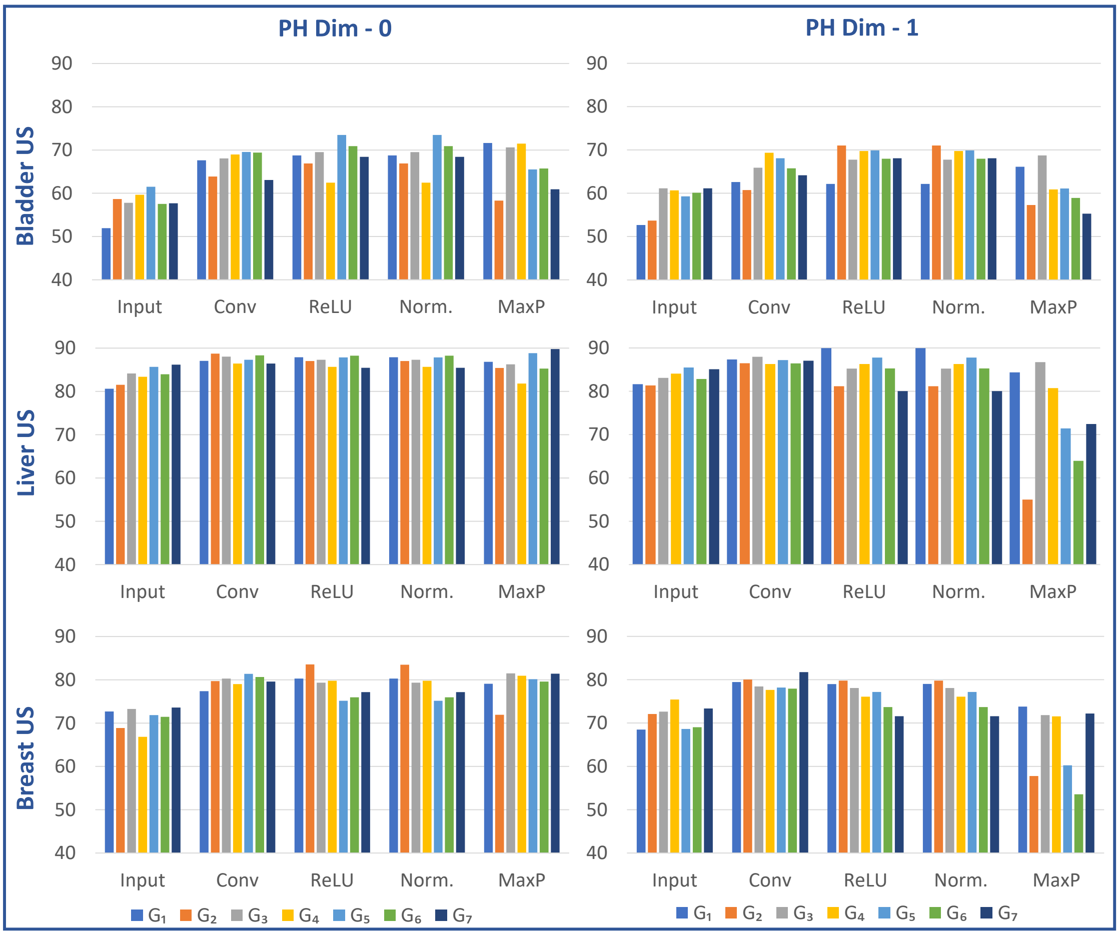


Figure 4.15: Classification accuracy - AlexNet 1^{st} convolutional layer.

Despite the reported drop in entropy in the previous section, the performance post-ReLU for all groups and homology dimensions is higher than the performance of the pre-convolution schemes. Upon close examination of the persistent homology (PH) features, it was observed that PH texture features vanish (resulting in empty landmark sets) for both cases when original images are convolved, bias values are added to feature maps, and the ReLU operation is applied. This improvement is linked to ReLU's impact, introducing sparsity in activation maps by zeroing out negative values and allowing positive values to pass through, thereby promoting the activation of more discriminative features. Despite a simplified data representation, ReLU's sparsity and selective feature extraction contribute to improved task-specific performance. Additionally, ULBP groups and PH vectors become empty for some feature maps produced by certain filters, especially those causing the entropy values to drop to 0 or near 0.

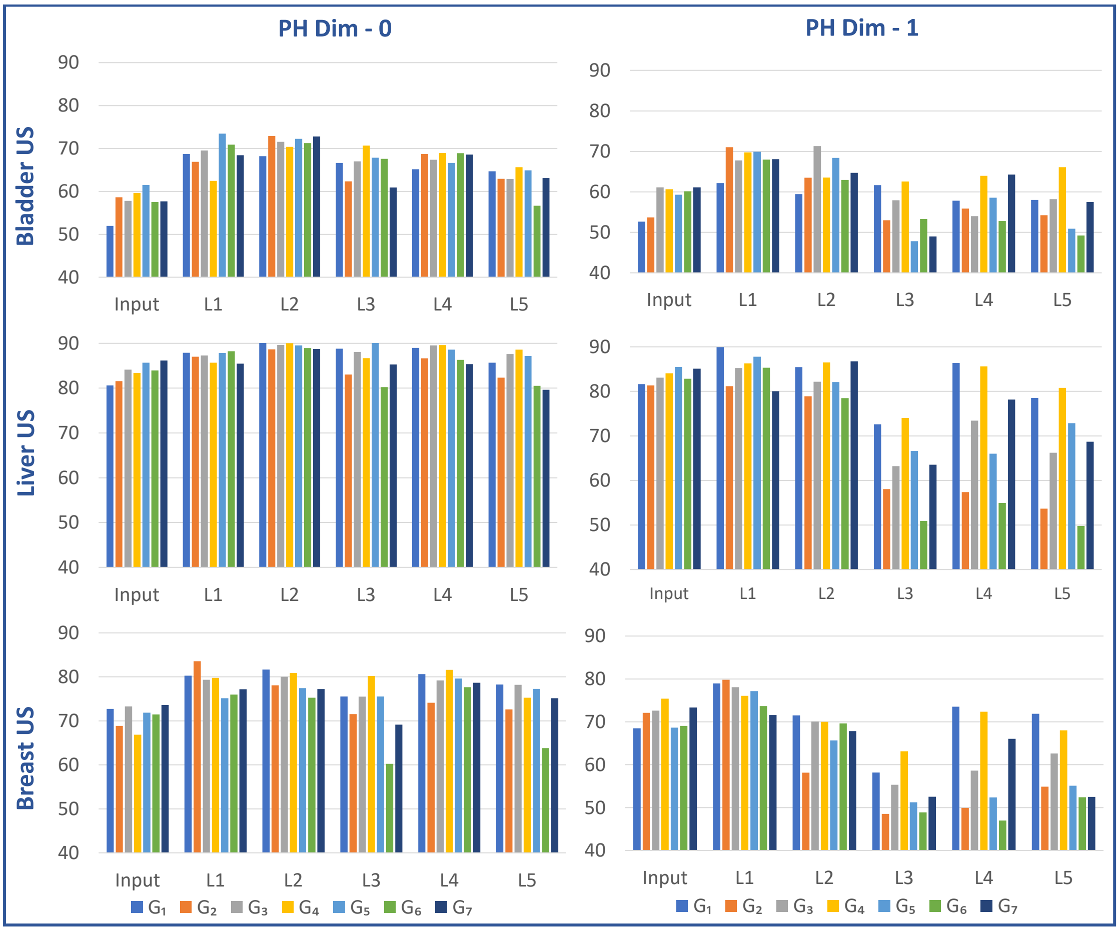


Figure 4.16: Classification accuracy - AlexNet convolutional layers.

4.5 Pruning convolutional filters

Pruning is used in deep neural networks to decrease the network's complexity and size by eliminating unnecessary connections and weights. The purpose of pruning is to identify the connections and weights in a CNN that make the least contribution to the network's accuracy. These connections and weights can be eliminated during training or removed entirely, resulting in a CNN with fewer parameters. This reduction in parameters leads to less memory usage and computational requirements during training and inference. There are several methods for pruning CNNs, including weight pruning, filter pruning, and neuron pruning. Weight pruning involves setting the smallest weights to zero and removing the corresponding connections, while filter pruning involves removing entire filters that have little impact on the network's accuracy. For instance, the authors in [105] proposed pruning filters based on $L1$ and $L2$-norm criteria for more efficient CNNs. The pruning was applied on VGG16 and ResNet-110 models trained CIFAR-10 and the inference costs reduced up to 34% and 38% for each model, respectively. Another example is integrating principal component analysis scheme to prune deep neural networks, [106].

To this extent, our investigations on exploring the impact of convolutional layers on input images by examining the algebraic parameters of initialised, pretrained, and transfer learning convolutional filters have led to proposing a simple filter pruning strategy based on the condition number. We implement pruning filters for a feedforward CL filters in AlexNet based on the condition number and percentage of filters. Pruning filters based on condition number may increase the robustness of the model. Filters with high condition numbers can be pruned at various percentages. In addition, we applied a filter pruning technique based on condition numbers, reducing pretrained filters by 50%. This led to a decrease in computational complexity while maintaining classification accuracy results that are similar to those achieved without pruning the filters (see Figure 4.17).

In order to explore the possibility of reducing the condition number of filters more thoroughly, examining the impact of backpropagation on pruning strategies, including the proposed filter pruning based on conditioning is essential. Furthermore, building upon a pilot study conducted earlier, it is important to investigate the feasibility of using different types of well-conditioned filter initialisation methods with the aim of decreasing redundancy and computational complexity.

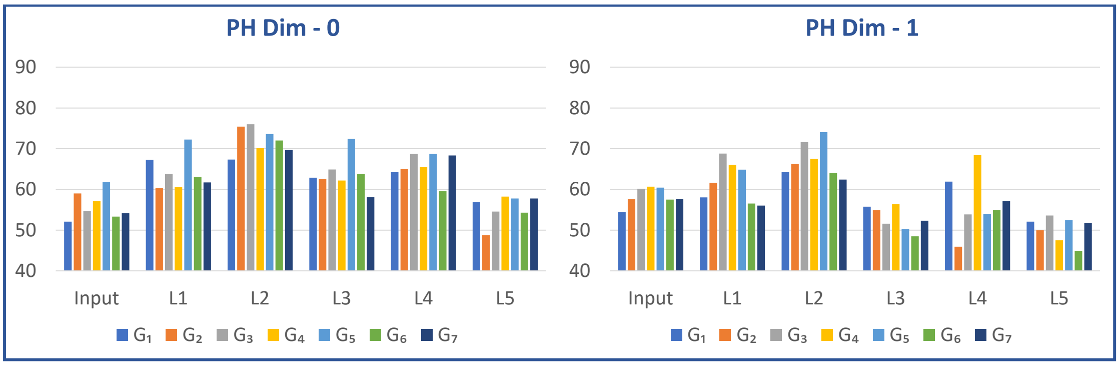


Figure 4.17: Classification accuracy after 50% pruning.

4.6 Summary and conclusion

In this chapter, we have established that utilising pretrained CNN models trained on natural images in transfer learning mode for the analysis of US images indeed results in a lack of robustness against natural perturbations and limited generalisation to unseen data. We conducted various investigations to understand these shortcomings by studying the effects of pretrained convolutional filters on feature maps in terms of conditioning of various filter sets. The aim was to gain a better understanding of the sources of overfitting in deep learning, and quantifying the amount of redundant information in the feature extraction process of AlexNet using the conventional and texture-based entropy values of the convolved images and feature maps. Additionally, we examined the effect of the convolutional layers on the preservation or enhancement of LBP texture features, which may not be evident from simple entropy quantification alone. To further understand the spatial distribution and topological features of feature maps across all convolutional layers, we deployed persistent homology.

Our findings suggest that the incorporation of convolution filters as feature extraction methods can enhance the discriminative capacity of PH based handcrafted features. Additionally, it enables the CNN model to more effectively distinguish between benign and malignant masses. We have gained valuable insights into the performance patterns across different layers, with potential implications for the design of CNN architectures tailored for ultrasound images. Our aim is to train these architectures from scratch, focusing on controlling or regularising the condition numbers of sets (tensors) of filters during training. This will be the focus of work in the next two chapters.

Chapter 5

Towards Slim, Robust and Generalisable CNNs for US Scans

In the last chapter, we established that existing state-of-the-art CNN models, that have been trained with natural images, in transfer learning mode for analysis of US images suffer significantly from lack of robustness against noise-based adversarial attacks, and overfitting effects manifested by the inability to generalise performances to unseen data. These problems are often attributed to the lack of availability of class labelled "good quality" US tumour image datasets that represent an i.i.d random sample of the unknown population. We have also uncovered empirical evidence for additional factors contributing to these limitations by (1) examining the algebraic and topological properties of pretrained and transferred learned point clouds of convolution filters, comparing them with those of initialised filters, and (2) linking these properties to their impact on the texture contents of US images. This chapter is designed to exploit knowledge gained so far in order to explore a viable strategy for overcoming these challenges. We shall demonstrate that controlling the condition numbers of convolution filters provides a suitable strategic framework for designing and testing customised slim CNN models[1], capable of addressing the aforementioned challenges in US image diagnostic tasks while achieving high performance.

Section 5.1, discusses the background for the work in this chapter with a review of existing work on developing customised CNN models trained from scratch on breast ultrasound images. Section 5.2, identified the basic requirements of intended customised models, describe our first attempted simple architectural structures of Slim CNNs, and investigate several approaches to generating convo-

[1] Slim CNN models are characterised by a reduction in parameters, achieved through reducing the number of filters and maintaining a shallow depth in convolutional and fully connected layers.

lution filters with emphasis on distribution of condition numbers. In Section 5.3, we test the performance of simple Slim CNN models on computer vision benchmark datasets such as Digits, MNIST, and CIFAR-10. Moreover, we investigate the instability of the condition numbers during the training in these natural image dataset experiments. In Section 5.4, we introduce an innovative filters weight initialisation to maintain equal condition numbers across the different channels of each layer to be used for refined models of the already suggested simple models. We then evaluate the performances of the simple CNN models, as well as their refined ones, for US tumour image diagnostics together with their robustness against tolerable data perturbations and ability to generalise to unseen data.

5.1 Introduction and background

The idea of designing customised CNN models for image analysis is not a new concept, and recently several such attempts have been made for a variety reasons and purposes. The need for such schemes arise for different reasons, mostly related to non-optimal performance of existing optimal CNN schemes retrained in transfer learning modes. Other reasons, that coincide with our main observation in relation to differences in the nature of images, under investigations, from the types of natural images used to build existing CNN models.

Yi-Cheng Huang et al., [107], designed customised Automated Optical Inspection (AOI) CNN models to identify defects on parts of metal surface that suffer from high reflection level. They justify the need for a customised scheme by the incompatibility of AOI requirements those of CNN algorithms and by the extremely size of residual networks (e.g. versions of ResNet and DarkNet). The improved customised metal surface defects detection scheme uses Grad–CAM to display the feature maps of the last layer for assessing the outcome.

Mobeen-ur-Rehman et al., [108], develop a customised CNN model for the classification of Diabetic Retinopathy Images (DRI) consisting of 2 CLs and 3 FC layers. Traditional Computer Aided Diagnostic schemes rely on detecting and assessing artefact-like feature known as Exudates formed by deposit of lipoprotein near leaking retinal capillaries. Their customised scheme is reported to outperform existing CNN models retrained on DRIs in transfer learning mode.

Osman Özkaraca et al., [109], note the challenge of achieving desired classification of health image data using CNN's in transfer learning modes. This is based on experimental work and review of the literature. That manuscript presents a comparative analysis of a list of several CNN and handcraft-feature classifications schemes trained and tested on MRI brain scans (including public Brats datasets) highlighting performances and pros and cons of each. They developed a customised scheme for classifying meningiomas and gliomas brain by combining multiclass handcrafted features-based schemes with CNN models retrained in transfer learning modes. The improved performance achieved comes at the expense of higher cost of processing time. The proposed scheme is presumed to be suitable for brain tumour as well as other chronic nerve diseases.

None of the above reviewed schemes, implicitly consider robustness or ability to generalise to unseen data which is the core concern of our research besides efficiency. While the DRI customised CNN model is obviously efficient, the Brain MRI combined classification scheme is not.

Robustness is often dealt with through optimisation of network architecture in relation to width[2], depth[3] and weight initialisation, [110]. While ability to generalisation in existing work often rely on data augmentation and/or regularisation methods to control the growth/vanishing of gradient decent during training. Practised approaches for achieving efficient CNN schemes include model compression, and filter dropping/exclusion.

Several recent research works have explored the use of orthogonality conditions on trainable deep learning model weights to improve the stability, robustness, and efficiency of CNNs. These include orthonormal and/or orthogonal weight initialisation techniques, regularisation, convolution, normalisation, and orthogonal DNN [102,111–117]. These studies indirectly support our hypothesis that there is a link between deep learning overfitting and the condition numbers of learnt convolution filters (few of these methods will be discussed further in the next chapter). Moreover, the emerging paradigm in these studies fit into spectral regularisation of neural network weight matrices. However, instability of weight matrices' condition numbers, that we established in the last chapter in terms of instability of filters condition numbers during training, are not discussed explicitly in these studies.

To overcome the vanishing gradient and/or overfitting issues, several techniques have been proposed, such as residual connections [118], batch normalisation [119], and skip connections [120]. These techniques can facilitate the flow of information across different layers and reduce the impact of the vanishing gradient problem, allowing deeper networks to be trained effectively through recovering texture features and/or reducing the effect of the data on adjusting the model

[2] Depth refers to the number of layers in the network.

[3] Width refers to the number of neurons in each layer.

parameters. Authors in [31] explored the deep neural networks with/without skip connections and revealed that without skip connections, DNNs encounter singularity issues as depth increases, causing hidden representations to lose information and making optimization challenging. In contrast, DNNs with skip connections avoid singularity issues as depth increases, maintaining complete information and resulting in better optimisation and generalisation.

A recent emerging customised CNN models developing strategy advocate the use of network architecture optimisation techniques such as neural architecture search (NAS) [121] to automatically discover efficient and robust architectures that balance depth and width, as well as other design criteria. These techniques can help to reduce time and cost of designing desired customised CNNs with improved performance. The Efficient Neural Architecture Search (ENAS) have been used, in [45,122], to automatically design CNN architecture specifically for Ten-D breast cancer classification from ultrasound images as part of Ten-D research project. Their initially generated CNN schemes (ENAS7 and ENAS17) outperformed the manually designed CNN architectures for US breast cancer classification, but do not generalise well to unseen datasets. They further investigated several approaches to improve the generalisation rate of these ENAS-based models including reduced model complexity, different data augmentation, and unbalanced dataset training. For more details, we refer the reader to [123].

5.2 Design requirements of customised CNN architectures

In this chapter, we investigate and explore the core ingredients towards constructing efficient customised convolutional neural networks, to be trained from scratch on ultrasound scans, that are robust against adversarial noise and able to generalise to unseen data. Guided by the intensive research in medical imaging tasks as well as our established results in Chapters 3 and 4, we focus on the main building blocks that are required to design a customised CNN model.

5.2.1 Characteristics of Slim CNN architectures

The depth and width of any CNN models form an efficiency characterising factor of their architectures and play a major role in achieving model performance, robustness, and ability to generalisation. Deeper neural networks are perceived to have more capacity to learn complex features and patterns in data. However, such architectures can also be more susceptible to overfitting, particularly in cases where the available dataset is small. Appropriate selection of the number of convolutional and fully connected layers in CNNs (i.e. their depth) should be based on the requirement of the task, characteristics of the image modality in terms of the discriminating features, required computational efficiency, the number of image classes, and the contribution to learning. Some of these factors also influence the decision width that represent the number of channels in the convolution layers as well as the number of neurons of FCL.

Our task relate to a binary classification (i.e. 2 classes: Benign and Malignant) of US tumour scan images, but we would not purposefully exclude their use for non-binary classification of other image modalities. Indeed, our design will be refined in light of their performances on benchmark datasets of natural images. Our strategy for deciding these architectural parameters for CNN models customised US tumour images will be guided by the knowledge, gained in the previous chapters. First of all, in Chapter 3, we established that texture content and spatial distribution of texture landmarks distinguish US images from natural images. In Chapter 4, Section 4.4, we also demonstrated that:

> *The spatial distribution of LBP texture landmarks via the PH-based classification scheme is improved markedly for all types of tissue through training in the first two layers of the pretrained CNN models compared to the original images, and then deteriorate subsequently indicating that texture feature pattern learning either stops or become very marginal.*

Therefore our strategy will be based on reduced depth CNN architectures. Furthermore, in Section 4.5, we have shown that:

> *Filters dropping by 50% based on their desirable condition numbers, maintains the discriminating power of PH texture landmark-based classification through all convolution layers.*

The above two displayed results were established in relation to the content of US tumour scan images being dominated by texture features. Accordingly:
First principal strategic design: Customised slim CNN for US images should have shallow depth (i.e. at most 3 layers) as well as narrow width (i.e. small number of filters per layer).

5.2.2 Filter initialisation requirements

Filters initialisation is a crucial step in the training of convolutional neural networks. There are several factors influencing our strategy of filters initialisation for our customised CNN architecture. These include: (1) determining matrix-related properties of the selected, (2) number of selected filters per convolution layers, and (3) what restrictions, if any, we should impose on the differences between filters of the different channels? All these choices have an impact on the features learning stage during training. This raises another issue on stability of the chosen matrix-related property during training.

The commonly popular approach to filter initialisation of CNN models use Random Gaussian Filters (RGFs) but the only property considered related to the standard deviation of the Gaussian function used. While effective and simple, this approach presents challenges for US images, in terms of their action on image texture landmarks being an important class discriminating image features the statistics of which are distinguishes US images from natural ones. RGFs are smoothing filters and images convolved by RGFs are sensitive to small/imperceptible textural changes (e.g. due to addition of noise) with sensitivity level depending on filter condition numbers. The condition number of a matrix measures the sensitivity of their action as linear transformation to changes in input domain. High condition numbers lead to ill-conditioned convolution that can cause numerical instabilities like vanishing or exploding gradients during training.

In fact, we already illustrated that convolution by ill-conditioned filters are highly likely to increase image textural artefacts which in turn contributed to lack of robustness against noise when the various pretrained CNN were used for retraining US datasets in transfer learning mode. The fact that most pretrained CNN filters, including those obtained through transfer learning, are highly ill-conditioned. When convolved with visually similar ultrasound images recorded in different clinical settings, this ill-conditioning likely increases the distances between them, thereby contributing to challenges in generalising to unseen data. In relation to issue (1), above, our initialisation strategy is to focus on filters condition numbers:

Only use reasonably well-conditioned Gaussian kernel.

In considering this strategy, we need to consider the task of generating for each convolution layers sufficient number of RGF's with the desired range of condition number to be used for specified number of channels. We shall first consider our strategy regarding the appropriate number of filters. In section 4.5, we have demonstrated that dropping the top 50% of ill-conditioned pretrained CNN filters had little or no adverse impact on the texture landmark-based PH scheme of US image diagnosis. In relation to issue (2), above, our initialisation strategy is to focus on filters condition numbers:

Use reasonably small number of RGFs per convolution layer.

Taking the above discussion into account, we stipulate that:
Second principal strategic design: Customised slim CNN for US images, should involve relatively small number of reasonably well-conditioned filters per convolution layer.

5.2.3 Simple customised CNN models

We shall now, propose first two simple Slim Customised CNN model architectures to be trained and tested primarily from scratch on US tumour image datasets.

1. ***Model-A***: consists of one convolutional layer (64 filters with dimensions $5 \times 5 \times 1$), ReLU/Tanh, maxpooling, one fully connected layer with 10 neurons, softmax, and classification layer.

2. ***Model-B***: consists of two convolutional layers (32 and 64 filters with dimensions $5 \times 5 \times 3$ and $5 \times 5 \times 32$), ReLU/Tanh, maxpooling, two fully connected layer with 64 and 10 or 2 neurons depending on the class size, softmax, and classification layer.

Model-A is specifically designed for binary classifications while the Model-B can also be used for multi-class purposes. However, we shall test viability of using both models for multi-class analysis of different non-US natural/synthetic datasets of images.

5.2.4 Initialisation of filter point clouds and their properties

In this section, we shall discuss the process of selecting the required number of filters in accordance with the specified architectures of the above two models. RGF matrices can have a wide range of condition numbers, but to select any number N of matrices that have a specific range of condition numbers, one needs to generate much greater than N filters and discard the surplus. In this work, we experimented with several randomly selected convolution filter sets of reasonably well-conditioned RGF matrices without imposing other criteria such as orthogonality.

In relation to the standard deviations of the associated Gaussian distribution, we conducted pilot experiments with three well-known types of weight initialisation techniques that control the standard deviation of the random Gaussian filters per layers.These weight initialisation known as Narrow Normal (NN) [29], Glorot/Xavier [68], and He [69]. We trained Model-A customised CNN scheme on benchmark datasets (DIGITS, MNIST and CIFAR-10, see descriptions later in Section 5.3) with rectified linear unit (ReLU) and hyperbolic tangent (Tanh) activation functions using randomly selected 96 (=32×3) weight initialisation, regardless of condition numbers, for a duration of 20 epochs. We observed similar pattern of behaviour, for the 3 datasets, in terms of the initial condition number and stability over the training. Irrespective of deployed weight initialisation and activation function, both well-conditioned and ill-conditioned initial filters may become unstable, but the fluctuation of well-conditioned filters are more likely to end up with acceptable condition number. In Figure 5.1, we present the observed condition numbers (in logarithmic scale) over the 20 epochs for two typical filters per initialisation scheme, one being unstable with a medium to high condition number and the other being stable with a low condition number when we trained Model-A on the DIGITS dataset. Our observations indicate that the effect of training the Digits dataset on the stability of filters is similar across the three filter models. Therefore, in the rest of the thesis we use the NN initialisation scheme for training customised CNNs.

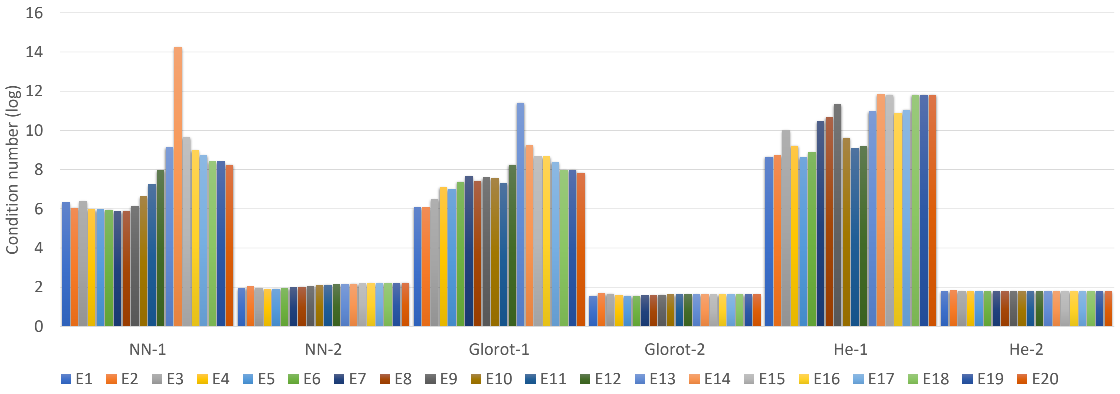


Figure 5.1: Condition number of stable and unstable filters over 20 epochs (E).

The following is a list of few sets of NN initialised convolution filters, used for training our proposed customised CNN models from scratch:

- $\mathcal{W}_D$: Default generate the exact number of filters regardless of conditioning.
- $\mathcal{W}_1$: select the exact number of filters per convolutional layer based on the

lowest condition number out of randomly generated 10^3 filters.

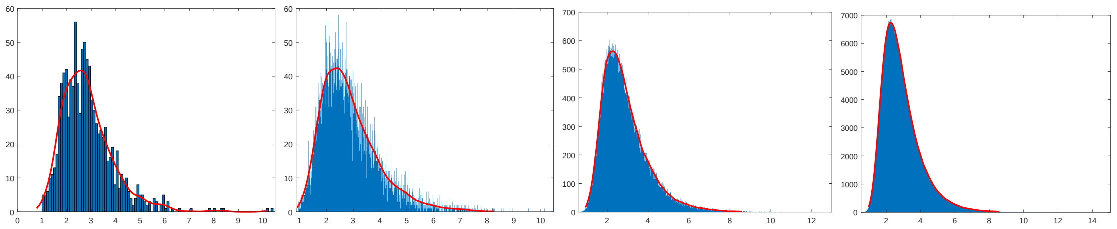
- $\mathcal{W}_2$: select the exact number of filters per convolutional layer based on the lowest condition number out of randomly generated one 10^6 filters.

- $\mathcal{W}_3$: selects twice the required number of well-conditioned filters out of 10^6, trains the model once, drop half of the filters that are less stable, and re-train the CNN model from scratch again with the retained filters. with For model-B, selected most stable 32 out of 64 filters for 1CL, and 64 out of 128 for 2CL. With model-A we selected the most stable 64 out of 128 filters.

The selected $n \times n$ filters in these sets differ in the range of condition numbers, and are expected to differ in their performance and behaviour during training and testing the proposed customised CNN models. In CNN setting, initialised filters can be treated as an $n \times n \times k$ tensor, where k is the number of channel-wise depth whose condition number is set to be the average condition number of the k $n \times n$ filters.

The probability of lower/upper bounding condition number of an identical and independent distribution (iid) $n \times n$ random matrices is well investigated in the literature (see [60, 64, 124, 125] for further detail). Understanding the lower and upper condition number probability bounds of the chosen criteria plays an important role in the way they affect the training image datasets, but the risk of instabilities of filters condition numbers during training cannot be determined a priori and maybe dependent on the training image datasets and the training procedure.

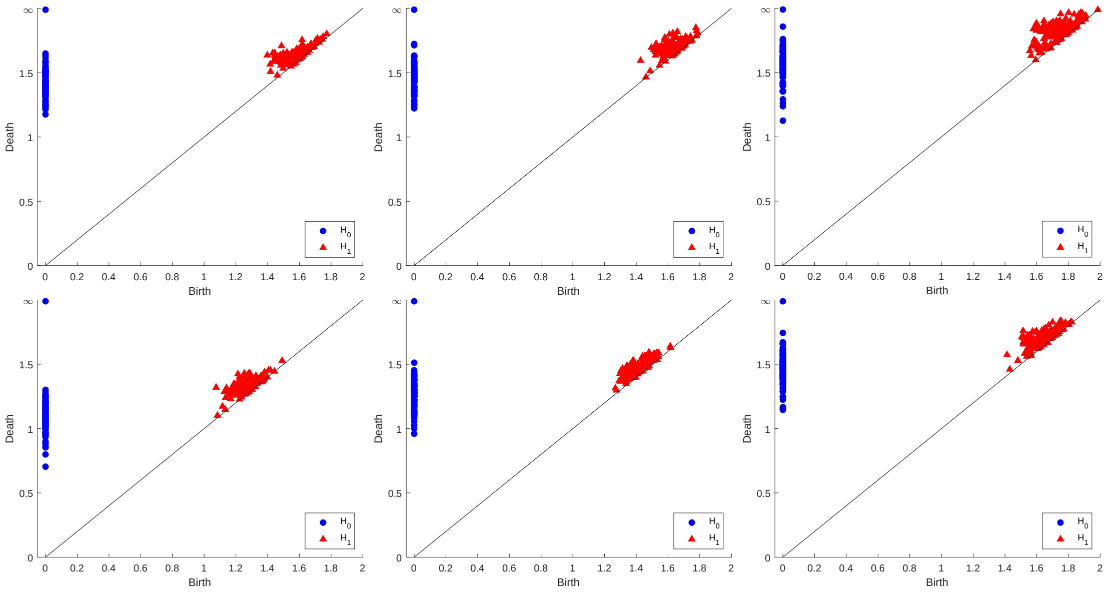
Figure 5.2, below, illustrates the distribution of N number of 5×5 random Gaussian matrices. Probability distributions of condition numbers for 3×3 filters are similar, see Figure A.4.



(a) $N = 10^3$ (b) N=10^4 (c) $N = 10^5$ (d) $N = 10^6$

Figure 5.2: Distribution of condition number (log) - N number of 5×5 RGFs.

In light of the observation in Figure 4.10 on the topological property of filters point clouds based on initialised, pretrained, and transfer learning filters, we close this section by computing the PD of point clouds of 5×5 filters in the sets $\mathcal{W}_D$, $\mathcal{W}_1$, and $\mathcal{W}_2$. Figure 5.3 summaries the PD representations of the spatial distribution of these point clouds and their inverses whereby the condition numbers are counted as the average over the corresponding layer channels (see Figures A.5 and A.6 for various 3×3 and 5×5 filter set conditioning). The PDs show clear differences between the persistence of their topological profiles. Not only the numbers of connected components and holes differ from one to another, but they differ in the range of their persistence life spans in relation to their birth times. Initially, members of the three point clouds join their neighbours forming local clusters that only start merging, to form bigger disjoint clusters after the proximity of threshold reaches around 1.175 for $\mathcal{W}_D$ and $\mathcal{W}_1$ (but slightly earlier around 1.125 for $\mathcal{W}_2$). The connected components continue to merge rapidly but become a single one just after threshold reaching 1.65 for $\mathcal{W}_D$ when $\mathcal{W}_1$ still have 2 component that only merge at 1.575 which is the time when $\mathcal{W}_2$ still have 2 component that merge at around 1.85. The holes start to appear for $\mathcal{W}_D$ at 1.4 marginally sooner that $\mathcal{W}_1$ but $\mathcal{W}_2$ larger number of holes do not emerge until 1.5. Moreover, the number of longer persisting holes increases in the order $\mathcal{W}_D$, $\mathcal{W}_1$, and $\mathcal{W}_2$. For $\mathcal{W}_D$ and $\mathcal{W}_1$, all holes become extinct around 1.8 when some holes in $\mathcal{W}_2$ continue to be generated for a while until 2. Reasonably similar comments can be made about patterns of differences between the topological profiles of the point clouds of inverses.



(a) $\mathcal{W}_D$ (b) $\mathcal{W}_1$ (c) $\mathcal{W}_2$

Figure 5.3: PD of point clouds of 5×5 filters (top) and their inverses (bottom).

5.3 Performance testing for natural image datasets

Given the knowledge we gained from previous chapters about the nature of ultrasound datasets, we test our hypothesis on well-known computer vision benchmark datasets as well as ultrasound datasets. In this section, we present the empirical investigations to test various filter initialisation approaches with customised CNN architectures trained on benchmark datasets Digits, MNIST, and CIFAR-10, but the same experimental work on the Ten-D dataset of US breast images will be given later in Section 5.4. These benchmark datasets differ among themselves, and from US image, in terms of their texture content. The performance of state-of-the-art CNN models are widely investigated. Moreover, Unlike these datasets, variation of lesion sizes and/or potential biases in the sampled US images may have an adverse impact overall model performance during training. These are the main motivations for including these experiments in a thesis dedicated to the analysis of US images. Besides testing the performance of the proposed Model-B on the training datasets, we shall also test their sensitivity to small perturbations of the input images.

5.3.1 The experimental benchmark datasets

All these benchmark datasets consist of 10 labelled classes, as shown in Figure 5.4, and their properties are described below:

- **Digits** [126]: consists of 10000 synthetic grayscale images of handwritten digits from 0-9 of size 28×28 pixels.
- **MNIST** [127]: consists of 70000 grayscale images of handwritten digits from 0 to 9 of size 28×28 pixels.
- **CIFAR-10** [128]: consists of 60000 colour images of natural images of size 32×32 pixels. The 10-classes are airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

Images in both Digits and MNIST datasets are grayscale and in such cases, the first convolution layer of the customised CNN models consists of single channel. Therefore, we shall only train and test Model-A for Digits and MNIST datasets. There obvious differences in the content and sources of the images in CIFAR-10 dataset that are all RGB images. Hence, it is highly unlikely that a single convolution layer is capable to do performing reasonably well. Accordingly, Model-B customised CNN will be used in this case the first layer of which should have 3 channels.

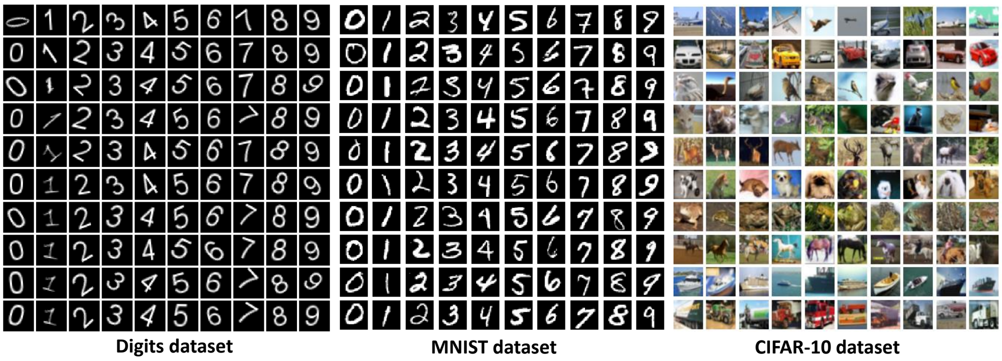
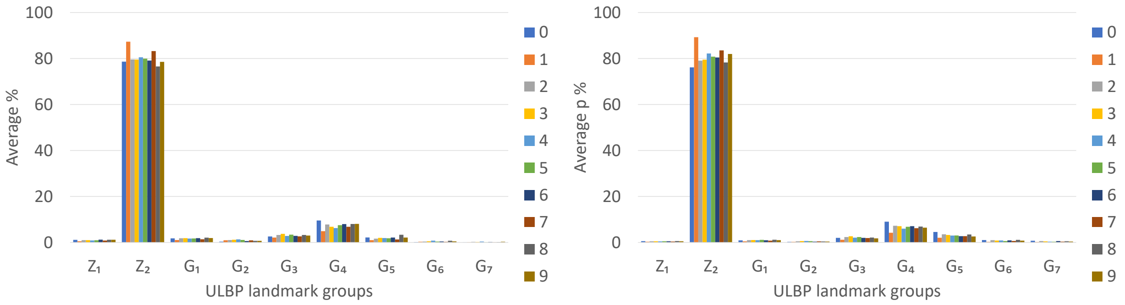


Figure 5.4: Digits, MNIST and CIFAR-10 benchmark dataset samples.

5.3.2 LBP descriptor statistics

We compute the statistical analysis on all images contained within three benchmark experimental datasets. The aim is to determine the proportions of the various LBP landmark groups present in each dataset similar to the statistical analysis of Ten-D ultrasound datasets[4]. The results obtained from this analysis provide valuable insights into the spatial distribution of geometric texture features within each of the benchmark datasets. Here, we shall only consider the statistics of ULBP landmarks, because the other landmarks are very rare in these natural image datasets. The percentage of the different ULBP landmark groups are clearly depicted in Figure 5.5, below, for the MNIST and DIGITS datasets. It shows that for all classes the dominant landmark set is the 0-transition group, Z_2 with approximately 80% of pixels in these datasets have the LBP code of 11111111. The remaining bins are distributed amongst other landmark groups. Furthermore, our analysis reveals that the distributions of LBP groups in digits and MNIST datasets exhibit a comparable pattern, with minor variations in percentage on average.



(a) Digits dataset

(b) MNIST dataset

Figure 5.5: ULBP descriptor statistics of 0,1,. . . ,9 digits.

[4] See Section 3.3.2 for further detail on the ultrasound LBP Statistics.

Figure 5.6, below, presents the statistical distributions of LBP landmarks in images of the CIFAR-10 dataset. These results demonstrate completely different patterns from the patterns observed in the Digits and MNIST datasets which reflects the difference in the nature of the image modalities. Specifically, the dominant groups in CIFAR-10 are G_4, G_5, and G_3, and the differences in the distributions between all classes are quite distinct for these landmarks. The discrepancies in the ULBP local texture feature distributions between CIFAR-10 and the two other similar datasets offer an exciting opportunity to explore the effects of CNN convolution layers on PH analysis post-convolution when compared to the original images. However, such an investigation falls outside the scope of this chapter's focus study as the LBP statistics of post-convolution feature maps are dependent on the filter stride, size, and activation function.

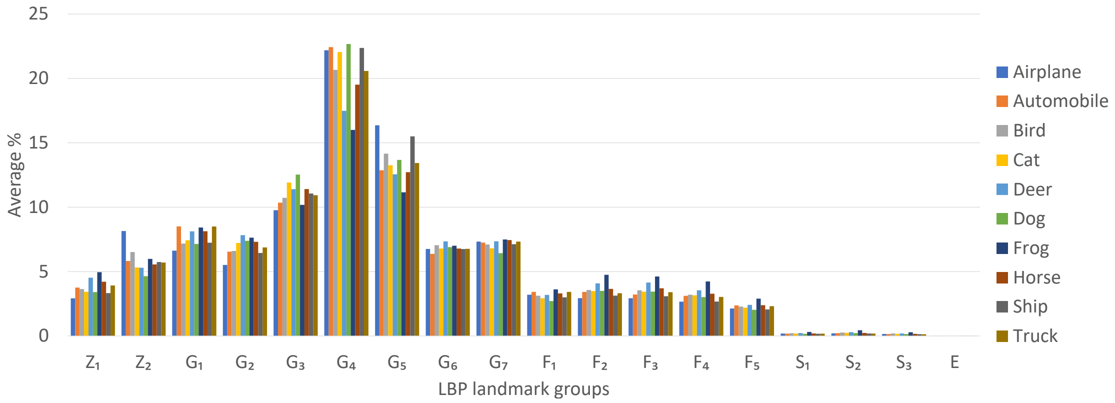


Figure 5.6: ULBP descriptor statistics of CIFAR-10 dataset.

It is worth noting that the distributions of the LBP landmarks in CIFAR-10 images may appear more similar to those in US images (refer to Section 3.3.2) when compared to Digits and MNIST. However, significant differences still exist. The percentage difference between dominant and non-dominant LBP landmark groups in US images is higher compared to natural images and lower compared to the digits datasets.

A key advantage of this type of analysis lies in the valuable textural insights it provides into datasets intended for CNN training from scratch or in transfer learning mode. For example, if a CNN model is trained with CIFAR-10, it can be re-trained with Digits or MNIST datasets and/or used as a feature extractor. However, the reverse process may not be as successful as the former due to the high variation in the texture. In light of this fact, [89] sheds light on the relationship between the topological structure of networks' ability to generalise across two datasets, handwritten digits (MNIST) and street view house numbers

(SVHN) [129], which feature distinct image classes from 0 to 9. In particular, the authors confirm the hypothesis that a network trained on the more diverse SVHN dataset exhibits better generalisation ability when evaluated on MNIST than vice versa, emphasising the relevance of dataset diversity for generalisation. We have observed that the distribution of LBP landmark of SVHN dataset is nearer to that of CIFAR-10, and it is much richer in texture than the MNIST dataset, see Figure A.2. In general, the statistical analysis presents an insight into the relationship between dataset diversity and generalisation performance, which have significant implications for the design and optimisation of neural networks for real-world applications. For example, before deploying a CNN model trained on a particular set of tumour images obtained from one or more hospitals, it may be helpful to conduct an analysis of LBP distributions in the new hospital's images to ensure that the model will perform well in that setting.

5.3.3 Training and testing the customised CNN models

We used the specified training benchmark datasets for training and validating the CNN models then testing them on the specified testing sets. Typically, k-fold cross-validation[5] technique is adopted to identify the best performed model. Digits and MNIST datasets are trained with Model-A, where the classification accuracy of validation and testing is around (97-99.5)%. Whereas CIFAR-10 and Ten-D datasets was trained with Model-B, achieving classification accuracy of validation and testing is around (65-67)%. Table 5.1 displays the full classification accuracy results, at the training & validation stage as well as when tested with an unseen subset of the datasets, for the Digits, MNIST, and CIFAR-10 datasets.

The results show that the test and validation accuracy are very similar reflecting the fact that the datasets are very large and the training as well as the testing sets are good random samples of their respective population. The results also demonstrate that datasets of images whose textural and structural contents are of limited variation can be classified almost optimally by a very slim customised CNN model with a single convolution layer and any relatively small set of convolutions filters, like our Model-A. However, datasets of images that vary widely in their textural and structural contents, require deeper CNN architecture and careful selection of convolution filters.

[5] k-fold cross-validation: is a statistical technique used to evaluate the performance of a machine learning model. The data is split into k subsets, and the model is trained on $k-1$ subsets and tested on the remaining subset. This process is repeated k times, and the results are averaged to estimate the model's performance.

Table 5.1: Classification accuracy of various customised CNN models.

Dataset	Digits		MNIST		CIFAR-10	
Weights	Val.	Test	Val.	Test	Val.	Test
$\mathcal{W}_D$	98.2	97.75	97.1	97.45	66.03	65.98
$\mathcal{W}_1$	99.33	98.85	97.2	97.46	65.82	66.82
$\mathcal{W}_2$	98.53	98.1	97.47	97.52	65.11	65.46
$\mathcal{W}_3$	97.73	97.95	97.38	97.44	66.11	66.03

While we have established all these schemes perform well with the benchmark dataset, our objectives in introducing our customised CNN schemes place more emphasis on robustness against various levels of natural perturbations similar to the image perturbations as well as the ability to generalise to unseen data. But next, we shall only investigate the robustness properties of the schemes for the benchmark datasets.

5.3.4 Robustness testing against random perturbation

Robustness is a critical aspect of image classification using convolutional neural networks. The performance of CNN models can be adversely affected by variations in the input data, leading to incorrect predictions and model failure. Therefore, evaluating the robustness of CNN models to variations in input data is crucial for their successful deployment, and this is nowhere more critical than in the analysis of medical images. Testing the performance of a CNN model on input data with natural perturbations is an effective way to evaluate its robustness, but appropriate evaluation metrics such as robustness score or adversarial accuracy should be used. The choice of natural perturbations for evaluation depends on the problem and the specific forms of perturbations likely to occur in real-world scenarios. In this section, we show the robustness of each of the customised CNN models in a simple adversarial setting, whereby we repeat the classification experiments after degrading the test images in the various datasets. We conducted this with different simple degradation procedures, but all have similar effects and therefore we only present the results for Gaussian noise degradation by adding Gaussian noise with 0 mean and variance in the range 10^{-4} to 0.5. We present the results for each dataset separately.

5.3.4.1 Digits dataset

We evaluated the robustness of Model-A to noise by introducing random 0-mean Gaussian noise with variances ranging from 10^{-4} to 0.1 on 1000 test images from the Digits dataset. Figure 5.7 presents the accuracy rates for 11 different noise levels. Intriguingly, we observed a slight improvement in testing accuracy upon adding low levels of noise (0.0001 and 0.001) to the input images. The most significant differences in testing accuracy between $\mathcal{W}_D$ and $\mathcal{W}_3$ were observed at variances of 0.01 and 0.03, with $\mathcal{W}_D$ weight initialisation dropped from 84% to 42.3% while $\mathcal{W}_3$ weight initialisation dropped from 89% to 70.4%, resulting in a difference of approximately 28% between $\mathcal{W}_D$ and $\mathcal{W}_3$ at $variance = 0.03$. In summary, our findings indicate that using $\mathcal{W}_3$ filters can enhance the noise tolerance of Model-A compared to other modes.

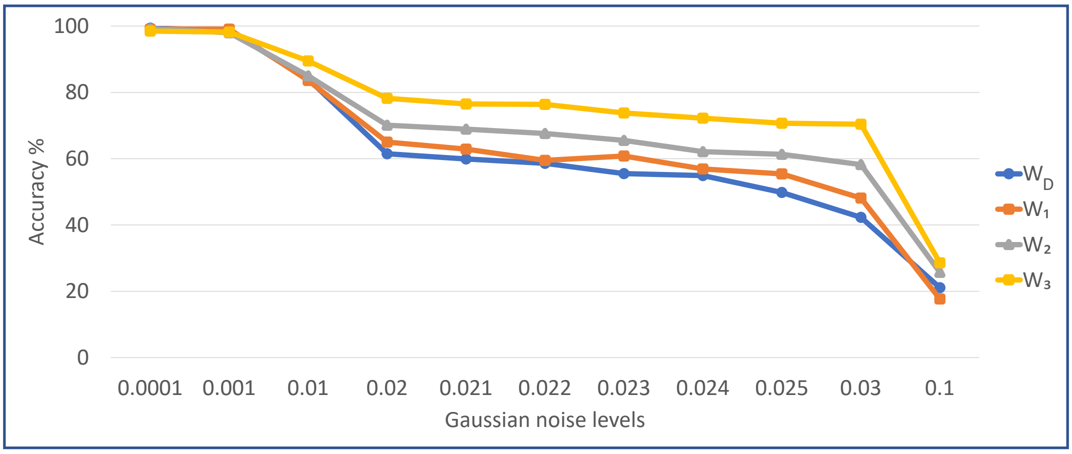


Figure 5.7: Robustness test of classification accuracy performance against natural perturbation on the Digits dataset.

5.3.4.2 MNIST dataset

To assess the effectiveness of Model-A when dealing with noisy data, we subjected one thousand test images from the MNIST dataset to random 0-mean Gaussian noise with varying degrees of variance (ranging from 10^{-4} to 0.5). The results of our experiments, presented in Figure 5.8, below, indicate that the accuracy of the model remained relatively stable until the noise level increased beyond 0.02. However, as the variance increased noise level to 0.1 and beyond, significant differences emerge in performance between the $\mathcal{W}_D$ and $\mathcal{W}_3$ sets. More strikingly, when the variance increased from 0.1 to 0.2, the accuracy of the $\mathcal{W}_D$ filters decreased to just over 20%, while the $\mathcal{W}_3$ scheme only experienced a decrease of less than 13%.

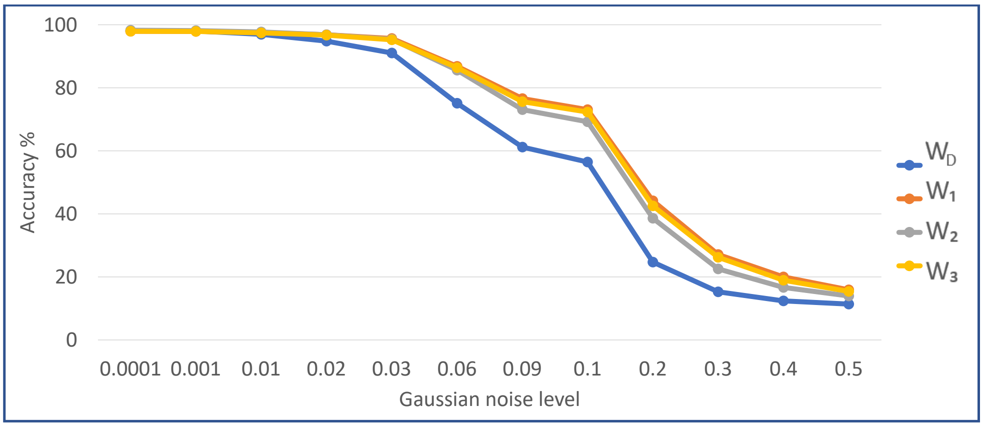


Figure 5.8: Robustness test of classification accuracy performance against natural perturbation on the MNIST dataset.

5.3.4.3 CIFAR-10 dataset

For this rather more complex dataset, we evaluated the performance of Model-B trained with the different initialisation sets post addition of noise. We subjected one thousand test images from the CIFAR-10 dataset to random 0 mean Gaussian noise with varying variances (ranging from 10^{-4} to 0.1). The results presented in Figure 5.9, below, again indicate that the accuracy of the model remained relatively stable when the noise level was low. However, unlike the results observed for other datasets, we found that $\mathcal{W}_1$ filters demonstrated better robustness to noise than $\mathcal{W}_3$ filters. Upon closer examination of the filters in $\mathcal{W}_3$, we discovered the presence of several stable filters that had high condition numbers. This seems to be the most plausible explanation for the observed discrepancy.

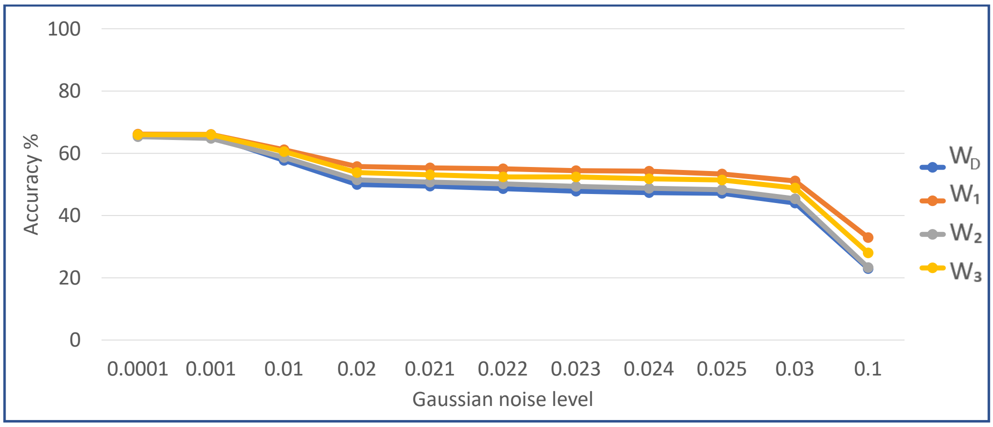


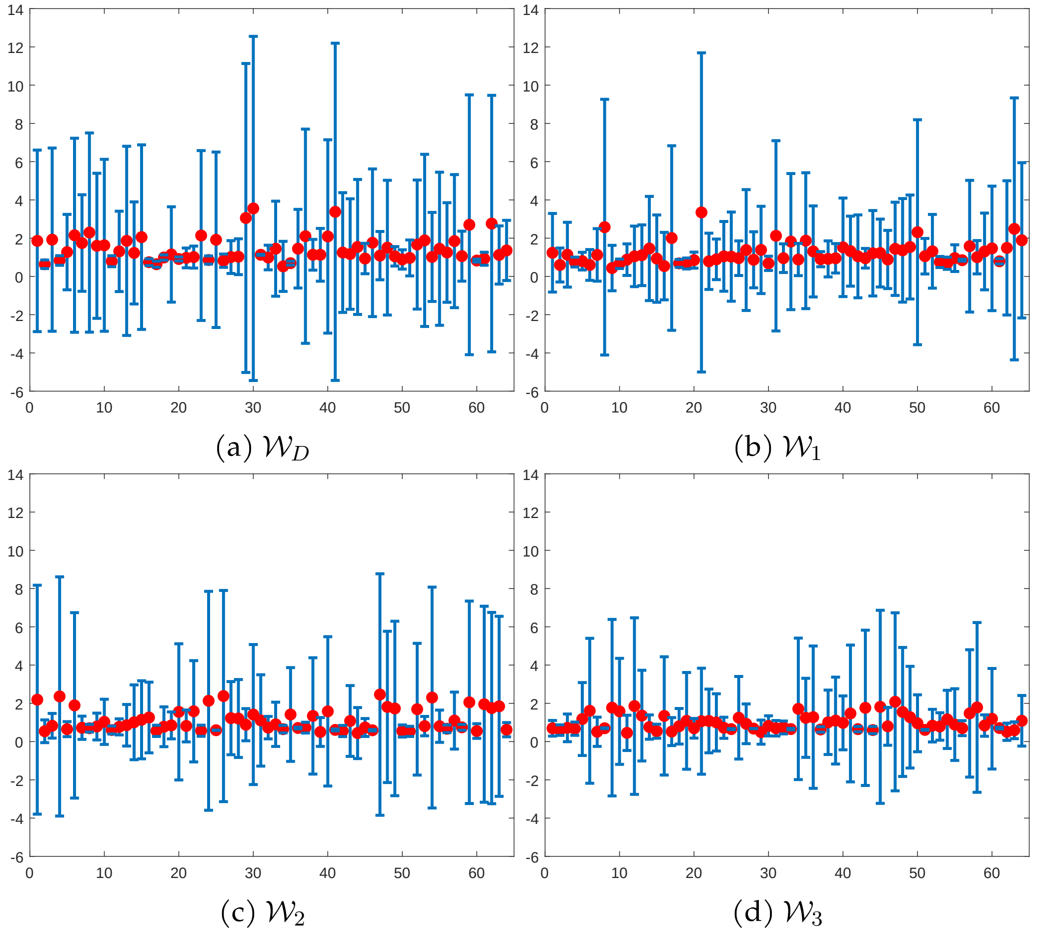
Figure 5.9: Robustness test of classification accuracy performance against natural perturbation on the CIFAR-10 dataset.

The results of these experiments provide strong evidence that well-conditioned filters are less susceptible to small perturbations than other filter modes. While customised CNN models with $\mathcal{W}_D$ exhibit high performance and accuracy during training, they experience a significant drop in performance when subjected to testing or even minor input changes.

5.3.5 Training impact on stability of filters condition numbers

Although, it seems that various weight initialisation techniques lead to similar classification accuracy results, but hidden properties such as robustness to small data input perturbation and the variability of filters condition number and persistent homology could provide a better insight of these CNN models. The purpose of conducting the above sets of experiments using the non-Ultrasound benchmark image datasets was not solely on achieving the highest classification accuracy. In Chapter 4, we argued that the instabilities of the already highly ill-conditioned pretrained convolution filters is among the most likely contributors to the lack of robustness and inability of generalisation of the several state-of-the-art CNN models. Hence, it was also necessary to monitor the training and document the progression of condition numbers of the convolutional layer filters. The lack of robustness even with the non-default initialisation sets despite the fact that their filters had reasonably low condition numbers, can benefit from analysing the documented progression of filters condition numbers. During the customised CNN models training, for all weight initialisation sets, we kept monitoring changes to the filters condition numbers with the aim of determining their stability over several training epochs. Figure 5.10, below, depicts the range of condition numbers of each of 96 (=32×3) filters over 10 epochs training of Model-A with the Digits dataset for all four types of filter initialisation sets. Similar patterns were noted with all the other datasets.

We observe that the numbers of filters with very small fluctuation range, increases steadily in the order $\mathcal{W}_D$, $\mathcal{W}_1$, $\mathcal{W}_2$ and $\mathcal{W}_3$. For the unstable filters in these four sets, the range of fluctuation decrease in the same order. These results are consistent with the robustness performance of the corresponding Model-A CNNs, on the Digits dataset, presented in Figure 5.7 where the lack of robustness decreases in the same order. However, the fact that there are still relatively many unstable filters in all these initialisation sets explains that all four schemes are not robust against addition of noise other than at low level. The fact that training with $\mathcal{W}_3$ resulted in a big drop (20%) in performance when the level of noise increased from 0.001 to 0.02, should be considered with the observation that several stable filters in $\mathcal{W}_3$ had very high condition numbers. Marginally different patterns of instability were noted, when the customised schemes with those four sets were



(a) $\mathcal{W}_D$

(b) $\mathcal{W}_1$

(c) $\mathcal{W}_2$

(d) $\mathcal{W}_3$

Figure 5.10: Condition numbers of different initialised filters, during training Model-A on Digits dataset.

trained and tested on the MNIST dataset. The alterations in filter parameters are influenced by the diverse patches of images encountered across batches and epochs during training. As the model processes various image patches over the course of training, the evolving patterns and features in these patches contribute to changes in the filter parameters. This dynamic interaction with different image subsets throughout batches and epochs plays a crucial role in shaping and adapting the model's filters, ultimately impacting its performance and robustness.

Finally, the stability records of the filters in the different initialisation sets, when Model-B was trained on the CIFAR-10 dataset, are less notable compared to the training of Model-A on the Digits or MNIST datasets. Recall that the images in CIFAR-10 are more complex in terms of textural as well as structural content than the other 2 datasets. We also observed that filters in $\mathcal{W}_3$ that were stable during the first training round may not remain stable during the second training round. This results from the fact that modifications to filter weights, enacted as a result of the backpropagation, are determined by the combined effects of all filters on diverse training image batches.

5.4 Performance of US customised CNNs

We repeat the experimental setting and implementation as well as testing the robustness and generalisation with similar settings to Section 5.3. However, we narrow the focus on the classification of US images by the two customised CNN models with the different convolution filter initialisation methods. Before proceeding, we consider the findings from the previous section concerning image modality and the stability/sensitivity of convolutional filters. Despite varying filter pools used for training across different datasets, filter condition number instabilities persisted throughout and post-training in the customised CNNs. Many unstable filters, mostly end with high condition number even when started with reasonably low condition number. In the next section, we shall propose using well established mathematical facts about matrix condition numbers to modify our initialisation procedure.

5.4.1 Refining convolution filter initialisation

Except for $\mathcal{W}_D$, all other initialisation techniques aim to select filters with the lowest condition numbers from a larger pool. However, as the number of channels increases, these methods often include filters with less favourable condition numbers. To address this, we leverage established mathematical insights regarding matrix condition numbers, whereby from given a matrix A of a known condition number various other matrices of the same size can be obtained while maintaining the identical condition number as A. Thus, we state the following lemma, [130].

Lemma 1. Let A be an $n \times n$ real matrix:

1. If $\alpha \neq 0$ is a real number then $\kappa(\alpha A) = \kappa(A)$.
2. If B is an $n \times n$ permutation matrix then $\kappa(BA) = \kappa(A) = \kappa(AB)$.

These observations offer valuable insights into the selection of a limited set of well-conditioned RGFs during the initialisation phase. For our specific objectives, we aim to identify a straightforward set of real-number parameters linked to the image while allowing some variation in condition number ranges. When employing real-number parameters, it is imperative to avoid significant increases or decreases in the entries. Instead of relying on random parameter choices (such as multipliers or permutations), we aim to formalise these parameter selections by considering desired or established relationships between channels within each layer. For instance, the necessity for three channels in the initial layer aligns with the RGB nature of the input images. Subsequent layers' channel counts are determined by the number of feature maps produced by preceding layer filters. Our approach formalises the generation of convolutional filter sets, multidimensional array, based on the following criteria:

$$\mathcal{F}^k_{nnm} = \beta_m \otimes F^k_{nn} \tag{5.1}$$

where each entry in $\beta_m \in \mathbb{R}$, m is the number of depth-wise channel per filter, F is an $n \times n$ random Gaussian matrix, and k in the number of convolutional filter. In special cases such as using 3D image for Model-B, the ***unified channel-wise*** initialisation is defined by the following filters selection:

First convolutional layer:

1. Select 32 filters from its respective pool.
2. For each filter create 3 filters by multiplication with β_3=[0.299, 0.587, 0.114].

The chosen multipliers, β_3, are those used in converting RGB images into grayscale images.

Second convolutional layer:

1. Select 64 filters from its respective pool.
2. For each filter create 32-channel filters by multiplying it with the convex combination coefficients:

   $$\beta_r = \frac{\kappa_r}{\sum_{i=1}^{32} \kappa_i}, \quad r = 1, \ldots, 32$$

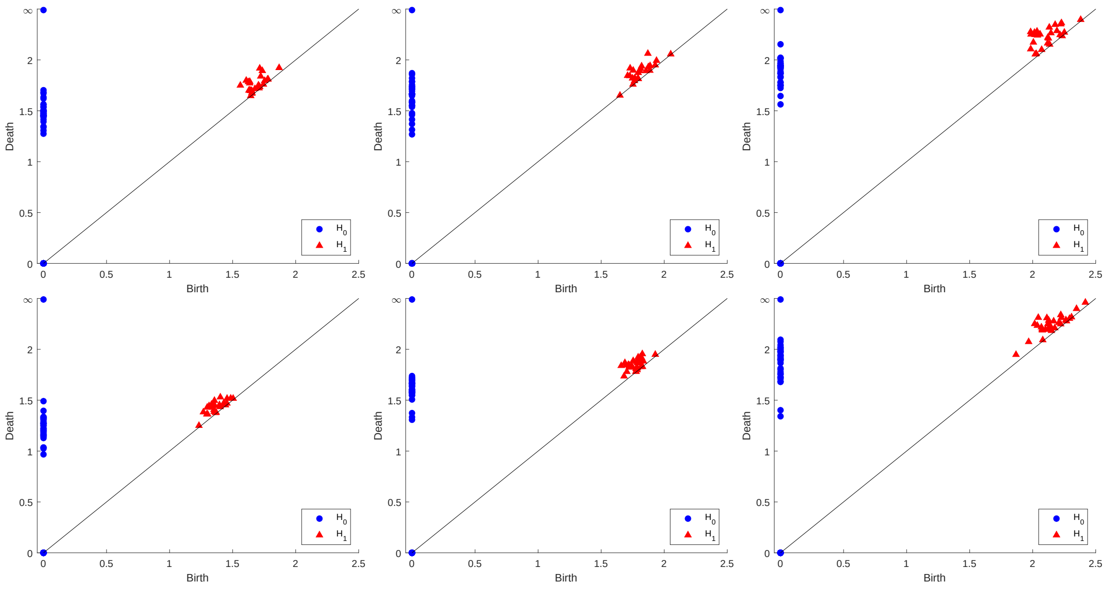
   where κ's are the condition numbers of 32 filters for the first convolutional layer.

Accordingly, we define a modified version of Model-B whereby the convolution filters are chosen according to the above scheme. We shall not apply this modification on the $\mathcal{W}_3$. We shall denote the three new filter initialisation sets as $\mathcal{F}_D$, $\mathcal{F}_1$, and $\mathcal{F}_2$, respectively. Here we use, 5×5 filters on both convolution layers.

Table 5.2: Proposed weight initialisation techniques - revisited.

Name	#channels $(1^{st}, 2^{nd})$ CL	Set size	Condition
$\mathcal{W}_D$	(3,32)	32/64	-
$\mathcal{W}_1$	(3,32)	10^3	Lowest condition number
$\mathcal{W}_2$	(3,32)	10^6	Lowest condition number
$\mathcal{F}_D$	(1,1)	32/64	-
$\mathcal{F}_1$	(1,1)	10^3	Lowest condition number
$\mathcal{F}_2$	(1,1)	10^6	Lowest condition number

Our interest in linking PH investigation to our proposed weight initialisation, stems from the significant differences between the topological behaviour (visualised by PDs) of well and ill-conditioned point clouds of filters as previously demonstrated in Section 5.2.4, Figure 5.11 shows at the top row PDs of the 3 point clouds of 5×5 filter initialisation sets $\mathcal{F}_D$, $\mathcal{F}_1$, and $\mathcal{F}_2$ and the PDs of their inverses at the bottom (see Figure A.7 filter sets). In comparison to Figure 5.3, the number of connected components and holes (i.e. homological features in dimensions zero and one) are significantly reduced for the same number of points in the point cloud, respectively. Therefore, imposing the exact condition number of filters across the channels at initialisation has led to reducing the complexity at the initial step of CNN model training as well as simple topological profiles of the filters point clouds.



(a) $\mathcal{F}_D$ (b) $\mathcal{F}_1$ (c) $\mathcal{F}_2$

Figure 5.11: PDs of point clouds of the unified channel-wise filters and their inverses.

5.4.2 Performance of the customised CNN models on US images

Along with testing our proposed procedure on weight initialisation, we incorporated batch normalisation (BN), [119], to our customised CNNs. BN is a popular technique used in DCNN to improve their accuracy and training efficiency by applying it to the output of each convolutional layer before applying the activation function to normalise the distribution of input values by subtracting the mean and dividing by the standard deviation of the batch of input values. The aim is to reduce the internal covariate shift problem that occurs due to changes in the input value distribution during training. The network also learns parameters for adapting to the specific characteristics of input data. The use of BN in CNNs improves the convergence speed and accuracy of the network, making it less sensitive to changes in input distribution. Hypothetically, this type is expected to work well on diverse image modalities such as medical images due to less adjusting CNN model parameters to the training dataset. However, we need to remember that their use for US image analysis may not have the same effect expected when used for natural images. This is due to the relatively small size ultrasound dataset which cannot be considered as an i.i.d. sample of the unknown population of US breast tumour images.

For testing this approach with Ten-D dataset, we integrate BN layer into few of our customised Model-B CNN architectures (namely Model-D-$\mathcal{F}_D$, Model-B-$\mathcal{F}_1$ and Model-B-$\mathcal{F}_2$), after both convolutional layers before applying ReLU layer. We train, validate, and test these as well as the others schemes that do not use BN, for the Ten-D ultrasound dataset, and the results are shown in Table 5.3. CNN models with batch normalisation layer outperformed those without BN layer by 5-6% with weight initialisation methods $\mathcal{F}_1$ and $\mathcal{F}_2$. Whereas the difference of default, $\mathcal{F}_D$, weight initialisation is less significant. These results are an obvious indication of the power of well-conditioned weight initialisation for training CNN models. As expected the performance of the customised CNNs are lower than the pre-trained CNN models except for the Efficientb0 model which is outperformed by all but the customised Model-B-$\mathcal{F}_2$. Both Model-B-$\mathcal{F}_1$+BN and Model-B-$\mathcal{F}_2$+BN achieve the best test accuracy of 94%. However, the difference between the sensitivity and specificity is increasing, in particular for models with batch normalisation layers, as the filter conditioning is more strict. This problem occurs because of the nature of texture features from malignant tumours and the BN parameters does not generalise well to adapt to various shapes and sizes of US tumours.

Table 5.3: Classification accuracy of various customised CNN model-B with-/without BN .

CNN model	Validation acc.	Test acc.	Sensitivity	Specificity
Model-B, $\mathcal{W}_D$	91.94	90.47	89.76	91.19
Model-B, $\mathcal{F}_D$	90.75	89.88	92.62	87.14
Model-B, $\mathcal{F}_D$, BN	90	91.19	92.38	90.00
Model-B, $\mathcal{W}_1$	90.75	91.42	88.81	94.05
Model-B, $\mathcal{F}_1$	89.55	89.4	84.76	94.05
Model-B, $\mathcal{F}_1$, BN	94.33	94.05	92.86	95.24
Model-B, $\mathcal{W}_2$	93.13	89.64	90.24	89.05
Model-B, $\mathcal{F}_2$	88.36	86.31	90.71	81.90
Model-B, $\mathcal{F}_2$, BN	94.03	94.05	90.95	97.14

5.4.2.1 Robustness testing against random perturbation

To evaluate the robustness of the customised CNN models, we conduct similar experiments to the robustness experiments of section 5.3.4, where we repeatedly degrade the testing images with various levels of natural perturbations. Our study aimed to investigate whether the use of well or ill-conditioned filters in model initialisation and imposing the same condition number over channel filters could facilitate learning of distinguishing features with varying sensitivity to reasonable perturbations. Specifically, we assessed the robustness of the models against small perturbations by adding Gaussian and Speckle noise to the testing dataset, with a mean of 0 and variance ranging between 0.0001 and 0.1.

Besides the customised CNN models, we recall the previous results from Section 4.2. The results presented in Figure 5.12 indicate that initialising with well-conditioned filters, $\mathcal{F}_1$ and $\mathcal{F}_2$, leads to more robust performance against small noise caused perturbations compared to the $\mathcal{F}_D$. Furthermore, the addition of batch normalisation to the model results in decreased sensitivity to the initialisation parameters. However, the condition number of the filters tends to increase after 2-4 epochs, with only a few stabilising thereafter. Model-B with BN achieved the highest performance when tested on the Ten-D dataset with/without noise, particularly when combined with $\mathcal{F}_1$ and $\mathcal{F}_2$ filters. Our customised CNNs trained from scratch on Ten-D US scans outperform those with transfer learning models in relation to Robustness.

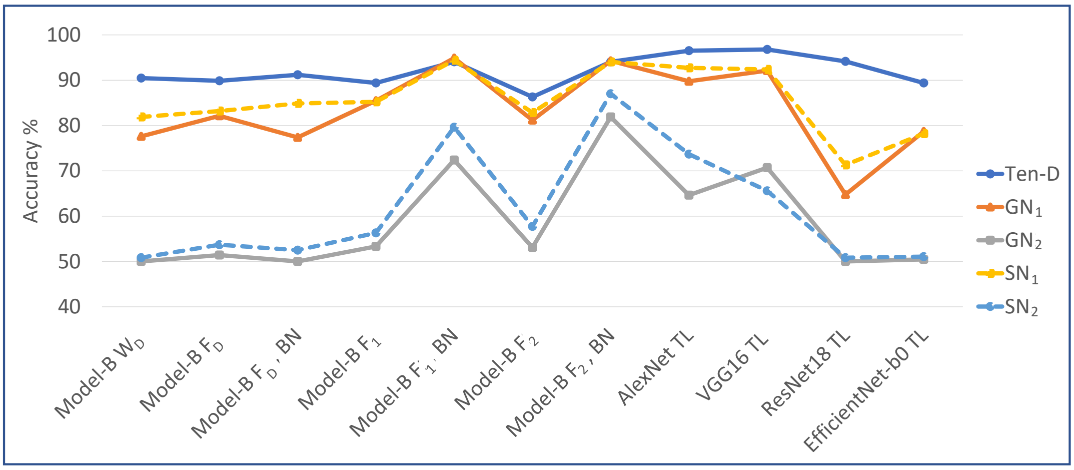


Figure 5.12: Ten-D breast US dataset robustness test.

5.4.2.2 Generalisability to unseen dataset

To assess whether our customised CNN models with the proposed weight initialisation techniques have successfully learnt to generalise to unseen data, we use BUSI dataset as the external testing dataset. We recall the results of the state-of-the-art architectures in the transfer learning mode when tested on BUSI dataset. Figure 5.13 shows the difference in performances of the various CNN models when tested on Ten-D and BUSI datasets. Notably, Model-B without BN are outperforming the rest of models. Imposing the well-conditioning property at the beginning leads to a better generalisation with small to no difference between the classification accuracy on both datasets. Whereas, use of BN layers add a higher risk of overfitting to the training set. This may be due to the higher condition numbers of the filters when using BN. The initial well-conditioned filters may not remain stable during training, leading to a lack of robustness and overfitting.

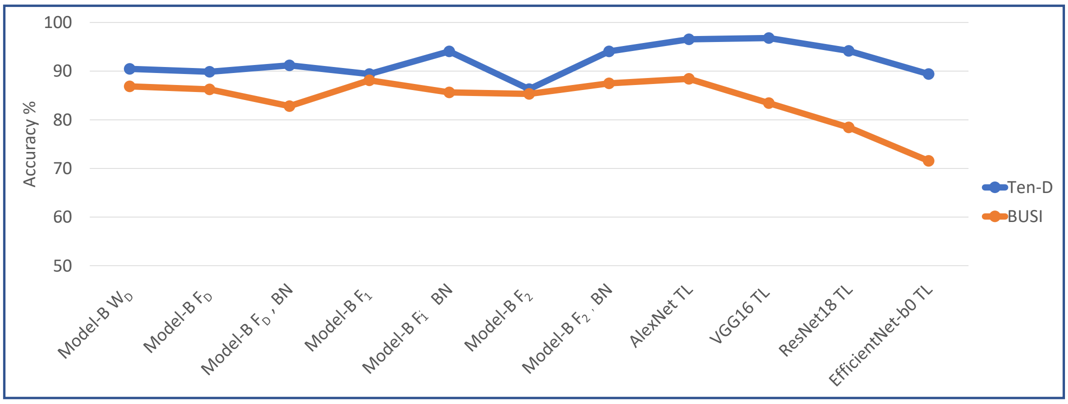


Figure 5.13: Model testing on Ten-D and BUSI breast US datasets.

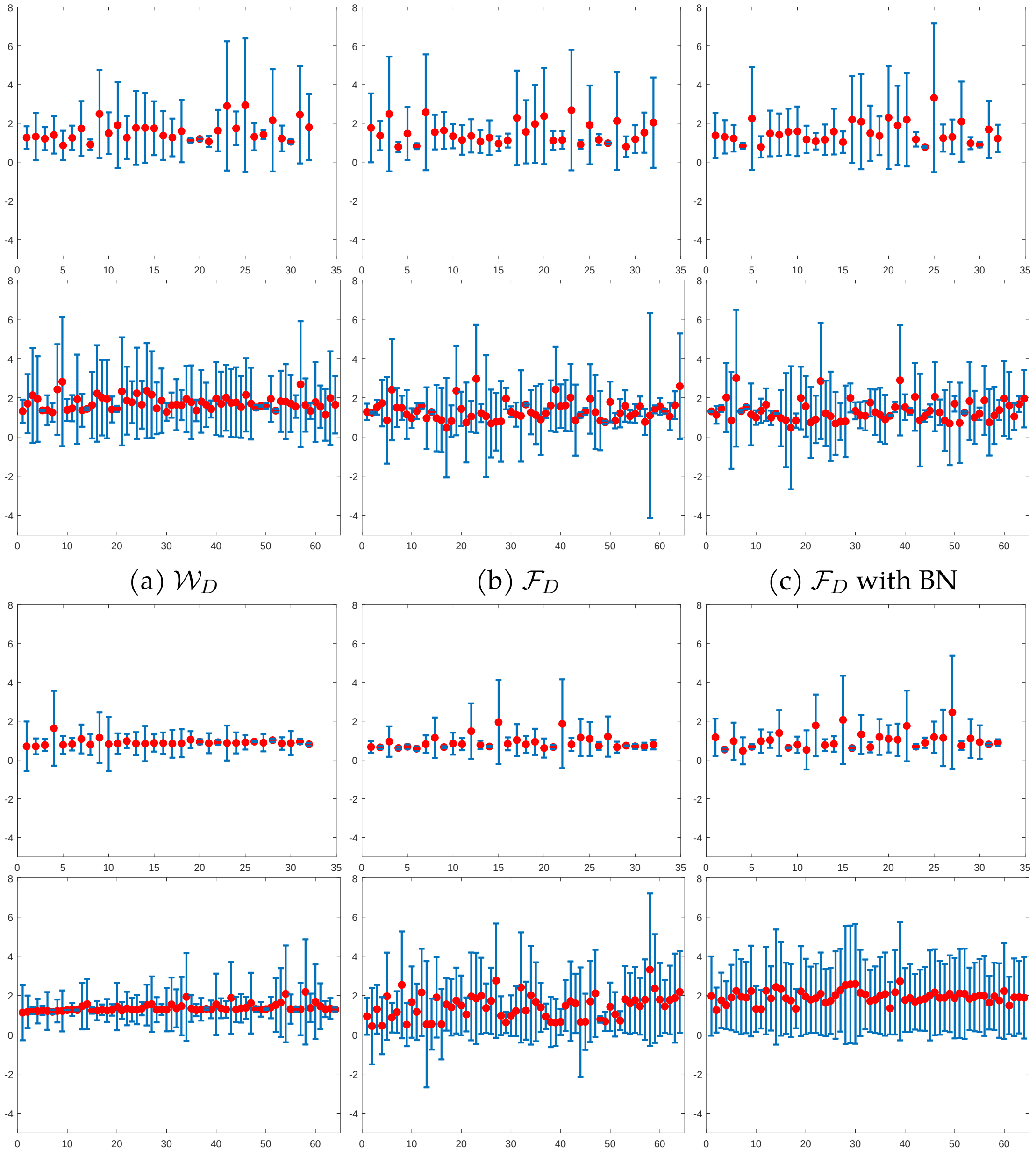
Transfer learning using all four deep learning architectures resulted in greater overfitting, measured by a drop in accuracy of 8.1-17.84%. This can be attributed to a significant increase in the condition numbers of the filters, compared to training the customised model from scratch.

5.4.3 Instability of filters' conditioning during training

Once again, we delve into examining the stability, or lack thereof, of the condition number of convolutional layer filters across a specified number of training epochs. Our focus is primarily on the dynamic behaviour of the condition number of filters during the training process, notwithstanding the final level of accuracy of the CNN models. To achieve this, we train various customised CNN models with a fixed set of parameters and hyperparameters, where the only distinction between them is the choice of filters. These filters are selected based on their initial condition numbers as described in the previous section and their stability over multiple training epochs. This approach enables us to examine the impact of the condition number on the stability, robustness, and generalisation of the models during and post the training process.

Figure 5.14, displays the stability records of (1) the $\mathcal{W}_D$ and $\mathcal{W}_1$ filters, their unified channel-wise versions $\mathcal{F}_D$ and $\mathcal{F}_1$, and the last two with BN. It is difficult to have a good assessment of the effect of the Unified-channel scheme on filter stability, because the charts in the original initialisations are based on the average of all channels of the 32 (or 64) filters. Nevertheless, we can see that for both initialisation sets the unified channel-wise conditioning create more relatively stable filters. Moreover, the BN procedure seem to create more filters with higher instability scales.

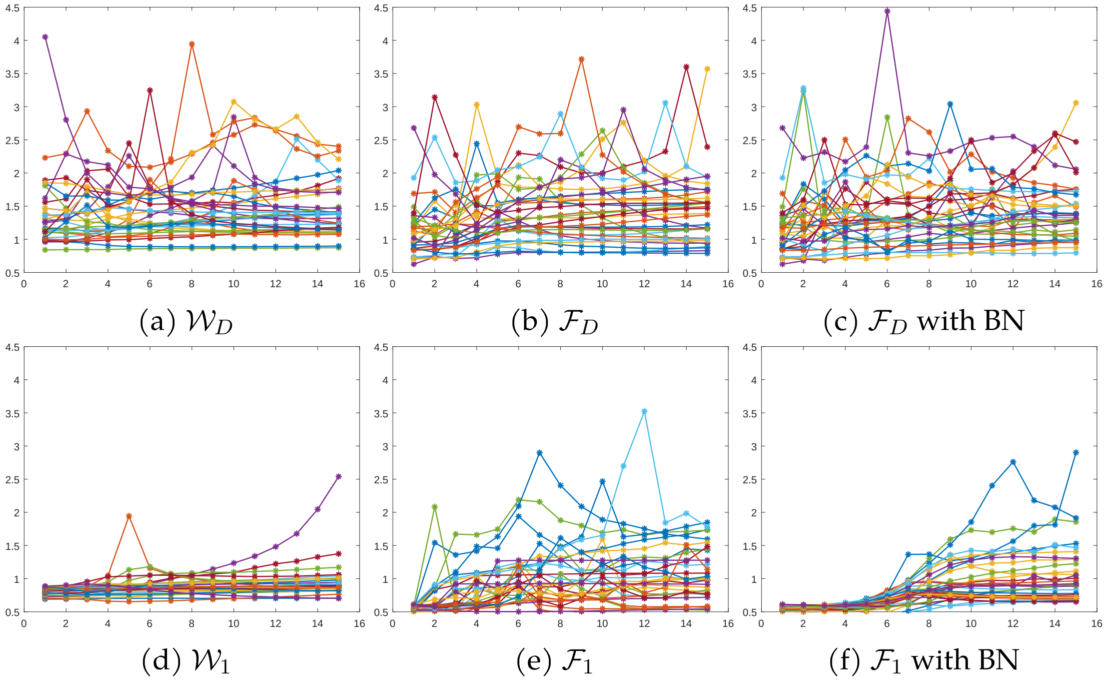
Figure 5.15, illustrates the first layer filters' condition number variability as the training procedure moves from one epoch to the next. It shows that in general, initialisation by selecting the lowest condition numbers from a sufficiently large pool result in filter stability over the first half of the epochs and this is more so with the application of BN. Moreover, more filters return to the what we may call the ***acceptable conditioning zone*** ($\text{Log}(\kappa(A)) \leq 1$) before the last epoch. This may explain the reasonable performances by the corresponding versions of Model-B in terms of accuracy, robustness and ability to generalise.



(a) $\mathcal{W}_D$ (b) $\mathcal{F}_D$ (c) $\mathcal{F}_D$ with BN

(d) $\mathcal{W}_1$ (e) $\mathcal{F}_1$ (f) $\mathcal{F}_1$ with BN

Figure 5.14: Filters condition number stability - Model-B.



(a) $\mathcal{W}_D$ (b) $\mathcal{F}_D$ (c) $\mathcal{F}_D$ with BN

(d) $\mathcal{W}_1$ (e) $\mathcal{F}_1$ (f) $\mathcal{F}_1$ with BN

Figure 5.15: Averaged condition number from initialisation to the last epoch - 1^{st} convolutional layer of Model-B.

In summary, the analysis of Figures 5.14 and 5.15 confirms that initialisation with well-conditioned filters using unified channels schemes provides better level of stability, but filter instability/stability is also dependent on the variation in contents of training images. This means that the problem of filter instability cannot be overcome by initially selecting lowest conditioned filters from a larger pool of RGF's. In fact, we tried much large pools of RGFs, than presented here but instability continued to plague the training, (see Table A.4). Instead, we need to reduce the effects of the training images by regularising the growth of filters condition numbers during training. This will be investigated in the next chapter.

5.5 Summary and conclusion

In this chapter, we progressed our investigation by designing a credible strategy to develop high performing slim US customised CNN architectures that are robustness against adversarial noise perturbation while generalise well to unseen data. This strategy is based on controlling the condition numbers of initialised convolution filters and their post training output filter sets. The viability of this strategy was demonstrated incrementally by developing a sequence of schemes that meet the requirements while being very slim using at most 2 convolution layers with reduced numbers of well-conditioned filters. We investigated the algebraic properties of the various filter sets in relation to cross -channels weight initialisation settings. We proposed a filter selection approach based on their condition numbers, and investigating both at their initial settings and throughout the training process, to address the instability of CNN models and lessen the chances of overfitting. Nevertheless, our findings indicate that there exists a trade-off between the CNN classification accuracy and its robustness, as the proposed condition number and stability-based method exhibited lower classification accuracy compared to other CNN models. This highlights the critical importance of striking a delicate balance between model performance and robustness while choosing filters in CNNs, thus indicating the need for further harnessing constrains on filters' condition number during training. On the other hand, the lower classification accuracy around 86-91% is aligning with radiologists' expectation in terms of recognising the type of tumour.

Our investigated filter selection approaches do yield well-conditioned and many stable filters for Model-A and Model-B. For more complex CNN architectures, using dropouts and regularisation techniques could help maintain well-conditioned and stable filters. However, incorporating regularisation does not mean controlling the condition number of filters precisely. Almost all filters of pretrained models retrained in transfer learning mode, were highly ill-conditioned, and their training made the final filters more ill-conditioned with few exceptions. While performing well on US datasets, they suffered from which resulted in lack of robustness and inability to generalisation.

In general, this chapter's investigations reveal a strong connection between filter's algebraic properties, robustness of the CNN model to adversarial behaviour, and generalisation to unseen data. This fits the widely assumed expectation that robust CNN models should be less prone to tiny perturbations to the input and is certainly linked to the potential of overfitting. The experimental results demonstrated that regardless of the pool from which well-conditioned are selected, training always produces unstable filters. A more realistic approach to stabilising filter conditioning should be based on regularising changes to filters, caused by the backpropagation procedure, throughout training.

Chapter 6

Stabilising Filters Conditioning by Matrix Surgery

In the previous chapter, we demonstrated the instabilities of filters condition number continue to fluctuate regardless of the filter's properties at initialisation and irrespective of the size of the RGF pool from which these filters are selected. The filters conditioning instabilities problem persisted, with lower severity, even when we used a unified-channel conditioning scheme or the dropping of 50% of filters that were more unstable during a first round of training. In this chapter, we introduce an innovative matrix surgery, based on their singular value decomposition, to reduce and control the instability of filters condition number. This approach is inspired by the concept of ***topological surgery*** of low dimensional manifolds considering the action of filter matrices as a linear transformation on their domains. It is also driven by our investigations of the links between convolution filters condition numbers and the distribution of extracted convolved US texture feature landmarks as well as the persistent homology of their point clouds. The matrix surgery approach is carried out through modifying the spread of the filters singular values determined by singular value decomposition.

In Section 6.1, we introduce basic matrix algebra concepts that describe the action of any matrix as a linear transformation of its domain in terms of its spectral analysis, and review existing work on controlling the ill-conditioning in CNNs and reducing the condition number techniques. Section 6.2, presents our singular value decomposition based matrix surgery approach for controlling and reducing filters' condition numbers. Section 6.3, shows the effect of the matrix surgery in large sets in terms of their distribution of condition numbers, distribution of singular values, and their topological profiles. Section 6.4 presents the application of the matrix surgery in the context of CNNs during initialisation (for pretrained models) and throughout the training process.

6.1 Introduction and related work

Any square $k \times k$ real matrix A defines a linear transformation of the Euclidean space $\mathbb{R}^k$ via matrix multiplications on the left, i.e. $\forall\ v \in \mathbb{R}^k$

$$A(v) := Av^T \tag{6.1}$$

where v^T is the transpose column vector of v. It defines another linear transformation by matrix multiplication on the right, i.e. $\forall\ v \in \mathbb{R}^k$

$$A(v) := vA \tag{6.2}$$

The action of A is uniquely determined by its mapping of the standard basis of $\mathbb{R}^k$, i.e.

$$\{e_1 = (1, 0, \dots, 0), \quad e_2 = (0, 1, \dots, 0), \quad e_k = (0, 0, \dots, 1)\}$$

Topologically, each of these transformations define a smooth map on the sphere S^{k-1} whose image in $\mathbb{R}^k$ is an ellipsoid whose major axes have their lengths are the magnitudes of the eigenvalues of A and their directions are determined by the corresponding eigenvectors of A. The condition number of the matrix is linked to the nature of the geometric distortion of the mapped ellipsoid near the smallest eigenvalue. For highly ill-conditioned matrices the smallest major axis is too small in comparison to the largest major axis, and the smaller it becomes the more ellipsoid distorted geometrically bordering on having a singularity near the end of the smallest major axis.

Reducing the condition number of such matrices amount to applying topological surgery on the ellipsoid, to remove the potential singularity, by pushing a sphere of a reasonably larger radius through the narrow tunnel. Here, we introduce a process of manipulating a matrix that results in mimicking such topological surgery, and we call this process as matrix surgery.

Accordingly, the inspiration for what we call matrix surgery can be attributed to the Surgery theory[1] on manifolds. Generally, topological surgery allows the construction of new manifolds (or homoeomorphic copies of a manifold) by modifying through surgery operations by the topological ***connected sum*** (#) operation. These operations usually involve removing a closed submanifold of a given

[1] Surgery theory was developed in the 1950s and 1960s by J. Milnor, A. Wallace, and others.

manifold, thereby creating new boundary components for the given manifold, and then gluing to it another manifold (that have the similar boundary components profile) along these common boundary components according to a certain controlled manner [131–134].

The most commonly known application of topological surgery, and easy to illustrate, is the process of creating closed connected *Riemann surfaces* X_g of any finite genus g, by repeatedly removing two closed discs from the surface of the 2-dimensional sphere S^2 and gluing a close bounded cylindrical surface along their bounding circles to the exposed closed circles of the sphere. Figure 6.1, below, is a simple illustration of this process. This process is similar to the well-known Dehn-Twists on the meridian curve of the above described annulus but without twisting, and it does not change the ellipsoid topologically, but changes the local geometry around the near-singularity region. Note that, Denn-Twists[2] along a closed curve on a Riemann surface X_g is implemented via topological surgery on an annulus around its meridian curve, [135].

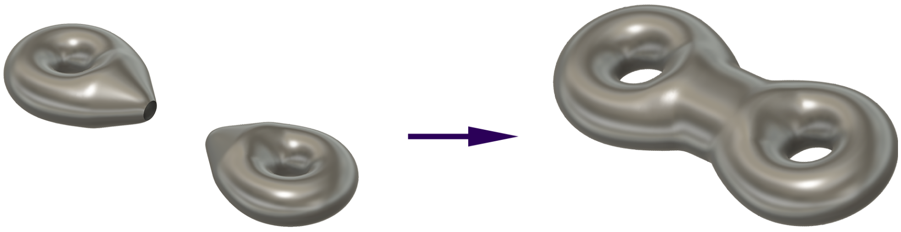


Figure 6.1: An illustration of gluing two surfaces.

Besides being instrumental in settling the Poincare Conjecture and classification of high dimensional manifolds, topological surgery is becoming a useful tool in a range of computer vision applications as diverse as black hole research and medical imaging. The medical imaging field benefited from the application of manifold surgery and has witnessed great success since then. The authors in [136] developed an "Automated manifold surgery" by maintaining both geometric accuracy and topological correctness after identifying the inaccurate connections between adjacent banks of a sulcus due to single voxel misclassification in the highly folded cerebral cortex, resulting in a topologically inaccurate model. Recently, the authors in [137] developed a deep learning based Rapid Reconstruction of Topologically-Correct Cortical Surfaces.

The concept of topological surgery may result in the modification and/or transformation of manifolds. However, our matrix surgery process do not change the

[2] The Dehn-Twists around $3g - 1$ non-separating closed curves on X_g, generate its ***mapping class group*** of homotopy classes of orientation preserving homeomorphisms of X_g.

topology of the mapped Ellipsoid but simply results in local geometry change. Matrix surgery in this thesis is a general term that refers to any technique that modifies its entries and results in reducing its condition number in such a way that the associated ellipsoid are less vulnerable to singularity, as explained above.

In terms of imposing orthogonality conditions on trainable DL model weights, several related research works that have investigated this approach in relation to the issues of underfitting/overfitting, instability of CNN performance, robustness, and generalisation to unseen data. These include orthonormal and orthogonal weight initialisation techniques, [111–113] orthogonal convolution [116], orthogonal regularizer [102], orthogonal deep neural networks [117], and orthogonal weight normalisation [114]. Since orthogonal/orthonormal matrices are optimally well-conditioned, our attempt to explain DL overfitting in terms of ill-conditioning of convolution filters learnt through training/retraining is consistent with and supported by these works. In most of these works, there is no explicit discussion about instability of convolution filters conditioning as a result of the model training procedure. However, these publications can be categorised within the emerging paradigm of spectral regularisation of NN layers weight matrices. For example, J. Wang et al., [116], assert that imposing orthogonality on convolutional filters is the appropriate mitigating tool of DCNN models training instability for improved performance. Interestingly, A. Sinha [102] point out that susceptibility of neural networks to adversarial attacks can be attributed to ill-conditioned learnt weight matrix. In fact, their orthogonal regularisation aims to keeping the learnt weight matrix's condition number sufficiently low and prove it improve adversarial accuracy when tested on the natural image datasets of MNIST and F-MNIST.

S. Li et al., in [117], note that existing spectral regularisation schemes, are mostly motivated to improve training for empirical applications, conduct a theoretical analysis of such methods using bounds the concept of Generalisation Error (GE) measures that is defined in terms of the training algorithms and the isometry of the application feature space. They conclude that optimal bound on GE is attained when each weight matrix of a DNN has a spectrum of equal singular values and call such models OrthDNNs. To overcome the high computation requirements of strict OrthDNNs, they define approximate OrthDNNs by periodically applying their Singular Value Bounding (SVB) scheme of hard regularisation. In general, controlling weights' behaviour during training has proven to accelerate the training process and reduce the likelihood of overfitting the model to the training set e.g. weight standardisation in [138], weight normalisation/reparameterization [139], centred weight normalisation [140], and using Newton's iteration controllable orthogonalization [115]. Imposing Lipschitz condition on convolution filters is less restrictive than orthogonality conditions have also been investigated, [141–144]. Like the methods used to impose orthognolity conditions, the method proposed by C. Runkel [144] to control Lipschitz constants result in reducing the condition numbers but more severely than our SVD-Surgery and have the potential to ignore the features extracted from the various batches.

Most of the above proposed techniques have been developed specifically to deal with trainable DL models for the analysis of natural images and one may assume that these techniques are used frequently during the training after each epoch/batch. However, none of the known state-of-the-arts DL models seem to implicitly incorporate these techniques. In fact, our investigations of these commonly used DL models revealed that the final convolution filters are highly ill-conditioned, [19]. Controlling the convolution filters norm doesn't necessarily control their condition numbers unless it is applied for the feedforward and back-propagation of CNNs. GradInit [145] and MetaInit [146] propose methods to control the norm of the network layer showing that these methods can accelerate convergence while improving model performance and stability. In both cases, the model requires additional trainable parameters and control of the condition numbers during training is not guaranteed.

Our literature review revealed that reconditioning and regularisation have long been used in analytical applications to reduce/control the ill-conditioning computations noted. In the late 1980's, E. Rothwell and B. Drachman, [147], proposed an iterative method to reduce the condition number in ill-conditioned matrix problem that is based on regularising the non-zero singular values of the matrix. At each iteration, each of diagonal entry in the SVD of matrix is appended with a ratio of a regularising parameter to the singular value. This algorithm is not efficient to be used for our motivating challenge. In addition, the change of the norm is dependent on the regularising parameter. Note that this iterative procedure amount to iterative matrix surgery.

Here, we introduce a singular value decomposition-based matrix surgery (SVD-Surgery) technique to modify matrix condition numbers that is suitable for stabilising the actions of ill-conditioned convolution filters on point clouds of image datasets. It decomposes square matrices by SVD factorisation, replaces the smaller singular value(s), and then reconstruct the original matrix with the resulting singular value diagonal matrix. SVD-Surgery preserves the norm of the input matrix while reducing the norm of its inverse. This means that SVD-Surgery make changes to the PH of the inverse matrices point clouds. We expect that PH analysis of point clouds of matrices (and those of their inverses) can provide an informative understanding of stability behaviour of DL models of image analysis.

6.2 SVD based matrix surgery

In this section, we introduce an innovative procedure to perform matrix surgery that aims to reduce the condition number of matrices. In the wide context, SVD-Surgery refers to the process of transforming ill-conditioned matrices to improve their condition numbers. In particular, the focus is on addressing matrices that deviate significantly from possessing orthogonality or orthonormality characteristics by replacing or approximating them with matrices that exhibit better conditioning. Since condition number of a matrix is defined in terms of it largest and smallest condition number then we need to factorise it by the well-known singular value decomposition. The surgery proceeds by modifying the eigenvalues and remultiply the left and right orthogonal singular vectors along with the new singular value diagonal matrix. Recalling that the singular value decomposition of a matrix $A \in \mathbb{R}^{m\times n}$ is defined by:

$$A = U\Sigma V^T \tag{6.3}$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are left and right orthogonal singular vectors (unitary matrices); diagonal matrix $\Sigma = diag(\sigma_1, ..., \sigma_n) \in \mathbb{R}^{n\times n}$ are singular values where $\Sigma = \sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n \geq 0$. Since the ratio of the largest and smallest singular values is determining the ill-conditioning of a matrix, we recall equation 2.9 in Section 2.3:

$$\kappa(A) = \sigma_1/\sigma_n$$

where σ_1 and σ_n are the largest and smallest singular values of A, respectively. Altering the singular values while keeping the left and right factor matrices of the SVD decomposition results in a new matrix with a different condition number. In order to reduce the condition number of the original matrix, the singular values alteration should preserve their monotonic order while reducing the distance between the largest and the smallest new singular values. Preserving monotonicity enables conducting topological surgery on the ellipsoid that corresponds to the original matrix by pushing spheres through the narrow tunnel. Increasing the last singular value alone may not be enough to make a significant/desired reduction in the matrix condition number, especially if the gap between it and the one above it is small compared to the gap between it and the first singular value. Accordingly, a significant reduction in conditioning of a matrix may require changing many more singular values and push them towards the most significant singular value, i.e. changing their spread. The extreme replacement of all the singular values with the most significant one produces orthogonal matrix with optimal condition number of 1. However, doing that for the convolution filters during training CNN models amounts to avoiding fitting the CNN model to the training image dataset.

SVD-Surgery can be customised in several ways depending on the desired characteristics of the output matrices to suit their intended application. The SVD-Surgery, described below, is equally applicable to rectangular matrices.
Given any matrix A, an SVD-Surgery on A outputs a new matrix of the same size as follows:

1. Compute its SVD decomposition,
2. From the diagonal matrix factor Σ construct another diagonal matrix $\tilde{\Sigma}$ by replacing the small singular value(s) while keeping their descendant order

$$\tilde{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & ... & 0 \\ 0 & \tilde{\sigma}_2 & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & \tilde{\sigma}_n \end{bmatrix}$$

   where the updated singular value $\tilde{\sigma}_i$'s are selected to maintain low condition number while the new diagonal entries remain monotonically decreasing, and
3. Reconstruct the output matrix $\tilde{A}$ as follows:

$$\tilde{A} = U\tilde{\Sigma}V^T$$

The modification of singular values leads to the adjustment of the matrix operation impact along the orthogonal singular vectors of U and V. The monotonicity requirement guarantees reasonable control over these adjustments. Although the orthogonal regularisation method proposed by [116] and the SVB method suggested by [117] can reduce condition numbers and may enhance overfitting control for deep learning models trained on natural images, they fail to satisfy the monotonicity condition. Furthermore, their effectiveness in training deep learning models for US image datasets cannot be guaranteed. While the SVB method is more restrictive than SVD-Surgery in controlling condition numbers, no analysis has been performed on the norm of these matrices or their inverses. Our proposed SVD-surgery approach is tailored to the intended application and aims to lower excessively high condition numbers while preserving the norm of original matrices. By replacing all singular values with the largest singular value, an orthogonal matrix with a condition number of one can be generated, but this approach overlooks significant variations in the training data along some of the singular vectors, leading to less effective learning. A more effective way is to try to change the spread of singular values by moving the lower ones upwards while keeping their non-ascending order so that all move nearer to the most significant one. Such a less drastic surgery strategy can include, but not limited to, changing the singular values in a formal way as follows:

Select a diagonal position j, $1 < j < n$, and let $\tilde{\sigma}$ be a convex linear combination:

$$\tilde{\sigma}_k = \sum_{k=j-1}^{n-1} \alpha_k \sigma_k, \quad where \quad \alpha_i \geq 0, \quad \sum \alpha_i = 1$$

and set $\tilde{\sigma}_k$ for each $j \leq k \leq n$.

The choice of j, and the linear combination parameters can be made application-dependent and possibly determined empirically. At its extreme, this approach allows for setting $\tilde{\sigma} = \sigma_j$, which is a conservative strategy compared to orthogonal regularisation methods, as it maintains the monotonicity of the singular values. For our intended application, layer-specific parameter choices should be made, but the linear combination parameters should not result in substantial rescaling of the training dataset features along the singular vectors. Although SVD surgery can be applied to inverse matrices, using the same replacement strategy and reconstruction may not result in a significant reduction in its condition number.

Example:

Suppose B is a square matrix with $n = 3$ drawn from the normal distribution with mean $\mu = 0$ and standard deviation $\sigma = 0.01$ as follow:

$$B = \begin{bmatrix} -0.01960899999 & 0.02908008031 & -0.01058180258 \\ -0.00197698226 & 0.00825218894 & -0.00468615581 \\ -0.01207845485 & 0.01378971978 & -0.00272469409 \end{bmatrix} \tag{6.4}$$

Singular values of B are $\Sigma = diag(\sigma_1, \sigma_2, \sigma_3)$ and to modify and reconstruct $\tilde{B}_1$, $\tilde{B}_2$, and $\tilde{B}_3$ by replacing one and/or two singular values s.t. $\tilde{\Sigma}_1 = diag(\sigma_1, \sigma_2, \sigma_2)$, $\tilde{\Sigma}_2 = diag(\sigma_1, \tilde{\sigma}_2, \tilde{\sigma}_3)$ and $\tilde{\Sigma}_3 = diag(\sigma_1, \sigma_1, \sigma_1)$, respectively. New singular values in $\tilde{\Sigma}_2$ are convex linear combinations s.t. $\tilde{\sigma}_2 = 2\sigma_1/3 + \sigma_2/3$ and $\tilde{\sigma}_3 = \tilde{\sigma}_2$. After replacement and reconstruction, the condition number of $\tilde{B}_1$, $\tilde{B}_2$, and $\tilde{B}_3$ are significantly lower compared to the original matrix as shown in Table 6.1, by using Euclidean norm.

Table 6.1: Norm and condition number of matrix B and post-surgery $\tilde{B}_i$.

	$\|A\|$	$\|A^{-1}\|$	$\kappa(A)$
B	0.041883482	2034.368572	85.20644044
$\tilde{B}_1$	0.041883482	199.5721482	8.358776572
$\tilde{B}_2$	0.041883482	30.36464182	1.271776943
$\tilde{B}_3$	0.041883482	23.87576058	1

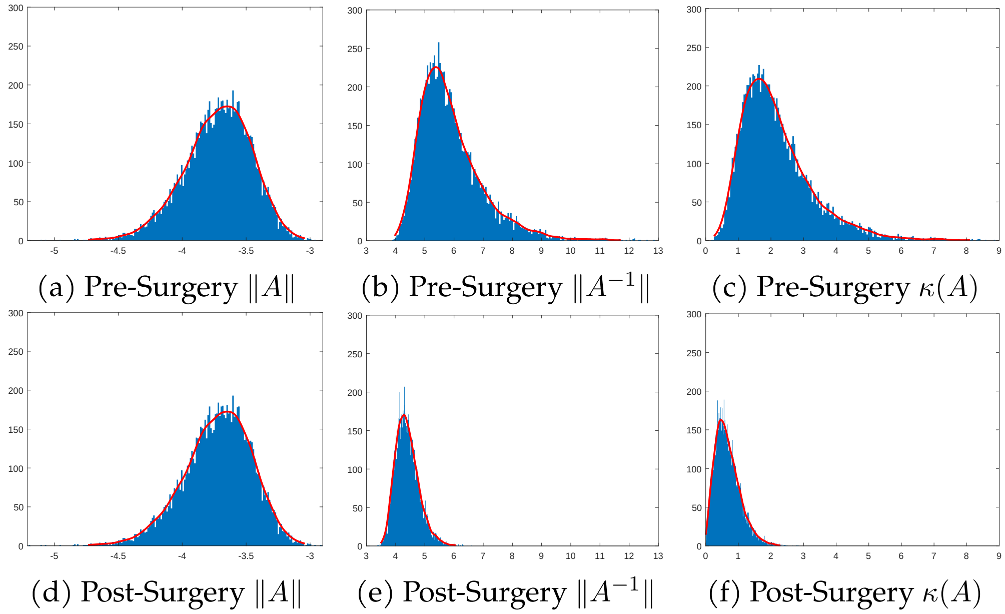
One way to control the condition number of CNN filters during training is by controlling their singular values. The implementation of this type of surgery can be integrated into customised CNN models for the analysis of natural and medical image datasets as a filter regularisation. This approach can be applied at the filter initialisation when training from scratch, on pretrained filters when training in the transfer learning mode, as well as at each epoch on filters that become ill-conditioned when modified during training by backpropagation post every batch/epoch. However, modification to convolution filters during training depends on the optimisation adopted by the trained CNN architecture, the convolution layer parameters and the training dataset batches. Therefore, the choice of appropriate SVD surgery per convolution layer, in terms of improved Generalisation and robustness performance, requires extensive training and retraining. Accordingly, we shall confine our investigation for this chapter on determining the effect of a reasonably simple SVD-surgery on the initialised/pretrained convolution filters. This will be done in Section 6.4, below. We shall next investigate the effect of SVD-surgery on various properties of point clouds of convolution filters.

6.3 Effects of SVD-Surgery on filter point clouds

The SVD-surgery operates on single matrices, and create new ones. Here, we investigate its impact on larger sets of kernel matrices in terms of their overall algebraic and topological behaviour. In particular, we present the impact of surgery (1) on the distribution of condition numbers along with its norms in Subsection 6.3.1, (2) on the distribution of eigenvalues in Subsection 6.3.2, and on their spatial distribution in Subsection 6.3.3.

6.3.1 Distribution of condition number of filter point clouds

To demonstrate the impact of SVD-Surgery on convolution filters point clouds empirically, we generate N sample of $n \times n$ random Gaussian matrices. We compute the norm of the original matrices, the norm of their inverses, and their condition number to observe and analyse changes to their distributions in terms of the level of SVD-surgery applied to each set. For simplicity of visualising the effect of gradual matrix surgery, we present our findings, in Figure 6.2, based on $N = 10^4$ samples of 3×3 random matrices selected from the normal distribution $\mathcal{N}(0, 10^{-4})$. We use the simple SVD-surgery by replacing the smallest singular value σ_3 with the new value $\tilde{\sigma}_3 = \sigma_2$. For graphs of matrix condition number, the x-axes is indexed by Log_{10}(condition numbers).



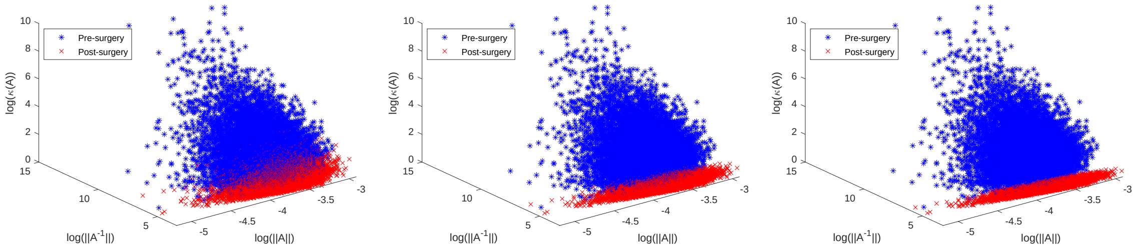
(a) Pre-Surgery $\|A\|$ (b) Pre-Surgery $\|A^{-1}\|$ (c) Pre-Surgery $\kappa(A)$

(d) Post-Surgery $\|A\|$ (e) Post-Surgery $\|A^{-1}\|$ (f) Post-Surgery $\kappa(A)$

Figure 6.2: Distributions of (Norm, Norm of inverse, condition number) of 3×3 matrices pre and post-surgery.

While, no change is detected in the distribution of the selected matrices norms post-surgery, these results reveal notable alterations in the distribution of the inverse matrices norms that in turn yielded significant change in the distribution of their condition numbers post-surgery. The condition number ranged from approximately 1.2 to 10256 in the original set, whereas after the replacement and reconstruction, the minimum and maximum numbers are [1.006, 17.14].

The changes in Figure 6.2, above, reflect the effects of the simple surgery that only altered the smallest singular value. It is expected that these distributions vary in terms of the level of matrix surgery. The use of a linear combination enables the maintenance of the condition number within a specified threshold, based on the singular value distribution. Figure 6.3, below, displays 3D visualisations of these distributions for 3 different surgery levels: (a) the above one, (b) where σ_2 and σ_3 are replaced with $\sigma_1/3 + 2\sigma_2/3$, and (c) where σ_2 and σ_3 are replaced with $(\sigma_1 + \sigma_2)/2$. After matrix surgery, the minimum and maximum condition number values for both sets are now [1.004, 2.687] and [1.003, 1.88], respectively.



(a) $\tilde{\sigma}_3 = \sigma_2$ (b) $\tilde{\sigma}_2$ and $\tilde{\sigma}_3 = \sigma_1/3{+}2\sigma_2/3$ (c) $\tilde{\sigma}_2$ and $\tilde{\sigma}_3 = (\sigma_1 + \sigma_2)/2$

Figure 6.3: 3D depiction of distribution of 3×3 RGF matrices pre- and post-surgery at 3 different levels.

In summary, different surgery results in different changes to the range of condition numbers but retains the norm of the original matrices. In fact, as the linear combination defining the SVD-surgery approaches a state where the smaller singular values become closer to the largest one, the range of conditioning numbers becomes tighter. These results emphasise the significance of the link between the distribution of singular values and the task of selecting an appropriate level of SVD-surgery for convolution filters. We shall discuss that link in the next section, but before doing that we shall return to the challenge of selecting and illustrate that determining the appropriate SVD-surgery for point clouds of convolution filters of different sizes, beyond 3×3.

Recall that in Chapter 4, we investigated the condition number of pretrained CNN convolutional layer filters and found that the majority of these filters were tending towards ill-conditioning regardless of the architecture design and the type of regularisation applied during training. This is a strong motivation for exploiting the advantages of SVD-Surgery to modify the pretrained convolutional filters before re-training for US scan images. Most state-of-the-art CNN models use filters of mixed sizes beyond 3×3. For example, the pretrained Alexnet convolution filters in the first layer uses 11×11, and it consists of filters of other smaller sizes in subsequent layers. The singular value distribution of the sets of pretrained filters per convolutional layer could guide the selection of appropriate levels of SVD-surgery in that layer. Those distributions reflect the collective spread of singular values of all the filters. For each single filter, the spread of its singular values determines the success any SVD-Surgery in controlling its condition number. However, for large filter sizes beyond 3x3 selecting appropriate linear combination of singular values for a point cloud of filters, of varying spread of singular values, is not a straightforward task. The following example, illustrate this challenge.

Example: Consider the two 3-channel 11×11 pretrained AlexNet filters used in the last two chapters:

$$w_1 = (w_{11}, w_{12}, w_{13}) \quad w_2 = (w_{21}, w_{22}, w_{23})$$

The various channel components of the two filters have a wide range of condition numbers. Now, iteratively apply a sequence of SVD-Surgery by simply modifying two consecutive singular values. For each $i = 1, \ldots, 11$, let Si be the simple SVD-Surgery defined by replacing the $(12-i)^{th}$ singular value with the $(11-i)^{th}$ singular value, i.e. the S_1 level of surgery $\tilde{\sigma}_{11} = \sigma_{11}$, ..., etc. Table 6.2, below, shows the condition number of operated on filters $w_{1,k}$ and $w_{2,k}$ per the channel $k = 1, 2, 3$ after each successive application of the $\{S_1, \ldots, S_i\}$.

Comparing the achieved condition number at each stage with that of the original pretrained filters, reveal that all the filters eventually become well-conditioned. However, there is no clear link between the original filter condition number and the speed with which it becomes reasonably well-conditioned. For instance, The condition number of the third channel of w_2 dropped by approximately 150 times the original in one step, whereas the dropping rate for the second channel is approximately 2 times. Additionally, the filters $w_{1,2}$ and $w_{2,1}$ have relatively similar condition numbers of 434.02 and 487.53, respectively, and yet their speed of reduction are far from each other. The different reduction rates, at any stage, is an indication of different spread of their original singular values is different. The overall condition number reduction rate, up to S_{10}, for the relatively well-

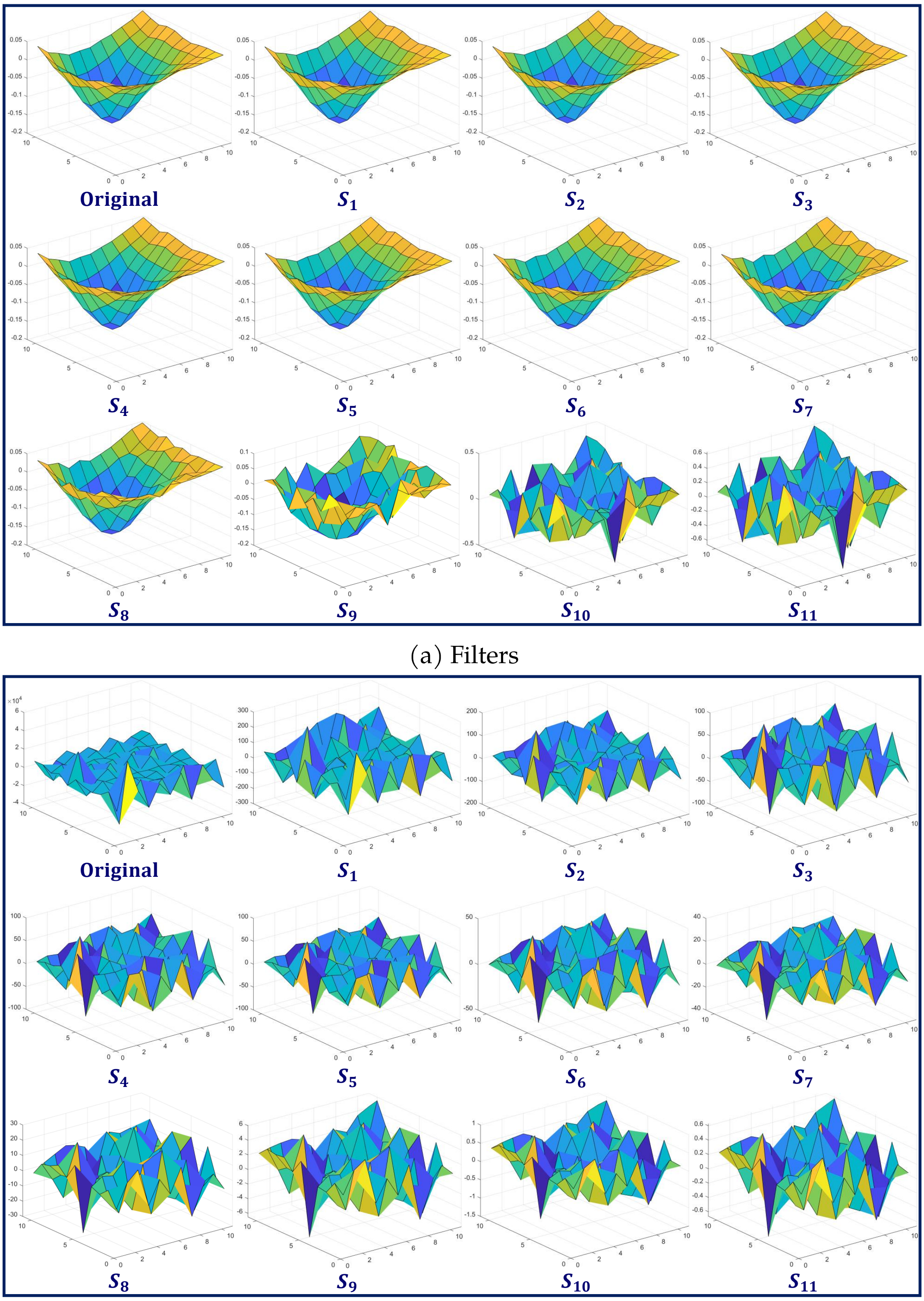
Table 6.2: Impact of iterative SVD-Surgery (S_i) on conditioning of filters w_1 & w_2.

Surgery level	$\kappa(w_{1,1})$	$\kappa(w_{1,2})$	$\kappa(w_{1,3})$	$\kappa(w_{2,1})$	$\kappa(w_{2,2})$	$\kappa(w_{2,3})$
Original	94.08	434.02	52.97	487.53	1903.70	59133.33
S_1	79.64	97.13	40.83	273.75	938.67	400.50
S_2	28.18	49.95	20.43	180.03	576.44	194.94
S_3	24.66	26.98	14.66	105.49	208.69	110.97
S_4	9.61	20.43	10.77	77.92	149.21	92.48
S_5	8.09	15.45	7.93	63.79	119.23	79.88
S_6	7.36	13.87	6.64	41.72	102.80	47.02
S_7	6.05	10.92	5.01	30.86	63.40	31.56
S_8	3.01	6.16	4.56	25.87	33.31	27.06
S_9	1.35	3.42	1.97	4.45	5.09	6.21
S_{10}	1	1	1	1	1	1
S_{11}	1	1	1	1	1	1

conditioned filter w_1 is faster than its counterpart w_2. As demonstrated in Section 6.2, replacing all singular values of a matrix with its largest singular value will lead to an orthogonal matrix. Therefore, the condition number of all matrices at S_{10} level of surgery is one, and the extremist level of the surgery S_{11} is replacing all singular values with 1 resulting in condition number being one. The overall structures of the original matrices' and their inverses remain the same to a certain level of the SVD-Surgery with a reduced scale of the matrix inverses. To illustrate the kind of changes to the entries of the filters and their inverses as a result of these surgeries, Figure 6.4 depicts the entries of the highly ill-conditioned filter $w_{2,3}$ and its inverse through this sequence of surgeries.

We first observe that $w_{2,3}$ post the first surgery S_1 yields significantly improved conditioning, and maintains its norm while the norm of its inverse is reduced as manifested by the reduced scale of entries from [-4,6] $\times 10^4$ to [-3,3] $\times 10^2$. Although, these kinds of changes are persistent until the S_7 level of surgery where the matrix entries start adapting to the reconstructed matrices.

In conclusion, deciding on the level of matrix surgery suitable for point clouds of pretrained filters, is a tough challenge, especially for layers that use filters of larger sizes than 3×3 sizes. Such a decision is dependent on the spread of the singular values of all the individual filters (i.e. on the distribution of eigenvalues of all filters) and this is closely related to the local geometry structures around lower major axes of their mapping of the unit sphere in its domain.



(a) Filters

(b) Inverse of filters

Figure 6.4: 3D Visualisation of entries of $w_{2,3}$ and its inverse post the sequence of S_i Surgeries S_i $(i = 1, \ldots, 11)$.

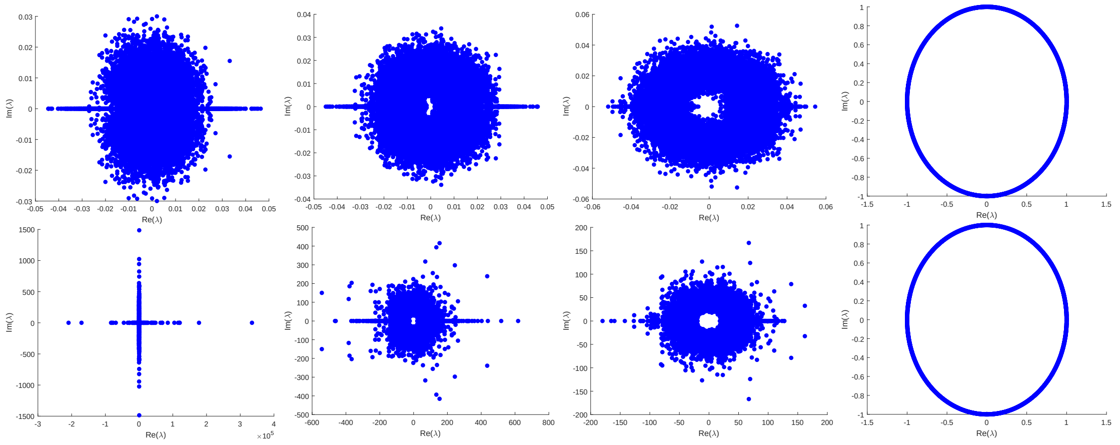
6.3.2 Impact of SVD-Surgery on eigenvalue distribution

The spread of eigenvalues of a matrix can give an indication of how well-conditioned or ill-conditioned the matrix is. In particular, a matrix with a large spread of eigenvalues (i.e., a large range of magnitudes) will have a high condition number, indicating that it is ill-conditioned and susceptible to numerical errors in computations. On the other hand, a matrix with a small spread of eigenvalues will have a low condition number, indicating that it is well-conditioned and less prone to such errors. For point clouds of RGF convolution filters, studying the spread of all the filter's eigenvalues can provide a valuable indicator of the stability/otherwise of the corresponding CNN models. As mentioned above, knowledge about the spread of eigenvalues provides a good guide in selecting an appropriate SVD-surgery level for CNN convolution filters. The fact that, counting multiplicities, $n \times n$ RGF's have n complex eigenvalues. Hence, visualising the set of eigenvalues as points in the complex plane provides a helpful tool in understanding the spread of eigenvalues of point clouds of filters [3].

Figure 6.5, illustrates the eigenvalue distribution in the complex plane of $N = 10^4$ randomly generated 3×3 matrices before and after matrix surgery. Eigenvalues in (a) are representing the original matrices A and A^{-1} before matrix surgery, and eigenvalues of $\tilde{A}_i$ and $\tilde{A}_i^{-1}$ for three different SVD-Surgery schemes: (b) replaces the smallest singular value with the second, i.e. $\tilde{\sigma}_3 = \sigma_2$; (c) the singular values σ_2 and σ_3 are replaced with σ_1, i.e. $\tilde{\sigma}_3 = \tilde{\sigma}_2 = \sigma_1$; and (d) the diagonal singular value matrix is replaced with identity matrix I_3. The last drastic surgery, which changes the eigenvalues is similar to the bounded singular value scheme in [117].

In the pre-surgery case, there is no hole around the origin, but as one applies more drastic SVD surgeries the whole around the origin for both the actual resulting matrices and their inverse point clouds starts to expand outward. An extended version of the above example of $N = 10^5$ randomly generated 5×5 matrices is presented in Figure 6.6. This illustrates the eigenvalue distribution before and after matrix surgery. Eigenvalues in (a) are representing the original matrices A and A^{-1} before matrix surgery, and eigenvalues of $\tilde{A}_i$ and $\tilde{A}_i^{-1}$ for five different SVD-Surgery schemes starting (b) by equating the last two eigenvalues; (c) by equating the last three eigenvalues; (d) by equating the last four eigenvalues; (e) by equating all the eigenvalues; and (f) by replacing the diagonal SVD factor with identity matrix I_5.

[3] For a deeper understanding of the study on the distribution of eigenvalues of large random matrices, interested readers are encouraged to explore [148–151].



(a) Pre-surgery A (b) Post-surgery $\tilde{A}_1$ (c) Post-surgery $\tilde{A}_2$ (d) Post-surgery $\tilde{A}_3$

Figure 6.5: Eigenvalue distribution of randomly generated 3×3 matrices before and after matrix surgery (top) and their correspondence inverses (bottom).

In general, the spread of eigenvalues in the complex plane is connected to matrix well- and ill-conditioning. The radii of the inner and outer circles, passing through the lowest and highest eigenvalues, respectively, provide insight into the sensitivity of the matrix to small changes and its conditioning. Recall, that during CNN models training/retraining at each layer, the convolution filters, are subject to changes dictated by the backpropagation procedure that attempts to fit the model to the training dataset.

The inner and outer radii of eigenvalues of orthogonal square matrices are equal to 1. As a result, orthogonal matrices have real eigenvalues with absolute values equal to 1. The inner radius of the eigenvalues of a square matrix is defined as the minimum distance from the origin to the eigenvalues of the matrix. For orthogonal matrices, all eigenvalues have absolute value 1, so the minimum distance from the origin to the eigenvalues is equal to 1, making the inner radius of the eigenvalues of orthogonal square matrices equal to 1. The inner radius of eigenvalues of well-conditioned non-orthogonal matrices is positive and close to 1, whereas it can be close to zero for large condition number values.

To determine the inner and outer circles of the well-condition filter after matrix surgery (condition number equal to 1 by keeping the first singular value only), for N number ($N = 10^4$) matrices with $n \times n$:

After matrix surgery to reduce all condition numbers to 1, the inner and outer radii of eigenvalues λ in the complex plane of $N \geq 2$ random Gaussian matrix samples are bounded and defined i.e. for any given matrix $D \in \mathbb{R}^{n\times n}$ the inner radius, r_{in}, is defined as:

$$r_{in} = \inf_{k \in N}\{D \in \mathbb{R}^{n \times n} : \|D_k\|_2\} \tag{6.5}$$

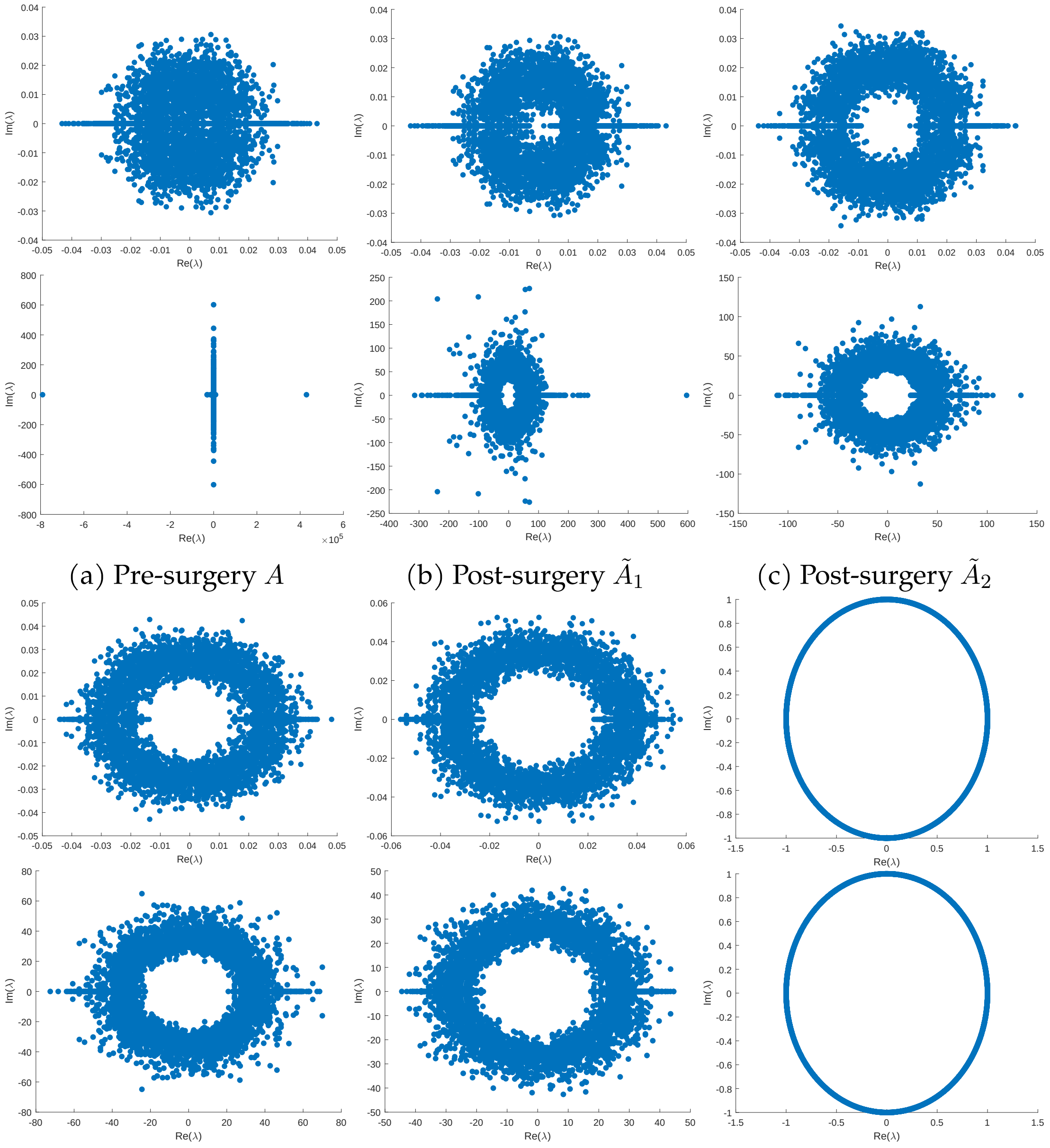
and the outer radius, r_{out}, is defined as:

$$r_{out} = \sup_{k \in N}\{D \in \mathbb{R}^{n \times n} : \|D_k\|_2\} \tag{6.6}$$

In both cases, the $\|D\|_2 = \sigma(D) = |\lambda(D)|$. The thickness of the ring is dependent on the spread of eigenvalues and singular value distribution. In fact, the width of the ring increases with larger sizes of D, and keeping the largest singular values reduces the thickness between the inner and outer circles i.e. $|r_{out} - r_{in}| \leq \varepsilon$. For instance, Figure 6.7 shows the radii of eigenvalue distribution in the complex plane before and after surgery with the significant difference between the largest singular value and eigenvalue, whereas the $\|D\|_2 = \sigma(D) = |\lambda(D)|$ when condition number is equal to 1. Moreover, it's also possible to control the thickness of the ring for non-orthogonal matrices by keeping the most top-relevant singular values for large matrices with keeping the condition number close to 1 or below 2, or simply dependent on the level of SVD-surgery. The radii of the eigenvalue distribution on the disc are defined as:

$$r_{in} = \inf_{k \in N}\{D \in \mathbb{R}^{n \times n} : |\lambda(D_k)|\} \tag{6.7}$$

$$r_{out} = \sup_{k \in N}\{D \in \mathbb{R}^{n \times n} : |\lambda(D_k)|\} \tag{6.8}$$



(a) Pre-surgery A

(b) Post-surgery $\tilde{A}_1$

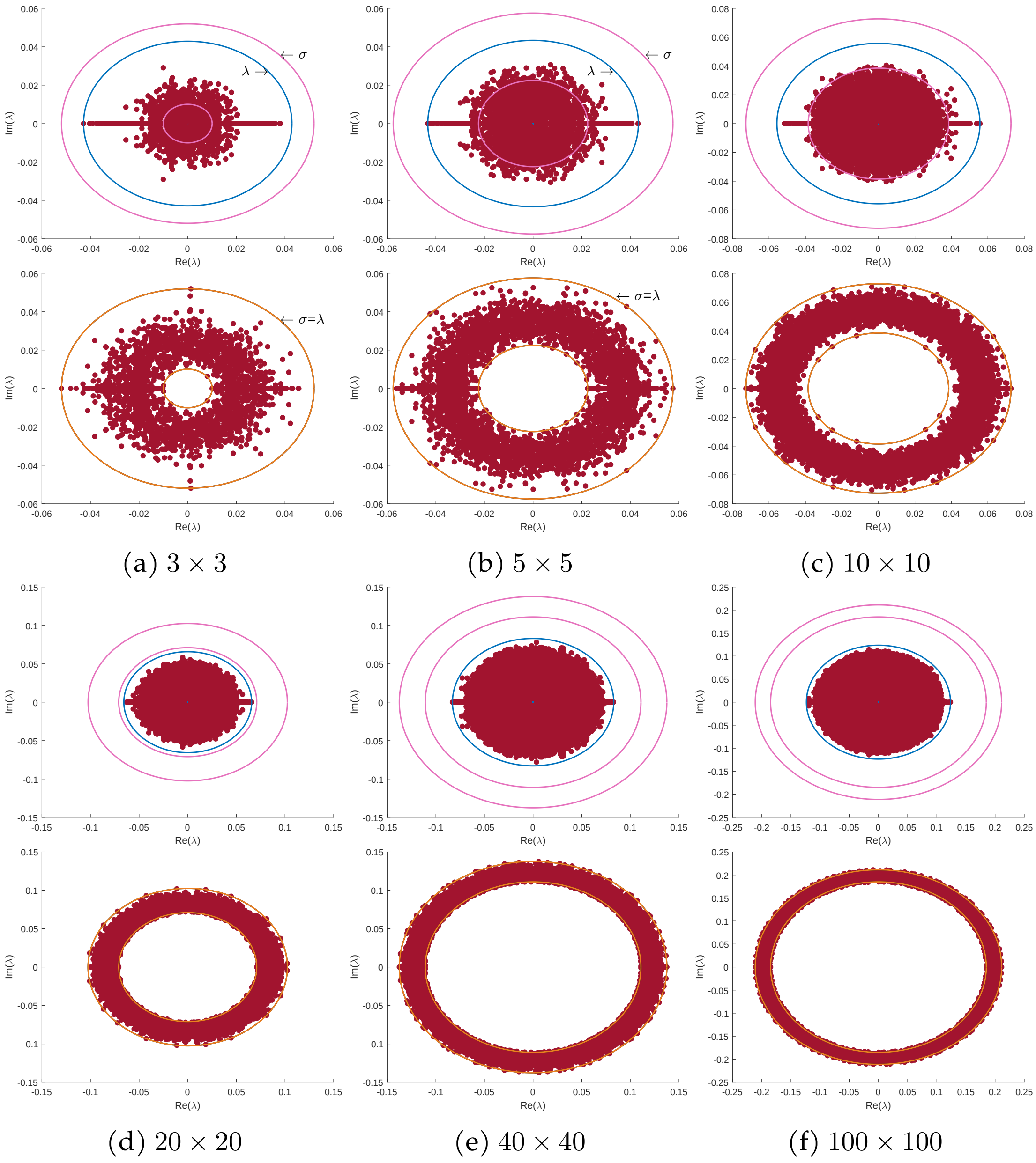
(c) Post-surgery $\tilde{A}_2$

(d) Post-surgery $\tilde{A}_3$

(e) Post-surgery $\tilde{A}_4$

(f) Post-surgery $\tilde{A}_5$

Figure 6.6: Eigenvalue distribution of randomly generated 5×5 matrices before and after matrix surgery (top) and their correspondence inverses (bottom).



(a) 3×3 (b) 5×5 (c) 10×10

(d) 20×20 (e) 40×40 (f) 100×100

Figure 6.7: Eigenvalue distribution of $N = 10^3$ randomly generated $n \times n$ matrices before and after matrix surgery.

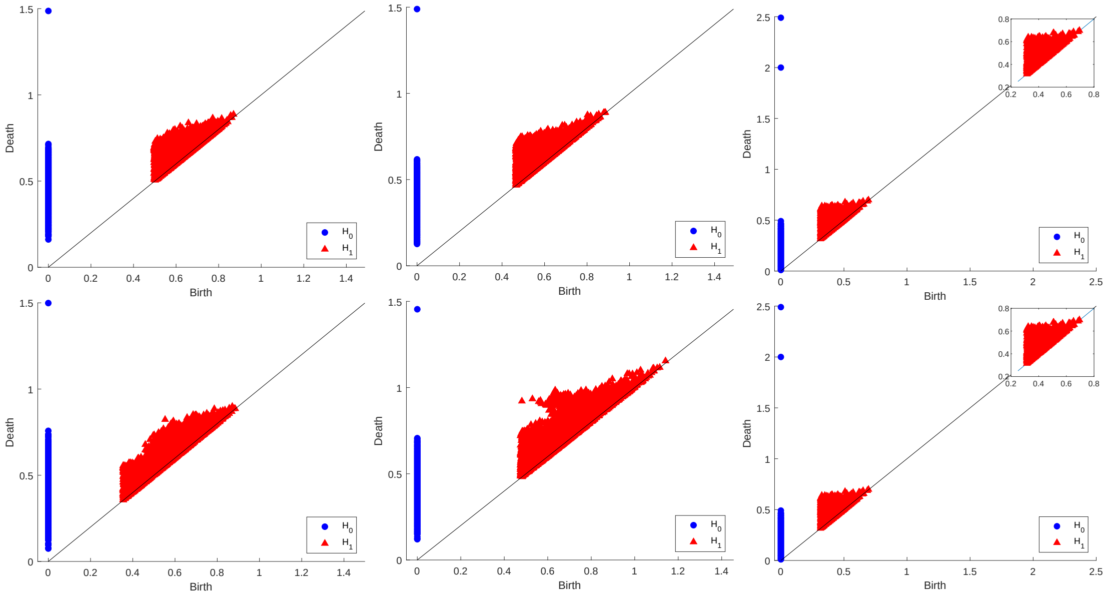
6.3.3 Impact of SVD-Surgery on PH of point clouds of filters

The interest in studying the impact on the topological profile of point clouds of filters post-surgery relates to the remarkable variations (established in the previous chapters) in the topological properties visualised through PDs between well-conditioned and ill-conditioned filters point clouds. We evaluate the level of SVD-Surgery of randomly generated matrices point clouds through the lens of persistent homology[4]. The PH of point clouds before and after surgery could provide insight into the geometrical and topological changes of their action on the unit sphere of their domain.

Investigating the PH changes of the point cloud of 3×3 randomly generated filters from the previous example, see Subsection 6.3.1, after entry normalisation results in a point cloud $\mathcal{A}$ on 8-dimensional sphere embedded in $\mathbb{R}^9$, and a similar process for the set of matrix inverses $\mathcal{A}^{-1}$. Figure 6.8 shows the persistence diagram in dimensions zero and one before and after applying matrix surgery for two simple linear combinations. The post-surgery point clouds $\tilde{\mathcal{A}}_1$ and $\tilde{\mathcal{A}}_2$ are representing a set of matrices $\tilde{A}_1$ and $\tilde{A}_2$ where the singular value $\tilde{\sigma_3} = \sigma_2$ and $\tilde{\sigma_3} = \tilde{\sigma_2} = \sigma_1$, respectively. The point clouds $\tilde{\mathcal{A}}_1$ (top) and $\tilde{\mathcal{A}}_1^{-1}$ (bottom) are composed of sets of matrices $\tilde{A}_1$ and $\tilde{A}_1^{-1}$, respectively, where replacing the smallest singular value σ_3 is resulting in a narrower range of condition numbers, shifting from $[1.2, 10256]$ to $[1.006, 17.14]$. The persistence diagrams are significantly changing compared to the pre-surgery $\mathcal{A}$ and $\mathcal{A}^{-1}$. The equivalence PDs of $\tilde{\mathcal{A}}_2$ (top) and $\tilde{\mathcal{A}}_2^{-1}$ (bottom) is a direct reflection of the particular SVD-Surgery that forces the orthogonality property of the matrices, which in this particular case, the inverse matrices are obtained simply by transposing the original ones.

In both dimensions, we observe contrasting PDs between well-conditioned and ill-conditioned matrices, as well as between the PDs of their respective inverses. However, when analysing the PDs of the original matrices and their inverse point clouds, we have observed minimal variation in the spatial distributions for well-conditioned matrices compared to the ill-conditioned ones in dimension 0. Our proposed matrix surgery will aim to control the differences between the PDs' of the output matrices and that of their inverse point clouds.

[4] See Subsection 2.2.3 for further detail.



(a) Pre-surgery $\mathcal{A}$ (b) Post-surgery $\tilde{\mathcal{A}}_1$ (c) Post-surgery $\tilde{\mathcal{A}}_2$

Figure 6.8: Persistence diagram of point clouds $\mathcal{A}$ (top) and $\mathcal{A}^{-1}$ (bottom) before and after SVD-Surgery.

6.4 Effects of SVD-Surgery on the CNN models

The various investigations in previous sections on the effects of matrix surgery have shown that the task of selecting appropriate SVD-Surgery by linear combination of singular values is somewhat daunting and is likely to be related to the dataset and the various parameters of the deployed CNN architecture. In this section, we shall investigate the impact of applying SVD-Surgery on convolution filters before training/pretraining CNN models and during the training process steps. The interesting impact is the ability to endow the resulting CNN schemes, for US tumour image analysis, with robustness and ability to generalise to unseen image data. But before conduction such experiments, we shall first attempt to answer the following question: "Does applying SVD-surgery at filter initialisation stage help stabilising and controlling filters condition numbers throughout the training procedure? This will be done in Subsection 6.4.1. In Subsection 6.4.3, we present the result of experiments, similar to those conducted in Chapter 5 with the various set of customised and state-of-the-art pretrained CNNs, but by applying a uniformly defined simple SVD-Surgery applied at their filter initialisation step. In the last subsection, we shall present the result of applying specially designed conditioning based SVD-Surgery that is applied both at the initialisation stage as well as at every intermediate epoch of the training stage. In both cases, the performance results will be compared with those obtained without any surgery in terms of robustness and generalisation ability.

6.4.1 Visualising effects of SVD-Surgery during training

We already established, in Chapters 4 and 5, the instabilities of filters' condition number during training our customised CNN models as well as retraining several pretrained of CNN models. Here we test the effect of applying simple SVD-Surgery, as a condition number regularisation, on the convolution filters during training our Model-B using the unified-channel default initialised filters, $\mathcal{W}_D$, while training the Ten-D US dataset. The aim of this experiment is to monitor the behaviour changes of the filters with and without integrating a simple level of SVD-Surgery whereby the last singular value is replaced with the one before the last. Figures 6.9, shows 3D illustration of the changes in norm of the first layer filters, norm of their inverses, and their condition number (in logarithmic scale) during training with no surgery over 15 epoch. Here, the graphs are labelled by the epochs number and they display the filters input into that epoch. The graph of Epoch 1 displays the initialised filters, and the graph of epoch (E_i), $i > 1$, displays the filters output from epoch E_{i-1}. Clearly, the training results in instability of the filters condition numbers. This is manifested by the observed fluctuation of condition numbers, that are initially in the high logarithmic range of 0-10 and though dropping into somewhat lower log range of 0-6 over the first 5 epochs only to fluctuate between these two ranges through the remaining epochs.

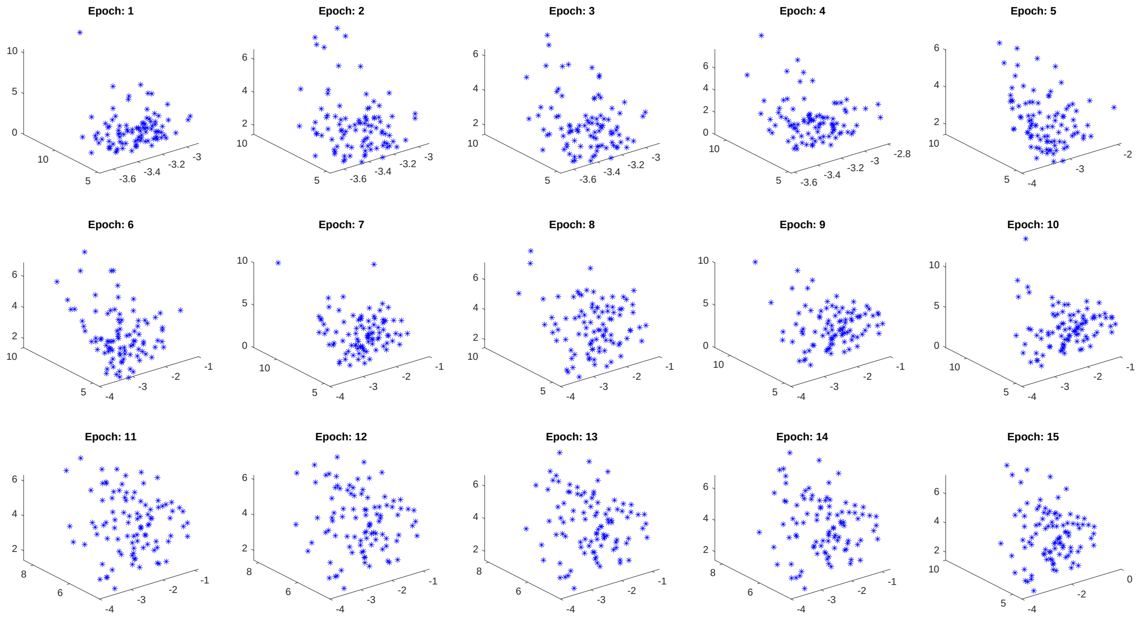


Figure 6.9: 3D graphs $(x, y, z) = (\|filter\|, \|filter^{-1})\|, \kappa(filter))$ for the 5×5 filters in $\mathcal{W}_D$.

Figure 6.10 showcases comparative graphs from a repeated training experiment, employing simple SVD-Surgery on the initialised filters in $\mathcal{W}_D$. In the graph labelled "Epoch1," blue dots represent the initialised filters, while red dots depict the same filters after applying the chosen SVD surgery. For subsequent epochs (E_i, $i > 1$), blue dots represent filters output from epoch E_{i-1}, and red dots signify the result of applying the selected surgery on these filters. Notably, filters at epoch 15 remain unaffected by surgery.

The figure illustrates a significant reduction in condition numbers during initialisation. While the condition numbers exhibit fluctuations within smaller logarithmic ranges for the first 7 epochs, they stabilise within the lower log range of 0-1.5 for the subsequent epochs. This is in contrast to the higher range of condition numbers (in log) observed in Figure 6.9. Towards the end of training, many output filters are reasonably well-conditioned within the log range of 0-0.5, while others are less well-conditioned. The observed effects are clearly linked to the chosen level of surgery, with fluctuations influenced by differences in data batches within the training set. A parallel behaviour is noted in the second convolutional layer set, both with and without SVD-Surgery, as depicted in Figures A.8 and A.9.

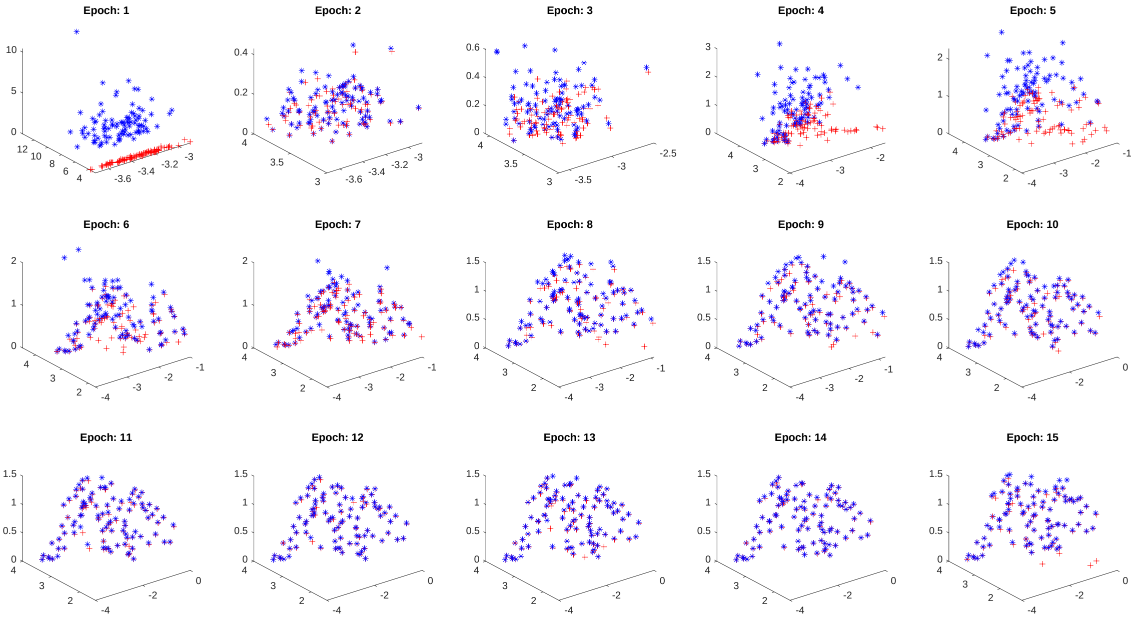


Figure 6.10: 3D illustration of the norm (x-axis), norm of inverse (y-axis) and condition number (z-axis) of a set of 5×5 weights per epoch (log) before and after SVD-Surgery.

6.4.2 SVD-surgery at initialisation or on pretrained filters

In this section, we present the results experiments, similar to those conducted in Chapter 5, with several customised and pretrained CNNs, but instead we apply a moderate uniformly defined SVD-surgery prior to training. This uniformity of the surgery here means that for different size filters the modified singular values are of the same proportion to the size. Here, we keep the largest two singular values and replace the rest of the singular values with the second largest singular value. Unless all singular values of a filter are equally significant, this surgery reduces its condition numbers relatively well but can only make it orthogonal if the first two singular values are equally significant.

Figure 6.11 shows the classification accuracy of the trained models from Chapters 4 and 5 with the modified trained CNNs after applying the above SVD-Surgery before starting the training process. The customised CNN Model-B is trained from scratch while for the pretrained CNNs are in the transfer learning mode after applying surgery on all the pretrained filter. Throughout this section, the displayed results gives the classification performance of the models with-/without the SVD-Surgery.

Figure 6.11, shows the overall all accuracy of three customised CNN Model-B variants with different filter initialisation schemes trained on the Ten-D breast US dataset, along with three pretrained CNN models retrained in transfer learning mode on the same dataset. Minimal differences are observed in their accuracy before and after surgery. There is only slight improvement resulting from surgery for Mode-B $\mathcal{W}_D$ and $\mathcal{F}_D$ as well as the ResNet18 Model, while all other schemes experience marginal declines due to the application of surgery.

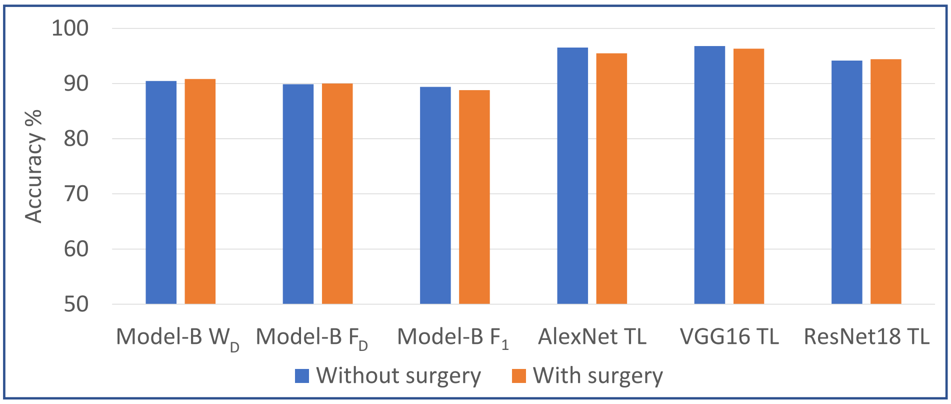


Figure 6.11: CNN classification accuracy before and after the uniform SVD-Surgery on initialised filters.

6.4.2.1 Robustness tests

To evaluate the robustness of the above CNN models, we apply the same noise levels and types as described in Subsection 4.2.2. Recalling the test results in Figures 4.2 and 5.12, we present those robustness results and compare the robustness performance when the newly trained/retrained CNNs after applying the SVD-Surgery (S) in Figure 6.12.

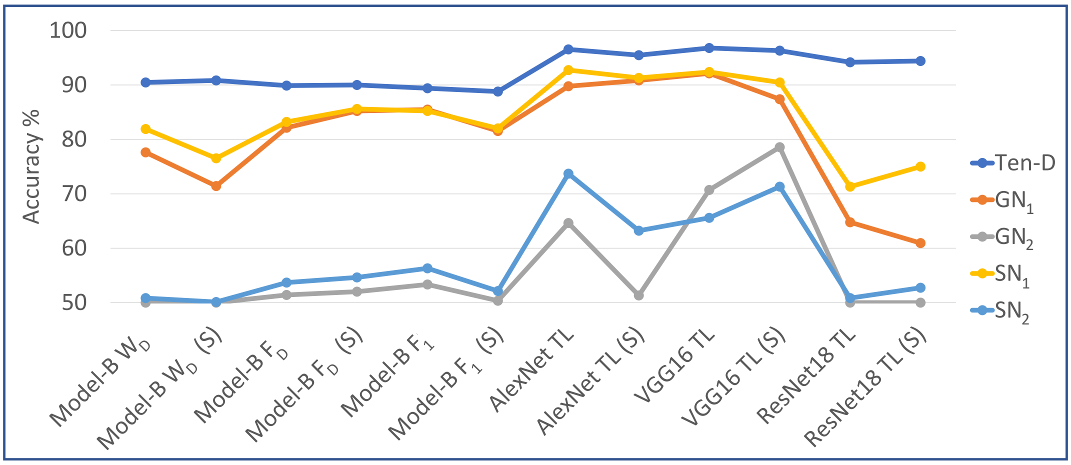


Figure 6.12: CNN models against Gaussian and Speckle noises before and after applying SVD-Surgery.

Generally, all models have disappointing robustness performance against the higher level of noise $(GN_2$ and $SN_2)$. And in that case it is checking the effect of surgery is worthless. However, with the exception of ResNet18 and Model-B Dw with surgery, all models have reasonable robustness tolerating the lower level of noise $(GN_1$ and $SN_1)$. And in that case, Model-B with $\mathcal{F}_D$ benefits from surgery by becoming reasonably more robust against GN_1 and SN_1 and AlexNet become marginally more robust against GN_1 only. The robustness of all other models deteriorate with variable rates as a result of SVD surgery.

6.4.2.2 Generalisation tests

To check the generalisability performance of the newly trained CNN models on Ten-D US datasets after reducing filters' condition number at initialisation and pretrained filters, we test the models on the BUSI US dataset[5]. Recalling the test results in Figures 4.3 and 5.13, we present all generalisation test results in Figure 6.13 for both US datasets and with/out the SVD-Surgery. The accuracy level of Model-B $\mathcal{F}_D$, VGG16 TL, and ResNet18 TL models on the BUSI dataset have all

[5] See Subsection 4.2.3 for further detail.

benefited from filter surgery by increased performance rate in the range 1.5-5% compared to no surgery. Model-B $\mathcal{F}_1$ and AlexNet performance on BUSI only benefited from surgery by a negligible amount of 0.31%.

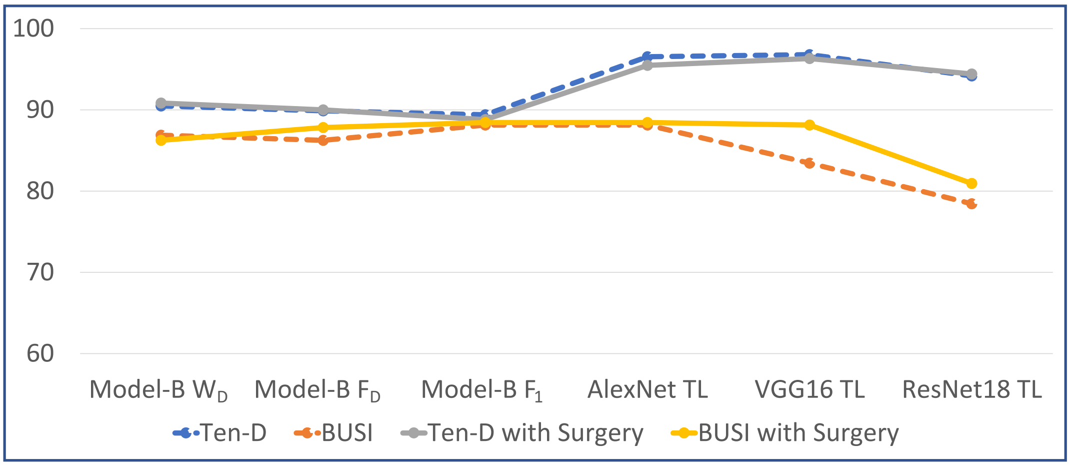


Figure 6.13: CNN performance on the unseen BUSI datasets pre- and post-surgery.

The experiments conducted in this section confirms that there is no one unified approach for SVD-Surgery across all CNN models. As mentioned in Subsection 6.4.1, applying surgery solely during the initialisation step does not effectively stabilise the condition numbers. The fluctuation in condition numbers is caused by the trainable parameters of the CNN architecture and variations between different batches of the training dataset.

Furthermore, considering the correlation between the distribution of condition numbers of filter point clouds and the spread of their singular values (as discussed in Subsections 6.3.1 and 6.3.2), it is reasonable to expect that a specific SVD-surgery may not uniformly enhance the conditioning of all filters. In light of these considerations, we conclude this chapter by testing a variable SVD-surgery scheme, where the level of surgery is determined based on the conditioning of the input filters. Furthermore, this method can be considered as a hyper-parameter for training CNN models.

Overall, the experiments in this section confirms that no one kind of SVD-Surgery can benefit all CNN models. In Subsection 6.4.1, above, we noted that applying surgery at the initialisation step only does not stabilise the condition numbers. Condition number fluctuation could arise as a result training parameters of the CNN architecture as well as the variation between the different batches of training dataset. Moreover, considering the link between the distribution of condition numbers of filter point clouds and the spread of their singular values (as discussed above in Subsections 6.3.1 and 6.3.2) we should also expect that one specific SVD-surgery can improve the conditioning of all filters in a similar way. Accordingly, we end this chapter by testing a variable SVD-surgery scheme whose level is based on the input filter conditioning.

6.4.3 Variable conditioning based SVD-Surgery

There are several major criteria that influence the performance of the CNN model with filter surgery. These include the imaging modality, the nature of the problem and the CNN architecture, parameter and hyper-parameter selection, the distribution of the eigenvalues and singular values, and the spatial distribution of the filter point clouds. Here, we shall propose a special variable SVD surgery whose level is defined in terms of the input filter condition number.

To test the viability of this variable SVD-surgery strategy, in relation to robustness and generalisability, we shall experiment on our Model-B with $\mathcal{F}_1$ initialised filters. We define the variable SVD-Surgery to be used on the filters of $\mathcal{F}_1$ before training and on the convolutional layer filters during training at each epoch. Realising the difficulty of determining a linear combination of a certain set of lower singular values to replace them all, we shall attempt to invoke the idea of changing their spread so that each is replaced by a proportion of the one above it. Accordingly, we define our special variable SVD-surgery as follows:

- At the initialisation stage, apply the simple surgery by modifying $\tilde{\sigma}_5 = \sigma_4$.
- During the training at each epoch, for any filter A use the surgery defined as follows:
  if $\kappa(A) \geq 10$ then:

  $$\tilde{\sigma}_3 = 0.5\sigma_2 \qquad \tilde{\sigma}_4 = 0.3\sigma_2 \qquad \tilde{\sigma}_5 = 0.2\sigma_2$$

  if $3 \leq \kappa(A) < 10$ then:

  $$\tilde{\sigma}_4 = 0.6\sigma_3 \qquad \tilde{\sigma}_5 = 0.4\sigma_3$$

  if $1.5 \leq \kappa(A) < 3$ then: $\tilde{\sigma}_5 = 0.95\sigma_4$
- Otherwise, do not apply surgery.

This type of surgery is less dependent of the singular value distributions. The various constant parameters are not meant to rigid but can be modified to suit the purpose of the surgery. The learning process may become slower if a large number of filters fall within the moderate surgery procedure when $\kappa(A) \geq 10$. However, the above procedure does not guarantee that the upper bound of the condition number as the ratio of σ_1 and σ_2 could be large.

Similar to the previous CNN model performances, we trained and tested the model with Ten-D breast US dataset and tested on BUSI US dataset for the generalisation and when different levels of noise are added to the Ten-D datasets. Table 6.3, below, summarises the classification performance of the Model-B $\mathcal{F}_1$ with the above described surgery scheme, for two distinct training batch sizes 50 and 80.

Table 6.3: Model-B classification performance with the variable SVD-Surgery.

Batch size	Testing set	Accuracy	Sensitivity	Specificity
50	TenD	87.26	85.48	89.05
	GN_1	87.74	78.81	96.67
	GN_2	87.50	85.48	89.52
	SN_1	88.33	80.24	96.43
	SN_2	87.38	88.33	86.43
	BUSI	87.62	85.71	89.52
80	TenD	87.62	81.43	93.81
	GN_1	87.86	84.05	91.67
	GN_2	83.10	93.33	72.86
	SN_1	88.33	84.76	91.90
	SN_2	82.62	93.81	71.43
	BUSI	83.10	93.57	72.62

These results demonstrate the success of using variable SVD-surgery, defined differently for each filter according to their condition numbers, throughout the entire training procedure. The developed model is optimally robust and generalisable to unseen data. In fact, it levelled up all different testing scenarios especially when the batch size is 50 i.e. moderate US image batch could help to tailor a better CNN with filter surgery that are robust to different noise levels and generalise well to unseen data. With batch size 50, the presence of low level noise (GN_1 and SN_1) there is a considerable gap between sensitivity and specificity rates which are reduced significantly when the level of noise increased to (GN_2 and SN_2). Interestingly, this pattern of gaps between sensitivity and specificity is reversed when training was conducted with batch size 80. This indicate that these gaps can be controlled by tweaking the training batch size.

The results from this and last section supports the idea of using variable SVD-surgery throughout training, the parameters are to be determined for each CNN model differently. The results also suggest that achieving adequate surgery parameters may be done in multiple steps of training ad retraining with different batch sizes.

6.5 Summary and conclusion

In this chapter, we proposed a convolution filters condition numbers regularisation scheme and adequately exploits the well-known link between the SVD of matrices and their mapping of the unit sphere in their domain when considered as linear transformations. The matrix SVD-Surgery concept is aimed at reducing and controlling the condition number of convolution filters through the training of CNN models. The SVD-Surgery applied on highly ill-conditioned matrices results in a reconstructed matrix of significantly lower condition number. We evaluated the various effects of the proposed strategy on different filter point clouds. We discovered that this approach effectively reduces the condition number, depending on the chosen parameters for the linear combination i.e. the level of the surgery. We evaluated its impact on large point clouds through the perspectives of persistent homology, distribution of condition numbers, and eigenvalues.

Addressing the initial motivation of the condition number regularisation, the SVD-Surgery approach is equally applicable to both training from scratch and transfer learning (fine-tuning) scenarios at initialisation and/or during the training process, without introducing additional complexity to the model or requiring the adjustment of extra parameters.

Although, the task of determining the adequate SVD-surgery parameters is somewhat daunting, we have demonstrated that it is possible to do and reap the benefits of developing customised or pretrained CNN models for diagnostic analysis of US tumour scan images that are robust against perturbation of data and conquer the overfitting problem by enabling generalisation to unseen data. However, it is important to consider that drastic changes in the filters particularly towards the final epochs of the training may lead to inconsistent learning or gradual de-learning of texture features. Therefore, the level of surgery across channels, convolutional layers, and per batches/epochs may influence various model performance criteria.

Chapter 7

Conclusion and Future Directions

This PhD research project arose within a wider collaborative initiative aimed to leverage the extraordinary successes of deep learning technologies in computer vision tasks and to develop CNN models for tumour diagnostic tasks using ultrasound tissue scan images. Established knowledge, at the time, highlighted several serious challenges in developing such models from scratch for any computer vision task without having access to a sufficiently large dataset of reliably class labelled image samples that are diverse in reflecting the wider population. The common suggested approach for small datasets was to select an existing CNN model that has been pretrained with huge samples of natural images, and retrain with US images for the purpose of efficiently tweaking the pretrained parameters (i.e. the sets of convolution filters and the FCL weight matrices) to learn discriminating US tumour image feature patterns.

The advice to use pretrained CNN models in transfer learning mode, comes with a number of warnings the most important ones are (1) to expect sub-optimal performance, (2) the model suffers from overfitting and may not generalise to unseen data, and (3) may not be robust against data perturbation. Sub-optimal performance may be acceptable as long its decisions are as good as those made by expert clinicians with reasonably long training. But for critical applications, like medical diagnostics, overfitting is a serious disadvantage that must be avoided. Robustness against the effect of tolerable level of perturbation by modest noise must be guaranteed for the model to be accepted for deployment. Accordingly, the aim and objectives of this thesis was set to (1) investigate CNN factors that can be associated with the causation of overfitting and/or lack of robustness, and (2) to use the gained knowledge to develop a strategy for designing CNN models whose factors/parameters/procedures help mitigating these shortcomings.

7.1 Conclusions

Post the retraining of a pretrained CNN model, the resulting convolution filters determine the feature maps that contain the learnt US images hidden feature patterns that discriminate between tumour classes. Therefore, our initial investigations focused on understanding the way commonly deployed convolution filters impact content and structure in terms of their algebraic and topological profiles. Such investigations are necessitated by the potential differences between natural images and US images in terms of convolutions on image content and texture descriptors including image entropy and distribution of texture landmarks.

These initial investigations were conducted on US images of different tumour tissue types using sets of random Gaussian filters of mixed conditioning. No easy to analyse patterns of impact were noted of image entropy variation in the convolved US images. However, prior to testing impact of convolution filters on distribution of texture landmarks in US images, our statistical analysis of the LBP texture landmark groups revealed significant differences between texture contents in US images and those in natural face images which was later noted for other natural image datasets. In fact, this was conformed when we investigated the statistical analysis of bladder, liver, and breast ultrasound scans as well as the benchmark datasets such as Digits, MNIST, and CIFAR-10.

This observation, that could be relevant to other modalities of medical images, serves as a general warning against high expectation from retraining pretrained CNN models of natural image analysis in terms of performance when deployed for US image analysis. It motivated the introduction of ***texture-based entropy*** (TE(LBP)) to quantify image texture feature content. It complements the empirical evidence, provided by Nicholas Konz et al. in [152], for differences between natural and radiological images, in terms of their intrinsic dimensions, and the associated difficulty of learning form radiology images.

Through these investigations, we highlighted a strong correlation between the condition numbers of convolution filters and their influence on traditional image entropy and TE(LBP) descriptors as a result of image resizing or noise presence. Furthermore, we noted substantial differences in the spatial distribution of texture landmarks within convolved US images using differently conditioned convolution filters, in the presence or absence of noise. These observations emphasise the necessity of conducting in depth research on the link between robustness against noise and the conditioning of convolution kernels as the main algebraic measure. Additionally, these observations also motivated the use of texture landmark based persistent homology to extract representation of spatial distribution of texture landmarks.

In Chapter 4, we set out to determine the level of success/failure of retraining several state-of-the-art CNN modes with US images in terms of robustness against data perturbation and ability to generalise to unseen data. These experiments established that while these schemes achieve high accuracy on the US dataset, they suffer considerably from lack of robustness (against noise perturbations) and generalisation into unseen data. Several investigations were conducted to have deeper understanding of the causes of these shortcomings by studying the effects of pretrained convolutional layers, on the conventional and texture-based entropy descriptors of the convolved images and feature maps, in terms of conditioning of various filter sets. We first noted that almost all pretrained filters were ill-conditioned with many being highly ill-conditioned. Moreover, the retraining on the US Images, made the final model filters even more ill-conditioned with few exceptions.

We also formally deployed the LBP landmark-based PH, to further understand the spatial distribution of texture landmarks in the feature maps across all convolutional layers. Our experimental work demonstrated that the incorporation of convolution filters helps increase the discriminative power of this PH as well as enable the CNN model to better distinguish between benign and malignant masses. The performance improvement of tumour classification, using the LBP landmark-based PH, was mostly noted in the first two layers of AlexNet. We also demonstrated that pruning 50% of the filters, to retain the better conditioned filters, maintained the pattern of accuracy of LBP landmark-based PH schemes at different layers. These results motivated the next step of our research into designing slim CNN architectures customised for ultrasound images to train from scratch.

Our strategy for constructing well performing slim US customised CNN architectures was designed to exploit the algebraic properties of filters and the spatial distribution of US image LBP landmarks. In order to ensure robustness against adversarial noise perturbation and ability to generalise well to unseen data, we deemed it necessary to control the condition numbers of the convolution filters as much as possible without jeopardising the learning rates. It is worth noting that, although we were not aware at the time with the work of Nicholas Konz et al. [152], our strategy is compatible with their advice to develop Deep learning architectures specifically tailored to the characteristics of radiological images.

The different proposed architectures are based on controlling the condition numbers of convolution filters both at initialisation and throughout training if necessary. The latter requirements were influenced by the instability of convolution filters' conditioning observed during training. Several customised CNN models were designed and tested in terms of the set objectives, insisting on very slim architectures using at most two convolution layers. The tested schemes adopted different filter initialisation including a default model as well as schemes of selecting the lowest set of well-conditioned RGF out of different pool sizes. Furthermore, to reduced the number of generated filters, we adopted the "unified channels" scheme whereby every filter of the multi-channel tensor is a scaled version of one chosen well-conditioned single filter. The experimental results were satisfactory, especially for the customised CNN models that involve better conditioned filters, in terms of overall accuracy, robustness to adversarial noise perturbation, and generalisation to unseen data. Except for the default model, most filter initialisation schemes maintained reasonable conditioning and many filters were stable.

The notable improved performance of the customised CNN models over the pretrained CNN models in terms of robustness against natural perturbations and generalisability to unseen data, demonstrated that stability of filters condition numbers is an important factor. Indeed, the fact that Batch Normalisation worsens overfitting and did not stabilise filters condition numbers confirms this conclusion

Despite these satisfactory success, the experimental results revealed a persistent instability in filters during training, albeit less severely when using well-conditioned filters selected from various RGF pools. This provided strong motive to search for algebraic schemes to stabilise filter condition numbers during training. Instead of imposing strict criteria for tightly controlling the convolution filters condition numbers such as orthogonality or Lipschitz bounds, in Chapter 6, we introduced SVD-Surgery as a condition number regularisation scheme. The matrix SVD-Surgery method leverages the established link between the SVD of matrices and their mapping of the unit sphere in their domain when considered as linear transformations. The aim is to reduce and control the filters condition number at each training epoch of CNN models.

The SVD-Surgery technique effectively reconstructs highly ill-conditioned matrices into considerably lower conditioned filters by surgically reshaping the singular values, spreading them closer to the largest singular value. This method may preserves the filters' norm depending on the surgery level while reducing the norm of their inverses. We extensively evaluated its impact on large point clouds through the lens of persistent homology, examining factors like distribution of condition numbers and eigenvalues.

Applying SVD-Surgery at initialisation and during the training process, does not introduce additional model complexity, and it is equally usable for retraining pretrained CNN models. Determining the adequate SVD-surgery parameters may require careful setting, but our experiments demonstrated that it is possible to do and reap the benefits of developing customised /pretrained CNN models for diagnostic analysis of US tumour scan images that are robust against perturbation of data and conquer the overfitting problem by enabling generalisation to unseen data.

In the various chapters, we computed persistent homology for the analysis of spatial distributions of different ingredients involved in our investigations for different purposes. These included the texture landmarks based US image based classifications before and at different CNN convolution layers, the spatial distributions of point clouds of convolution filters of different types (together with those of their inverses) at various training stages. All these investigations revealed strong links between the topological profiles of these ingredients and the performance of the corresponding CNN models. Indeed, SVD-Surgery emerged as a powerful technique that enabled the regularisation of highly ill-conditioned convolution filters, effectively mitigating potential near singularities during back-propagation in training.

We close this section by listing the most important conclusions and observations that are implicitly discernible from the above detailed statements.

- The impact of the convolution filter's conditioning and their stability during training on robustness and generalisation to unseen data sets is significant. This is demonstrated well by the performance of the customised CNN models that are driven by the characteristics of the images, even when relatively modest SVD-Surgery was implemented during training.

- When a model overfits during any training epoch, (i.e. the training and validation loss divert), the overall condition number of filters get extremely higher than in previous epochs. This is an indicator of the variation between the different training batches, and using SVD-Surgery reduces the overfitting of the model to those batches. Note that Batch Normalisation does not help in preventing overfitting.

- The combined operations applied on input data within the convolutional layers are enhancing the discrimination power of different classes in terms of their texture feature and PH characteristics. In many cases, this improvement in discriminating benign from malignant tumours are due to the complete wipe out of textures in the benign images by the convolution filters as compared to images of malignant images. This may be explained by the medical fact that cancer cells spread out fact and create a highly dense cellular connections.

- The spatial distribution of cancerous tissues in liver and bladder ultrasound scans exhibits distinctive features when compared to breast ultrasound scans, allowing for effective differentiation between malignant and benign classes. These differences primarily stem from variations in tissue sizes, organ tissue deformation, cancer stage, and the constraints imposed by limited sample sizes representing the overall population.

- The integration of TDA and DL can enhance the analysis of US images that differ in their textural complexity from natural images, by providing a more comprehensive understanding of the underlying structure in images feature maps. The link between these two fields lies in their complementary strengths. TDA can provide insights into the geometric structure of high-dimensional data, which can be difficult to understand using traditional statistical methods, while deep learning can learn meaningful representations of the data.

- Uncovering the connection between matrix condition numbers and the persistent homology of matrix point clouds revealed intriguing insights. Initially, we established that the topological characteristics of CNN filter point clouds are intricately linked to the algebraic properties of individual filters. Subsequently, we utilised PH to evaluate the extent of SVD-Surgery and its consequential effects on the topological profiles of point clouds and their inverses. This approach allowed us to unravel the intricate interplay between algebraic stability, induced by SVD-Surgery, and the resulting topological features within the context of CNN filter representations.

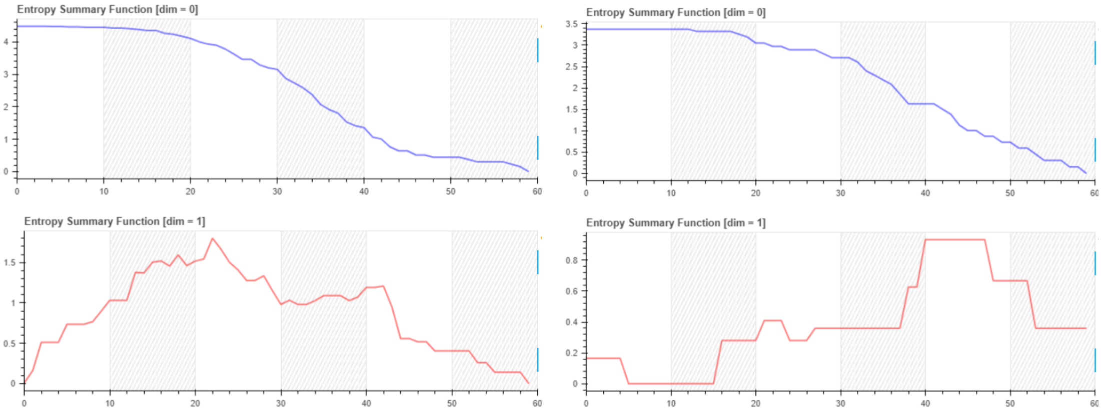
7.2 Future research directions and challenges

The nature of this research project and the topics covered are very broad involving concepts of linear algebra, topology and geometry besides Deep learning technologies. The wealth of knowledge established over centuries in these diverse fields of Mathematics and the many of the issues/challenges encountered in this thesis are likely to have been investigated for other purposes. This observation opens the door to many potential future works as well as challenges that could impact the future of AI in healthcare system. Here, we shall describe a few of these problems below:

1. **The full PH profile of point clouds of filters.**
   For practical reasons related to matrix inversion, our investigations of the topological profiles of point clouds of well/ill-conditioned filters were mostly confined to homology invariants of dimensions 0 and 1. Since convolution filters are point in $\mathbb{R}^{k\times k}$, with $k \geq 3$, then there is an obvious need to extend our PH investigation to the topological profiles to H_2 and beyond. Our early investigation revealed that the computation of H_2 for a relatively small point cloud of random matrices using javaPlex [153] is possible, especially when the matrices are well-conditioned or when both types are combined. However, the javaPlex solution is not feasible in the case of point clouds of the inverse filters for large point clouds of ill-conditioned filters. The proposed SVD-Surgery provides a potential remedy for this situation. In fact, we conducted a pilot study on the topological profiles of point clouds of 3×3 (or 5×5) matrices post SVD-surgery, we were able to compute the H_2 invariants for both the original and inverse point clouds, see Figures A.10 and A.11 We note that the (dis)appearance of holes in H_2 of the inverses of ill-conditioned matrices is dependent on the surgery level that change inverses norms.

2. **Effect of SVD-Surgery on PD Vectorisation of Convolution filters.**
   Current investigations on the statistical interpretation and feature vectorisation of the topological profiles are based solely on the convolution filters point clouds without consideration of the point clouds of their inverses that reflect their conditioning status. For instance, the persistent entropy-based PD vectorisation approach, [154], applied to filter point clouds can easily highlight the differences, if any, before and after SVD-Surgery. Figure 7.1 illustrates a significant difference and impact of improved condition of the

filters[1]. The bottleneck distances can be used to evaluate these differences for each weight initialisation technique as well as for filter point clouds instability during the training process. Ultimately, this could provide a potential stopping criteria when no changes in PD and/or condition number instability are detected during training instead of focusing on the overall model performance.



(a) Before SVD-Surgery

(b) After SVD-Surgery

Figure 7.1: Persistent entropy before and after applying SVD-Surgery on filters point clouds.

3. **Geometric properties of point cloud of US image patches.**
   Inspired by existing work in the literature on the geometry and topology of the space of high-density patches in natural images, (e.g. [86,156]), our research efforts on deployment of DL models of US tumour scan images is expected to benefit from extending the existing work into point clouds of US image patches. However, the feasibility of this investigation is a challenge due to the fact that natural image patches investigations used a set of 8×10^6 or more for mapping the high contrast patches onto the Kline Bottle. This challenge can be overcome, in different ways including the extraction of large number of US image patches from images of a variety tissues without restricting the selection to patches from tumour RoIs. Moreover, the abundances of certain groups of Uniform Local Binary Patterns (ULBP) within US images can provide an analogous concept to that of "high contrast patches" used in the natural image work in terms of their topological structure. Our preliminary empirical investigations show good synergy between both approaches. Besides providing interesting insight into deep learning

[1] The PH computation of the PH entropy-based approach is implemented with [155]

approach to US image analysis, such investigations helps checking the viability of developing special US-related convolution filters analogously to that of the Klein Filters recently proposed in [97]. Furthermore, the outcome from this study help shed a light on the other important challenge of CNN decision interpretation. This work can be extended to other radiological image analysis.

4. **SVD-Surgery for Tensor-based convolution filters.**
   In our unified channel conditioning problem[2] avoids the use of different condition number filters in different channels. However, there's a need to investigate and develop SVD-Surgery types for multi-channel convolution filter channels without imposing such strict condition to observe their topological behaviour. Our unified-channel convolution filters condition was imposed only at the initialisation step, but the training does not maintain this property throughout the successive epochs. Attempting to impose this condition over the different epochs is not only cumbersome but may slow down or impede high learning rates of CNN models. The proposed investigation could benefit from the existing work regarding condition numbers for tensor rank decomposition problems in [157, 158]. This also provide a chance to link our intended SVD surgery to the work on Lipschitz condition of neural networks as discussed in [141–144]. In [144], Christina Runkel et al., introduce an efficient method for spectral normalisation of depth wise separable convolutions. These investigations should be extended to cover the stability of the fully connected layer weights during the training process. For this we can exploit the Singular Value Representation (SVR) method, [159], to our customised models.

5. Does US tumour image datasets satisfy the "**Manifold Hypothesis**"?
   The manifold hypothesis formulates a long held mathematically sound view that data of high dimension have much lower intrinsic dimension bounded by the degrees of freedom in its coordinate, i.e. the records are samples of points on/near a manifold of much lower dimension, [46]. However, many recently encountered datasets emerging form genuine data analysis tasks that does not align with this assertion, whose elements are sampled from non-manifold, or manifolds with few exceptional singularities, [160, 161]. This proposed project is motivated by, and extends, the work of Nicholas Konz et al., [152], regarding the relationship between intrinsic dimension-

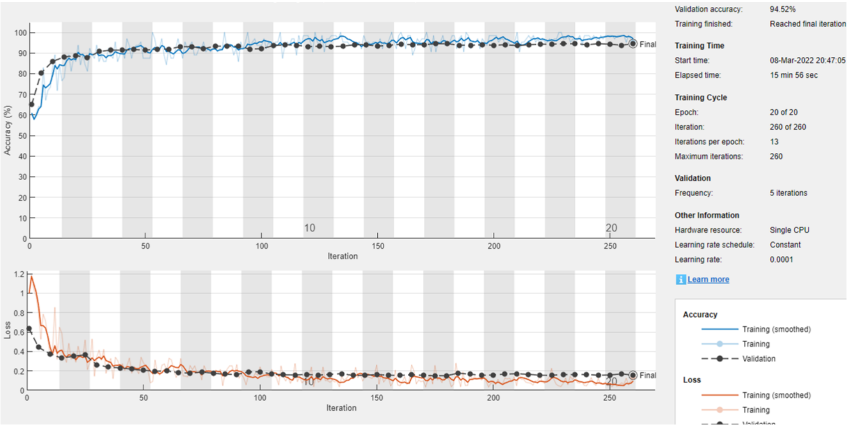
[2] See Subsection 5.4.1 for further detail.

ality and generalisation ability of CNN models. Although, US images are not included in their study, they establish that not only the intrinsic dimensions of radiological images are much lower than natural images, but the relationship between generalisation ability and intrinsic dimensionality is much stronger for radiological images due to the difficulty of learning intrinsic features in radiological images. This challenge is somewhat difficult due to the fact that US datasets are not only small but do not reflect the diversity of the rather unknown population. However, it would be useful to turn the question around to replace such US dataset with the corresponding significantly larger dataset of their feature maps post the convolution layers of customised/pretrained CNN models. The answer to this question is expected to shed light on the sensitivity of the corresponding FCL decisions in general, but is also relevant to adversarial attacks. Also, the answer is expected to depend on the algebraic and topological properties of the convolution filters. The dependency on these properties of filters can be used as a mechanism of protecting the medical CNN models against adversarial tampering with the filters, as well as trustworthiness of AI in the clinical settings.

Appendix A

Additional Information and Results

A.1 Chapter 4



Validation accuracy: 94.52%

Training finished: Reached final iteration

Training Time

Start time: 08-Mar-2022 20:47:05

Elapsed time: 15 min 56 sec

Training Cycle

Epoch: 20 of 20

Iteration: 260 of 260

Iterations per epoch: 13

Maximum iterations: 260

Validation

Frequency: 5 iterations

Other Information

Hardware resource: Single CPU

Learning rate schedule: Constant

Learning rate: 0.0001

Figure A.1: Performance of AlexNet trained on breast ultrasound dataset.

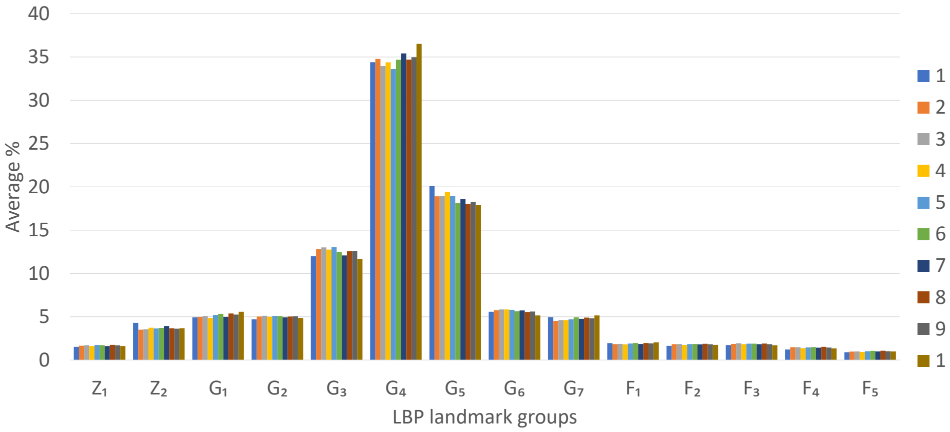


Figure A.2: LBP descriptor statistics of SVHN dataset.

Table A.1: Classification accuracy, sensitivity, and specificity of PH based ULBP landmarks in dimensions zero and one of bladder US dataset.

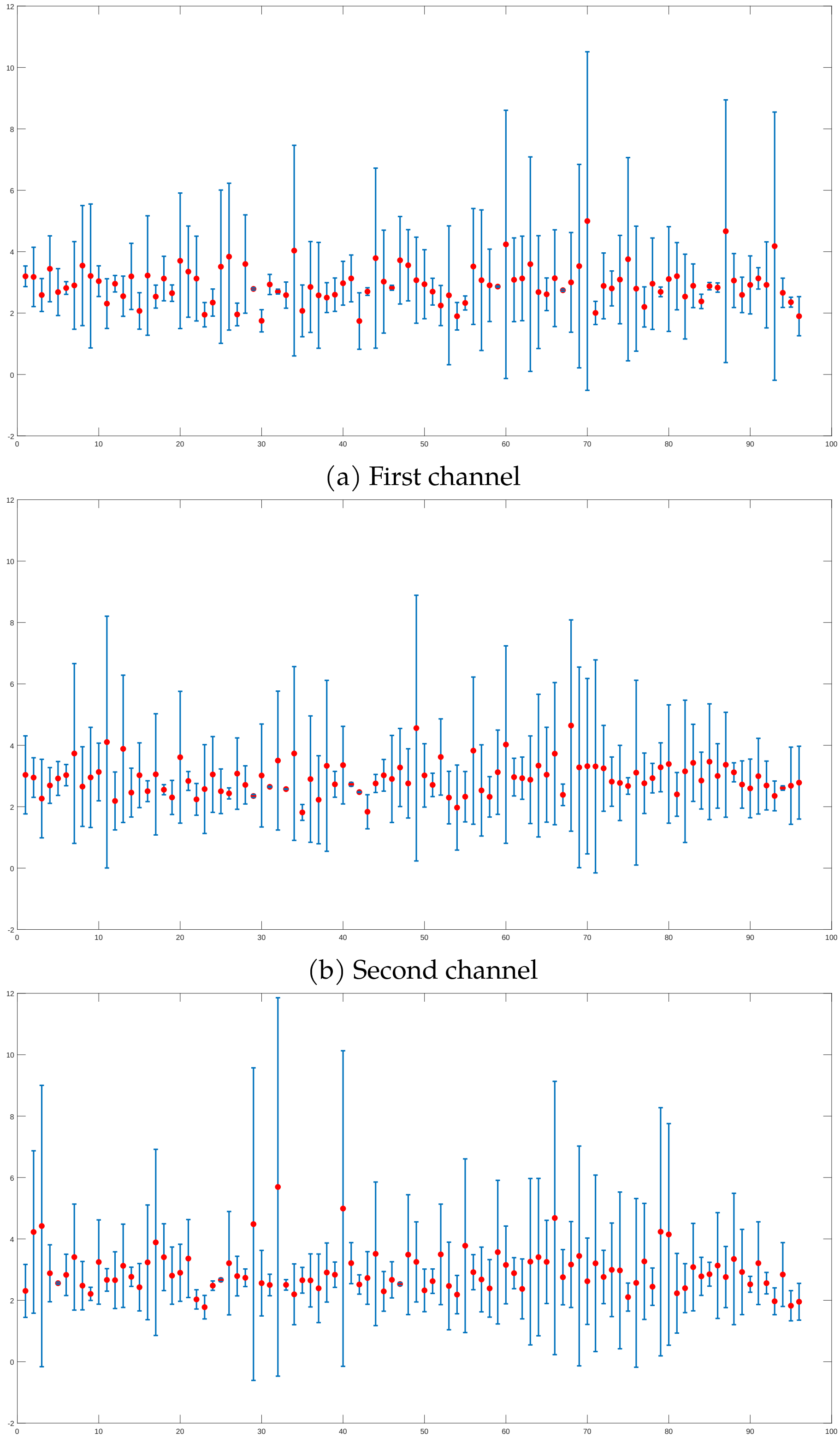
			G_1	G_2	G_3	G_4	G_5	G_6	G_7
PH - Dimension 0	Image	Sensitivity	48.91	57.64	57.23	60.14	62.32	56.55	55.82
		Specificity	55.00	59.68	58.36	59.14	60.68	58.55	59.55
		Accuracy	51.95	58.66	57.80	59.64	61.50	57.55	57.68
	1^{st} CL	Sensitivity	69.50	67.91	71.91	64.32	76.45	71.86	69.09
		Specificity	68.00	65.91	67.14	60.59	70.50	69.91	67.77
		Accuracy	68.75	66.91	69.52	62.45	73.48	70.89	68.43
	2^{nd} CL	Sensitivity	71.55	77.86	77.36	76.14	76.41	75.86	77.05
		Specificity	64.86	67.95	65.77	64.64	68.09	66.64	68.59
		Accuracy	68.20	72.91	71.57	70.39	72.25	71.25	72.82
	3^{rd} CL	Sensitivity	67.45	64.91	71.32	72.23	69.18	64.82	62.36
		Specificity	65.82	59.82	62.68	69.09	66.50	70.36	59.45
		Accuracy	66.64	62.36	67.00	70.66	67.84	67.59	60.91
	4^{th} CL	Sensitivity	65.14	68.50	68.32	70.64	67.91	67.64	73.05
		Specificity	65.18	69.00	66.45	67.27	65.41	70.18	64.14
		Accuracy	65.16	68.75	67.39	68.95	66.66	68.91	68.59
	5^{th} CL	Sensitivity	66.00	64.23	62.95	66.55	64.14	57.55	64.36
		Specificity	63.36	61.64	62.86	64.77	65.68	55.77	61.91
		Accuracy	64.68	62.93	62.91	65.66	64.91	56.66	63.14
PH - Dimension 1	Image	Sensitivity	36.18	40.59	62.32	59.95	59.73	60.36	61.14
		Specificity	69.14	66.77	59.95	61.36	58.86	59.91	61.14
		Accuracy	52.66	53.68	61.14	60.66	59.30	60.14	61.14
	1^{st} CL	Sensitivity	61.27	71.64	66.59	75.64	70.18	68.45	65.18
		Specificity	63.05	70.41	68.95	63.86	69.64	67.50	70.95
		Accuracy	62.16	71.02	67.77	69.75	69.91	67.98	68.07
	2^{nd} CL	Sensitivity	64.41	53.91	73.45	71.68	64.55	50.64	65.41
		Specificity	54.50	73.09	69.14	55.36	72.23	75.27	64.00
		Accuracy	59.45	63.50	71.30	63.52	68.39	62.95	64.70
	3^{rd} CL	Sensitivity	67.64	37.18	54.77	61.86	37.23	80.91	44.68
		Specificity	55.68	68.82	61.09	63.27	58.45	25.68	53.32
		Accuracy	61.66	53.00	57.93	62.57	47.84	53.30	49.00
	4^{th} CL	Sensitivity	59.09	46.14	45.50	63.68	52.09	14.09	73.59
		Specificity	56.55	65.59	62.55	64.23	65.05	91.50	54.95
		Accuracy	57.82	55.86	54.02	63.95	58.57	52.80	64.27
	5^{th} CL	Sensitivity	57.64	85.86	50.59	64.64	48.73	2.45	59.09
		Specificity	58.41	22.64	65.82	67.55	53.05	95.95	55.91
		Accuracy	58.02	54.25	58.20	66.09	50.89	49.20	57.50

Table A.2: Classification accuracy, sensitivity, and specificity of PH based ULBP landmarks in dimensions zero and one of breast US dataset.

			G_1	G_2	G_3	G_4	G_5	G_6	G_7
PH - Dimension 0	Image	Sensitivity	72.50	69.55	72.00	69.05	71.95	71.09	72.95
		Specificity	72.91	68.18	74.55	64.64	71.73	71.82	74.27
		Accuracy	72.70	68.86	73.27	66.84	71.84	71.45	73.61
	1^{st} CL	Sensitivity	80.64	85.55	81.18	80.05	74.55	77.64	77.86
		Specificity	79.91	81.59	77.50	79.50	75.77	74.32	76.50
		Accuracy	80.27	83.57	79.34	79.77	75.16	75.98	77.18
	2^{nd} CL	Sensitivity	82.64	77.82	79.41	81.91	77.05	77.59	79.55
		Specificity	80.68	78.32	80.64	79.82	77.82	72.91	74.91
		Accuracy	81.66	78.07	80.02	80.86	77.43	75.25	77.23
	3^{rd} CL	Sensitivity	72.82	69.45	74.95	77.68	72.05	57.73	71.55
		Specificity	78.27	73.59	76.09	82.73	79.05	62.73	66.73
		Accuracy	75.55	71.52	75.52	80.20	75.55	60.23	69.14
	4^{th} CL	Sensitivity	79.55	73.05	78.05	79.77	78.50	76.50	78.18
		Specificity	81.68	75.18	80.32	83.41	80.77	78.82	79.14
		Accuracy	80.61	74.11	79.18	81.59	79.64	77.66	78.66
	5^{th} CL	Sensitivity	73.59	68.68	75.50	71.50	74.05	60.86	73.95
		Specificity	82.95	76.55	80.86	79.05	80.45	66.73	76.36
		Accuracy	78.27	72.61	78.18	75.27	77.25	63.80	75.16
PH - Dimension 1	Image	Sensitivity	65.91	68.23	71.73	77.00	68.86	67.55	71.95
		Specificity	71.09	75.95	73.50	73.82	68.41	70.55	74.73
		Accuracy	68.50	72.09	72.61	75.41	68.64	69.05	73.34
	1^{st} CL	Sensitivity	79.05	81.14	79.41	78.68	79.09	74.68	68.82
		Specificity	78.91	78.45	76.73	73.45	75.23	72.68	74.32
		Accuracy	78.98	79.80	78.07	76.07	77.16	73.68	71.57
	2^{nd} CL	Sensitivity	72.55	63.45	71.05	75.77	73.00	81.59	73.41
		Specificity	70.45	52.86	69.14	64.23	58.32	57.68	62.23
		Accuracy	71.50	58.16	70.09	70.00	65.66	69.64	67.82
	3^{rd} CL	Sensitivity	61.05	9.36	35.23	57.55	44.55	7.50	73.50
		Specificity	55.32	87.68	75.41	68.73	57.91	90.27	31.55
		Accuracy	58.18	48.52	55.32	63.14	51.23	48.89	52.52
	4^{th} CL	Sensitivity	72.27	29.95	49.23	68.05	36.91	8.82	73.23
		Specificity	74.77	69.86	68.05	76.64	67.86	85.18	58.86
		Accuracy	73.52	49.91	58.64	72.34	52.39	47.00	66.05
	5^{th} CL	Sensitivity	69.86	23.18	58.59	66.41	57.09	11.09	54.18
		Specificity	73.91	86.59	66.68	69.64	53.14	93.77	50.77
		Accuracy	71.89	54.89	62.64	68.02	55.11	52.43	52.48

Table A.3: Classification accuracy, sensitivity, and specificity of PH based ULBP landmarks in dimensions zero and one of liver US dataset.

			G_1	G_2	G_3	G_4	G_5	G_6	G_7
PH - Dimension 0	Image	Sensitivity	76.95	79.23	85.00	82.05	84.95	85.95	85.73
		Specificity	84.27	83.86	83.27	84.73	86.36	81.95	86.64
		Accuracy	80.61	81.55	84.14	83.39	85.66	83.95	86.18
	1^{st} CL	Sensitivity	86.50	88.18	88.09	86.27	88.32	87.77	82.95
		Specificity	89.27	85.86	86.50	85.05	87.36	88.73	87.95
		Accuracy	87.89	87.02	87.30	85.66	87.84	88.25	85.45
	2^{nd} CL	Sensitivity	92.27	90.32	91.59	92.32	92.05	89.59	89.73
		Specificity	88.45	87.00	87.68	87.73	87.05	88.32	87.73
		Accuracy	90.36	88.66	89.64	90.02	89.55	88.95	88.73
	3^{rd} CL	Sensitivity	90.59	80.68	88.36	88.23	91.86	80.68	86.18
		Specificity	87.00	85.36	87.77	85.18	88.68	79.77	84.36
		Accuracy	88.80	83.02	88.07	86.70	90.27	80.23	85.27
	4^{th} CL	Sensitivity	89.32	88.50	91.59	91.82	89.45	86.36	88.55
		Specificity	88.64	84.82	87.45	87.36	87.73	86.27	82.14
		Accuracy	88.98	86.66	89.52	89.59	88.59	86.32	85.34
	5^{th} CL	Sensitivity	84.95	82.59	86.64	89.68	88.55	81.32	77.41
		Specificity	86.36	82.05	88.59	87.50	85.82	79.68	81.82
		Accuracy	85.66	82.32	87.61	88.59	87.18	80.50	79.61
PH - Dimension 1	Image	Sensitivity	75.82	75.95	82.95	84.36	85.91	84.77	83.73
		Specificity	87.41	86.73	83.18	83.73	85.05	80.86	86.41
		Accuracy	81.61	81.34	83.07	84.05	85.48	82.82	85.07
	1^{st} CL	Sensitivity	87.00	84.27	87.27	89.59	91.41	85.41	76.18
		Specificity	92.86	78.05	83.18	83.00	84.14	85.14	83.86
		Accuracy	89.93	81.16	85.23	86.30	87.77	85.27	80.02
	2^{nd} CL	Sensitivity	92.82	81.50	93.59	97.23	93.00	88.27	90.45
		Specificity	78.05	76.32	70.73	75.73	71.18	68.64	83.00
		Accuracy	85.43	78.91	82.16	86.48	82.09	78.45	86.73
	3^{rd} CL	Sensitivity	82.82	28.73	42.23	74.73	56.14	9.82	85.95
		Specificity	62.41	87.36	84.18	73.32	77.05	92.00	41.09
		Accuracy	72.61	58.05	63.20	74.02	66.59	50.91	63.52
	4^{th} CL	Sensitivity	89.14	37.27	71.05	86.77	63.50	15.18	91.09
		Specificity	83.55	77.50	75.82	84.45	68.50	94.73	65.18
		Accuracy	86.34	57.39	73.43	85.61	66.00	54.95	78.14
	5^{th} CL	Sensitivity	80.05	66.59	61.68	80.41	74.91	56.18	77.95
		Specificity	76.95	40.77	70.73	81.09	70.77	43.41	59.41
		Accuracy	78.50	53.68	66.20	80.75	72.84	49.80	68.68



(a) First channel

(b) Second channel

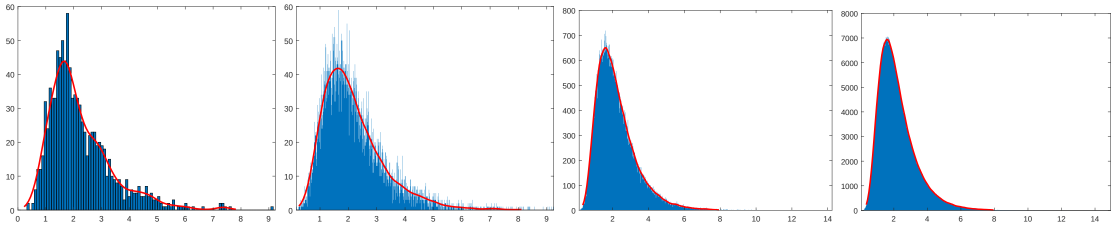
(c) Third channel

Figure A.3: Instability of 1^{st} convolutional layer filter's condition number per channel - pretrained AlexNet.

A.2 Chapter 5

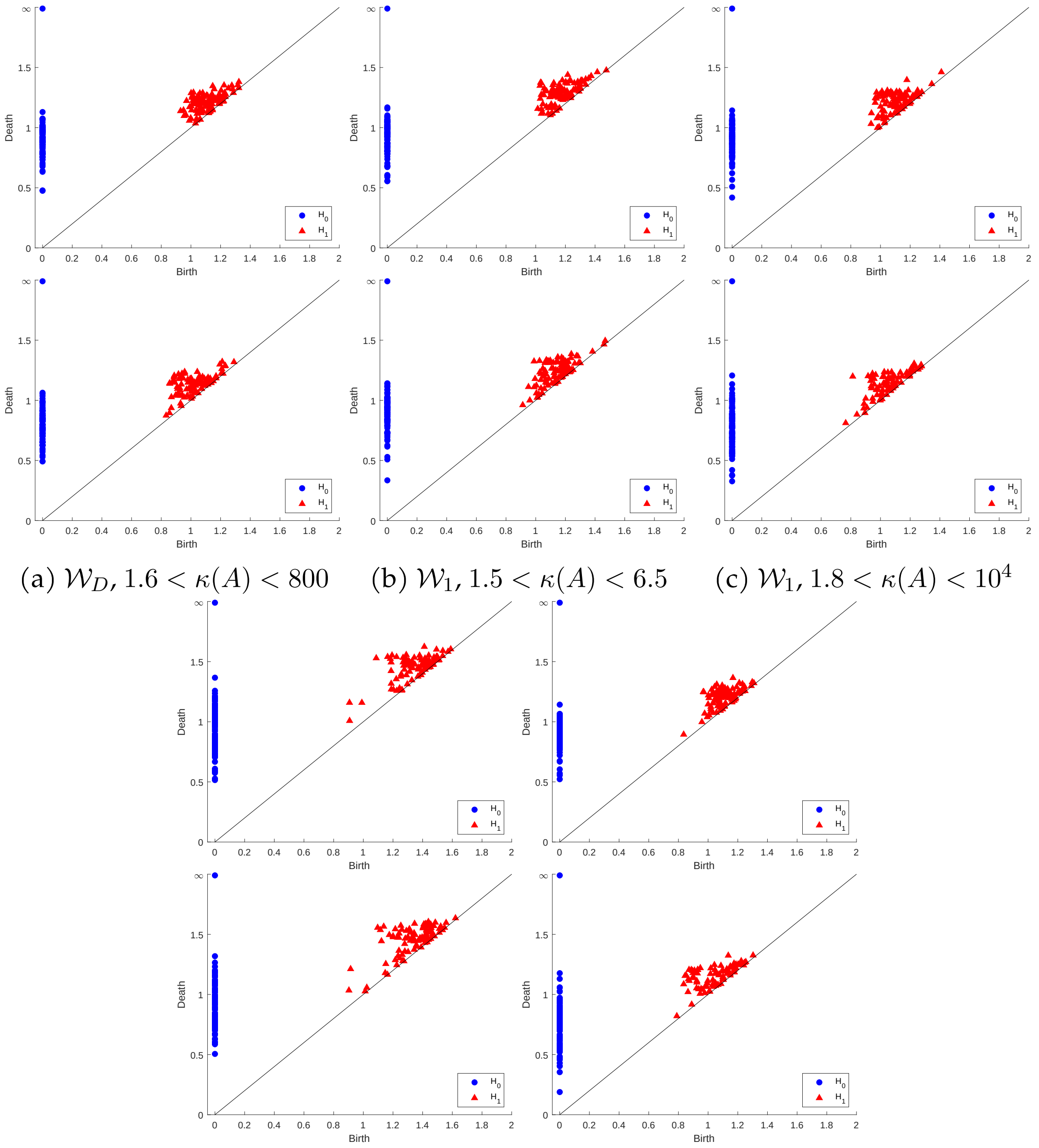
Table A.4: Classification accuracy of various customised CNN model-B with-/without BN .

CNN model	Validation acc.	Test acc.	Sensitivity	Specificity
Model-B, $\mathcal{W}_D$	91.94	90.47	89.76	91.19
Model-B, $\mathcal{F}_D$	90.75	89.88	92.62	87.14
Model-B, $\mathcal{F}_D$, BN	90	91.19	92.38	90.00
Model-B, $\mathcal{W}_1$	90.75	91.42	88.81	94.05
Model-B, $\mathcal{F}_1$	89.55	89.4	84.76	94.05
Model-B, $\mathcal{F}_1$, BN	94.33	94.05	92.86	95.24
Model-B, $\mathcal{W}_2$	93.13	89.64	90.24	89.05
Model-B, $\mathcal{F}_2$	88.36	86.31	90.71	81.90
Model-B, $\mathcal{F}_2$, BN	94.03	94.05	90.95	97.14
Model-B, $\mathcal{F}_3$	89.85	87.98	85.71	90.24
Model-B, $\mathcal{F}_3$, BN	95.07	93.69	95.00	92.38
Model-B, $\mathcal{F}_4$	72.09	72.14	76.67	70.48
Model-B, $\mathcal{F}_4$, BN	94.93	94.88	95.48	94.29
Model-B, $\mathcal{F}_5$	87.16	87.02	92.62	81.43
Model-B, $\mathcal{F}_5$, BN	93.88	94.64	94.76	94.52
Model-B, $\mathcal{F}_6$	90.6	89.4	88.33	90.48
Model-B, $\mathcal{F}_6$, BN	94.63	94.64	94.05	95.24



(a) $N = 10^3$ (b) $N = 10^4$ (c) $N = 10^5$ (d) $N = 10^6$

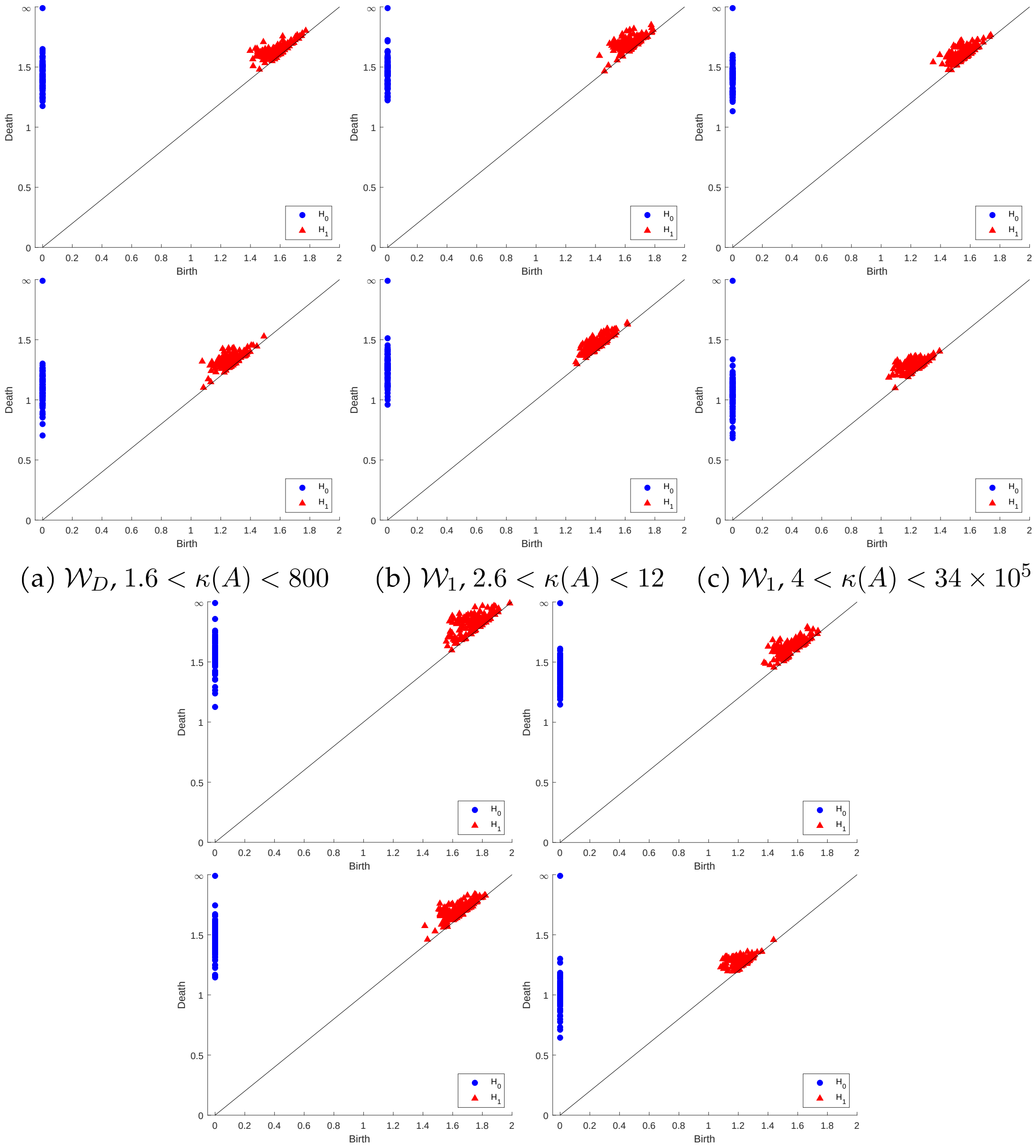
Figure A.4: Distribution of condition number (log) - N number of 3×3 RGFs.



(a) $\mathcal{W}_D$, $1.6 < \kappa(A) < 800$ (b) $\mathcal{W}_1$, $1.5 < \kappa(A) < 6.5$ (c) $\mathcal{W}_1$, $1.8 < \kappa(A) < 10^4$

(d) $\mathcal{W}_2$, $1.2 < \kappa(A) < 2.5$ (e) $\mathcal{W}_2$, $1.3 < \kappa(A) < 4\times10^6$

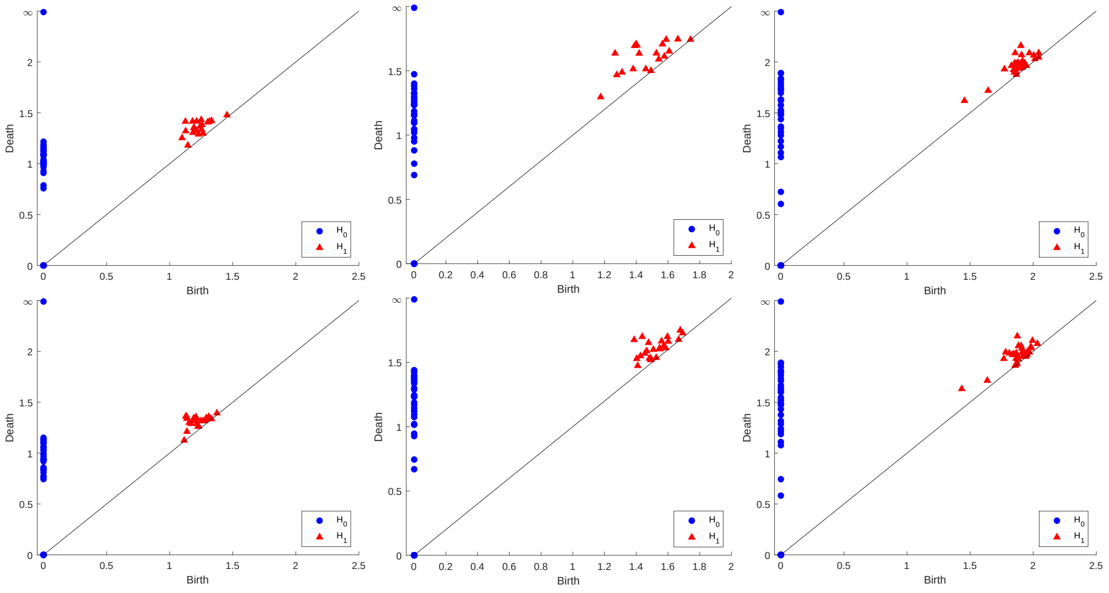
Figure A.5: PD of point clouds of 3×3 filters (top) and their inverses (bottom).



(a) $\mathcal{W}_D$, $1.6 < \kappa(A) < 800$ (b) $\mathcal{W}_1$, $2.6 < \kappa(A) < 12$ (c) $\mathcal{W}_1$, $4 < \kappa(A) < 34 \times 10^5$

(d) $\mathcal{W}_2$, $2 < \kappa(A) < 5.5$ (e) $\mathcal{W}_2$, $2.8 < \kappa(A) < 7\times10^6$

Figure A.6: PD of point clouds of 5×5 filters (top) and their inverses (bottom).



(a) $\mathcal{F}_D$ (b) $\mathcal{F}_1$ (c) $\mathcal{F}_2$

Figure A.7: PD of point clouds of 3×3 unified-channel filters (top) and their inverses (bottom).

A.3 Chapter 6

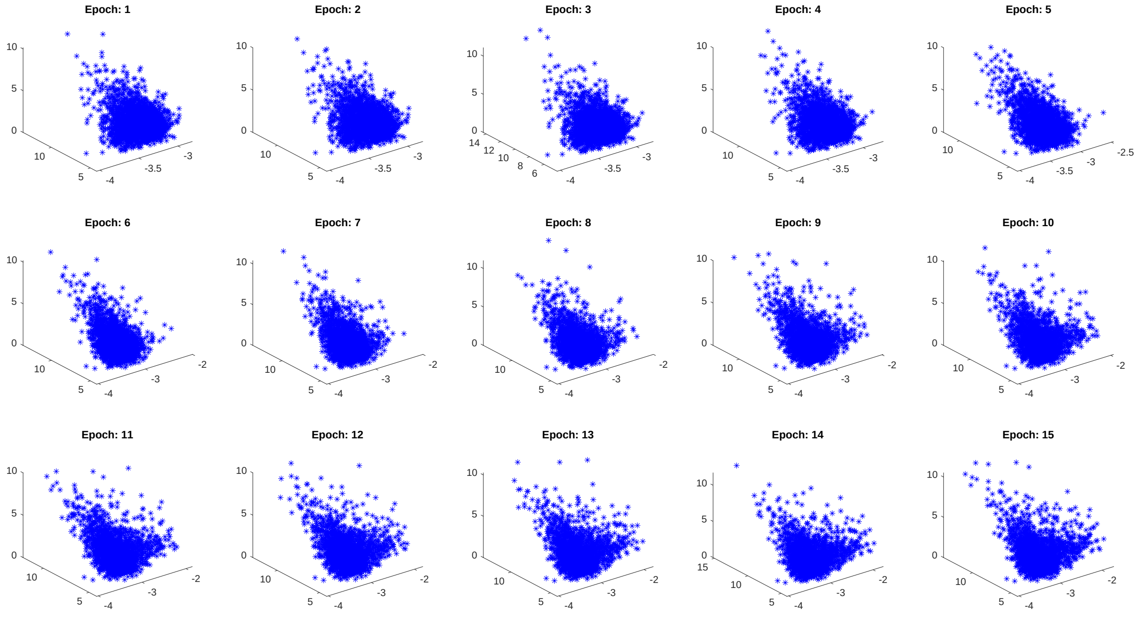


Figure A.8: 3D graphs $(x, y, z) = (\|filter\|, \|filter^{-1})\|, \kappa(filter))$ for the 2^{nd} CL 5×5 filters in $\mathcal{W}_D$.

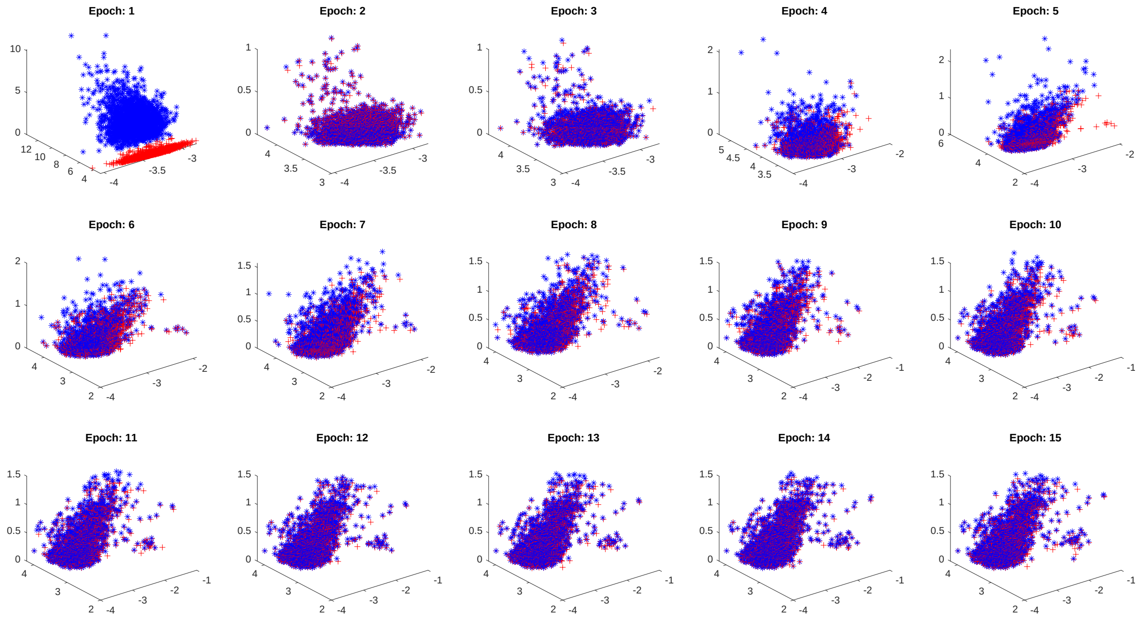
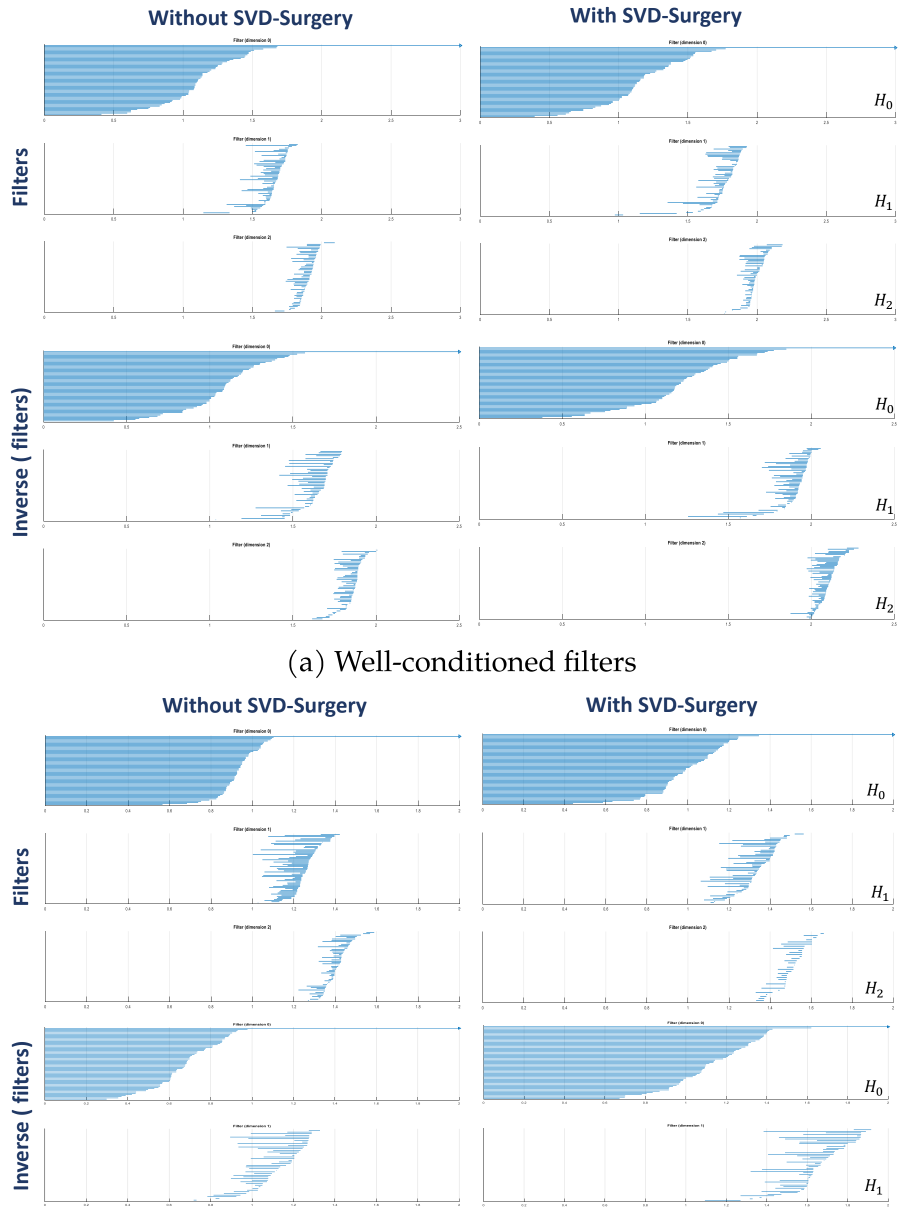


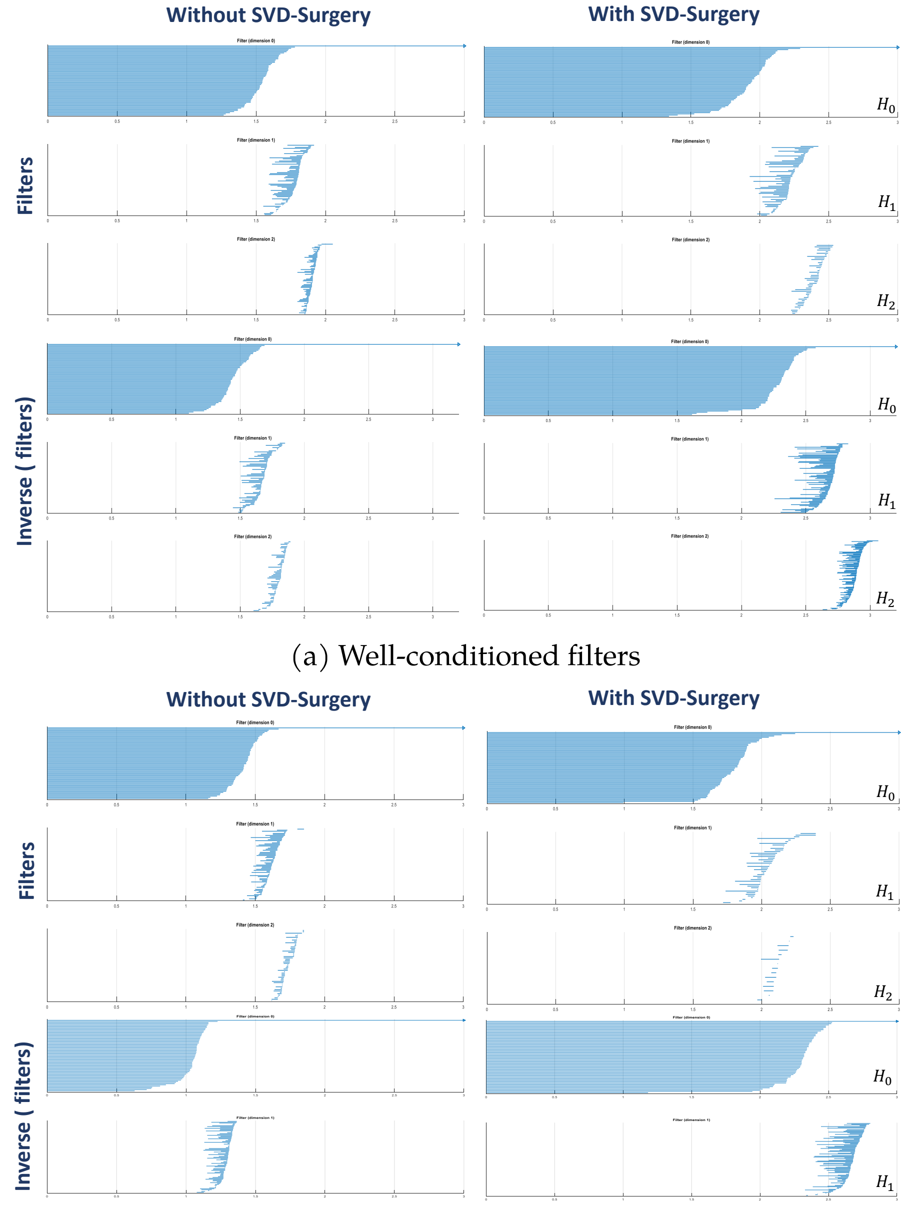
Figure A.9: 3D illustration of the norm (x-axis), norm of inverse (y-axis) and condition number (z-axis) of the 2^{nd} CL 5×5 weights per epoch (log) before and after SVD-Surgery.



(a) Well-conditioned filters

(b) Ill-conditioned filters

Figure A.10: Persistent barcode representations of 3×3 matrices.



(a) Well-conditioned filters

(b) Ill-conditioned filters

Figure A.11: Persistent barcode representations of 5×5 matrices.

Bibliography

[1] M. J. Colbrook, V. Antun, and A. C. Hansen, "The difficulty of computing stable and accurate neural networks: On the barriers of deep learning and smale's 18th problem," *Proceedings of the National Academy of Sciences*, vol. 119, no. 12, p. e2107151119, 2022.

[2] D. Heaven, "Why deep-learning AIs are so easy to fool," *Nature*, vol. 574, pp. 163–166, 10 2019.

[3] A. Bastounis, A. C. Hansen, and V. Vlačić, "The mathematics of adversarial attacks in ai–why deep learning is unstable despite the existence of stable neural networks," *arXiv preprint arXiv:2109.06098*, 2021.

[4] Y.-C. Zhu, A. AlZoubi, S. Jassim, Q. Jiang, Y. Zhang, Y.-B. Wang, X.-D. Ye, and D. Hongbo, "A generic deep learning framework to classify thyroid and breast lesions in ultrasound images," *Ultrasonics*, vol. 110, p. 106300, 2021.

[5] N. Byrne, J. R. Clough, I. Valverde, G. Montana, and A. P. King, "A persistent homology-based topological loss for cnn-based multiclass segmentation of cmr," *IEEE Transactions on Medical Imaging*, vol. 42, no. 1, pp. 3–14, 2022.

[6] L. J. Brattain, B. A. Telfer, M. Dhyani, J. R. Grajo, and A. E. Samir, "Machine learning for medical ultrasound: status, methods, and future opportunities," *Abdominal Radiology*, vol. 43, pp. 786–799, 4 2018.

[7] S. Khazendar, *Computer-aided diagnosis of gynaecological abnormality using B-mode ultrasound images*. PhD thesis, University of Buckingham, 2016.

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, International Conference on Learning Representations, ICLR, 12 2015.

[9] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 86–94, 10 2016.

[10] S. Y. Khamaiseh, D. Bagagem, A. Al-Alaj, M. Mancino, and H. W. Alomari, "Adversarial deep learning: A survey on adversarial attacks and defense mechanisms on image classification," *IEEE Access*, 2022.

[11] R. R. Wiyatno, A. Xu, O. Dia, and A. de Berker, "Adversarial examples in modern machine learning: A review," *arXiv preprint arXiv:1911.05268*, 2019.

[12] I. Y. Tyukin, D. J. Higham, and A. N. Gorban, "On adversarial examples and stealth attacks in artificial intelligence systems," in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6, IEEE, 2020.

[13] X. Ma, Y. Niu, L. Gu, Y. Wang, Y. Zhao, J. Bailey, and F. Lu, "Understanding adversarial attacks on deep learning based medical image analysis systems," *Pattern Recognition*, vol. 110, p. 107332, 2021.

[14] K. D. Apostolidis and G. A. Papakostas, "A survey on adversarial deep learning robustness in medical image analysis," *Electronics*, vol. 10, no. 17, p. 2132, 2021.

[15] G. Bortsova, C. González-Gonzalo, S. C. Wetstein, F. Dubost, I. Katramados, L. Hogeweg, B. Liefers, B. van Ginneken, J. P. Pluim, M. Veta, *et al.*, "Adversarial attack vulnerability of medical image analysis systems: Unexplored factors," *Medical Image Analysis*, vol. 73, p. 102141, 2021.

[16] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, and I. S. Kohane, "Adversarial attacks on medical machine learning," *Science*, vol. 363, pp. 1287–1289, 3 2019.

[17] S. Whelan, "Benign vs Malignant Tumors | Technology Networks."

[18] N. J. Higham, *Accuracy and stability of numerical algorithms*. SIAM, 2002.

[19] J. Ghafuri, H. Du, and S. Jassim, "Topological aspects of CNN convolution layers for medical image analysis," in *Mobile Multimedia/Image Processing, Security, and Applications 2020*, vol. 11399, pp. 229–240, SPIE, 2020.

[20] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[21] J. Berner, P. Grohs, G. Kutyniok, and P. Petersen, *The Modern Mathematics of Deep Learning*, p. 1–111. Cambridge University Press, 2022.

[22] C. F. G. D. Santos and J. P. Papa, "Avoiding overfitting: A survey on regularization methods for convolutional neural networks," *ACM Computing Surveys (CSUR)*, vol. 54, no. 10s, pp. 1–25, 2022.

[23] J. Ghafuri, H. Du, and S. Jassim, "Sensitivity and stability of pretrained CNN filters," in *Multimodal Image Exploitation and Learning 2021*, vol. 11734, pp. 79–89, SPIE, 2021.

[24] J. Ghafuri, H. Du, and S. Jassim, "Impact of Convolutional Layer Filters' Instability on Robustness of Classification Decisions for Tumour Diagnosis from Ultrasound Images," 26^{th} UK conference on Medical Image Understanding and Analysis, Cambridge, UK,2022.

[25] J. Ghafuri and S. Jassim, "Singular value decomposition based matrix surgery," *arXiv preprint arXiv:2302.11446*, 2023.

[26] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[28] M. H. Herzog and A. M. Clarke, "Why vision is not both hierarchical and feedforward," *Frontiers in computational neuroscience*, vol. 8, p. 135, 2014.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[30] O. K. Oyedotun, K. Al Ismaeil, and D. Aouada, "Training very deep neural networks: Rethinking the role of skip connections," *Neurocomputing*, vol. 441, pp. 105–117, 2021.

[31] O. K. Oyedotun, K. Al Ismaeil, and D. Aouada, "Why is everyone training very deep neural network with skip connections?," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[32] F. Lei, X. Liu, Q. Dai, and B. W.-K. Ling, "Shallow convolutional neural network for image classification," *SN Applied Sciences*, vol. 2, pp. 1–8, 2020.

[33] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International conference on machine learning*, pp. 7472–7482, PMLR, 2019.

[34] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," *Advances in neural information processing systems*, vol. 28, 2015.

[35] X. Yu, T. Liu, X. Wang, and D. Tao, "On compressing deep models by low rank and sparse decomposition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7370–7379, 2017.

[36] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang, "Pruning and quantization for deep neural network acceleration: A survey," *Neurocomputing*, vol. 461, pp. 370–403, 2021.

[37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2016.

[40] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, pp. 10691–10700, 5 2019.

[41] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[42] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.

[43] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8697–8710, 2018.

[44] H. Pham, M. Guan, B. Zoph, Q. Le, and J. Dean, "Efficient neural architecture search via parameters sharing," in *Proceedings of the 35th International Conference on Machine Learning* (J. Dy and A. Krause, eds.), vol. 80 of *Proceedings of Machine Learning Research*, pp. 4095–4104, PMLR, 10–15 Jul 2018.

[45] M. Ahmed, H. Du, and A. AlZoubi, "An enas based approach for constructing deep learning models for breast cancer recognition from ultrasound images," *arXiv preprint arXiv:2005.13695*, 2020.

[46] G. Carlsson, "Topology and data," *Bulletin of the American Mathematical Society*, vol. 46, no. 2, pp. 255–308, 2009.

[47] H. Edelsbrunner, D. Letscher, and A. Zomorodian, "Topological persistence and simplification," in *Proceedings 41st annual symposium on foundations of computer science*, pp. 454–463, IEEE, 2000.

[48] R. Ghrist, "Barcodes: the persistent topology of data," *Bulletin of the American Mathematical Society*, vol. 45, no. 1, pp. 61–75, 2008.

[49] N. Otter, M. A. Porter, U. Tillmann, P. Grindrod, and H. A. Harrington, "A roadmap for the computation of persistent homology," *EPJ Data Science*, vol. 6, pp. 1–38, 2017.

[50] H. Edelsbrunner and J. L. Harer, *Computational topology: an introduction*. American Mathematical Society, 2022.

[51] U. Bauer and H. Edelsbrunner, “The morse theory of čech and delaunay complexes,” *Transactions of the American Mathematical Society*, vol. 369, no. 5, pp. 3741–3762, 2017.

[52] R. Turkeš, G. Montúfar, and N. Otter, “On the effectiveness of persistent homology,” *arXiv preprint arXiv:2206.10551*, 2022.

[53] T. K. Dey and Y. Wang, *Computational topology for data analysis*. Cambridge University Press, 2022.

[54] A. Asaad, *Persistent Homology Tools for Image Analysis*. PhD thesis, University of Buckingham, 2020.

[55] G. Singh, F. Mémoli, and G. E. Carlsson, “Topological methods for the analysis of high dimensional data sets and 3d object recognition.,” *PBG@ Eurographics*, vol. 2, pp. 091–100, 2007.

[56] F. Belchí, J. Brodzki, M. Burfitt, and M. Niranjan, “A numerical measure of the instability of mapper-type algorithms,” *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 8347–8391, 2020.

[57] P. Y. Lum, G. Singh, A. Lehman, T. Ishkanov, M. Vejdemo-Johansson, M. Alagappan, J. Carlsson, and G. Carlsson, “Extracting insights from the shape of complex data using topology,” *Scientific reports*, vol. 3, no. 1, pp. 1–8, 2013.

[58] F. Chazal and B. Michel, “An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists,” *Frontiers in Artificial Intelligence*, vol. 4, 10 2017.

[59] A. M. Turing, “Rounding-off errors in matrix processes,” *The Quarterly Journal of Mechanics and Applied Mathematics*, vol. 1, no. 1, pp. 287–308, 1948.

[60] A. Edelman, *Eigenvalues and Condition numbers*. PhD thesis, MIT, 1989.

[61] J. Todd, “The condition of a certain matrix,” in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 46, pp. 116–118, Cambridge University Press, 1950.

[62] A. Edelman, “On the distribution of a scaled condition number,” *Mathematics of computation*, vol. 58, no. 197, pp. 185–190, 1992.

[63] J. R. Rice, “A theory of condition,” *SIAM Journal on Numerical Analysis*, vol. 3, no. 2, pp. 287–310, 1966.

[64] J. W. Demmel, “The geometry of ill-conditioning,” *Journal of Complexity*, vol. 3, no. 2, pp. 201–229, 1987.

[65] D. J. Higham, “Condition numbers and their condition numbers,” *Linear Algebra and its Applications*, vol. 214, pp. 193–213, 1995.

[66] A. Sankar, D. A. Spielman, and S.-H. Teng, "Smoothed analysis of the condition numbers and growth factors of matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 28, no. 2, pp. 446–476, 2006.

[67] V. Klema and A. Laub, "The singular value decomposition: Its computation and some applications," *IEEE Transactions on automatic control*, vol. 25, no. 2, pp. 164–176, 1980.

[68] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, vol. 9, pp. 249–256, 2010.

[69] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, vol. 2015 Inter, pp. 1026–1034, 2 2015.

[70] M. V. Narkhede, P. P. Bartakke, and M. S. Sutaone, "A review on weight initialization strategies for neural networks," *Artificial intelligence review*, vol. 55, no. 1, pp. 291–322, 2022.

[71] Y. Wen, L. Chen, Y. Deng, and C. Zhou, "Rethinking pre-training on medical imaging," *Journal of Visual Communication and Image Representation*, vol. 78, p. 103145, 2021.

[72] H. Zhang, L. Feng, X. Zhang, Y. Yang, and J. Li, "Necessary conditions for convergence of cnns and initialization of convolution kernels," *Digital Signal Processing*, p. 103397, 2022.

[73] A. Carovac, F. Smajlovic, and D. Junuzovic, "Application of ultrasound in medicine," *Acta Informatica Medica*, vol. 19, no. 3, p. 168, 2011.

[74] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[75] G. Castellano, L. Bonilha, L. Li, and F. Cendes, "Texture analysis of medical images," *Clinical radiology*, vol. 59, no. 12, pp. 1061–1069, 2004.

[76] L. Armi and S. Fekri-Ershad, "Texture image analysis and texture classification methods-a review," *arXiv preprint arXiv:1904.06554*, 2019.

[77] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognition*, vol. 29, pp. 51–59, 1 1996.

[78] B. S. Vidya and E. Chandra, "Entropy based local binary pattern (elbp) feature extraction technique of multimodal biometrics as defence mechanism for cloud storage," *Alexandria Engineering Journal*, vol. 58, no. 1, pp. 103–114, 2019.

[79] J. Raghavan and M. Ahmadi, "Performance evaluation of entropy based lbp for face recognition," in *2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 241–245, IEEE, 2021.

[80] A. Asaad and S. Jassim, "Topological data analysis for image tampering detection," in *International Workshop on Digital Watermarking*, vol. 10431, pp. 136–146, Springer, Springer, 2017.

[81] M. Ploquin, A. Basarab, and D. Kouamé, "Resolution enhancement in medical ultrasound imaging," *Journal of Medical Imaging*, vol. 2, no. 1, pp. 017001–017001, 2015.

[82] H. Temiz and H. S. Bilge, "Super resolution of b-mode ultrasound images with deep learning," *IEEE Access*, vol. 8, pp. 78808–78820, 2020.

[83] T. Hassan, A. AlZoubi, H. Du, and S. Jassim, "Towards optimal cropping: breast and liver tumor classification using ultrasound images," in *Multimodal Image Exploitation and Learning 2021*, vol. 11734, pp. 111–122, SPIE, 2021.

[84] T. Hassan, A. AlZoubi, H. Du, and S. Jassim, "Ultrasound image augmentation by tumor margin appending for robust deep learning based breast lesion classification," in *Multimodal Image Exploitation and Learning 2022*, vol. 12100, pp. 80–89, SPIE, 2022.

[85] D. Al-Karawi, D. Ibrahim, H. Al-Assam, H. Du, and S. Jassim, "A model-based adaptive method for speckle noise reduction in ultrasound images of ovarian tumours: a new approach," in *Multimodal Image Exploitation and Learning 2021*, vol. 11734, pp. 133–145, SPIE, 2021.

[86] G. Carlsson, T. Ishkhanov, V. De Silva, A. Zomorodian, G. Carlsson, T. Ishkhanov, V. De Silva, and A. Zomorodian, "On the local behavior of spaces of natural images," *International Journal of Computer Vision*, vol. 76, pp. 1–12, 1 2008.

[87] G. Carlsson and R. B. Gabrielsson, "Topological approaches to deep learning," in *Topological data analysis*, pp. 119–146, Springer, 2020.

[88] A. B. Lee, K. S. Pedersen, and D. Mumford, "The nonlinear statistics of high-contrast patches in natural images," *International Journal of Computer Vision*, vol. 54, pp. 83–103, 2003.

[89] R. B. Gabrielsson and G. Carlsson, "Exposition and interpretation of the topology of neural networks," in *2019 18th ieee international conference on machine learning and applications (icmla)*, pp. 1069–1076, IEEE, 2019.

[90] S. Chowdhury, T. Gebhart, S. Huntsman, and M. Yutin, "Path homologies of deep feedforward networks," 10 2019.

[91] A. Rathore, N. Chalapathi, S. Palande, and B. Wang, "Topoact: visually exploring the shape of activations in deep learning," in *Computer Graphics Forum*, vol. 40, pp. 382–397, Wiley Online Library, 2021.

[92] M. Gabella, "Topology of learning in feedforward neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3588–3592, 2020.

[93] T. Lacombe, Y. Ike, M. Carriere, F. Chazal, M. Glisse, and Y. Umeda, "Topological uncertainty: Monitoring trained neural networks through persistence of activation graphs," *arXiv preprint arXiv:2105.04404*, 2021.

[94] Y. Zhao and H. Zhang, "Quantitative performance assessment of cnn units via topological entropy calculation," *arXiv preprint arXiv:2103.09716*, 2021.

[95] R. Ballester, X. A. Clemente, C. Casacuberta, M. Madadi, C. A. Corneanu, and S. Escalera, "Towards explaining the generalization gap in neural networks using topological data analysis," *arXiv preprint arXiv:2203.12330*, 2022.

[96] G. Jorgenson, H. Kvinge, T. Emerson, and C. Olson, "Random filters for enriching the discriminatory power of topological representations," in *Topological, Algebraic and Geometric Learning Workshops 2022*, pp. 183–188, PMLR, 2022.

[97] E. R. Love, B. Filippenko, V. Maroulas, and G. Carlsson, "Topological convolutional layers for deep learning," *Journal of Machine Learning Research*, vol. 24, no. 59, pp. 1–35, 2023.

[98] A. Eskandari, H. Du, and A. AlZoubi, "Towards linking cnn decisions with cancer signs for breast lesion classification from ultrasound images," in *Medical Image Understanding and Analysis: 25th Annual Conference, MIUA 2021, Oxford, United Kingdom, July 12–14, 2021, Proceedings 25*, pp. 423–437, Springer, 2021.

[99] M. E. Aktas, E. Akbas, and A. E. Fatmaoui, "Persistence homology of networks: methods and applications," *Applied Network Science*, vol. 4, no. 1, pp. 1–28, 2019.

[100] G. Carlsson, T. Ishkhanov, V. De Silva, and A. Zomorodian, "On the local behavior of spaces of natural images," *International journal of computer vision*, vol. 76, pp. 1–12, 2008.

[101] W. Al-Dhabyani, M. Gomaa, H. Khaled, and A. Fahmy, "Dataset of breast ultrasound images," *Data in Brief*, vol. 28, p. 104863, 2 2020.

[102] A. Sinha, M. Singh, B. Krishnamurthy, A. Sinha, B. Krishnamurthy, M. Singh, and B. Krishnamurthy, "Neural networks in an adversarial setting and ill-conditioned weight space," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, vol. 11329 LNAI, pp. 177–190, Springer, Springer, 2018.

[103] The Mathworks, "Pretrained AlexNet convolutional neural network - MATLAB alexnet," 2017.

[104] U. Bauer, "Ripser: efficient computation of Vietoris–Rips persistence barcodes," *Journal of Applied and Computational Topology*, vol. 5, pp. 391–423, 9 2021.

[105] H. Li, A. Kadav, I. Durdanovic, H. Samet, H. P. Graf, A. Kadav, I. Durdanovic, H. P. Graf, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," *arXiv preprint arXiv:1608.08710*, 8 2016.

[106] M. Riera, J.-M. Arnau, and A. Gonzalez, "(Pen-) Ultimate DNN Pruning," *arXiv preprint arXiv:1906.02535*, 2019.

[107] Y.-C. Huang, K.-C. Hung, C.-C. Liu, T.-H. Chuang, and S.-J. Chiou, "Customized convolutional neural networks technology for machined product inspection," *Applied Sciences*, vol. 12, no. 6, p. 3014, 2022.

[108] S. H. Khan, Z. Abbas, S. D. Rizvi, *et al.*, "Classification of diabetic retinopathy images based on customised cnn architecture," in *2019 Amity International conference on artificial intelligence (AICAI)*, pp. 244–248, IEEE, 2019.

[109] O. Özkaraca, O. İ. Bağrıaçık, H. Gürüler, F. Khan, J. Hussain, J. Khan, and U. e. Laila, "Multiple brain tumor classification with dense cnn architecture using brain mri images," *Life*, vol. 13, no. 2, p. 349, 2023.

[110] Z. Zhu, F. Liu, G. Chrysos, and V. Cevher, "Robustness in deep learning: The good (width), the bad (depth), and the ugly (initialization)," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36094–36107, 2022.

[111] A. M. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," *arXiv preprint arXiv:1312.6120*, 12 2013.

[112] D. Mishkin and J. Matas, "All you need is a good init," *arXiv preprint arXiv:1511.06422*, 11 2015.

[113] D. Xie, J. Xiong, and S. Pu, "All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6176–6185, 3 2017.

[114] L. Huang, X. Liu, B. Lang, A. Yu, Y. Wang, and B. Li, "Orthogonal weight normalization: Solution to optimization over multiple dependent stiefel manifolds in deep neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[115] L. Huang, L. Liu, F. Zhu, D. Wan, Z. Yuan, B. Li, and L. Shao, "Controllable Orthogonalization in Training DNNs," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 6428–6437, 4 2020.

[116] J. Wang, Y. Chen, R. Chakraborty, and S. X. Yu, "Orthogonal Convolutional Neural Networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6 2020.

[117] K. Jia, S. Li, Y. Wen, T. Liu, D. Tao, K. Jia, Y. Wen, T. Liu, and D. Tao, "Orthogonal Deep Neural Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, pp. 1352–1368, 4 2021.

[118] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[119] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, pmlr, 2015.

[120] D. Wu, Y. Wang, S.-T. Xia, J. Bailey, and X. Ma, "Skip connections matter: On the transferability of adversarial examples generated with resnets," *arXiv preprint arXiv:2002.05990*, 2020.

[121] Z. Yue, B. Lin, Y. Zhang, and C. Liang, "Effective, efficient and robust neural architecture search," in *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2022.

[122] M. Ahmed, A. AlZoubi, and H. Du, "Improving generalization of enas-based cnn models for breast lesion classification from ultrasound images," in *Medical Image Understanding and Analysis: 25th Annual Conference, MIUA 2021, Oxford, United Kingdom, July 12–14, 2021, Proceedings 25*, pp. 438–453, Springer, 2021.

[123] M. Ahmed, *Automatic Convolutional Neural Network Architecture Search for Breast Lesion Classification from Ultrasound Images: An ENAS Bayesian Optimization approach*. PhD thesis, University of Buckingham, 2022.

[124] G. W. Stewart, "Perturbation theory for the singular value decomposition," tech. rep., 1998.

[125] J.-M. Azaïs and M. Wschebor, "Upper and lower bounds for the tails of the distribution of the condition number of a gaussian matrix," *SIAM Journal on Matrix Analysis and Applications*, vol. 26, no. 2, pp. 426–440, 2004.

[126] MathWorks, "Data Sets for Deep Learning - MATLAB & Simulink."

[127] C. C. Yann LeCun and C. Burges, "MNIST handwritten digit database."

[128] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[129] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[130] S. J. Leon, L. De Pillis, and L. G. De Pillis, *Linear algebra with applications*. Pearson Prentice Hall Upper Saddle River, NJ, 2006.

[131] V. V. Isaeva, N. V. Kasyanov, E. V. Presnov, *et al.*, "Topology in biology: singularities and surgery transformations in metazoan development and evolution," *Applied Mathematics*, vol. 5, no. 17, p. 2664, 2014.

[132] A. Ranicki, "An introduction to algebraic surgery," *Surveys on surgery theory*, vol. 2, pp. 81–163, 2000.

[133] S. Lambropoulou, S. Antoniou, and N. Samardzija, "Topological surgery and its dynamics," *arXiv preprint arXiv:1406.1106*, 2014.

[134] P. M. Alsing, H. A. Blair, M. Corne, G. Jones, W. A. Miller, K. Mischaikow, and V. Nanda, "Topological signals of singularities in ricci flow," *Axioms*, vol. 6, no. 3, p. 24, 2017.

[135] W. B. Lickorish, "A finite set of generators for the homeotopy group of a 2-manifold," in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 60, pp. 769–778, Cambridge University Press, 1964.

[136] B. Fischl, A. Liu, and A. M. Dale, "Automated manifold surgery: constructing geometrically accurate and topologically correct models of the human cerebral cortex," *IEEE transactions on medical imaging*, vol. 20, no. 1, pp. 70–80, 2001.

[137] A. Hoopes, J. E. Iglesias, B. Fischl, D. Greve, and A. V. Dalca, "Topofit: Rapid reconstruction of topologically-correct cortical surfaces," *Proceedings of machine learning research*, vol. 172, p. 508, 2022.

[138] S. Qiao, H. Wang, C. Liu, W. Shen, and A. Yuille, "Micro-batch training with batch-channel normalization and weight standardization," *arXiv preprint arXiv:1903.10520*, 2019.

[139] T. Salimans, D. P. Kingma, T. S. Openai, and D. P. K. Openai, "Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks," *Advances in Neural Information Processing Systems*, vol. 29, pp. 901–909, 2 2016.

[140] L. Huang, X. Liu, Y. Liu, B. Lang, and D. Tao, "Centered weight normalization in accelerating training of deep neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2803–2811, 2017.

[141] A. Virmaux and K. Scaman, "Lipschitz regularity of deep neural networks: analysis and efficient estimation," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[142] C. Anil, J. Lucas, and R. Grosse, "Sorting out lipschitz function approximation," in *International Conference on Machine Learning*, pp. 291–301, PMLR, 2019.

[143] M. Fazlyab, A. Robey, H. Hassani, M. Morari, and G. Pappas, "Efficient and accurate estimation of lipschitz constants for deep neural networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[144] C. Runkel, C. Etmann, M. Möller, and C.-B. Schönlieb, "Depthwise separable convolutions allow for fast and memory-efficient spectral normalization," *arXiv preprint arXiv:2102.06496*, 2021.

[145] C. Zhu, R. Ni, Z. Xu, K. Kong, W. R. Huang, and T. Goldstein, "GradInit: Learning to Initialize Neural Networks for Stable and Efficient Training," *Advances in Neural Information Processing Systems*, vol. 20, pp. 16410–16422, 2 2021.

[146] Y. N. Dauphin and S. Schoenholz, "Metainit: Initializing learning by learning to initialize," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[147] E. Rothwell and B. Drachman, "A unified approach to solving ill-conditioned matrix problems," *International Journal for Numerical Methods in Engineering*, vol. 2, pp. 609–620, 3 1989.

[148] J. Shen, "On the singular values of gaussian random matrices," *Linear Algebra and its Applications*, vol. 326, no. 1-3, pp. 1–14, 2001.

[149] E. Meckes, "The eigenvalues of random matrices," *arXiv preprint arXiv:2101.02928*, 2021.

[150] J. Fischmann, W. Bruzda, B. A. Khoruzhenko, H.-J. Sommers, and K. Życzkowski, "Induced ginibre ensemble of random matrices and quantum operations," *Journal of Physics A: Mathematical and Theoretical*, vol. 45, no. 7, p. 075203, 2012.

[151] J. A. Fischmann, *Eigenvalue distributions on a single ring.* PhD thesis, Queen Mary University of London, 2013.

[152] N. Konz, H. Gu, H. Dong, and M. A. Mazurowski, "The intrinsic manifolds of radiological images and their role in deep learning," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 684–694, Springer, 2022.

[153] A. Tausz, M. Vejdemo-Johansson, and H. Adams, "JavaPlex: A research software package for persistent (co)homology," in *Proceedings of ICMS 2014* (H. Hong and C. Yap, eds.), Lecture Notes in Computer Science 8592, pp. 129–136, 2014. Software available at `http://appliedtopology.github.io/javaplex/`.

[154] M. Rucco, R. Gonzalez-Diaz, M.-J. Jimenez, N. Atienza, C. Cristalli, E. Concettoni, A. Ferrante, and E. Merelli, "A new topological entropy-based approach for measuring similarities among piecewise linear functions," *Signal Processing*, vol. 134, pp. 130–138, 2017.

[155] D. Ali, A. Asaad, M.-J. Jimenez, V. Nanda, E. Paluzo-Hidalgo, and M. Soriano-Trigueros, "A survey of vectorization methods in topological data analysis," *arXiv preprint arXiv:2212.09703*, 2022.

[156] A. B. Lee, K. S. Pedersen, and D. Mumford, "The nonlinear statistics of high-contrast patches in natural images," *International Journal of Computer Vision*, vol. 54, no. 1-3, pp. 83–103, 2003.

[157] N. Vannieuwenhoven, "Condition numbers for the tensor rank decomposition," *Linear Algebra and its Applications*, vol. 535, pp. 35–86, 2017.

[158] C. Beltrán, P. Breiding, and N. Vannieuwenhoven, "The average condition number of most tensor rank decomposition problems is infinite," *Foundations of Computational Mathematics*, vol. 23, no. 2, pp. 433–491, 2023.

[159] D. Meller and N. Berkouk, "Singular value representation: A new graph perspective on neural networks," in *International Conference on Artificial Intelligence and Statistics*, pp. 3353–3369, PMLR, 2023.

[160] B. J. Stolz, J. Tanner, H. A. Harrington, and V. Nanda, "Geometric anomaly detection in data," *Proceedings of the national academy of sciences*, vol. 117, no. 33, pp. 19664–19669, 2020.

[161] J. von Rohrscheidt and B. Rieck, "Topological Singularity Detection at Multiple Scales," *arXiv preprint arXiv:2210.00069*, 2022.