Breast Lesion Classification from Bi-modal Ultrasound Images by Convolutional Neural Network

A thesis submitted to the
Department of Electrical and Electronic Engineering (EEE)
of
Bangladesh University of Engineering and Technology (BUET)

In partial fulfillment of the requirement for the degree of
Master of Science in Electrical and Electronic Engineering

Submitted by
Md. Shamim Hussain
(ID: 0417062229)

Department of Electrical and Electronic Engineering (EEE)
Bangladesh University of Engineering and Technology (BUET)
January 2019

The thesis titled “Breast Lesion Classification from Bi-modal Ultrasound Images by Convolutional Neural Network” submitted by Md. Shamim Hussain, Roll No.: 0417062229, Session: April, 2017, has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Master of Science in Electrical and Electronic Engineering on January 26, 2019.

BOARD OF EXAMINERS

1.	Dr. Mohammad Ariful Haque Professor Department of EEE, BUET, Dhaka-1205	Chairman
2.	Dr. Md. Shafiqul Islam Professor and Head Department of EEE, BUET, Dhaka-1205	Member (Ex- officio)
3.	Dr. Md. Aynal Haque Professor Department of EEE, BUET, Dhaka-1205	Member
4.	Dr. Mohammad Rakibul Islam Professor Department of EEE, Islamic University of Technology, Board Bazar, Gazipur-1704, Bangladesh.	Member (External)

CANDIDATE'S DECLARATION

It is hereby declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Shamim Hussain

Md. Shamim Hussain

DEDICATION

I dedicate this thesis to my respectable parents whose constant support has been the greatest source of inspiration in my life.

“Intelligence is the ability to adapt to change.”

-Stephen Hawking

Table of Contents

List of Figures

List of Tables

List of Abbreviations of Technical Symbols and Terms

US	Ultrasound B-mode
UE	Ultrasound Quasi-static Elastography
CADx	Computer Aided Diagnosis
ROI	Region of Interest
ML	Machine Learning
DNN	Deep Neural Network
CNN	Convolutional Neural Network
DL	Deep Learning
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

Acknowledgement

I would like to express my sincerest gratitude to my supervisor Dr. Mohammad Ariful Haque, Professor, Dept. of EEE, BUET for his continuous support, patience, motivation and immense knowledge. His guidance helped me all the time of the research and in writing of this thesis. He shared his knowledge with me in interpreting subject topics and also valued my way of thinking to synthesize those topics. His suggestions pushed me towards the direction of better thinking, his brilliant reviews refined me in working out my research problems, and his support gave me spirit to continue my work at the time of disappointment. I am extremely thankful and indebted to him for sharing his expertise and for offering his valuable guidance and encouragement.

This work of research is supported by Higher Education Quality Enhancement Project, CPSF – 096 for which I served as a research engineer. I would like to thank my Sub-Project Manager Dr. Md. Kamrul Hasan, Professor, Dept. of EEE, BUET for giving me the opportunity to be a part of this project and also for his valuable guidance and continuous support.

I would also like to thank Dr. Sharmin R. Ara, Assistant Professor, Dept. of EEE, East West University and Senior Research Assistant, DSP Research Lab for her assistance and valuable insights which greatly helped me in my research.

I am grateful to the Head of the department of Electrical and Electronic Engineering, BUET, for all time lab facilities in the DSP Research Lab which have been vital in performing this research.

I would like to deeply thank my parents and my elder sister for the unceasing encouragement, support and attention they provided me with. They always believe in me and inspire me to work for good. I am grateful to them for their never-ending support and inspiration.

I also place on record, my sense of gratitude to one and all, who directly or indirectly, have assisted in this thesis work.

Abstract

Ultrasound imaging provides a convenient and easily accessible means for breast cancer detection. Quasi-static Elastography is a very useful imaging modality which can be combined with conventional B-mode imaging to implement a non-invasive lesion classification system. Computer Aided Diagnosis (CADx) can provide an objective opinion alongside the radiologist's diagnosis to increase the reliability of such a system. Traditionally, CADx based systems have relied on statistical features derived from the morphology and/or texture of the lesions which are fitted to a machine learning model– to classify the lesions into either malignant or benign category. The performance of this approach is highly dependent on the selection of an appropriate set of features which is found to be a difficult task. The segmentation process required for feature extraction is time-consuming and introduces subjectivity in the classification process.

Although a Computer Aided Diagnosis system based on object recognition techniques by deep Convolutional Neural Networks (CNN) holds the possibility of real-time lesion classification directly from images, this approach faces the difficulty of gathering enough data for training such a network from scratch. In this work, we investigated the use of transfer learning to alleviate this difficulty. We show that a CNN trained on ImageNet can be used as a starting point to design a deep CNN which can be trained easily on a small dataset of lesions. Also, we integrate both ultrasound B-mode and elastography images in a single unified network for lesion classification that can be trained end-to-end. On a dataset of 217 clinically proven cases, our approach achieves >91% accuracy, >88% sensitivity and >92\% specificity.

Apart from achieving satisfactory classification performance on our dataset, the proposed method shows indications of improvement with increasing dataset size. This approach, which is based on transfer learning, is applicable to a dataset of any reasonable size and also maintains the scalability and flexibility of deep learning. Furthermore, this method is completely objective, requires no segmentation of the lesion or ROI selection and is suitable for a real-time classification system. Additionally, we show that classification results can be further improved by multi-task learning of relevant tasks or inclusion of additional qualitative features of the lesions.

Chapter 1
Introduction and Motivation

In this chapter we first introduce the concern of breast cancer among women worldwide and in the developing countries. Then we show how a two-in-one ultrasound imaging system can provide cheap and reliable screening facilities for early detection of breast cancer. Then we introduce Computer Aided Diagnosis as a simple and inexpensive solution to improving the reliability of breast cancer diagnosis and discuss the previous works done thereupon. Later we discuss about our motivation for designing a breast lesion classification system based on convolutional neural network and the objective of this thesis.

1.1 Background

Breast cancer is the most prevalent form of cancer in women worldwide and is the cause of second highest mortality rate from cancer [1]. According to [2], it is estimated that around 508,000 women died worldwide in 2011 alone due to breast cancer across the world. Nearly 1.7 million new cases were diagnosed in 2012 [3]. Breast cancer has been deemed as the number one cancer killer among women, beating cervix cancer in the USA [4]. Although, breast cancer rates are higher among women in the developed countries, the rate of mortality from breast cancer in the developing countries is almost twice that of developed countries [5]. The higher rate of mortality from breast cancer in underdeveloped and developing countries can be largely attributed to lack of adequate screening facilities and late detection. Apart from prevention, early detection is the most import measure in reducing mortality rates from breast cancer. A lot of research has been performed to develop and test different forms of imaging for the detection and diagnosis of breast cancer. Most noticeable of them being mammography, ultrasound and Magnetic Resonance Imaging (MRI). None of these imaging modalities can independently ensure reliable detection of breast cancer. Although mammography is the primary modality of imaging for diagnosis of breast cancer, it is age restrictive and less accessible to the patients in developing countries. Ultrasound is the cheapest and the most easily accessible modality of imaging for cancer detection which can be used for non-invasive diagnosis. However, sensitivity, specificity and accuracy from conventional B-mode ultrasound image alone is found not to be satisfactory [6], [7]. Other modalities of ultrasound imaging, such as elastography can help improve the classification performance. In recent years quasi-static ultrasound elastography [8] and shear-wave elastography (SWE) [9] alongside ultrasound B-Mode images have been used in the classification of tumors from ultrasound images.

1.2 Ultrasound Imaging for Breast Cancer Detection

Ultrasound imaging has found numerous medical applications in gynecology, pulmonology, ophthalmology, urology and many other areas because of its simplicity, versatility, mobility, high resolution, non-ionizing nature and low cost. Ultrasound imaging can play a major role in reducing deaths from breast cancer due to its accessibility and low cost. Also unlike mammography, ultrasound imaging is not age restrictive. According to a recent study [10], mammography does not contribute to the reduction of mortality rate of cancer patients. The tumor for dense breast is hard to identify with mammogram [11] and this modality is not effective for detection and diagnosis of breast tumor below a certain age [12]. Moreover, in the context of developing countries, MRI and mammography may not be easily accessible. On the other hand, because of the fact that ultrasound examination is age independent and high-resolution imaging is possible using the state-of-the-art ultrasound scanners due to the availability of the high-frequency probes, it has become an integral part of evaluation and diagnosis of breast tumors in recent years [13]. Conventional ultrasound imaging portrays the tissue attenuation whereas ultrasound based elastography measures the relative stiffness by comparing a region of interest in the target lesion with the fatty tissue in the breast. As discussed further in the following subsections, together these two modalities can provide reliable screening of breast lesions and thus reduce the rate of death and also unnecessary biopsies from breast cancer, especially in the developing countries where more expensive imaging systems are not always available to the patients.

We briefly will discuss two ultrasound imaging modalities which are relevant to our work.

1.2.1 B-mode Imaging

The most commonly used ultrasound B-mode (Brightness Mode) images are formed by taking envelope of the Radio Frequency (RF) data and interpreting the amplitudes as brightness. The complete imaging system includes several preprocessing and post processing steps to produce a good quality 2D image. In the context of breast cancer detections, B-mode ultrasound can be readily used to view a tumor/lesion because high frequency ultrasound can easily penetrate soft tissue and produce high resolution images which can point out different characteristics or features of the tumor/lesion. However, it is often used alongside other modalities, like mammography to increase the reliability of the diagnosis.

1.2.2 Quasi-static Elastography

Elastography [14] (also known as strain imaging) is an imaging modality which is used for visualizing the stiffness distribution in the soft tissue. It is useful because of the fact that tissue stiffness has deep correlations with the pathology change of the tissue. Even before the advent of tissue elastography, physicians used manual palpitation to estimate the stiffness of tissue for diagnostic purpose. Elastography provides a means of a more accurate and objective quantification of the stiffness properties of the tissue, even for internal organs like liver. So, Elastography is often used alongside conventional imaging modalities like B-mode ultrasound imaging to facilitate a more reliable diagnosis. The elastography techniques can be loosely categorized according to how the stress is applied and how the tissue deformation is measured. For example, stress can be applied externally by means of the probe [14] or internally via acoustic radiation force [15] (ARFI). On the other hand, tissue deformation can be measured using MRI or ultrasound.

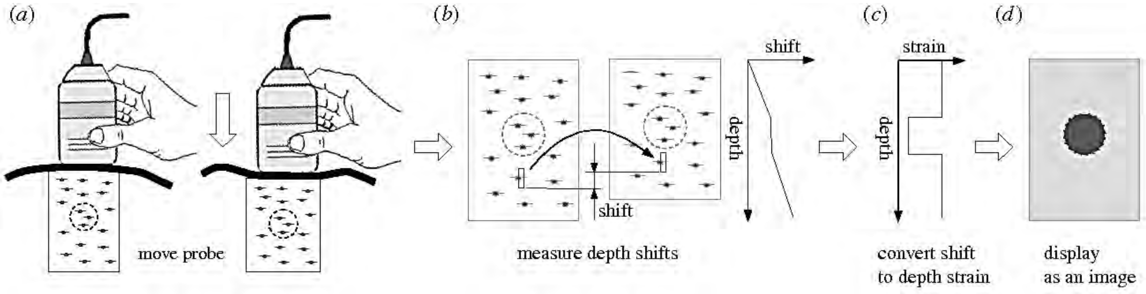
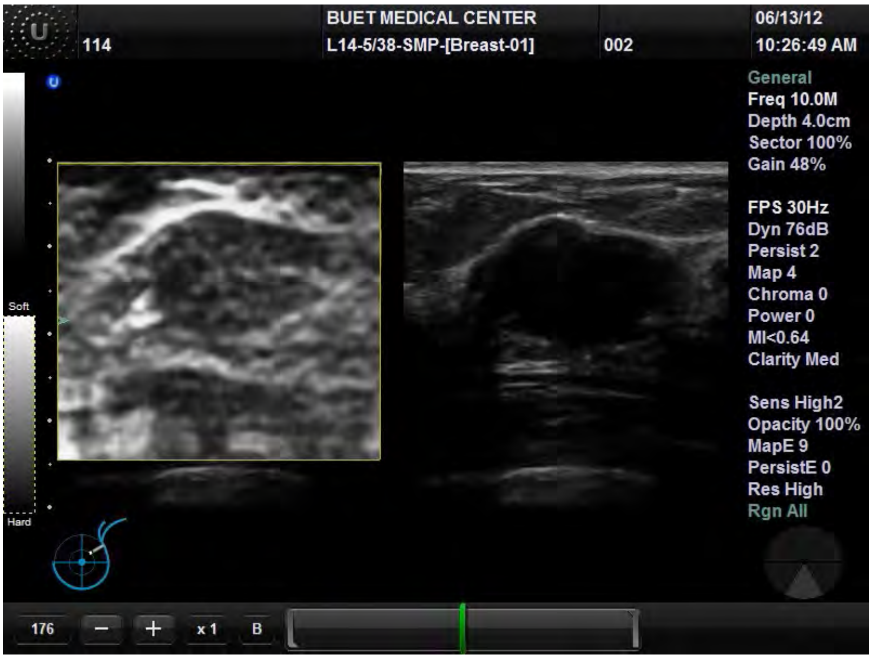


Figure 1.1 Overview of the quasi-static elastography process [16]. (a) The anatomy is scanned with a conventional ultrasound probe, which is moved very slightly up and down (much less than is shown here). In this example, the dashed circle is stiffer than the surrounding tissue.

In quasi-static ultrasound elastography (Figure 1.1), the deformation is caused by applying external pressure on the tissue surface with the transducer, and estimated using ultrasound backscatter echoes. In quasi-static elastography, the time domain post-compression RF signal is modeled as a compressed and delayed version of the pre-compression RF signal. To ascertain the displacement between these two signals which is eventually used to calculate strain, the correlation of these pre- and post-compression signals are analyzed. Though, there are more quantitative techniques based on shear wave speed [17], quasi-static elastography is a qualitative technique that is unsuitable for measuring absolute tissue stiffness. However, it is possible to relate stiffness of a lesion to that of the background tissue. In addition, it has a high spatial resolution, is real time, and does not require any modifications to conventional ultrasound hardware.

1.2.3 Two-in-one Imaging System for Breast Cancer Diagnosis

As mentioned before, it is hard to predict the malignancy of a breast lesion reliably from only conventional ultrasound B-mode images. However, in addition to B-mode imaging, quasi-static elastography can provide vital information regarding the stiffness distribution of a lesion. Other elastography imaging systems like Shear Wave Elastography (SWE) could also be used for this purpose, but quasi-static elastography has the convenience that no additional hardware is required to form these images, only a software implementation of the elastography algorithm, which makes the diagnostic process cheaper and more easily available to the patients in the developing countries. Quasi-static strain imaging can be implemented in any ultrasound imaging system capable of producing a B-mode image and the two imaging modalities can be viewed side-by-side in real time which would provide the radiologists a better insight about the pathology of the lesion.


BUET MEDICAL CENTER
114 L14-5/38-SMP-[Breast-01] 002
06/13/12
10:26:49 AM

General
Freq 10.0M
Depth 4.0cm
Sector 100%
Gain 48%

FPS 30Hz
Dyn 76dB
Persist 2
Map 4
Chroma 0
Power 0
MI<0.64
Clarity Med

Sens High2
Opacity 100%
MapE 9
PersistE 0
Res High
Rgn All

Figure 1.2 A two-in-one imaging system whereby both ultrasound B-mode and quasi-static elastography can be viewed side by side

Strain images produced by elastography reveals the stiffness distribution of the lesion. Malignant lesions are known to be stiffer and have irregular stiffness distributions [18]. On the other hand, benign lesions are usually less stiff and the stiffness is uniformly distributed. This knowledge of elasticity, i.e. stiffness can be vital in distinguishing between malignant and benign lesions. Although in case of quasi-static elastography, absolute value of stiffness i.e. elastic modulus cannot be measured, the strain within a lesion can be compared to strain of the healthy surrounding tissue to produce a quantitative measure called strain ratio, which is found to be a very effective in distinguishing malignant lesions from benign ones. The relative sizes of the lesion in the two imaging modalities can also provide indications regarding malignancy. Malignant lesions are usually surrounded by stiffer tissue and thus occupy a slightly larger area in the elastography than in the B-mode image. Another important clue that elastography can provide is the presence of necrosis, i.e. dead tissue within a malignant lesion. Due to all these advantages a two-in-one imaging system can provide a much more reliable diagnosis than that using only B-mode elastography.

1.3 Computer-Aided Diagnosis (CADx) for Breast Cancer Detection

Reliability is a big issue when it comes to medical diagnosis. Reliability of a diagnostic process, for example in our case, the detection of breast cancer from ultrasound images can be increased by reducing inter-observer variation. Often, double-reading whereby the same lesion is examined by two different radiologists can be used to reduce this variation and thus produce a more reliable and objective diagnosis [19]. But this requires extra work and time and incurs more cost. So, rather than taking opinion from a second radiologist, automated Coputer-Aided Diagnosis system could be used to produce a second opinion free of cost and thus support the diagnostic procedure. Computer-Aided Diagnosis (CADx) is a computerized procedure to provide a second objective opinion for the assistance of medical image interpretation and diagnosis. One of the major CADx applications is the differentiation of malignancy/benignancy for tumors/lesions. Several studies have suggested that the incorporation of the CADx system into the diagnostic process can improve the performance of image diagnosis by decreasing inter-observer variation [20], [21] and providing the quantitative support for the clinical decision like biopsy recommendations [22]. Specifically, the CADx systems were shown to be effective to assist the diagnostic workup for the reduction of unnecessary false-positive biopsies [22] and thoracotomy [22]. However, it should be emphasized the objective of a CADx system is to supplement/support the diagnosis by human radiologists rather than replace them.

1.4 Feature Based Breast Lesion Classification

Traditionally, a CADx system is known to be comprised of three steps – feature extraction, feature selection and classification. Engineering of effective feature extraction step for each specific problem is regarded as the one of the most important issues in CADx. Extraction of discriminative features could potentially ease the latter steps of feature selection and classification. Traditionally, CADx systems for breast lesion classification have relied on statistical features derived from the morphology [23] and/or texture [24] of the lesions which are fitted to a machine learning model [8], [25], [26] to classify the lesions into either malignant or benign category. The extraction of effective features is a complicated task that involves many image processing steps. These image processing steps include morphological feature computing [20], [27], [28], which is still difficult to solve [29], and image decomposition [30], [31], followed by statistical summaries and presentations for the calculation of textural features[32]. In feature integration by classifier, the widely used techniques are based on the KNN (k-nearest neighbor) method [33], [34], LDA (linear discriminant analysis) [35], [36] and SVM (support vector machines) [37], [38].

1.4.1 Breast Lesion Classification from Ultrasound Features

Although, ultrasound B-mode is the most accessible imaging modality, the sensitivity, specificity and accuracy of conventional B-mode US alone is not yet satisfactory as shown in a recent study (88.5%, 42.9%, and 53.6%, respectively) [6] based on the Breast Imaging-Reporting and Data System (BI-RADS) specified criteria (developed by the American College of Radiology (ACR 2003)). According to [7], six B-mode ultrasound features (e.g., orientation, undulation, angularity, average gradient, gradient variance and intensity variance) based classification resulted in sensitivity, specificity, and accuracy of 70.6%, 89.4%, and 82.3%, respectively. In [23], a morphometric parameter-based (e.g., form factor, roundness, aspect ratio, convexity, solidity) breast tumor classification scheme with 88.89% sensitivity, 92.50% specificity and 90.95% accuracy has been reported.

As B-mode alone is not adequate to avoid unnecessary biopsies, ultrasound elastography has become popular in the diagnosis of breast tumors [39]. The reported sensitivity are 100%, 83.8%, 70.1%, specificity, 73.8%, 87.6%, 93.0%, and accuracy, 80%, 86.2%, 87.1%, for the UE features, `area ratio(AR)' [6], `strain ratio (SR)' [7] and `elasticity scoring' [40], respectively.

To further improve the accuracy of diagnosis, the bi-modal imaging, i.e., combined B-mode and elastography has been considered in the previous studies [6]–[8], [41]. The efficacy of combined B-mode and elastogram has been verified in [41], [42], where the conventional ultrasonic images of invasive ductal carcinoma, fibroadenoma, hematoma, malignant lymph node and cyst of breast are compared with elastogram, based on the lesion size, stiffness, and contrast measured from the strain images. An overall sensitivity of 95% and specificity of 85% were recorded for differentiating malignant and benign breast lesions. In [6], the reported sensitivity, specificity and accuracy of the combined B-mode and elastogram have been found to be 88.50%, 78.60%, and 80.90%, respectively, and 95.60%, 87.60%, and 90.60%, respectively, in [7]. [8] used an EMD-DWT based transformed domain reduced feature space technique and reported a accuracy, sensitivity and specificity of 98.21%, 97.93% and 98.01% respectively.

Apart from morphological features discussed above, textural features from B-mode images can also be used to classify lesions. The ultrasound textures can be regarded as the regional intensity distribution features characterizing the scattering properties of ultrasonic RF echoes in B-mode images of breast tissues [43].Traditionally, the texture feature descriptors are calculated using a variety of statistical, structural, spectral and model based techniques, such as auto-covariance coefficients [44], [45], gray level co-occurrence matrix (GLCM) [46], [47], block difference of inverse probabilities [45], [46], block variation of local correlation coefficients [46], fractal dimension [46] and complexity curve [47], [48]. These methods can represent the statistical characteristics of gray level distribution in certain region of interest (ROI). [24], [25] used shearlet based textural features which showed better classification performance. [25] used stacked deep polynomial network for classification from textural features and reported accuracy, sensitivity and specificity of 92.40%, 92.67% and 91.36% respectively.

1.4.2 Limitations of the Feature Based Classification Approach

Although most of CADx systems proposed so far have relied on feature based classification, this approach has some major limitations. The extraction of meaningful features is highly dependent on the quality of each intermediate result in the image processing steps [27], [28], [32], which often requires recursive trial and error to obtain satisfactory results. Thus, it is time-consuming and very difficult to design and tune the overall performance of a conventional CADx framework to get a satisfactory result because many image processing steps need to be considered at the same time. Also, the performance of this approach is highly dependent on the set of selected features. It is found to be difficult to choose an optimal set of features from a wide range of features having different performance levels [8]. Also, to derive morphological features a lesion boundary needs to be identified which can be problematic if the boundary of a lesion is not distinguishable from the ultrasound image. If automated segmentation of ultrasound images is used, the performance level of the whole process depends on the segmentation algorithm used. Manual segmentation is more accurate, but is time-consuming and introduces subjectivity in the classification process. Although, textural features do not require accurate segmentation, the type of image decomposition method used dictates the performance these features which are not guaranteed to be best/optimal for the problem. Moreover, the scalability of such feature based approach is not proven for bigger datasets i.e. whether they would be able to learn or take advantage of the diverse information present in a bigger dataset has not yet been studied.

1.5 Computer Aided Diagnosis Using Deep Neural Networks

Due to all the aforementioned inconveniences of feature based classification approach, there has been a growing interest in using Deep Neural Networks (DNN) for classification of tumors/lesions directly from images without any intermediate feature engineering [9], [49], [50]. A Deep Neural Network (DNN) [51] consists of multiple layers of neurons stacked on top of one another. Higher layers process the outputs from lower layers and thus form a more abstract representation of the input data. Multiple layers of abstraction allow the network to extract complex features inherently. Two broad classes of deep architectures, namely convolutional (CNN) [52] and recurrent (RNN) [53] networks stand out as most successful ones for solving diverse classes of problems in different fields.

DNNs use multiple layers of processing to inherently extract task-relevant features directly from the input data [54]. This provides immense flexibility to this method and allows similar Deep Learning (DL) approaches to be applied to a variety of machine learning tasks [51], if sufficient data is available. For the problem of lesion classification, deep neural networks can alleviate the difficulty of engineering features. The classification algorithm can be applied directly on ultrasound images, without any need for segmentation or image processing steps. This approach is also scalable when more data becomes available as the network size can be increased to take advantage of the information available in big datasets, resulting in graceful increase in classification performance with size of the training dataset.

Deep Neural Networks have found numerous applications and revolutionized various machine learning tasks such as object recognition, audio classification and segmentation, speech recognition, machine translation and robotics. Apart from these applications, deep learning methods have been introduced to medical imaging with promising results in various medical applications, such as the computerized prognosis or diagnosis for Alzheimer's disease [55]–[57] and mild cognitive impairment [58], organ segmentation [59] and detection [60] ultrasound standard plane selection [61], tissue classification in histological and histopathological images [62], [63] and knee cartilage segmentation [64], among others.

1.5.1 Convolutional Neural Networks

In recent years deep Convolutional Neural Networks (CNN) [51], [65] have been shown to produce outstanding classification results on big labeled datasets such as ImageNet [66] while outperforming feature based learning by a huge margin. CNNs are a class of feed-forward neural networks, which implement one or more convolutional layers. A convolutional layer consists of a set of filters, each of which shares its weights across spatial/temporal dimensions of the input. This ensures a shift-invariant approach which is effective for feature extraction from a broad range of data types such as image, audio and text.

CNNs have achieved remarkable success in image classification [67], object recognition [68] and segmentation [69] . However, training deep CNNs from scratch usually requires big datasets which poses a problem for medical applications where there is a scarcity of data [70]. So, the deep learning approaches that operate directly on ultrasound images need to either use a big dataset or limit the network size to prevent overfitting. However, as we will discuss later, transfer learning [71] can be used to alleviate this problem to a considerable extent.

1.6 Related Works: Classification of Lesions Directly from Ultrasound Images

The interest of researchers to apply deep neural networks directly on ultrasound images is fairly recent. Cheng et al. [28] first used Stacked Denoising Auto-Encoder (SDAE) to pretrain a 2-hidden-layer deep neural network on B-mode breast lesion images (flattened to an array of pixel-wise intensity values), which was consequently fine-tuned for lesion classification. Their dataset contained 275 benign and 245 malignant lesions and they achieved an accuracy of 82.4%. Zhang et al. [9] used a Pointwise Gated Restricted Boltzman Machine to extract relevant features from Shear Wave Elastography images of lesions and applied a conventional RBM and then a Support Vector Machine (SVM) on the extracted task-relevant features to classify lesions. They reported a classification accuracy of 93.4% on a dataset of 135 benign and 92 malignant lesions. Although these methods used DNNs to classify breast lesions, the networks they used had a densely connected architecture, rather than a convolutional architecture. Only Han et al. [49] have reported to be able to successfully train a deep convolutional neural network, GoogleNet [72] on ultrasound B-mode images. However, this method utilized a very big dataset of lesions, containing 4254 benign lesions and 3154 malignant lesions. They reported a classification accuracy of 91.23%.

Apart from breast lesions, deep neural networks have been used to classify liver lesions from US images as well [73]–[75]. Among these works, the findings by Meng et al. [73] is highly relevant to our work as they utilized transfer learning technique to overcome the limitation of a small dataset and to successfully train a deep neural network (VGGNet + FCNet) end-to-end to classify liver fibrosis from B-mode images. We independently found out the utility of transfer learning for our task i.e. detecting malignancy in bi-modal US images of breast lesions.

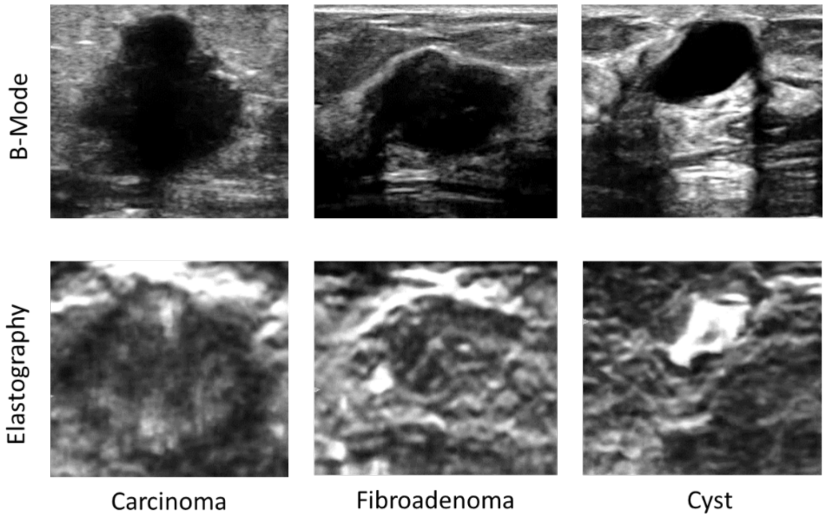


Figure 1.3 Lesion classification from visual clues. The malignant (carcinoma) lesion has a vertical orientation (B-mode) and shows presence of necrosis (elastography). The fibroadenoma lesion has an elliptic shape (B-mode) and has almost homogenious strain distribution (elastography). The cyst lesion has distinctive capsulation (B-mode) and filled with fluid (elastography).

1.7 Motivation

For image classification and object detection Convolutional Neural Networks work analogous to the human eye in that, they look for visual clues and form representations of different structures present in the image such as edges, shapes and textures. Radiologists have used visual clues such as vertical orientation, capsulation, posterior acoustic shadowing, area ratio etc. to classify lesions (Figure 1.3). Although these clues are often quantified in the form of quantitative morphological and textural features, this is an artificial processing step that could introduce bias in the classification process. So, instead of engineering features, it is more intuitive to train a model to learn visual clues directly from the images. This process is similar to how a radiologist would learn to classify lesions, except a machine learning algorithm would be much more accurate, less biased and more discerning. So, it seems very intuitive to train a convolutional neural network to classify breast lesions directly from ultrasound images.

However, the main difficulty in training a deep CNN on ultrasound images of breast lesions is the unavailability of a big and diverse dataset of labeled ultrasound images (such as ImageNet for object recognition). We used transfer learning [71] to circumvent this problem. Transfer learning is the process of transferring required knowledge that is common between two machine learning tasks, from a source task with sufficient available data to a similar/related target task where there is a shortage of available data (Figure 1.4). This, in turn, relaxes the requirement for a higher amount of data and also facilitates more robust learning from a small dataset for the target task.

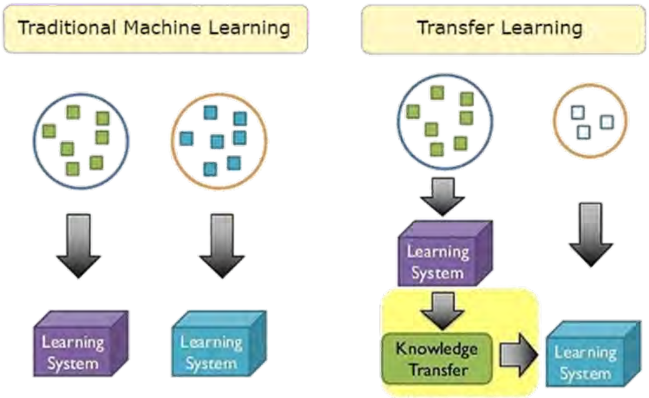


Figure 1.4 Comparison between transfer learning and traditional machine learning. In transfer learning, scarcity of trainable data is handled by first training a model on a relevant source task with adequate training data, and then transfering the knowledge learned to the target task which is used as a starting point for training on the target task. This in turn facilitates learning from a smaller dataset on the target task.

Deep CNNs trained on a diverse labeled dataset such as ImageNet are known to learn rich feature representations for visual recognition tasks [76], [77]. Features produced by these deep CNNs are found to be flexible enough to be transferred among varieties of image classification/ segmentation [78] tasks. Intermediate feature maps from CNNs trained on ImageNet have been used for tasks such as perceptual evaluation [79], style transfer [80] and also for classification of audio from spectral representations [81], [82]. We are motivated to investigate the utility and the versatility of these features for lesion classification in ultrasound images.

In addition, we want to include both ultrasound B-mode and elastography images in the classification process, because, as mentioned in previous sections, it has proven to improve classification performance of both human radiologists and feature based machine learning models. So, we can expect a performance increase in case of classification by convolutional neural network as well by using both B-mode and quasi-static elastography images for the classification process.

1.8 Objective of This Thesis

The objectives of this thesis are as follows –

- To design a convolutional neural network based system for classification of breast lesions into malignant and benign categories that avoids feature engineering and feature selection, and operates directly on images.
- To investigate the use of transfer learning to train a deep convolutional neural network on a small dataset of breast lesions.
- To combine two imaging modalities – ultrasound B-mode and quasi-static elastography in a single end-to-end trainable model for higher classification performance.
- To classify lesions directly from image frames/video obtained from the ultrasound machine, in a single step without any need for segmentation or ROI selection. The only requirement would be that the lesion be observable in the selected frames/video. This would make the proposed approach suitable for a real-time classification system.
- To make the classification model scalable in the sense that, as the amount of available data increases, the network can be fine-tuned further to achieve better results and thus fully utilizing the diverse information present in bigger datasets.
- To facilitate the inclusion of additional information when available in the classification/ training process.

1.9 Thesis Layout

- In Chapter 2 we will present the conceptual and theoretical background of convolutional neural networks.
- In Chapter 3 we will discuss the methodology of our work – the nature of the data collected, how they are collected, processed and also about the proposed design of the CNN architecture and the rationale and considerations regarding this design.
- In Chapter 4 we will discuss how we evaluated the learning algorithm, i.e. how its performance was measured. Then we will present the results obtained in our experiments and their implications.
- Finally, in chapter 5 we end our discussion with some conclusive remarks and point out the prospects of future work.

Chapter 2
Deep Learning and Convolutional Neural Networks

In this chapter, first we present a brief discussion on supervised machine learning and also the important concepts of model capacity, overfitting and underfitting which are very relevant for machine learning tasks on small datasets. Then we introduce deep neural network as a supervised learning algorithm. Next, we discuss convolutional neural network which is the type of architecture that plays the central role in the proposed model (discussed in Chapter 3). Next we briefly discuss ways of training and regularizing deep neural networks in the context of our work. Finally, we discuss transfer learning in convolutional neural networks and its utility.

2.1 Machine Learning

Machine Learning (ML) is the study of algorithms and statistical models related to data-driven learning. It is often thought as a sub-field of Artificial Intelligence (AI) that facilitates learning from experience i.e. data rather than by explicit programming by human. Here learning is defined as follows – a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T , as measured by P , improves with experience E.” Here, T is usually some desired outcome from the model such as classification, regression, anomaly detection or clustering; E is the training data and P is some objective function such as mean squared error or cross-entropy. Machine learning facilitates solving complicated task-specific problems that are too difficult or cumbersome to solve for a human programmer by hand. Machine learning is said to have 3 main branches – supervised learning, unsupervised learning and reinforcement learning.

2.1.1 Supervised Learning

A supervised learning task involves learning to predict a desired output given some input. The training data contains both inputs and desired outputs for each data-point. The objective of the learning algorithm is to predict the outputs for new data (i.e. outside training data) as accurately as possible. So, a supervised learning algorithm must be able to generalize to data it has never seen before, rather than only performing well on the training data. Two of the well-known supervised learning tasks are regression and classification. In case of regression a numerical value is predicted by the algorithm. On the other hand, in case of classification a class label is predicted. Support Vector Machine, K Nearest Neighbors, Decision Trees, linear regression and classification algorithms are some well-known supervised learning algorithms. Feed forward neural networks are also primarily used for supervised learning tasks. However, different variants of neural networks in general (like auto-encoders, Restricted Boltzmann Machine, auto-regressive models etc.), have been used for unsupervised, semi-supervised learning, generative modeling and density estimation as well.

2.2 Capacity, Overfitting and Underfitting

As mentioned before, a machine learning algorithm must perform well not only on the training set, but also on new data points. So, it has two objectives-

i. Reduce error on the training set
ii. Reduce the difference between training and test error

Objective (i) is often explicitly accomplished by an optimization algorithm, however, objective (ii) cannot be accomplished explicitly. It requires that we make some assumptions about the data generating process. According to statistical learning theory, we can make such assumptions given that that the examples in each dataset are independent from each other, and that the train set and test set are identically distributed, drawn from the same probability distribution as each other. These assumptions in turn dictate how flexible our choice of the model is, which is known as the capacity of the model. Informally, a model's capacity is its ability to fit a wide variety of functions. It may depend one various factors such as the choice of the model, optimization tactics, regularization and constraints. In case of deep neural networks, often the size, i.e. the number of parameters is taken as an indicator of the capacity. However, in practice, the estimation of capacity is much more complicated.

The two objectives mentioned before often are at odds with each other, i.e. if we try to reduce the training error excessively, we may end up increasing the generalization error. The extremes of the two objectives result in two unwanted phenomena know as overfitting and underfitting. Underfitting occurs when the model is not able to obtain a sufficiently low error value on the training set. Models with low capacity may struggle to fit the training set. Models with high capacity can overfit by memorizing properties of the training set that do not serve them well on the test set. Figure 2.1 demonstrates the effect of model capacity and the phenomenon of overfitting and under fitting for simple polynomial regression.

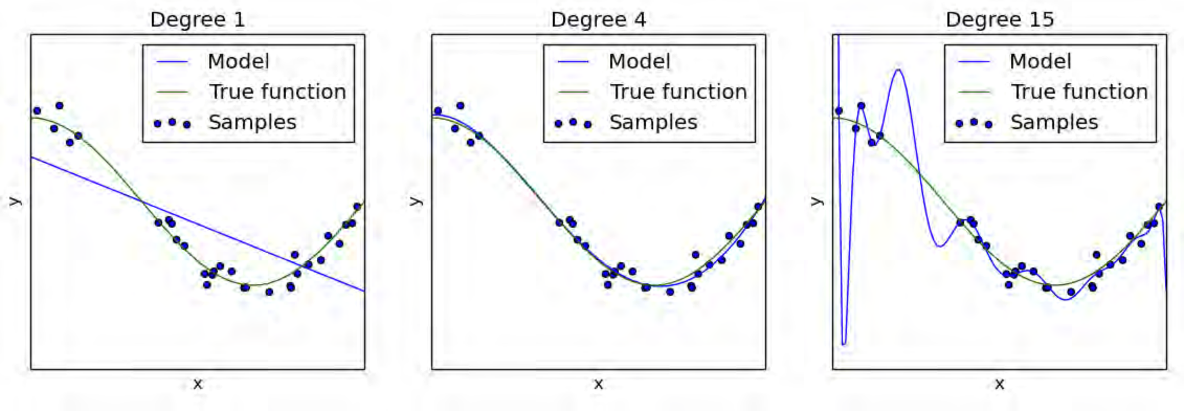


Figure 2.1 We fit three models to an example training set. (Left)A linear function fit to the data suffers fromunderfitting—it cannot capture the curvature that is present in the data. (Center). A polynomial of degree 4 fit to the data generalizes well to unseen points. It does not suffer from a significant amount of overfitting or underfitting. (Right)A polynomial of degree 9 fit to the data suffers fromoverfitting.

2.3 Deep Neural Networks

2.3.1 Artificial Neural Networks

A neural network is a computation model loosely based on the biological neural networks in animal brains. It consists of individual signals/variables called neurons connected by weighted connections, called edges. Typically, neurons are organized in layers. Different layers may perform different kinds of transformations on their inputs. The activity of the neurons are modeled by applying a nonlinearity to the weighted sum of its inputs. A neural network can learn to solve a problem by adjusting its weights based on the training data.

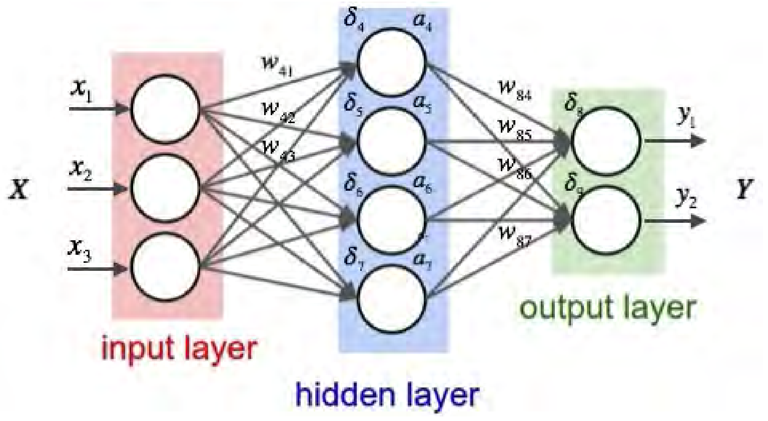


Figure 2.2 A neural network with a single hidden layer.

2.3.2 Deep Feedforward Neural Network

A Deep Neural Network (DNN) consists of one or more hidden layers (Figure 2.2) which mediate the connection among the input and the output layers. Each hidden layer is connected to the previous one and to the next one. The later (deeper) layers process the outputs of the previous layers.

The goal of a feedforward network is to approximate some function f^*. For example, for a classifier, $y = f^*(x)$ maps an input x to a category y. A feedforward network defines a mapping $y = f(x;\, \theta)$ and learns the value of the parameters θ that result in the best function approximation. These models are called feedforward because information flows through the function being evaluated from x, through the intermediate computations used to define f , and finally to the output y. There are no feedback connections in which outputs of the model are fed back into itself. When feedforward neural networks are extended to include feedback connections, they are called recurrent neural networks

Deep feedforward networks perform a chain of transformations (i.e. $f \equiv f_n \circ \ldots \circ f_2 \circ f_1$) on the input to produce the outputs. In a densely connected layer each transformation is simply an affine transformation followed by a non-linearity.

$$\delta_{i+1} = \sigma(W_i \delta_i + b_i) \tag{2.1}$$

Where, δ_i is the input of the layer, and δ_{i+1} is the output or activation of the layer of the layer. W_i is known as the weight matrix and b_i is known as the bias vector. σ is the non-linear activation function.

2.4 Convolutional Neural Network

Convolutional Neural Networks (CNN) are a type of deep feedforward neural network specialized for handling data that has high degree of temporal or spatial correlation, i.e. 1D time-series data like audio, text and 2D spatial data like image. A convolutional neural network uses convolutional layers containing filters that capture spatially distributed features more effectively and efficiently. Apart from convolutional layers a CNN may include pooling layers to achieve shift invariance. Convolutional neural networks have been tremendously successful at speech and image recognition, segmentation, image and audio generation, machine translation and even at playing games by reinforcement learning.

In a CNN the input image is processed by multiple convolutional layers that gradually form more abstract representations of the image. This is partially inspired by the visual cortex in the animal brain. Lower layers learn representations of basic primitives of the image like edges and textures. Higher layers connect these primitives to form representations of higher level primitives like shapes and eventually more complex parts like human eyes, noses and lips and finally for complete human faces.

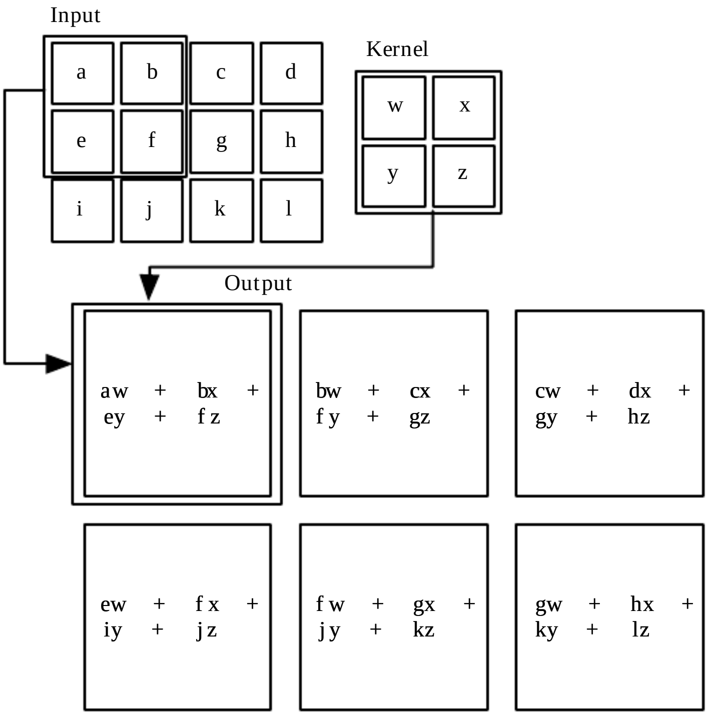


Figure 2.3 An example of 2-D convolution without kernel-flipping. We draw boxes with arrows to indicate how the upper-left element of the output tensor is formed by applying the kernel to the corresponding upper-left region of the input tensor.

2.4.1 Convolution

Convolutional networks use convolution in place of general matrix multiplication in their convolutional layers. Convolutional for 1D discrete time signal is defined as –

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(a-t) \qquad (2.2)$$

Where, x is the input and w is the kernel which are both functions of t. However, in practice the input is 2D and usually finite and the kernel a multidimensional array of parameters that are adapted by the learning algorithm. And the operation is implemented as correlation rather than convolution (Figure 2.3). The mathematical expression of this operation is given by -

$$S(i,j) = \sum_{m}\sum_{n} I(i+m, j+n)K(m,n) \tag{2.3}$$

Where I is the input channel, K is the filter, known as “kernel” and S is the output channel. In practice there are multiple input and output channels, where each output channel accumulates (adds) contributions from all input channels.

Discrete convolution can be viewed as multiplication by a matrix. However, the matrix has several entries constrained to be equal to other entries. For example, for univariate discrete convolution, each row of the matrix is constrained to be equal to the row above shifted by one element. This is known as a Toeplitz matrix. In two dimensions, a doubly block circulant matrix corresponds to convolution. In addition to these constraints that several elements be equal to each other, convolution usually corresponds to a very sparse matrix. This is because the kernel is usually much smaller than the input image.

2.4.2 Convolutional Layers and Motivation

Convolution leverages three important ideas that can help improve a machine learning system: sparse interactions, parameter sharing and equivariant representations. Moreover, convolution provides a means for working with inputs of variable size. We now describe each of these ideas in turn.

Sparse interactions: Traditional neural network layers use matrix multiplication by a matrix of parameters with a separate parameter describing the interaction between each input unit and each output unit. This means every output unit interacts with every input unit. Convolutional networks, however, typically have sparse interactions (also referred to as sparse connectivity or sparse weights) (Figure 2.4). This is accomplished by making the kernel smaller than the input. For example, when processing an image, the input image might have thousands or millions of pixels, but we can detect small, meaningful features such as edges with kernels that occupy only tens or hundreds of pixels. This means that we need to store fewer parameters, which both reduces the memory requirements of the model and improves its statistical efficiency. It also means that computing the output requires fewer operations.

Parameter Sharing: Parameter sharing refers to using the same parameter for more than one function in a model. In a traditional neural net, each element of the weight matrix is used exactly once when computing the output of a layer. It is multiplied by one element of the input and then never revisited. As a synonym for parameter sharing, one can say that a network has tied weights, because the value of the weight applied to one input is tied to the value of a weight applied elsewhere. Ina convolutional neural net, each member of the kernel is used at every position of the input (except perhaps some of the boundary pixels, depending on the design decisions regarding the boundary). The parameter sharing used by the convolution operation means that rather than learning a separate set of parameters for every location, we learn only one set. It also further reduces the storage requirements of the model parameters.

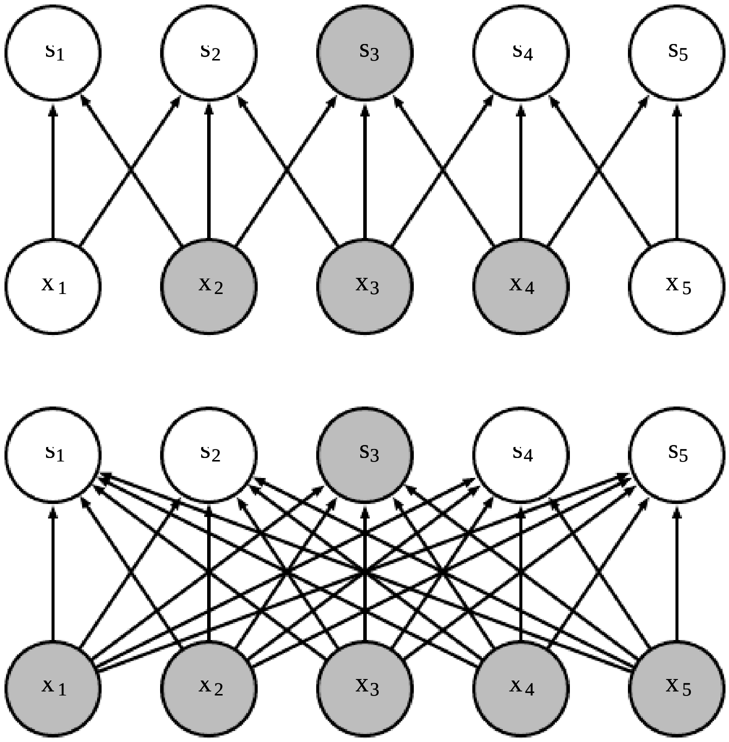


Figure 2.4 Sparse connectivity, viewed from above: We highlight one output unit, s3, and also highlight the input units in x that affect this unit. These units are known as the receptive field of s_3. (Top)When s is formed by convolution with a kernel of width 3, only three inputs affect s_3. (Bottom)When s is formed by matrix multiplication, connectivity is no longer sparse, so all of the inputs affect s_3.

Equivariance: In the case of convolution, the particular form of parameter sharing causes the layer to have a property called equivariance to translation. To say a function is equivariant means that if the input changes, the output changes in the same way. Specifically, a function $f(x)$ is equivariant to a function g if $f(g(x)) = g(f\ (x))$. In the case of convolution, if we let g be any function that translates the input, i.e., shifts it, then the convolution function is equivariant to g. For example, let I be a function giving image brightness at integer coordinates. Let g be a function mapping one image function to another image function, such that I'=g(I) is the image function with $I'(x, y) = I(x - 1, y)$. This shifts every pixel of I one unit to the right. If we apply this transformation to I, then apply convolution, the result will be the same as if we applied convolution to I', then applied the transformation g to the output. When processing time series data, this means that convolution produces a sort of timeline that shows when different features appear in the input. If we move an event later in time in the input, the exact same representation of it will appear in the output, just later in time. Similarly with images, convolution creates a 2-D map of where certain features appear in the input. If we move the object in the input, its representation will move the same amount in the output. This is useful for when we know that some function of a small number of neighboring pixels is useful when applied to multiple input locations. For example, when processing images, it is useful to detect edges in the first layer of a convolutional network. The same edges appear more or less everywhere in the image, so it is practical to share parameters across the entire image.

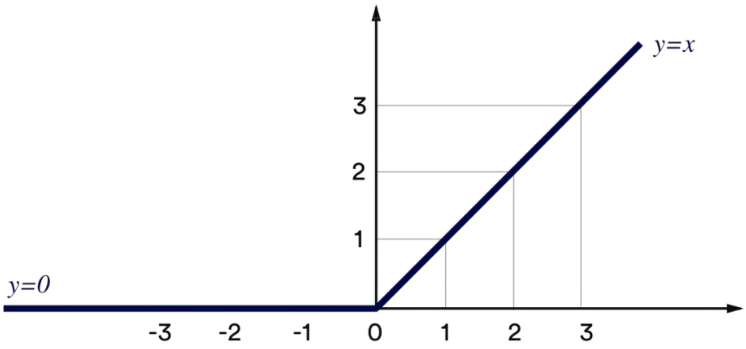


Figure 2.5 The ReLU activation.

2.4.3 Activation Function: ReLU

The activation function is a non-linear function applied to the output of each neuron in a layer. In case of convolutional neural networks, Rectifying Linear Units (ReLU) is the most commonly used non-linearity. For a given input x the output of ReLU is simply defined as

$$y = x^{+} = \max(x, 0) \qquad (2.4)$$

That is, if x is positive, it is passed without any modification. However, if x is negative, zero is passed instead. So, the ReLU function is simply a rectifier that clips the values below zero. It has been shown that this type of activation perform best for convolutional neural networks

[83]. They are shown to solve the vanishing gradient problem in deep architectures and thus facilitates better learning [84]. The ReLU function (y-axis) is plotted against its input (x-axis) in **Figure 2.5**.

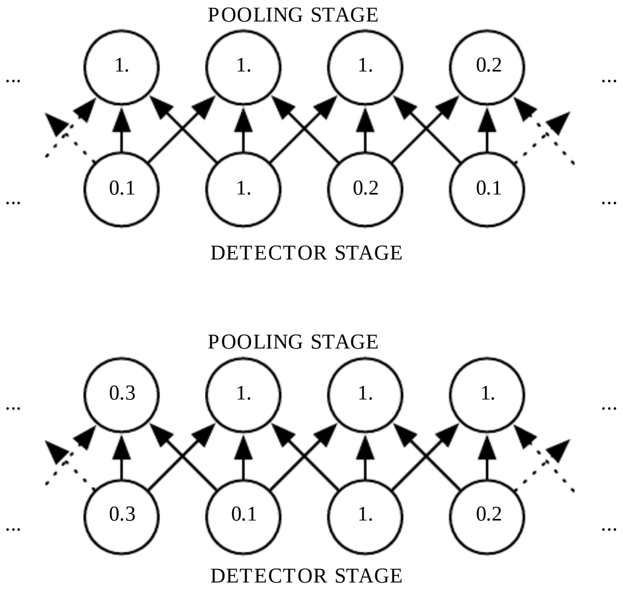


Figure 2.6 Max pooling introduces invariance. (Top)A view of the middle of the output of a convolutional layer. The bottom row shows outputs of the nonlinearity. The top row shows the outputs of max pooling, with a stride of one pixel between pooling regions and a pooling region width of three pixels. (Bottom)A view of the same network, after the input has been shifted to the right by one pixel. Every value in the bottom row has changed, but only half of the values in the top row have changed, because the max pooling units are only sensitive to the maximum value in the neighborhood, not its exact location.

2.4.4 Pooling

A pooling function replaces the output of the net at a certain location with a summary statistic of the nearby outputs. For example, the max pooling [85] operation reports the maximum output within a rectangular neighborhood. Other popular pooling functions include the average of a rectangular neighborhood, the L^2 norm of a rectangular neighborhood, or a weighted average based on the distance from the central pixel. In all cases, pooling helps to make the representation become approximately invariant to small translations of the input. Invariance to translation means that if we translate the input by a small amount, the values of most of the pooled outputs do not change (Figure 2.6). Invariance to local translation can be a very useful property if we care more about whether some feature is present than exactly where it is. For example, when determining whether an image contains a face, we need not know the location of the eyes with pixel-perfect accuracy, we just need to know that there is an eye on the left side of the face and an eye on the right side of the face.

For many tasks, pooling is essential for handling inputs of varying size. For example, if we want to classify images of variable size, the input to the classification layer must have a fixed size. This is usually accomplished by varying the size of an offset between pooling regions so that the classification layer always receives the same number of summary statistics regardless of the input size. For example, the final pooling layer of the network may be defined to output four sets of summary statistics, one for each quadrant of an image, regardless of the image size.

2.4.5 Typical Architecture of Convolutional Networks

A typical layer of a convolutional network consists of three stages (Figure 2.7). In the first stage, the layer performs several convolutions in parallel to produce a set of linear activations. In the second stage, each linear activation is run through a nonlinear activation function, such as the rectified linear activation function. This stage is sometimes called the detector stage. In the third stage, we use a pooling function to modify the output of the layer further.

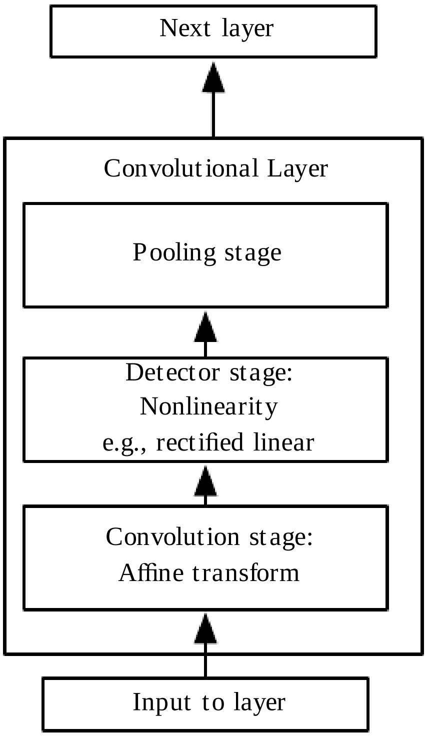


Figure 2.7 The components of a typical convolutional neural network layer.

2.5 Output Units of a Classifier

For a DNN classifier, usually two types of output units are commonly used, namely sigmoid and softmax, depending on whether the classification task is binary or multi-class. However, these non-linearities used in these outputs are not necessarily restricted to the output of a network; they are often used as activation functions in hidden layers within a network as well.

2.5.1 Sigmoid Output Unit for Binary Classification

For binary classification, given and input x the network needs to output the probability of a Bernoulli variable y which represents the output class and can take two values, 0 or 1. To parameterize this Bernoulli distribution, a sigmoid function is applied to the output of the network. A sigmoid function is defined as follows-

$$\sigma(z) = \frac{1}{1 + \exp(-z)} \tag{2.5}$$

Where, σ represents the non-linear sigmoid function and z is the input. As shown in Figure 2.8, this function maps the entire number line to a range of $[0,1]$ and has a finite gradient everywhere, so it is appropriate for representing the output class probability $P(y = 1|x)$.

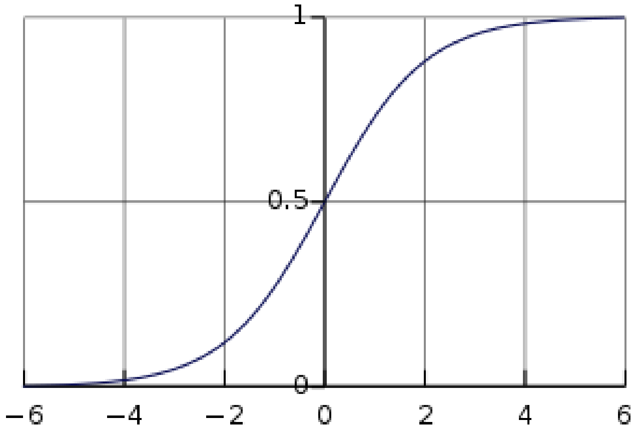


Figure 2.8 The logistic sigmoid function.

2.5.2 Softmax Output Units for Multiclass Classification

For multi-class classification probabilities over class label y need to estimated by the network for a given input x, where y can take m different values (m is the number of classes). Softmax activation achieves these by the following non-linear function

$$softmax(z)_i = \frac{\exp(z_i)}{\sum_{j=1}^{m} \exp(z_j)} \tag{2.6}$$

As, can be seen from the expression $sofmax(z)_i$ is always in the range $[0, 1]$ and $\sum_{i=1}^{m} softmax(z)_i = 1$. Thus it can effectively produce a categorical distribution for $P(y = y_i|x)$.

2.6 Cost functions of a Classifier: Maximum Likelihood Estimation

Under maximum likelihood estimation the model parameters θ (weights of the network) are chosen so that the likelihood of the given data X according to the model, is maximized. In case of classification we need to maximize the conditional likelihood $P_{model}(Y|X; \theta)$. Here, P_{model} is a function parameterized by the deep neural network. The optimum parameters θ_{ML}^{*} under maximum likelihood scheme is given by.

$$\theta_{ML}^{*} = \arg\max_{\theta} P_{model}(Y|X; \theta) \tag{2.7}$$

This problem can be converted into a stochastic optimization problem if we assume that x and y are drawn from the data generating distribution. Then we can maximize the expected log probability rather than the actual probability.

$$\theta_{ML}^{*} = \arg\max_{\theta} \underset{(x,y)\sim P_{data}}{E} logP_{model}(y|x; \theta) \tag{2.8}$$

$$\Rightarrow \ \theta_{\mathrm{ML}}^{*} = \arg\min_{\theta} \underset{(x,y)\sim P_{data}}{-E} logP_{model}(y|x; \theta) \tag{2.9}$$

Here, we simply took the negative value to convert the maximization problem into a minimization problem. Here,

$$J(\theta) = H(P_{data}, P_{model}) = \underset{(x,y)\sim P_{data}}{-E} logP_{model}(y|x; \theta) \tag{2.10}$$

Is known as the cross-entropy and used as a cost function (also called a loss function or simply “loss”) for classification, which needs to be minimized by an optimization algorithm. In case of binary classification y is either 0 or 1. So, we can simplify the cost function as follows

$$J(\theta) = \underset{(x,y)\sim P_{data}}{-E} [y \log P_{model}(y = 1|x;\theta) + (1 - y) \log\{1 - P_{model}(y = 1|x;\theta)\}] \quad (2.11)$$

This cost function is known as binary cross entropy. The network only needs to output a single probability value $P_{model}(y = 1|x;\theta)$, usually with a sigmoid output unit.

2.7 Optimization Strategies for Deep Neural Network

Deep feedforward neural networks, like the convolutional neural network are trained using a gradient based optimization technique, called the stochastic gradient descent. However, before we can apply gradient descent, we need to evaluate the gradient of the cost function w.r.t. each element of the parameter set θ of the network. This is done efficiently by an analytical tool called the backpropagation algorithm.

2.7.1 The Backpropagation Algorithm

When we use a feedforward neural network to accept an input x and produce an output $\hat{y}$, information flows forward through the network. The inputs x provide the initial information that then propagates up to the hidden units at each layer and finally produces $\hat{y}$. This is called forward propagation. During training, forward propagation can continue onward until it produces a scalar cost $J(\theta)$. The back-propagation algorithm [53], often simply called backprop, allows the information from the cost to then flow backwards through the network, in order to compute the gradient. Computing an analytical expression for the gradient is straightforward, but numerically evaluating such an expression can be computationally expensive. The back-propagation algorithm does so using a simple and inexpensive procedure.

The chain rule of calculus is used to compute the derivatives of functions formed by composing other functions whose derivatives are known. Back-propagation is an algorithm that computes the chain rule, with a specific order of operations that is highly efficient.

Let x be a real number, and let f and g both be functions mapping from a real number to a real number. Suppose that $y = g(x)$ and $z = f(g(x)) = f(y)$. Then the chain rule states that

$$\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx} \quad (2.12)$$

If x, y are vectors, as in intermediate values or weights of a neural network.

$$\nabla_x z = \left(\frac{\partial y}{\partial x}\right)^T \nabla_y z \quad (2.13)$$

Here $\frac{\partial y}{\partial x}$ is the Jacobian matrix of g.

Using the chain rule, it is straightforward to write down an algebraic expression for the gradient of a scalar with respect to any node in the computational graph that produced that scalar. To compute the gradient of some scalar z with respect to one of its ancestors x in the graph, we begin by observing that the gradient with respect to z is given by $\frac{dz}{dz}$ = 1. We can then compute the gradient with respect to each parent of z in the graph by multiplying the current gradient by the Jacobian of the operation that produced z. We continue multiplying by Jacobians traveling backwards through the graph in this way until we reach x. For any node that may be reached by going backwards from z through two or more paths, we simply sum the gradients arriving from different paths at that node.

2.7.2 Stochastic Gradient Descent

Nearly all of deep learning is powered by one very important algorithm: stochastic gradient descent or SGD [86]. A recurring problem in machine learning is that large training sets are necessary for good generalization, but large training sets are also more computationally expensive. The cost function used by a machine learning algorithm often decomposes as a sum over training examples of some per-example loss function. For example, the negative conditional log-likelihood of the training data can be written as

$$J(\theta) = \underset{x,y \sim p_{data}}{E} L(x, y, \theta) \tag{2.14}$$

where, L is the per-example loss $L(x, y, \theta) = -\log p_{model}(y|x;\theta)$

The insight of stochastic gradient descent is that the gradient is an expectation. The expectation may be approximately estimated using a small set of samples. Specifically, on each step of the algorithm, we can sample a minibatch of examples $\boldsymbol{B} = \{x^{(1)}, \ldots, x^{(m')}\}$ drawn uniformly from the training set. The minibatch size m' is typically chosen to be a relatively small number of examples, ranging from 1 to a few hundred. Crucially, m' is usually held fixed as the training set size m grows. We may fit a training set with billions of examples using updates computed on only a hundred examples.

The estimate of the gradient is formed as

$$g = \frac{1}{m'} \nabla_\theta \sum_{i=1}^{m'} L(x^{(i)}, y^{(i)}, \theta) \tag{2.15}$$

The stochastic gradient descent algorithm then follows the estimated gradient downhill:

$$\theta \leftarrow \theta - \epsilon\theta \qquad (2.16)$$

where ϵ is the learning rate. Many improvements on the basic stochastic gradient descent algorithm have been proposed and used. In particular, in machine learning, the need to set a learning rate (step size) has been recognized as problematic. Setting this parameter too high can cause the algorithm to diverge; setting it too low makes it slow to converge. A conceptually simple extension of stochastic gradient descent makes the learning rate a decreasing function of the iteration number, giving a learning rate schedule, so that the first iterations cause large changes in the parameters, while the later ones do only fine-tuning. Modern algorithms like Adam [87] use a more sophisticated approach to calculate updates in each iteration and adapt them based on the gradients calculated in not only current step but also the previous steps.

2.8 Regularization Methods for Deep Neural Networks

As mentioned in section 2.2, a central problem in machine learning is how to make an algorithm that will perform well not just on the training data, but also on new inputs. Many strategies used in machine learning are explicitly designed to reduce the test error, possibly at the expense of increased training error. These strategies are known collectively as regularization. There are a many forms of regularization available for deep neural networks. We will discuss only a few of them, which are related to our work.

2.8.1 Weight Regularization and Constraint

A common form of regularization, known as L_2 regularization or weight decay works by penalizing the L_2 norm of a set of weights. This is done by adding a penalty term $\Omega(\theta)$ to the loss function $J(\theta; X, y)$. The modified loss function

$$\tilde{J}(\theta; X, y) = J(\theta; X, y) + \Omega(\theta) \qquad (2.17)$$

Where, for L_2 regularization,

$$\Omega(\theta) = \lambda\|w\|_2^2 \qquad (2.18)$$

Sometimes we may wish to use explicit constraints rather than penalties. We can modify algorithms such as stochastic gradient descent to take a step downhill on J(θ) and then project θ back to the nearest point that satisfies $\Omega(\theta) < k$. This can be useful if we have an idea of what value of k is appropriate and do not want to spend time searching for the value of α that corresponds to this k. Another reason to use explicit constraints and reprojection rather than enforcing constraints with penalties is that penalties can cause non-convex optimization procedures to get stuck in local minima corresponding to small θ. Finally, explicit constraints with reprojection can be useful because they impose some stability on the optimization procedure. When using high learning rates, it is possible to encounter a positive feedback loop in which large weights induce large gradients which then induce a large update to the weights. If these updates consistently increase the size of the weights, then θ rapidly moves away from the origin until numerical overflow occurs. Explicit constraints with reprojection prevent this feedback loop from continuing to increase the magnitude of the weights without bound.

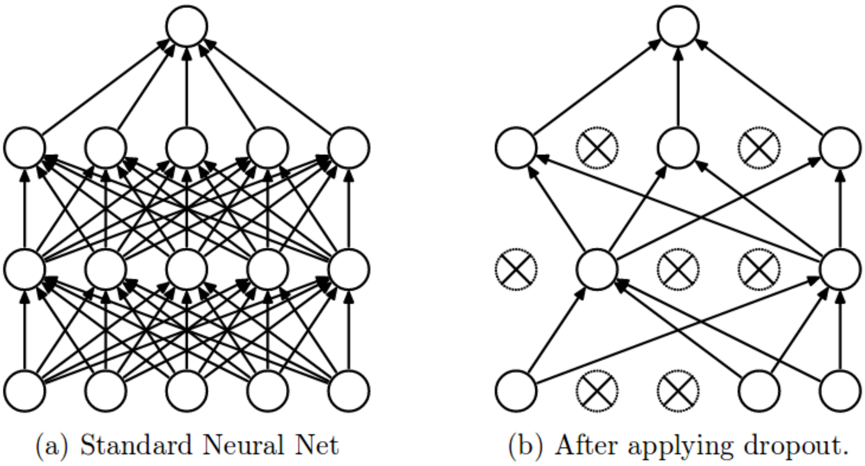


Figure 2.9 Dropout in deep neural network. Dropout can be easily implemented by randomly disconnecting some neurons of the network. (a) The complete network (b)one possible network formed by dropout at training time. If the model has n neurons, there are 2^n potential models.

2.8.2 Dropout

Dropout [88] provides a computationally inexpensive but powerful method of regularizing a broad family of models. To a first approximation, dropout can be thought of as a method of making bagging practical for ensembles of very many large neural networks. Bagging involves training multiple models, and evaluating multiple models on each test example. This seems impractical when each model is a large neural network, since training and evaluating such networks is costly in terms of runtime and memory. Dropout provides an inexpensive approximation to training and evaluating a bagged ensemble of exponentially many neural networks.

Specifically, dropout trains the ensemble consisting of all sub-networks that can be formed by removing non-output units from an underlying base network (Figure 2.9). We can effectively remove a unit from a network by multiplying its output value by zero. Dropout regularizes each hidden unit to be not merely a good feature but a feature that is good in many contexts.

Dropout is more effective than other standard computationally inexpensive regularizers, such as weight decay, filter norm constraints and sparse activity regularization [88]. Dropout may also be combined with other forms of regularization to yield a further improvement. One advantage of dropout is that it is very computationally cheap. Using dropout during training requires only $O(n)$ computation per example per update, to generate n random binary numbers and multiply them by the state.

2.8.3 Early Stopping

When training large models with sufficient representational capacity to overfit the task, we often observe that training error decreases steadily over time, but validation set error begins to rise again, as shown in Figure 2.10. This behavior occurs very reliably. This means we can obtain a model with better validation set error (and thus, hopefully better test set error) by returning to the parameter setting at the point in time with the lowest validation set error. Every time the error on the validation set improves, we store a copy of the model parameters. When the training algorithm terminates, we return these parameters, rather than the latest parameters. This strategy is known as early stopping. It is probably the most commonly used form of regularization in deep learning. Its popularity is due both to its effectiveness and its simplicity.

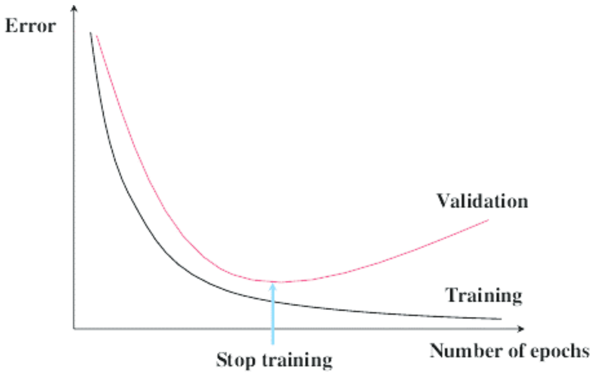


Figure 2.10 Typical loss/error behavior in DNNs. The training objective decreases consistently over time, but the validation set average loss eventually begins to increase again, forming an asymmetric U-shaped curve.

2.8.4 Multi-task Learning

Multi-task learning [89] is a way to improve generalization by pooling the examples arising out of several tasks. In the same way that additional training examples put more pressure on the parameters of the model towards values that generalize well, when part of a model is shared across tasks, that part of the model is more constrained towards good values (assuming the sharing is justified), often yielding better generalization.

Figure 2.11 illustrates a very common form of multi-task learning, in which different supervised tasks (predicting $y^{(i)}$ given x) share the same input x, as well as some intermediate-level representation $h^{(shared)}$ capturing a common pool of factors. The model can generally be divided into two kinds of parts and associated parameters

1. Task-specific parameters (which only benefit from the examples of their task to achieve good generalization). These are the upper layers of the neural network.
2. Generic parameters, shared across all the tasks (which benefit from the pooled data of all the tasks). These are the lower layers of the neural network.

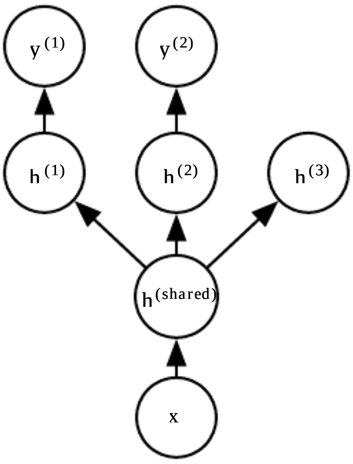


Figure 2.11 A common setting for multi-task learning.

Improved generalization can be achieved because of the shared parameters, for which statistical strength can be greatly improved (in proportion with the increased number of examples for the shared parameters, compared to the scenario of single-task models). Of course this will happen only if some assumptions about the statistical relationship between the different tasks are valid, meaning that there is something shared across some of the tasks. From the point of view of deep learning, the underlying prior belief is the following: among the factors that explain the variations observed in the data associated with the different tasks, some are shared across two or more tasks.

2.9 Transfer Learning in Convolutional Neural Networks

We introduced transfer learning in Section 1.7. Transfer learning allows the training of deep networks using significantly less data than that would be needed it were trained from scratch. With transfer learning, we can in effect transfer the “knowledge” that a model has learned from a source task, to a target task. The idea behind transfer learning is that the two tasks are not totally disjoint, and as such we can leverage whatever network parameters that model has learned through its extensive training, without having to do that training ourselves. Transfer learning has been consistently proven to boost model accuracy and reduce the required training time. In case of DNNs, the most common method of transfer learning simply involves transferring the learned weights/parameters from the source task to the destination task. On the destination task these learned weights can either be fine-tuned or kept fixed (to prevent overfitting) depending on how much data is available.

Convolutional Neural Networks trained on diverse datasets like the ImageNet dataset learns robust feature representations of images that can be used for other classification tasks [77]. Lower layers pick up information about edges which are processed by the upper layers to detect shape primitives. Going deeper they start to combine and form complex shapes (such as faces) and texture (such as fur) representations [90]. Deeper layers also become invariant to perturbations such as shift, small rotations, scaling, cropping, color distortions and geometric distortions [79]. These features/feature maps can be used to learn from very few training examples where there is a shortage of training data [91]. This method also increases the robustness of the model if properly applied. Feature representations from deep CNNs have practically replaced conventional feature descriptors used in computer vision, such as HOG and SIFT features [77], [92]. These features have also been utilized for perceptual evaluation[79], style transfer [80], feature inversion [93] and texture synthesis [94]. These experimental findings imply that pretrained deep CNNs are very good at disentangling information about the structure and morphology of the input image. These disentangled feature representations can ease further training of ensuing layers that have the necessary information about the input in an easily interpretable form. The invariances built into pretrained networks can also be useful where they are relevant to the target task and alleviates the need for data augmentation.

2.10 Chapter Summary

In this chapter we discussed a range of topics, beginning with machine learning in general. Then we moved on to deep neural networks and the convolutional architecture for DNNs that has been proven to be successful on many high dimensional datatypes like images and audio. Finally, we presented specialized optimization and regularization strategies for training these networks effectively. We ended with a brief discussion about transfer learning in CNNs which plays a key role in our work.

Chapter 3
Breast Lesion Classification by Convolutional Neural Network

In this chapter we present the methodology of our work. First we discuss how the data is acquired and processed. Next we discuss about the factors we considered in designing the proposed CNN model. Finally, we describe the proposed network architecture in details and also point out the reasons behind the design choices.

3.1 Data Acquisition Method

A dataset of 239 lesions from 188 patients was collected. Written consent was obtained from all patients prior to collection of data and all the data were deidentified for the preservation of patient privacy. The patients' ages ranged from 13 to 75 years with a mean age of 35.27 years and standard deviation of 12.18 years. Of the 188 patients, 176 proceeded to pathological testing. So, we report our test results only on the pathologically confirmed 217 lesions collected from 176 patients. However, the diagnoses of the rest of the lesions were based on strong radiological evidence and we included them in the training phase. The lesions are broadly categorized into carcinoma, fibroadenoma, cyst, and inflammation. The details of the diagnoses are summarized in Table 3.1 and Table 3.2.

Table 3.1 Test data

Lesion type	No. of lesions	No. of patients	Mean Age ± SD	Method of Confirmation
Carcinoma	61	45	44.91±1.89	Biopsy
Fibroadenoma	85	74	27.48±9.13	FNAC/Biopsy
Cyst	44	34	39.13±8.9	FNAC
Inflammation	27	23	35.43±12.71	FNAc/Biopsy
Total	217	176	-	-

Table 3.2 All data (used in training phase)

Lesion type	No. of lesions	No. of patients
Carcinoma	66	47
Fibroadenoma	98	82
Cyst	48	36
Inflammation	27	23
Total	239	188

A commercial SonixTOUCH Research (Ultrasonix Medical Corporation, Richmond BC, Canada) scanner integrated with a linear array transducer, L14-5/38, operating at 10 MHz with a sampling frequency of 40 MHz was used to perform the B-mode (US) and elastography (UE) examinations (Figure 3.1). The dataset is composed of videos containing ultrasound B-mode (US) and elastogram (UE) frames of each lesion. We extracted multiple frames per lesion to reduce the variance in quasi-static elastography frames due to variations in the movement of the probe.

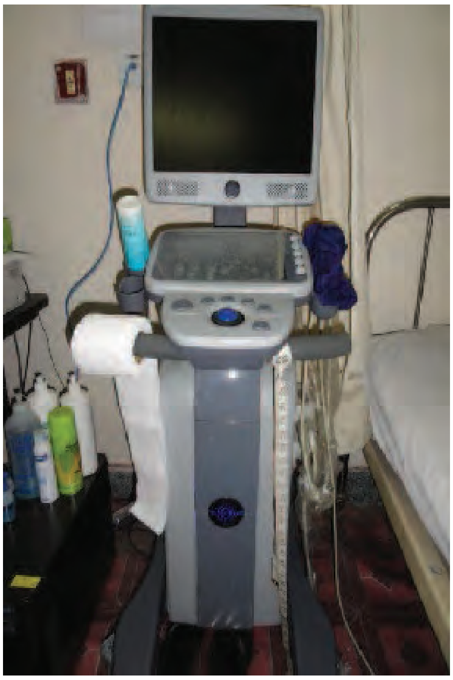


Figure 3.1 SonixTOUCH RP for performing US and UE at BUET medical center.

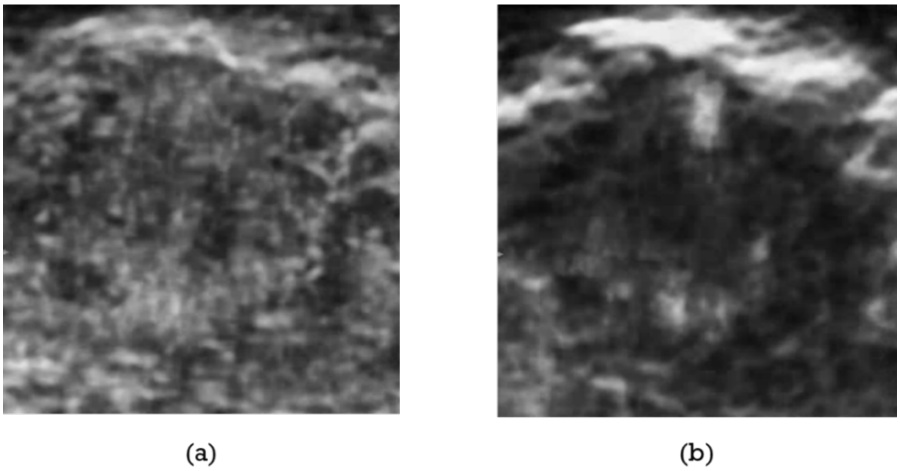


Figure 3.2 Necessity of selecting appropriate UE frames from video (a) a noisy frame (b) a good quality frame of the same lesion.

3.2 Frames Extraction from Ultrasound Video

Multiple (14-28) US and UE frames were extracted per lesion from ultrasound videos. Temporal and spatial correspondences were maintained between US and UE images i.e. they were taken from the same frame of the video and would align perfectly when overlapped.

Quasi-static elastography frames are inherently noisy because of the indeterministic nature of the movement of the probe that in turn causes random movement in the tissue. A big portion of the UE frames are obscure and do not contain any usable information about the lesion (Figure 3.2). However, we found that almost all of the obscure frames can be discarded by monitoring the focal measure of the images. We deployed a curvature based focal measurement [95] system of the UE frames to select the most informative UE frames which best represented the stiffness distribution. Under this focal measure scheme gray values $g(x, y)$ are assumed to correspond to a 3D surface $\big(x, y, g(x, y)\big)$. The surface is parameterized as $f(x, y) = p_0 x + p_1 y + p_2 x^2 + p_3 y^2$. The coefficients are found from the image $I(x, y)$ using filters g_0 and g_2.

$$g_0 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad g_2 = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \tag{3.1}$$

$$P = \left[\frac{g_0 * I}{6}; \frac{g_0^T * I}{6}; \frac{3g_2 * I}{10} - \frac{g_2^T * I}{5}; -\frac{g_2 * I}{5} + \frac{3g_2^T * I}{10}\right]^T \tag{3.2}$$

The focal measure is calculated by taking the absolute sum of the coefficients

$$FM_C = |p_0| + |p_1| + |p_2| + |p_3| \tag{3.3}$$

The sharpest frames i.e. the frames where the lesion most prominent, were found to correspond to the extrema (i.e. minima and maxima, as shown in Figure 3.3) of the measure. The frames corresponding to the most prominent extrema conveys the most information. These frames (as shown in Figure 3.4) were selected for the classification task. The selection process up to this point is fully automated. However, we noticed that a few obscure frames would occasionally be selected by the algorithm. So, to further ensure the selection of the clearest images, a manual selection was carried out from the automatically selected frames. At this stage, manual selection was much less time consuming because there were only a few frames to select from. After the selection of the UE frames, corresponding US frames from the video were selected.

Relative sizes of the lesion in US and UE images is an important indicator of malignancy of a tumor [96]. To capture this information we produced a difference image by subtracting US frames from corresponding UE frames and then rescaling the intensity of the resultant image as shown in Figure 3.5. The difference images also capture the positional correspondence between US and UE images. The intensities of the US and UE images were also rescaled to cover the whole allowed range of pixel intensity levels (e.g. 0 to 255 for 8-bit image). The dimensions (i.e. width and height) of the frames were rescaled by a fixed factor of ½ which approximately brought the lesions within the receptive field of the CNN. Our prepared dataset contained three types of frames namely, US, UE and difference with different sizes for different lesions. The variable image size was acceptable because of the nature of the convolutional architecture we used. The workflow of the data preparation process is shown in Figure 3.6.

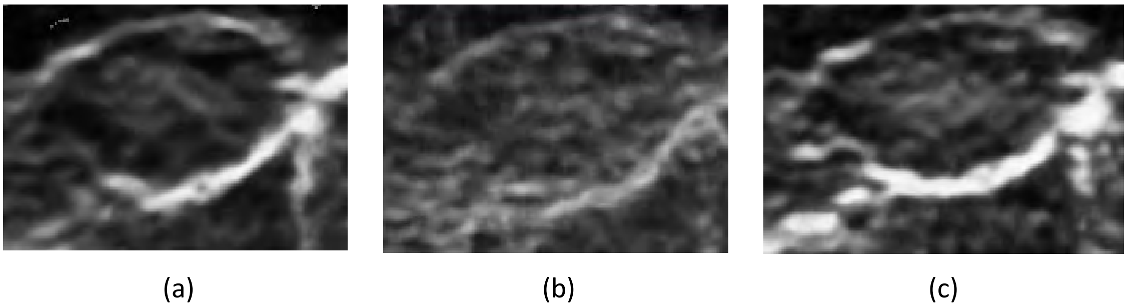

(a) (b) (c)

Figure 3.3 UE frames of a lesion, corresponding to 3 different cases of focal measure (a) a prominent minimum, (b) neither minimum nor maximum (c) a prominent maximum. The minima and the maxima correspond to frames of better quality.

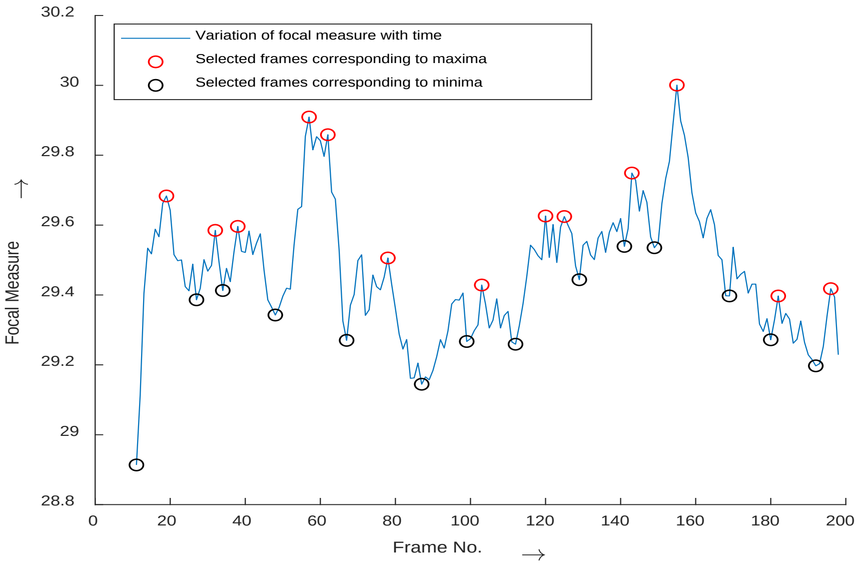


Figure 3.4 Selection procedure of the UE frames from the video. The best frames are found to correspond to the prominent minima and maxima of the focal measure. Based on this observation, the most prominent extrema are chosen by the algorithm.

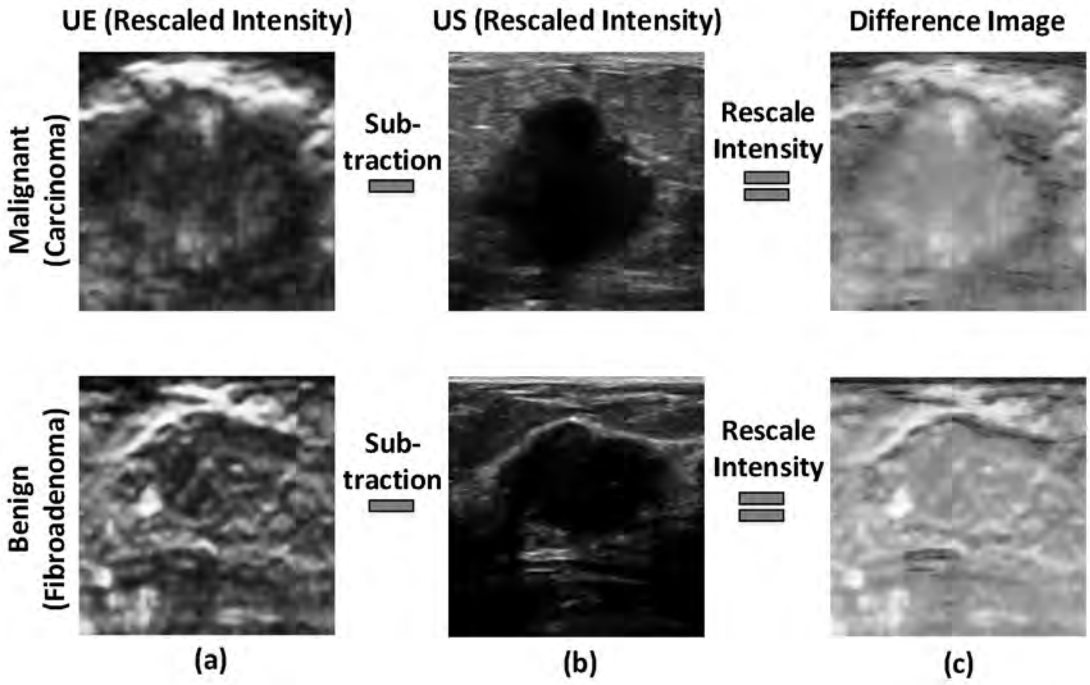


Figure 3.5 Process of creating difference images. The intensity of both ultrasound and elastography frames are rescaled prior to subtraction. The enlargement of the malignant lesion in the elastography image results in a dark region around the lesion in the difference image. The benign lesion shows no such region.

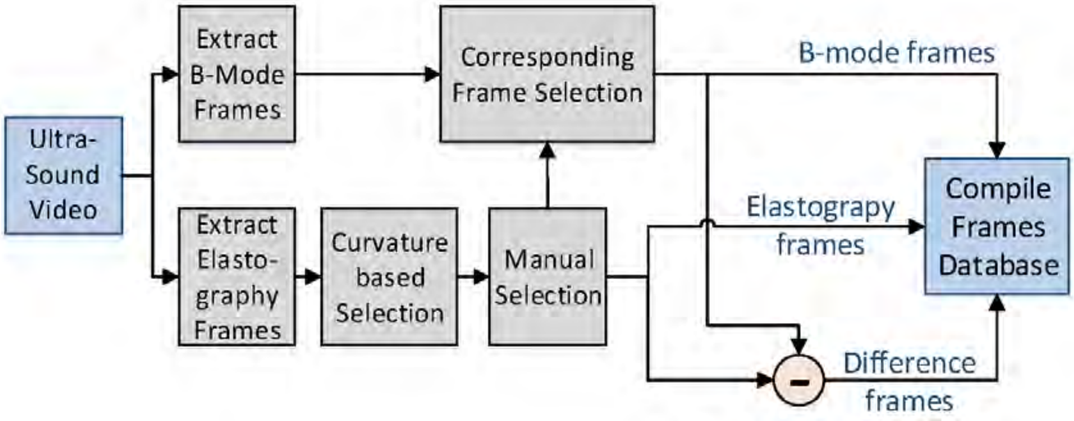


Figure 3.6 Block diagram showing data preparation process. Temporal correspondence is maintained among extracted B-mode, elastography and difference images. Together they form a triplet.

3.3 Notes on ROI Selection

Most of the previous deep learning models on breast lesion classification from ultrasound images, e.g. [9] and [28] require a carefully selected rectangular ROI of the lesion. Although it is easier to select such ROIs compared to full segmentation, it still introduces some subjectivity in the classification process, increases risk of miss detections and also require reasonable experience on part of the radiologists to identify such ROIs. Also, a tightly bounded ROI may leave out important information present in the surrounding tissue. Furthermore, in some cases, as shown in Figure 3.7, it can be difficult to identify the lesion boundary for malignant and inflammatory lesions, which often have no well-defined boundary and also because of shadowing or echoes in the images. This requirement of an ROI centered around a lesion is due to the densely-connected nature of the networks used in these works, which are not inherently invariant to changing positions of the lesions in image frames. However, we used a pretrained convolutional neural network in the primary feature extraction stage of the proposed model, which is mostly invariant to positional shifts and able to extract the features of an object regardless of its position. So, as long as the image contains the lesion, it is able to extract relevant features. So, the proposed method requires no ROI selection and we directly used image frames from the machine. This approach makes the proposed method more objective, faster and more convenient for real-time implementation. Also, it is able to use features from surrounding tissue to come to a decision even when the lesion boundary is not distinct.

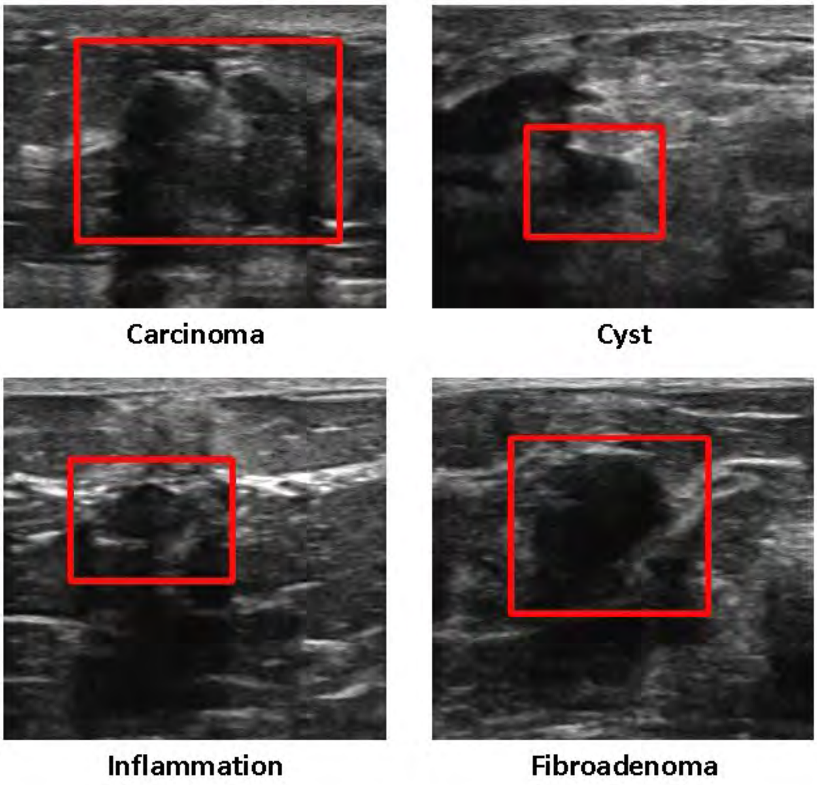


Figure 3.7 The proposed method uses raw images rather than selected ROIs (red rectangle). These images demonstrate the difficulty in selecting an ROI. The malignant and inflammatory lesions have no distinct boundary. The boundary of the fibroadenoma lesion is obscured due to shadowing.

3.4 Notes on Data Augmentation

We did not use any form of data augmentation. This is due to the fact that, we used a deep CNN (VGG-16) trained on a big and diverse dataset (ImageNet) as a fixed feature extractor, which has naturally become invariant to transformations such as small shifts and zoom, slight rotations, flips and small elastic deformations which are used as conventional means of image data augmentation. So, we did not get any significant change/improvement in classification results by data augmentation.

3.5 Additional Features

Image classification often benefits from side information. Similarly, as an additional investigation, we experimented with the impact of the inclusion of several additional features in the classification process. These features were extracted from the ultrasound images of a lesion by visual inspection. These additional features, which we will call “qualitative features” (contrary to quantitative/numerical features) require no segmentation and can be easily identified by a trained radiologist. A summary of the qualitative features extracted are given below:

Quadrant: Determined from the clock position of the lesion – in which of the 4 quadrants of the breast the lesion is located.

Capsulation: Whether capsulation was (i) present / (ii) not present / (iii) ambiguous

Shape: Shape of the lesion (i) oval / (ii) round / (iii) irregular

Orientation: Orientation of the lesion (i) horizontal (parallel to skin) / (ii) vertical / (iii) indifferent

Lesion Boundary: Nature of lesion boundary (i) hyper halo / (ii) thin Capsule / (iii) not present

Posterior Acoustic Shadowing: Nature/presence of posterior acoustic shadowing (i) shadowing / (ii) enhancement / (iii) not present

Echo Pattern: Nature of echo pattern (i) Complex / (ii) Homogeneous

Hyper-echogenic Spots: (i) Present / (ii) absent

Surrounding Tissue: Nature of the tissue surrounding the lesion (i) complex / (ii) distorted / (iii) indistinct

Calcification: (i) Present / (ii) absent

Strain Ratio: Although quantitative in nature, this feature has high variability. So, it’s values were discretized in 5 different ranges.

A more in-depth description of these features can be found in [97]. Apart from using these features as inputs we also used them (all except quadrant) as additional targets alongside the principle target of malignant/benign, in a multi-task learning setting. We must mention that these features are slightly subjective in nature, i.e. dependent on the decision of the radiologist. So, our primary results are focused on classification without inclusion of these features.

3.6 Considerations in Network Design and Training

3.6.1 Transfer Learning: Factors of Transferability

For incorporating transfer learning in a model, several choices need to be made [91]. The first is, of course, which source task to train on and which network to train on the source task. It has been seen that the more closely related the source task is to the target task, the better the knowledge transfer can occur. Deeper architectures usually learn richer feature representations than shallower ones. Another concern is to decide which layer of the network to extract features from. It is observed that features from the lower layers are more general-purpose because they represent the basic structural elements of the image like edges, basic shapes and textures. The representations learned by deeper layers are more abstract and task specific. So, if two tasks are very similar, the task-specific features from deeper layers would produce good results. Otherwise, the more general features from the lower layers are found to perform better. The layers from the pretrained network can be made untrainable (i.e. fixed weight) or they can be fine-tuned for the target data.

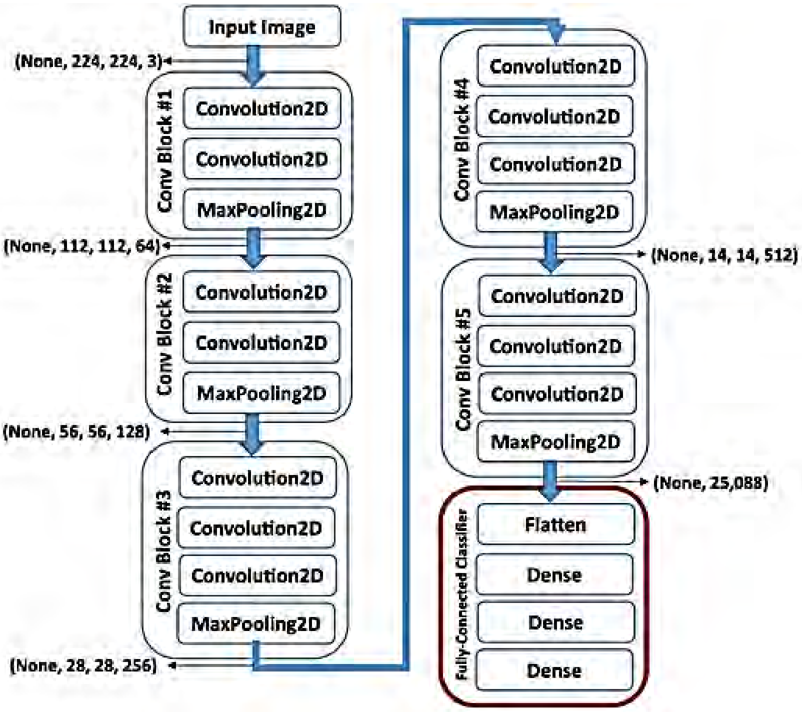


Figure 3.8 The VGG-16 architecture

3.6.2 Network Selection

For our purpose we experimented with the VGG (16 and 19) [98], Inception (v1 and v2) [72] and ResNet (50, 100 and 150) [99] and found the VGG architectures i.e. VGG-16 and VGG-19 most suitable. More specifically, we used the feature representations produced by `relu_4_3' layer of VGG-16 as the starting-point of the proposed network, i.e. we built additional layers on top of these feature maps. We predict that, since US and UE images are very different from natural images, these intermediate-level feature representations worked well for our purpose, rather than deeper representations. Since the proposed network was trained on very few datapoints (239 lesions) we refrained from fine-tuning the VGG-16 layers which would increase the overall capacity of the model and result in overfitting. So, these VGG-16 layers were used as fixed feature extractors for our purpose.

The VGG-16 Network has a total of 13 convolutional layers arranged in 5 blocks separated by Max-Pooling Layers as shown in Figure 3.8. After flattening the outputs of the final convolutional block 3 additional dense layers are added before ending in a softmax output. All the convolutional filters are 3x3 and all the activations applied (except for softmax) are ReLU. More details on the network architecture can be found in [98].

3.7 Description of the Network Architecture

The proposed network can be thought of as having three parts, each serving a particular functionality. The preliminary part of the network extracts features from the images, the intermediate part combines these features in a meaningful way and the final part uses the combined features for lesion classification. The detailed architecture is shown in Figure 3.9 -- Figure 3.13.

3.7.1 Feature Extraction Stage

As we have 3 types of images we used 3 branches with identical structures for feature extraction. In each branch, the input image was fed as a grayscale image to a VGG-16 feature extractor. This method is justified by the fact that, the feature representations formed in the deeper layers of VGG-16 are invariant to image color-space. However, the first convolutional layer of VGG-16 takes 3 input channels (R, G, B). So, it was modified to take single channel input. The ImageNet weights were loaded into the VGG-16 layers. For the first layer, the convolutional kernel from ImageNet weights had to be modified as follows to apply to a single channel input - if for the m'th input channel and n'th output channel/feature map, the kernel coefficient at position (i,j) is then the modified kernel coefficients $K'_{1,n,i,j}$ are calculated by summing over the 3 input channels.

$$K'_{1,n,i,j} = \sum_{m=1}^{3} K_{m,n,i,j} \tag{3.4}$$

As mentioned previously we used the feature representations produced by `relu_4_3' layer, so in total 10 (out of 13) convolutional layers were used from VGG-16. The weights of these layers were kept fixed during training. The VGG-16 block produces 512 feature maps. Next, we added a trainable convolutional layer with ReLU activation which further processes the feature representations and also reduces the number of feature maps to 256. The reduction in feature space is done intentionally to limit the capacity of the network and prevent overfitting. However, it may not be necessary for bigger datasets. We did not get any improvement by incorporating additional convolutional layers. Next, we applied global average pooling to the activations of the convolutional block. We chose to end the convolutional part of the network by global average pooling [100] instead of flattening or other types of pooling (e.g. spatial pyramidal pooling [101]. This is done, to regularize the network by keeping the feature space small. Also, global average pooling is proven to be a general way of ending convolutional parts of a network in many well-established architectures [72], [99]. As an additional benefit, it allows the network to accept images of variable input shapes, which is useful for our case, because of the irregular shapes and sizes of different input frames. The feature extraction stage of the network is shown in Figure 3.9.

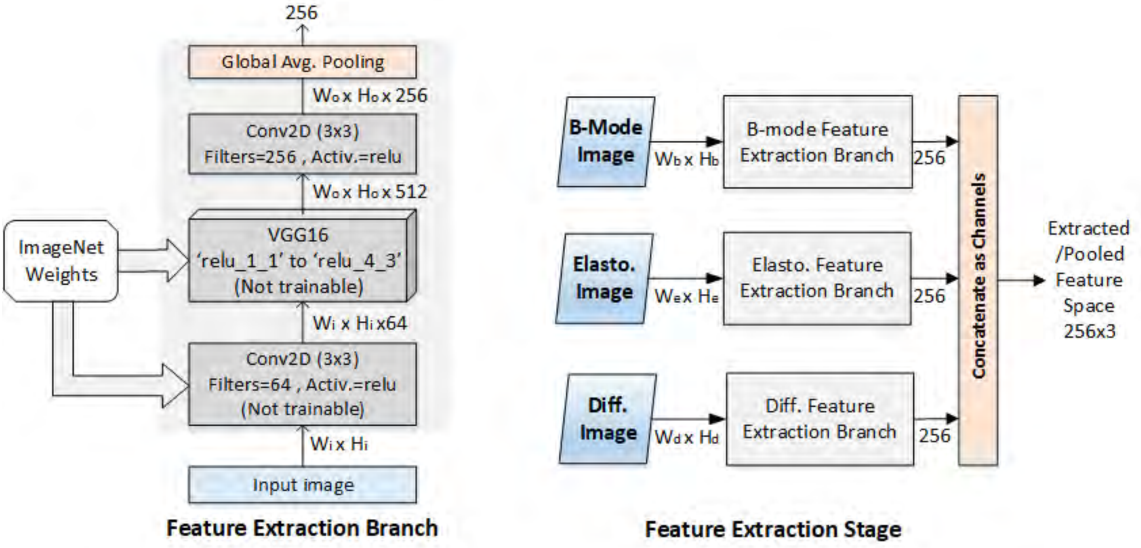


Figure 3.9 Feature extraction stage of the network. Each type of image has its own feature extraction branch. This stage takes images of variable input sizes and outputs a set of pooled features which proceeds to the merging stage.

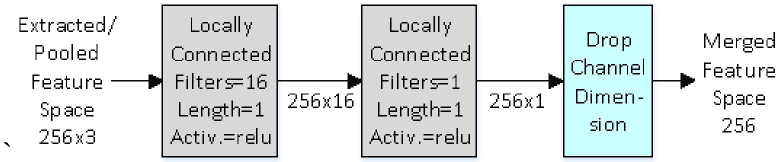

Extracted/ Pooled Feature Space 256x3
Locally Connected Filters=16 Length=1 Activ.=relu
256x16
Locally Connected Filters=1 Length=1 Activ.=relu
256x1
Drop Channel Dimen-sion
Merged Feature Space 256

Figure 3.10 Feature merging stage of the network. The merged features proceed to the classification stage.

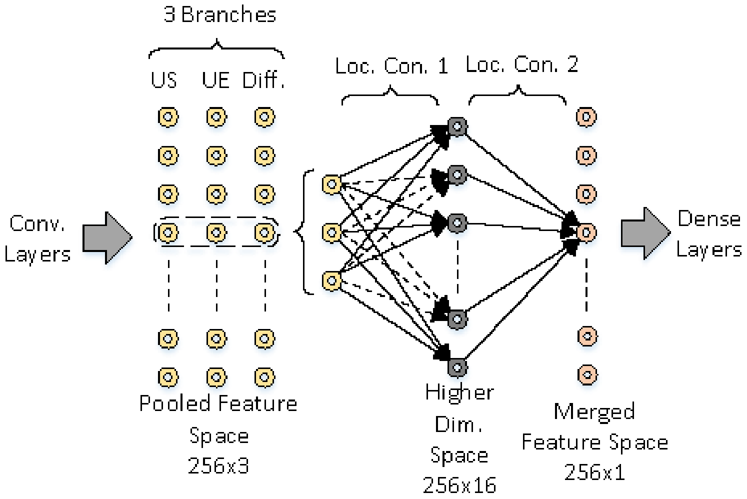


Figure 3.11 Process of merging features. The first layer puts each triplet of corresponding features (from US, UE and Diff. branches) to a higher dimension and the second layer merges them to a single feature.

3.7.2 Feature Merging Stage

After global average pooling, we get three sets of (3x256) features from three branches of the network operating on three types of images. We, hypothesize that each corresponding feature from three branches can be merged together independently of all other features; before further processing. This way, we can effectively reduce the feature space to form a single set of features, while keeping the network capacity in check. This is done by processing through two consecutive layers with ReLU activations. Structurally this is identical to applying two consecutive 1D locally connected layers with filter length 1 where the 3 branches are treated as input channels. The output of the second layer is only one feature map. These layers take each triplet of features, map them to a higher (16x256) dimensional space and then reduce them to a single (1x256) dimensional space and thus effectively merging them. By using two layers with non-linearity and a higher intermediate dimensionality we ensure the merging process takes into account complex relationships among the three branches. The feature merging stage of the network is shown in Figure 3.10 and merging process is demonstrated symbolically in Figure 3.11.

3.7.3 Classification Stage

After the features from 3 input branches are merged together they are passed through 2 dense layers with ReLU activations whose weights are constrained to have a maximum norm of 2. The final classification layer has a single output with sigmoid activation, indicating the probability of malignancy. As a method of regularization, we added dropout after each trainable layer. However, dropout layers are not shown in the figure because they are optional and may not be necessary for bigger datasets.

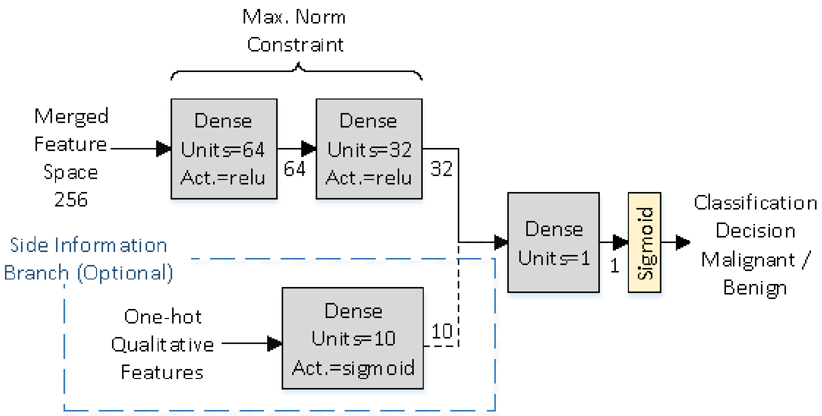

Max. Norm Constraint
Merged Feature Space 256
Dense Units=64 Act.=relu
64
Dense Units=32 Act.=relu
32
Dense Units=1
1
Sigmoid
Classification Decision Malignant / Benign
Side Information Branch (Optional)
One-hot Qualitative Features
Dense Units=10 Act.=sigmoid
10

Figure 3.12 Classification stage of the network. The side information branch is included when we want to use additional features in the classification process.

If some side-information such as the qualitative features are to be included in the classification process, another input branch is added. In case of the qualitative features, inputs are put into a one-hot format and concatenated together. We used only a single hidden layer in this branch with sigmoid activation. Outputs of the hidden layer are additionally transferred to the last dense layer which takes the classification decision. Figure 3.12 shows the final classification stage of the network.

3.8 Notes on Regularization

In order to train the proposed network on a small dataset, we had to carefully regularize the network. The capacity of the network is limited by fixing the weights of the pretrained convolutional layers. However, in the presence of more data, some of these layers may be made trainable. Further regularization effect is brought about by the norm constraints put on the dense layers, which may be relaxed for a bigger dataset.

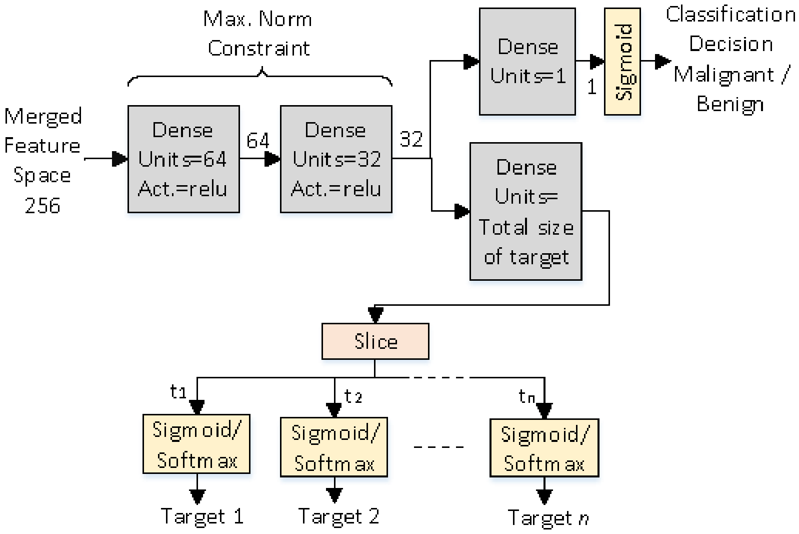


Figure 3.13 Multi-task learning setting. An output branch predicts the additional targets.

In addition to the usual forms of regularization, multi-task learning can also facilitate better generalization. Under a multi-task learning setting, a model not only learns to do a single principle task but also some additional relevant tasks. To this end, in deep neural networks, multiple branches are derived from a shared common branch. The derived branches specialize in specific tasks, whereas the common part of the network tries to learn more generalized features. The pressure to learn parameters that generalize well for multiple relevant tasks serves as a form of regularization [102]. In our case, we show that, in addition to the principle task, i.e. detection of malignancy, asking the network to predict some qualitative features of the lesion, which are identified and used by the radiologists for grounding their decisions on, results in better classification performance. To achieve this, we branched off the network just before the final dense layer, to another dense layer. After applying appropriate activation (sigmoid for binary targets and softmax for categorical targets), the outputs are used for predicting the additional targets. Figure 3.13 shows the multi-task learning setting.

3.9 Chapter Summary

In this chapter we discussed the methodology of our approach to breast lesion classification step by step. First we discussed how we extracted and processed the image frames from the ultrasound video. We mentioned why our approach does not require ROI selection or data augmentation. We discussed about some additional features that could optionally be included in the classification process. Finally we discussed about the proposed network architecture in details along with the specific design choices we made and also the reasoning behind those choices.

Chapter 4
Results and Discussions

In this chapter we present our experimental setup and evaluation strategy for measuring the performance level of our classification algorithm. Next we present the results from different experiential setups. We end this chapter by discussing the implications of the results obtained in various experiments.

4.1 Experimental Setup

We evaluated the classification performance by a 5 fold stratified cross-validation. By stratification, we imply that the compositions of types of lesions (i.e. Carcinoma, Fibroadenoma, Cyst and Inflammatory) were same in the training and test sets and they were also kept equal for each cross-validation step (Figure 4.1). As we have multiple triplets (i.e. US, UE and difference) of frames per lesion and prediction is done individually for each triplet, we need to combine these predictions to get a decision for the lesion. We found the median of the frame-wise predictions (probabilities of malignancy) produced the most robust decision statistic for the lesion. At each cross-validation step, we recorded the predictions for the test lesions and the final result is produced by combining the predictions from all steps. However, as we mentioned before, the pathologically unconfirmed cases were excluded from the final test results.

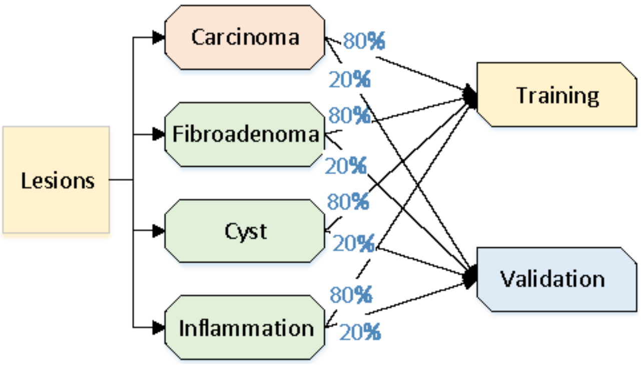


Figure 4.1 Stratified cross validation. At each validation step data from is each lesion type is split into training and validation sets maintaining the same ratio.

4.1.1 Evaluation Strategy and Classification Metrics

We calculated 5 classification metrics for evaluation of classification performance, namely accuracy, sensitivity, specificity, Positive Predictive Value (PPV) and Negative Predictive Value (NPV). To define these parameter we need to define 4 terms –

$$\mathrm{TP} = \mathrm{No.\ of\ True\ Positives\ i.e.\ the\ malignant\ lesions\ classified\ as\ malignant}$$
$$\mathrm{TN} = \mathrm{No.\ of\ True\ Negatives\ i.e.\ the\ benign\ lesions\ classified\ as\ benign}$$
$$\mathrm{FP} = \mathrm{No.\ of\ False\ Positives\ i.e.\ the\ benign\ lesions\ classified\ as\ malignant}$$
$$\mathrm{FN} = \mathrm{No.\ of\ False\ Negatives\ i.e.\ the\ malignant\ lesions\ classified\ as\ benign}$$

Then,

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

$$\mathrm{Sensitivity} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{4.2}$$

$$\mathrm{Specificity} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{4.3}$$

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{4.4}$$

$$\mathrm{NPV} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FN}} \tag{4.5}$$

To compute these 5 parameters and their sum, we took a probability threshold of 0.5 to declare a lesion either malignant or benign. For computation of the Area Under the Receiver Operating Characteristic Curve (AUROC, or AUC in short) raw probability values were used.

4.2 Baseline

As a baseline to compare our results to, we produced results by training a Stacked Denoising Auto-Encoder (SDAE) on the B-mode images in our dataset and then fine-tuning it for classification, following the method described in [28]. We chose this method because it is the only method that deals with the classification of lesions from ultrasound B-mode images on a small dataset of comparable size to ours. We tried to train a network like GoogleNet or even smaller networks like AlexNet [65] from scratch like the method described in [49] but they resulted in severe overfitting, which is to be expected as such a method requires a considerable amount of data. We could not compare our results to the method used by Zhang et al. [9] because we did not have access to Sheer Wave Elastography images for the lesions in our database.

We had to select rectangular ROIs for the lesions in order to implement the SDAE method. We found that the model failed to learn any meaningful representation without carefully selected ROI, which is to be expected from a densely connected network. The implementation method was exactly as described in [28] except we had to tune some hyper-parameters like learning rate and the number of training epochs for the best result. We also augmented the data by random flips, slight shifts and zoom which improved result. The cross-validation and evaluation procedure was exactly same a sour method, with multiple images used per lesion for better prediction

4.3 Optimization Strategy

At each cross-validation step, training is done on a set of triplets of frames obtained by combining all triplets from the lesions in the training set. The ratio of the number of malignant frames to the number of benign frames was approximately 1:3. So, to balance the classes during training we used a class weight of 3 for malignant cases and 1 for benign cases. Binary cross-entropy was used as the objective/loss function for training. In the case of multi-task learning, binary cross-entropy was used for binary targets and categorical cross-entropy was used for categorical targets. The total loss in a multi-task setting is defined as -

$$L = \lambda_{c} L_c + \lambda_{t_1} L_{t_1} + \lambda_{t_2} L_{t_2} + \cdots + \lambda_{t_n} L_{t_n} \tag{4.6}$$

Where, L_c is the classification loss, the L_{t_i} are the losses for additional tasks, and the λ are the weights for each individual loss term. In our experiments, the losses for the additional tasks were given a weight ¼ of the weight of principle classification task, i.e. $\lambda_{t_1} = \lambda_{t_2} = \cdots = \lambda_{t_n} = {}^{1}\!/_{4}\,\lambda_{c}$.

The rate of convergence was slow due to the intensive regularization and the constraints put on the network. We found that the Adam [87] optimizer worked best for speeding up the rate of convergence. We used a batch size of 1 and a very low learning rate of 10^{-5}. A single triplet of frames per batch enabled us to pass variable sized input images to the network. Also, it increased the number of stochastic updates per epoch which, for our task, increased the rate of convergence. However, the network was not trained to full convergence. This is known as early stopping [102], which is a method of regularization for neural networks. We trained our network for 14 epochs with 3000 updates per epoch. In case of multi-task setting, we trained for 18 epochs. For the last 4 epochs, the learning rate was decayed by a factor of $e^{1/2}$ each epoch. The same set of hyper-parameters and training settings were used for all cross-validation steps.

4.4 Tools and Resources Used for Implementation

The images were extracted and processed by Matlab 2017a [103]. The DNN models were implemented using python libraries Tensorflow [104] and Keras [105]. Tensorflow is a numerical library suited for large scale implementation of machine learning algorithms. It has built in automatic gradient calculation features using the backpropagation algorithm and also provides efficient implementation of convolutions, matrix multiplications and many other operations related to deep neural networks on NVIDIA CUDA [106] enabled GPUs. Keras provides a simplified (layered) interface to Tensorflow for Deep Neural Networks. The models were trained on a PC with a 6-core Intel Core i7 CPU and an NVIDIA GTX 1080ti as GPU. The convolutional part of the model was placed on GPU while the dense part was placed on CPU in order to make the implementation faster for a batch size of 1. The Imagenet weights for the VGG networks were collected from a release by Oxford University under Creative Commons License.

4.5 Classification Results

The classification results for the proposed method are presented in Table 4.1. To verify the necessity of including three branches in the proposed network to extract features from three types of images (US, UE and Difference) and to demonstrate the importance of including elastography in classification first we trained with only US branch, then with US and UE branches and finally with all three branches, while keeping the network structure similar. We show classification results for each case.

Figure 4.2 shows the Receiver Operating Characteristic curves for different models.

Table 4.1 Classification results for the proposed model

Network Branches	Accuracy (%)	Sensitivity (%)	Specificity (%)	PPV (%)}	NPV (%)	Sum (%)	F_1 (%)	AUC
US	84.332	83.607	84.615	68.000	92.958	413.511	75.000	0.90889
UE	86.175	83.607	87.179	71.831	93.151	421.943	77.273	0.89082
US+UE	88.479	88.525	88.462	75.000	95.172	435.637	81.203	0.94567
US+UE+ Diff.	91.244	88.525	92.308	81.818	95.364	449.259	85.039	0.95660

Table 4.2 Classification results for SDAE (B-mode images)

Data	Accuracy (%)	Sensitivity (%)	Specificity (%)	PPV (%)	NPV (%)	Sum (%)	F₁ (%)	AUC
All Data	82.790	61.017	91.026	72.000	86.061	392.894	72.000	0.84289
Without Inflamm-ation	85.106	69.492	92.248	80.392	86.861	414.099	74.545	0.87387

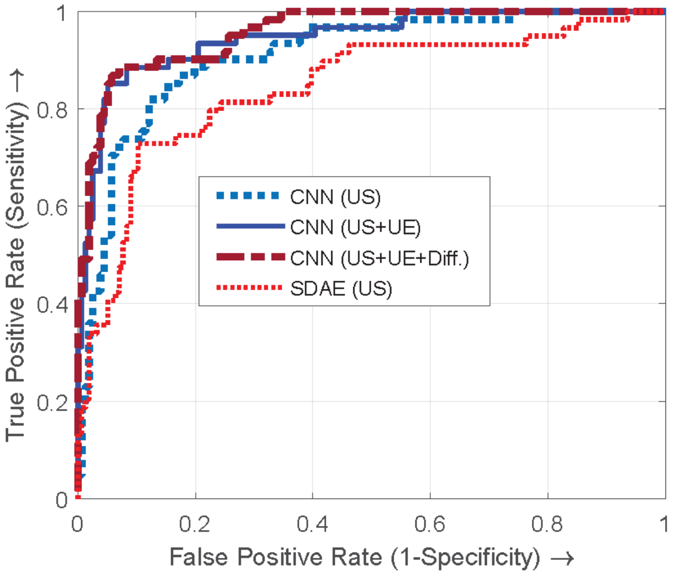


Figure 4.2 ROC curves. The area under the curves increases with each additional branch, showing their usefulness.

4.5.1 Baseline Results

Table 4.2 shows the results produced by the SDAE method. Only ROIs from US frames are used in this method, in accordance with the original work. It also shows the results produced by the SDAE method when inflammatory lesions are excluded from the dataset. This is done to verify the impact of inflammatory lesions on classification performance.

4.6 Network Performance with Representations from Different Layers of VGG-16

We stated in subsection 3.6.1 that representations of intermediate layers of VGG-16 work best for our task. To demonstrate this fact, in Figure 4.3 we have shown the classification performance of the proposed network where we used feature representations from different layers of VGG-16 architecture. We have shown four performance metrics - AUC, F_1 scores, accuracy and average (sum/5) of five parameters (Accuracy, Sensitivity, Specificity, PPV, and NPV) for each case, all of which show a clear trend. The classification result is poor when representations from very low layers are used. As we go higher the results gradually improve. We get the best results using representations from `relu_4_3'. If we go even higher, performance starts to drop. These results are consistent with the observations made in [91].

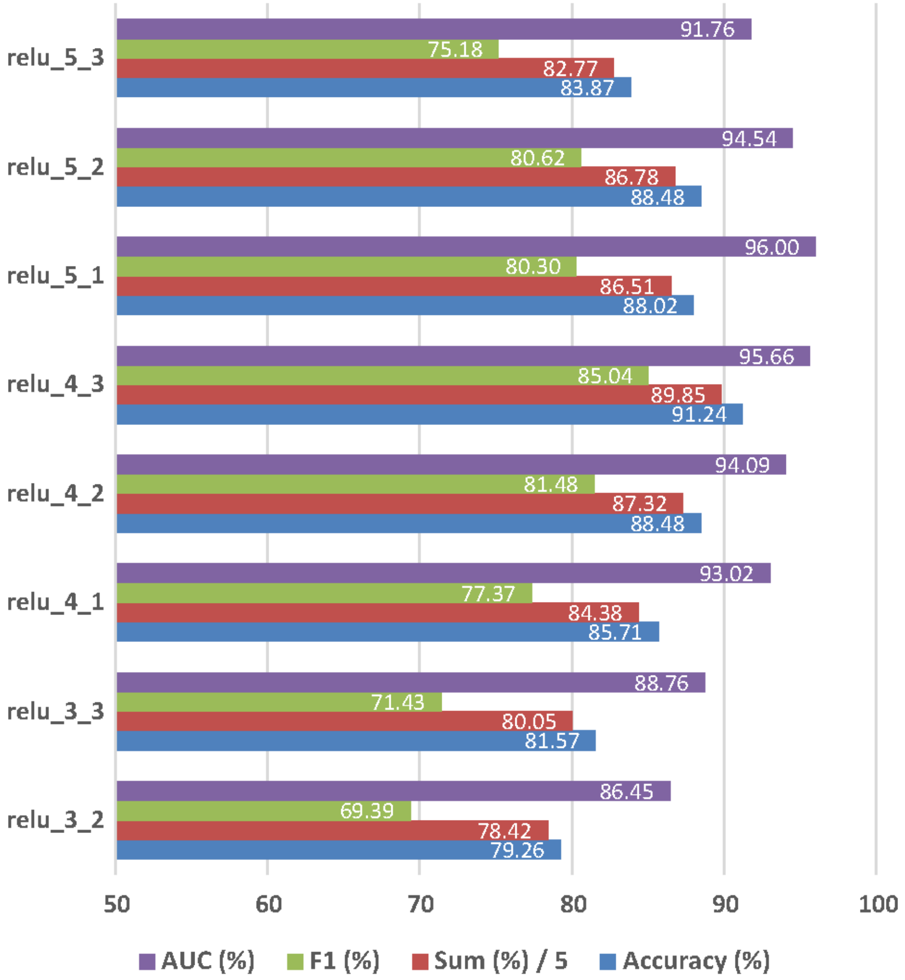


Figure 4.3 Classification results using representations from different layers of VGG-16. Best results are achieved for 'relu_4_3'.

4.7 Results Showing Impact of Different Types of Lesions

Different types of lesions possess different features, so they pose different levels of difficulty in classification. To investigate their impact, we experimented with different combinations of lesion types for classification. The results of these experiments are summarized in Table 4.3.

4.8 Results Using Additional Features and Multi-task Learning

The qualitative features were not available for all lesions in our dataset, so we had to evaluate the results on a subset of our original database for which they were present. Table 4.4 shows the results on this subset of database. It also shows the classification performance when we only use qualitative features for classification (i.e. no image). These two results show performances of image-only and feature-only methods. Table 4.4 also presents the results when the qualitative features are used as additional targets in a multi-task learning setting and when these features are used as additional inputs to the classification model.

Table 4.3 Impact of different types of lesions on classification results (Mal. = Malignant, Fib. = Fibroadenoma, Inf. = Inflammatory)

Network Branches	Accuracy (%)	Sensitivity (%)	Specificity (%)	PPV (%)}	NPV (%)	Sum (%)	F_1 (%)	AUC
Mal. vs. Fib	91.096	90.164	91.765	88.710	92.857	454.591	89.431	0.96779
Mal. vs. Cyst	93.333	93.443	93.182	95.000	91.111	466.069	94.215	0.96684
Mal. vs. Fib. and Cyst	91.579	90.164	92.248	84.615	95.200	453.806	87.302	0.96480
Mal. vs. Fib., Cst and inf.	91.244	88.525	92.308	81.818	95.364	449.259	85.039	0.95660

Table 4.4 Additional classification results

	No Qualitative Features, No Multi-task Learning	Only Qualitative Features (No Images)	Multi-task Learning of Qualitative Features	Qualitative Features as Additional Inputs
Accuracy (%)	89.302	90.698	91.163	92.019
Sensitivity (%)	86.667	86.667	88.333	88.136
Specificity (%)	90.323	92.256	92.256	93.506
PPV (%)}	77.612	81.250	81.538	83.871
NPV (%)	94.595	94.702	95.333	95.364
Sum (%)	438.498	445.574	448.625	452.896
F_1 (%)	82.645	83.871	84.800	85.039
AUC	0.98838	0.95903	0.95280	0.94035

4.9 Discussions

4.9.1 Remarks on Classification Results

As can be seen from the results presented in Table 4.1, including elastography frames in the Network, results in a significant increase in classification performance. This justifies our assumption that, in addition to the B-mode images, the stiffness distribution conveyed by quasi-static elastography images is useful for lesion classification. Adding difference images further increases performance, which demonstrates the necessity of comparative information between US and UE.

Regarding the baseline results produced in Table 4.2 for the SDAE method, we see that the accuracy of the method is comparable to that reported in [28], however, the sensitivity is much lower. This may be due to three reasons. Firstly, the dataset used in the original work contained malignant and benign lesions in almost equal proportions, whereas our dataset is greatly imbalanced with benign lesions far outnumbering malignant ones. Secondly, our dataset size is smaller than that used in the original work. And thirdly, our dataset contained many malignant lesions with obscure lesion boundaries. Although we have chosen appropriate ROIs, the obscure lesions may hamper the overall classification performance. Table 4.2 also shows the results without inflammatory lesions. Inflammatory lesions, like malignant lesions, often have obscure lesion boundaries and are harder to distinguish from malignant lesions. We see that both specificity and overall classification performance improves upon exclusion of inflammation lesions.

The relative performance levels of different approaches are apparent from the ROC curves presented in Figure 4.2 ROC curves. The area under the curves increases with each additional branch, showing their usefulness.. There is a significant increase in Area Under the Curve when CNN (the proposed method) is used instead of SDAE. The Area further increases with each additional branch of the CNN (i.e. US, UE and Difference image).

4.9.2 Impact of Different types of Benign Lesions on Classification Results

From the results presented in Table 4.3, it is evident that each type of benign lesion considered, poses different levels of difficulty in classification. As can be seen from the results, against only Fibroadenoma, which is the most prevalent type of benign lesions, malignant lesions are more easily detectable, as evidenced by the higher sensitivity and PPV which are almost equal to specificity and NPV respectively. The classification results are best for malignant vs Cyst. This is due to the fact that, Cyst lesions are easily distinguishable from malignant lesions from US and UE images.

Inflammatory lesions are known to be harder to distinguish from malignant lesions [107]. Adding them to the dataset results in a drop in sensitivity and PPV. This is due to the fact the ultrasound images of inflammatory lesions share some common features with malignant lesions. So, without them, the network faces less ambiguity and thus, predicts malignant lesions more confidently.

4.9.3 Impact of Multi-task Learning

As shown in Table 4.4, incorporating multi-task learning improves classification performance. We must emphasize that in this method the additional features are used as targets, not as inputs. So, they are not required during test/deployment. They only serve as a means of training the network to learn a more generalized model. The classification procedure remains fully objective and automated.

4.9.4 Impact of Additional Qualitative Features

By using qualitative features as inputs, we introduce subjectivity in the classification process. Another downside is that the classification process no longer remains fully automated.

However, as apparent from Table 4.4, combining them with images results in better classification performance. This performance is better than the performance levels of both image-only and feature-only approaches. We may think of this approach as combining both machine and human intuition to make classification more reliable.

4.9.5 Impact of Dataset size

To predict how the proposed network will perform for bigger datasets and to verify the generality of our approach we created shorter datasets of sizes 70-100% by keeping the composition of different types of lesions same. Also, to further verify the impact of inflammatory lesions we created shorter datasets without inflammatory lesions in a similar way. Figure 4.4 Variation of AUC with dataset size. It shows a trend of increasing AUC value with increasing dataset size. shows the impact of dataset size on classification performance from which we see a clear upward trend in AUC both with and without inflammatory lesions. This result indicates that in presence of more data our approach can achieve higher classification performance. This also justifies the scalability and generality of our approach for bigger datasets.

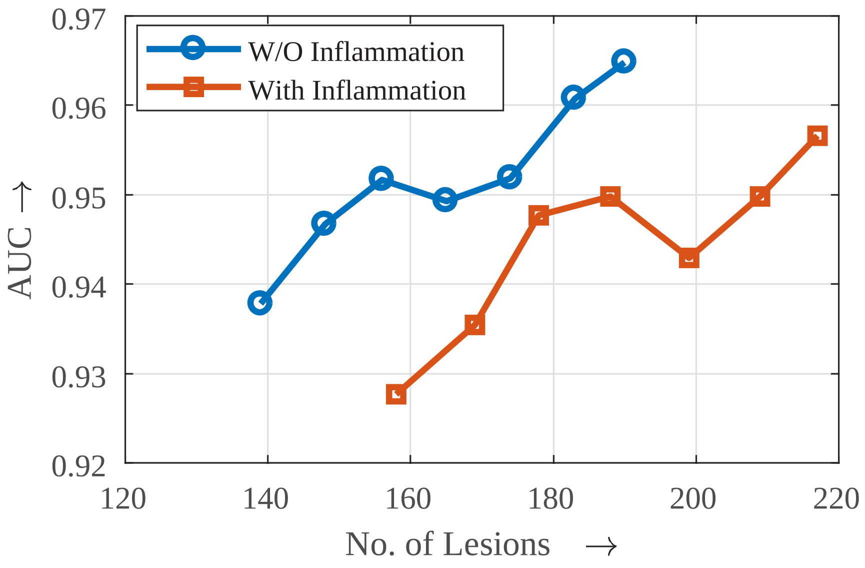


Figure 4.4 Variation of AUC with dataset size. It shows a trend of increasing AUC value with increasing dataset size.

4.10 Additional Results and Discussions

4.10.1 Learning Curves and Convergence

Due to the small size of our dataset, the proposed model faced the risk of overfitting. We had to use extensive regularization and other techniques to ensure that the model was actually learning useful features rather than memorizing the training data. Whether the model is learning as expected, can be ensured by monitoring the training and validation losses.

The training and validation losses against training steps are plotted in Figure 4.5-Figure 4.7. As we can see, both training and validation losses decrease steadily with training steps for all cross-validation steps, which proves that the model is indeed learning useful solution to the classification problem. Initially the training loss is shown to be higher than the validation loss, because of dropout, which makes the network make noisier predictions during training time. However, at the end of the training the training loss goes below validation loss, which is to be expected, because the network is capable of overfitting the data. This is where learning rate reduction process begins and slows down the overfitting process. This is evidenced by the fact that the validation loss curves flatten out at the end of the training. This also proves the effectiveness of the regularization measures taken. If we keep on training the loss curves keep oscillating unpredictably, so early stopping was used to stop the training beforehand.

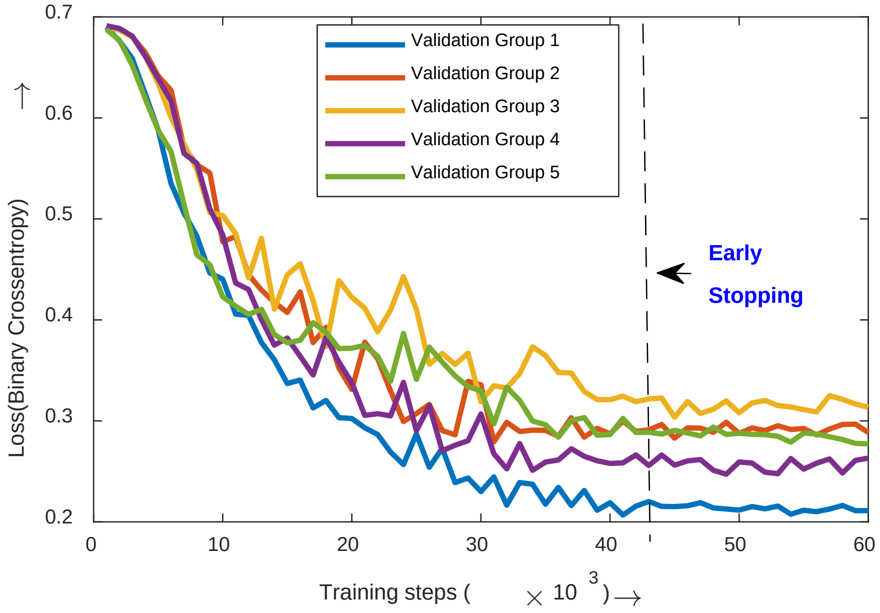


Figure 4.5 Validation loss curves for all cross-validation steps

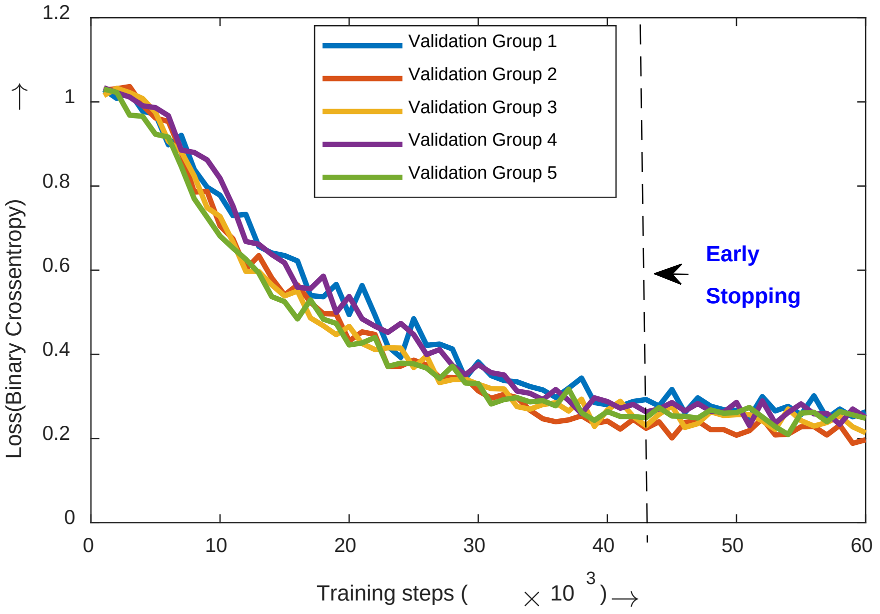


Figure 4.6 Training loss curves for all cross-validation steps

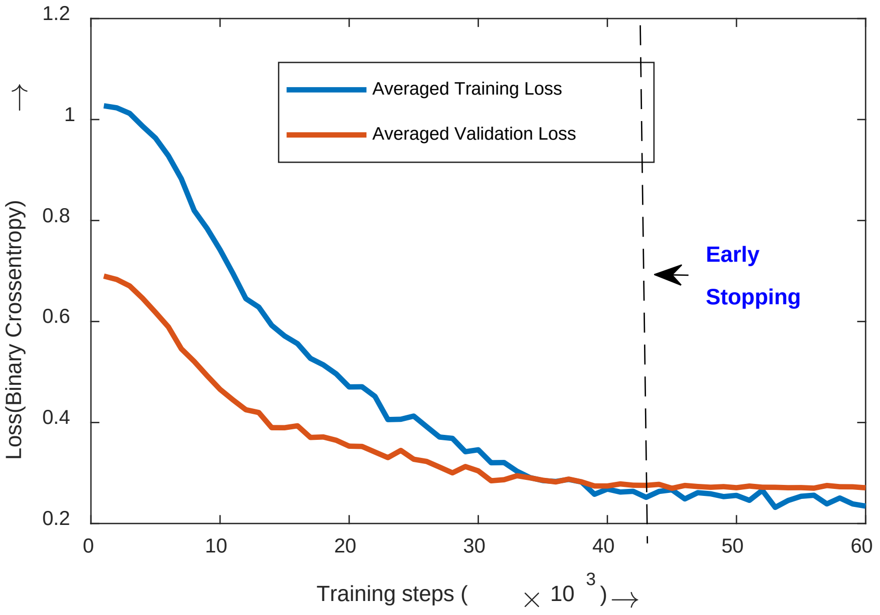


Figure 4.7 Training and validation loss curves averaged over all cross-validation steps

4.10.2 Utility of Multiple Frames per Lesion

We used multiple frames per lesions to get a more reliable prediction for each lesion and thus get a better classification performance. To verify if the use of multiple frames per lesion is justified, we chose a single triplet per lesion with most prominent UE frames and tried two approaches. In one approach we based the predictions on the most prominent frames only, in another approach we both trained and tested on the most prominent frames.

Table 4.5 Impact of multiple frames on result

	Train on single frame, test on single frame	Train on multiple frames, test on single frame	Train on multiple frames, test on multiple frames
Accuracy (%)	87.558	87.558	91.244
Sensitivity (%)	83.607	86.885	88.522
Specificity (%)	89.103	87.821	92.308
PPV (%)}	75.000	73.611	81.818
NPV (%)	93.289	94.483	95.364
Sum (%)	428.555	445.574	449.259
F_1 (%)	79.070	430.357	85.039
AUC	93232	0.94252	0.95660

As we can see from Table 4.5, by training and predicting on multiple frames, the classification performance improves significantly. This is due to the fact that the prediction becomes more robust and less prone to frame-wise variations.

4.10.3 Variations within Frame-wise Predictions

As, we have mentioned before, use of multiple frames per lesion improves the classification performance. However, to understand the impact of frame-wise variations in predictions (i.e. predicted probabilities of malignancy), we conducted a few analyses. As presented in Table 4.6, the variations in frame-wise predictions is higher for incorrectly predicted (False Positive and False Negative) results than correctly predicted (True Positive and True Negative) results.

Table 4.6 Variations in frame-wise predictions for different result types

Result type	Average Standard Deviation of frame-wise predicted probability of malignancy	Average Inter-Quartile Range (IQR) of frame-wise predicted probability of malignancy
TP	0.087217	0.078442
TN	0.069818	0.055674
FP	0.216626	0.153949
FN	0.154305	0.111173

This observation indicates that the model is "confused" about the lesions it incorrectly predicted and if we exclude the lesions with higher variance in frame-wise predictions, we may get better classification results. Figure 4.8 shows that indeed this is the case. If we set a threshold for Inter-Quartile Range of frame-wise predicted probability of malignancy, we can discard the most unreliable results, and the classification performance improves gradually as this threshold is tightened. This observation can be useful to verify whether a prediction is reliable, which is a vital requirement for medical diagnosis.

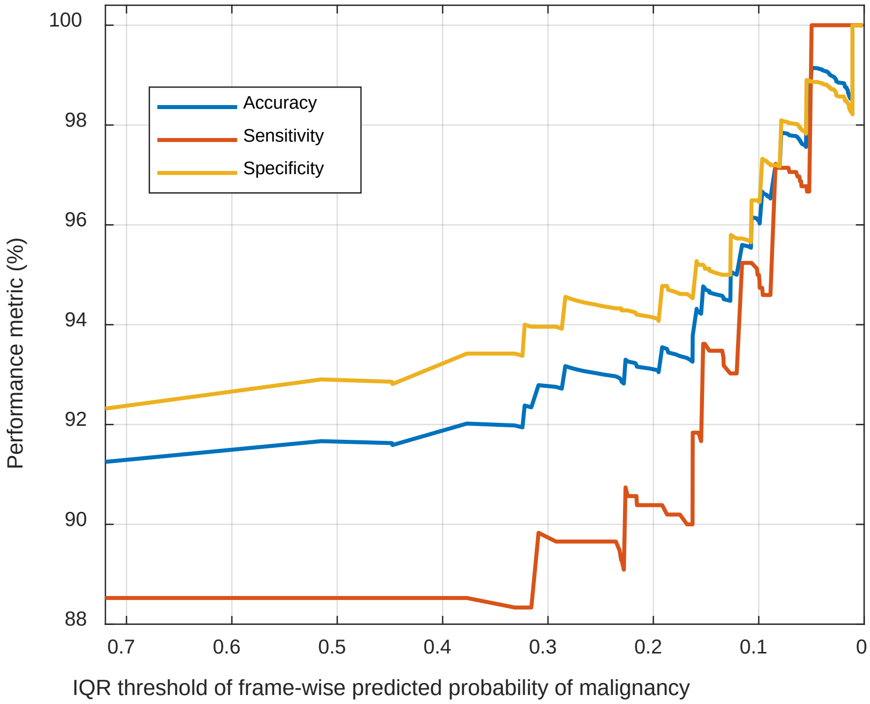


Figure 4.8 Classification performance improvement with the threshold in frame-wise variation (IQR) in predictions.

This observation leads to a question - what fraction of lesions under test would be discarded if we aimed for more reliable predictions? We hypothesize that, it would depend on the quality of the videos the frames are extracted from. If the video has frames that consistently show the important features of a lesion, the model would be less confused about that lesion. In our experiment we observed that, if we discard the 25% lesions with highest variance in frame-wise predictions, we may get very reliable predictions with accuracy, sensitivity and specificity of approximately 95%. A detailed plot is shown in Figure 4.9.

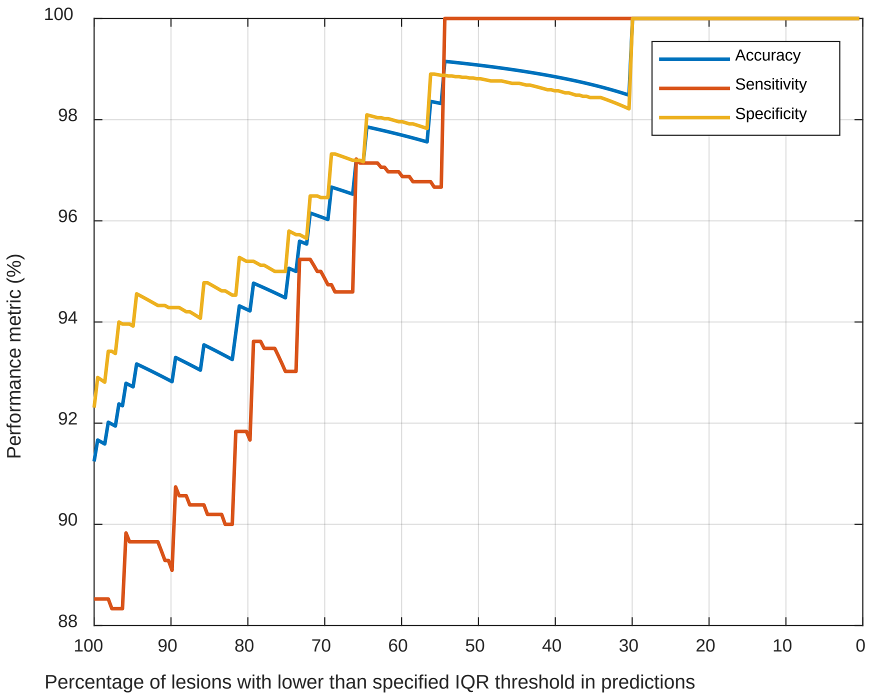


Figure 4.9 As we applied different threshold levels to the IQR of predicted probabilities of malignancy for each lesion, we analyzed what portion of the dataset satisfied that condition and what performance level was achieved within that part of the dataset.

4.10.4 Impact of Clinically Unproven Data

As we have mentioned before, we included clinically unproven data with strong radiological evidence in the training step. This was done to increase the training dataset size and thus achieve better results. However, the performance was evaluated only on clinically proven data. To justify this approach, in Table 4.7 we presented three cases. First we notice that, if we train on all data, including the clinically unproven data, but test on only the clinically proven data, better results are achieved compared to that obtained by training on only clinically proven data.

This result confirms that the extra data, even though clinically unproven, helps the model to learn more. The final set of results show that, even if we included the clinically unproven data in our evaluation, the results would not be much different from that achieved on only clinically proven data. This observation confirms that, the predictions for these lesions are consistent with the assumed label based on radiological evidence.

Table 4.7 Impact of Clinically Unproven Data on Classification Results

	Train and test on clinically proven data	Train on all data, test on only clinically proven data	Train and test on all data
Accuracy (%)	90.323	91.244	91.213
Sensitivity (%)	88.525	88.522	87.879
Specificity (%)	91.026	92.308	92.486
PPV (%)}	79.412	81.818	81.690
NPV (%)	95.302	95.364	95.238
Sum (%)	444.586	449.259	448.506
F_1 (%)	83.721	85.039	84.672
AUC	0. 95628	0.95660	0.95665

4.11 Chapter Summary

In this chapter we discussed how we evaluated the performance of the proposed model and presented the results obtained in our experiments. Then we interpreted the results to gain more insights about the impact of different choices regarding our methodology. We presented intuitive explanations of different observations and also drew different important conclusions based on those observations.

Chapter 5
Conclusions and Future Work

Breast cancer is the most frequent form of cancer in women and second most common type of cancer around the world. Detection and computer aided differentiation of breast tumors is an integral part of successful treatment strategy for patients with breast tumors. Bi-modal radiology-based imaging, i.e., conventional B-mode ultrasound and elastography, when combined, promise to reduce the number of biopsies significantly because of the detection of false malignant cases with high accuracy. We proposed a Convolutional Neural Network based Computer Aided Diagnosis system to aid lesion classification from this Bi-modal imaging system. In this chapter we review and summarize the work done so far. Then we will clarify the limitations of the proposed method and point out future prospects of improvement and expansion.

5.1 Summary

In this work, we propose a method of breast-lesion classification based on convolutional neural network that can be used for implementation of a real-time non-invasive breast lesion classification system. Such a system will provide a completely objective second opinion free of cost, directly from ultrasound B-mode and elastography images in order to help improve the reliability of breast cancer diagnosis. The proposed method requires no segmentation or feature engineering, only image frames obtained directly from the machine. We evaluated the proposed method on a clinically proven dataset of 217 patients. We explained how we exploited transfer learning from the ImageNet dataset to train the proposed network on a very small number of datapoints. Also, the experimental results indicate that our approach should achieve higher classification performance on bigger datasets. The deep learning framework with tunable regularization ensures the scalability of our approach. We have also discussed the impact of different types of benign lesions on classification performance. Finally, we trained the proposed model in a multi-task learning setting with qualitative features as additional targets, which is shown to improve the generalization of the trained model and thus to improve classification results. We also showed that including qualitative features as additional inputs further increases classification performance.

5.2 Limitations of Our Approach

The most prominent limitation of our approach is that its classification performance is lower compared to traditional morphological feature based approaches. This is due to the fact that, the proposed model is learning to differentiate lesions directly from images, rather than a handful of quantitative features, which makes it a much more difficult task to learn. To reach the performance level of traditional approach, the proposed model would need a much bigger and diverse dataset. Another limitation of our approach is that it still depends on the radiologist to correctly distinguish a lesion from normal breast tissue, since it has not been trained to automatically identify a lesion from healthy breast tissue. This was done to make the classification process simpler i.e. we wanted to limit the classification between malignant and benign lesions, and also because of the insufficient data for healthy breast tissue. Finally, to better evaluate the proposed model it should be tested on a large dataset. Classification results on a bigger and more diverse dataset would be a more reliable indicator of the true performance level of the proposed method.

5.3 Future Works

We propose some prospects of improvement and elaboration of the model –

- We showed that by combining two ultrasound imaging modalities the proposed model can achieve better classification performance. We could combine other modalities such as mammogram, MRI or shear-wave elastography with the two modalities considered in this work and see if the classification performance improves any further.
- We showed that the proposed model shows improvement in classification performance with increase in dataset size. We want to verify if this trend holds for very large datasets. More specifically, we used extensive regularization methods and avoided fine-tuning for lack of more data. If such a big dataset were available, these regularization methods could be relaxed and the previously fixed parameters/weights could be fine-tuned for better classification performance.
- Class labels for lesions are rare and difficult to collect as it requires that the patient proceed to clinical verification such as biopsy or FNAC. However, the proposed method showed improvement in classification performance with multi-task learning. If we can gather more data with the additional features, then we can train the relevant part of the network with this information even without class label (i.e. malignant/benign). This could potentially improve the classification performance for the original classification task, because the shared portion of the network would learn more generalized representations.

- Our method showed improvement in classification performance with inclusion of additional qualitative features. However, these features are subjective in nature and introduces human intervention. On a bigger dataset we could try to train a network to predict these features from images rather than collecting them from radiologists. If that could be done with satisfactory accuracy, those predictions from the auxiliary network could be used to improve the performance of the original network, while keeping the whole process free of human intervention.
- We claimed that the proposed method could be integrated in an imaging system for real-time classification. We could verify that by implementing a software version of the proposed method in an ultrasound system.

References

[1] C. DeSantis, J. Ma, L. Bryan, and A. Jemal, “Breast cancer statistics, 2013,” *CA. Cancer J. Clin.*, vol. 64, no. 1, pp. 52–62, 2014.

[2] World Health Organization, “Global Health Estimates, 2016,” 2016. [Online]. Available: https://www.who.int/cancer/detection/breastcancer/en/index1.html.

[3] World Cancer Research Fund, “Breast cancer How diet, nutrition and physical activity affect breast cancer risk.” [Online]. Available: https://www.wcrf.org/dietandcancer/breast-cancer.

[4] Centers for Disease Control and Prevention, “Leading Cancer Cases and Deaths, Female, 2015.” [Online]. Available: https://gis.cdc.gov/cancer/USCS/DataViz.html.

[5] M. P. Coleman *et al.*, “Cancer survival in five continents: a worldwide population-based study (CONCORD),” *Lancet Oncol.*, vol. 9, no. 8, pp. 730–756, 2008.

[6] L. C. H. Leong *et al.*, “A prospective study to compare the diagnostic performance of breast elastography versus conventional breast ultrasound,” *Clin. Radiol.*, vol. 65, no. 11, pp. 887–894, 2010.

[7] W. K. Moon *et al.*, “Analysis of elastographic and B-mode features at sonoelastography for breast tumor classification,” *Ultrasound Med. Biol.*, vol. 35, no. 11, pp. 1794–1802, 2009.

[8] S. R. Ara, S. K. Bashar, F. Alam, and M. K. Hasan, “EMD-DWT based transform domain feature reduction approach for quantitative multi-class classification of breast lesions,” *Ultrasonics*, vol. 80, pp. 22–33, 2017.

[9] Q. Zhang *et al.*, “Deep learning based classification of breast tumors with shear-wave elastography,” *Ultrasonics*, vol. 72, pp. 150–157, 2016.

[10] A. B. Miller, C. Wall, C. J. Baines, P. Sun, T. To, and S. A. Narod, “Twenty five year follow-up for breast cancer incidence and mortality of the Canadian National Breast Screening Study: randomised screening trial,” *Bmj*, vol. 348, p. g366, 2014.

[11] H. Zhi, B. Ou, B.-M. Luo, X. Feng, Y.-L. Wen, and H.-Y. Yang, “Comparison of ultrasound elastography, mammography, and sonography in the diagnosis of solid breast lesions,” *J. ultrasound Med.*, vol. 26, no. 6, pp. 807–815, 2007.

[12] L. W. Bassett, M. Ysrael, R. H. Gold, and C. Ysrael, “Usefulness of mammography and sonography in women less than 35 years of age.,” *Radiology*, vol. 180, no. 3, pp. 831–835, 1991.

[13] B. Alacam, B. Yazici, N. Bilgutay, F. Forsberg, and C. Piccoli, “Breast tissue characterization using FARMA modeling of ultrasonic RF echo,” *Ultrasound Med. Biol.*, vol. 30, no. 10, pp. 1397–1407, 2004.

[14] J. Ophir, I. Cespedes, H. Ponnekanti, Y. Yazdi, and X. Li, “Elastography: a quantitative method for imaging the elasticity of biological tissues,” *Ultrason. Imaging*, vol. 13, no. 2, pp. 111–134, 1991.

[15] T. Sugimoto, S. Ueha, and K. Itoh, “Tissue hardness measurement using the radiation force of focused ultrasound,” in *Ultrasonics Symposium, 1990. Proceedings., IEEE 1990*, 1990, pp. 1377–1380.

[16] G. Treece, J. Lindop, L. Chen, J. Housden, R. Prager, and A. Gee, “Real-time quasi-static ultrasound elastography,” *Interface Focus*, vol. 1, no. 4, pp. 540–552, 2011.

[17] J. Bercoff, M. Tanter, and M. Fink, “Supersonic shear imaging: a new technique for soft tissue elasticity mapping,” *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 51, no. 4, pp. 396–409, 2004.

[18] B. Zoph and Q. V Le, “Neural architecture search with reinforcement learning,” *arXiv Prepr. arXiv1611.01578*, 2016.

[19] L. Tabár *et al.*, “Swedish two-county trial: impact of mammographic screening on breast cancer mortality during 3 decades,” *Radiology*, vol. 260, no. 3, pp. 658–663, 2011.

[20] B. Sahiner *et al.*, “Malignant and benign breast masses on 3D US volumetric images: effect of computer-aided diagnosis on radiologist accuracy,” *Radiology*, vol. 242, no. 3, pp. 716–724, 2007.

[21] S. Singh, J. Maxwell, J. A. Baker, J. L. Nicholas, and J. Y. Lo, “Computer-aided classification of breast masses: performance and interobserver variability of expert radiologists versus residents,” *Radiology*, vol. 258, no. 1, pp. 73–80, 2011.

[22] M. L. Giger, N. Karssemeijer, and J. A. Schnabel, “Breast image analysis for risk assessment, detection, diagnosis, and treatment of cancer,” *Annu. Rev. Biomed. Eng.*, vol. 15, pp. 327–357, 2013.

[23] R.-F. Chang, W.-J. Wu, W. K. Moon, and D.-R. Chen, “Automatic ultrasound segmentation and morphology based diagnosis of solid breast tumors,” *Breast Cancer Res. Treat.*, vol. 89, no. 2, p. 179, 2005.

[24] S. Zhou, J. Shi, J. Zhu, Y. Cai, and R. Wang, “Shearlet-based texture feature extraction for classification of breast tumor in ultrasound image,” *Biomed. Signal Process. Control*, vol. 8, no. 6, pp. 688–696, 2013.

[25] J. Shi, S. Zhou, X. Liu, Q. Zhang, M. Lu, and T. Wang, “Stacked deep polynomial network based representation learning for tumor classification with small ultrasound image dataset,” *Neurocomputing*, vol. 194, pp. 87–94, 2016.

[26] M. Abdel-Nasser, J. Melendez, A. Moreno, O. A. Omer, and D. Puig, “Breast tumor classification in ultrasound images using texture analysis and super-resolution methods,” *Eng. Appl. Artif. Intell.*, vol. 59, pp. 84–92, 2017.

[27] S. Joo, Y. S. Yang, W. K. Moon, and H. C. Kim, “Computer-aided diagnosis of solid breast nodules: use of an artificial neural network based on multiple sonographic features,” *IEEE Trans. Med. Imaging*, vol. 23, no. 10, pp. 1292–1300, 2004.

[28] J. Z. Cheng *et al.*, “Computer-Aided Diagnosis with Deep Learning Architecture: Applications to Breast Lesions in US Images and Pulmonary Nodules in CT Scans,” *Sci. Rep.*, vol. 6, 2016.

[29] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, “Contour detection and hierarchical image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, 2011.

[30] M.-C. Yang *et al.*, “Robust texture analysis using multi-resolution gray-scale invariant features for breast sonographic tumor diagnosis,” *IEEE Trans. Med. Imaging*, vol. 32, no. 12, pp. 2262–2273, 2013.

[31] T. Sun, R. Zhang, J. Wang, X. Li, and X. Guo, “Computer-aided diagnosis for early-stage lung cancer based on longitudinal and balanced data,” *PLoS One*, vol. 8, no. 5, p. e63559, 2013.

[32] G. D. Tourassi, “Journey toward computer-aided diagnosis: role of image texture analysis,” *Radiology*, vol. 213, no. 2, pp. 317–320, 1999.

[33] S. Şahan, K. Polat, H. Kodaz, and S. Güneş, “A new hybrid method based on fuzzy-artificial immune system and k-nn algorithm for breast cancer diagnosis,” *Comput. Biol. Med.*, vol. 37, no. 3, pp. 415–423, 2007.

[34] N. Holsbach, F. S. Fogliatto, and M. J. Anzanello, “A data mining method for breast cancer identification based on a selection of variables,” *Cien. Saude Colet.*, vol. 19, no. 4, pp. 1295–1304, 2014.

[35] N. Pérez, M. A. Guevara, and A. Silva, “Improving breast cancer classification with mammography, supported on an appropriate variable selection analysis,” in *Medical Imaging 2013: Computer-Aided Diagnosis*, 2013, vol. 8670, p. 867022.

[36] N. Pérez, M. A. Guevara, A. Silva, I. Ramos, and J. Loureiro, “Improving the performance of machine learning classifiers for Breast Cancer diagnosis based on feature selection,” in *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*, 2014, pp. 209–217.

[37] M. F. Akay, “Support vector machines combined with feature selection for breast cancer diagnosis,” *Expert Syst. Appl.*, vol. 36, no. 2, pp. 3240–3247, 2009.

[38] D. Wang, L. Shi, and P. A. Heng, “Automatic detection of breast cancers in mammograms using structured support vector machines,” *Neurocomputing*, vol. 72, no. 13–15, pp. 3296–3302, 2009.

[39] A. Thomas, F. Degenhardt, A. Farrokh, S. Wojcinski, T. Slowinski, and T. Fischer, “Significant differentiation of focal breast lesions: calculation of strain ratio in breast sonoelastography,” *Acad. Radiol.*, vol. 17, no. 5, pp. 558–563, 2010.

[40] H. Zhi, X.-Y. Xiao, H.-Y. Yang, B. Ou, Y.-L. Wen, and B.-M. Luo, “Ultrasonic elastography in breast cancer diagnosis: strain ratio vs 5-point scale,” *Acad. Radiol.*, vol. 17, no. 10, pp. 1227–1233, 2010.

[41] S. R. Ara, F. Alam, M. H. Rahman, S. Akhter, R. Awwal, and M. K. Hasan, “Bimodal multiparameter-based approach for benign–malignant classification of breast tumors,” *Ultrasound Med. Biol.*, vol. 41, no. 7, pp. 2022–2038, 2015.

[42] D. T. Ginat, S. V Destounis, R. G. Barr, B. Castaneda, J. G. Strang, and D. J. Rubens, “US elastography of breast and prostate lesions,” *Radiographics*, vol. 29, no. 7, pp. 2007–2016, 2009.

[43] H.-W. Lee *et al.*, “Breast tumor classification of ultrasound images using a reversible round-off nonrecursive 1-D discrete periodic wavelet transform,” *IEEE Trans. Biomed. Eng.*, vol. 56, no. 3, pp. 880–884, 2009.

[44] W.-J. Wu and W. K. Moon, “Ultrasound breast tumor image computer-aided diagnosis with texture and morphological features,” *Acad. Radiol.*, vol. 15, no. 7, pp. 873–880, 2008.

[45] D.-R. Chen, Y.-L. Huang, and S.-H. Lin, “Computer-aided diagnosis with textural features for breast lesions in sonograms,” *Comput. Med. Imaging Graph.*, vol. 35, no. 3, pp. 220–226, 2011.

[46] H.-D. Cheng, J. Shan, W. Ju, Y. Guo, and L. Zhang, “Automated breast cancer detection and classification using ultrasound images: A survey,” *Pattern Recognit.*, vol. 43, no. 1, pp. 299–317, 2010.

[47] A. V. Alvarenga, W. C. A. Pereira, A. F. C. Infantosi, and C. M. Azevedo, “Complexity curve and grey level co-occurrence matrix in the texture evaluation of breast tumor on ultrasound images,” *Med. Phys.*, vol. 34, no. 2, pp. 379–387, 2007.

[48] F. Gómez and E. Romero, “Rotation invariant texture characterization using a curvelet based descriptor,” *Pattern Recognit. Lett.*, vol. 32, no. 16, pp. 2178–2186, 2011.

[49] S. Han *et al.*, “A deep learning framework for supporting the classification of breast lesions in ultrasound images,” *Phys. Med. Biol.*, vol. 62, no. 19, p. 7714, 2017.

[50] M. M. Jadoon, Q. Zhang, I. U. Haq, S. Butt, and A. Jadoon, “Three-class mammogram classification based on descriptive CNN features,” *Biomed Res. Int.*, vol. 2017, 2017.

[51] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, p. 436, 2015.

[52] Y. LeCun and Y. Bengio, “Convolutional networks for images, speech, and time series,” *Handb. brain theory neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[53] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, p. 533, 1986.

[54] Y. Bengio, “Learning deep architectures for AI,” *Found. trends® Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[55] H.-I. Suk, S.-W. Lee, D. Shen, and A. D. N. Initiative, “Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis,” *Neuroimage*, vol. 101, pp. 569–582, 2014.

[56] H.-I. Suk and D. Shen, “Deep learning-based feature representation for AD/MCI classification,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2013, pp. 583–590.

[57] F. Li, L. Tran, K.-H. Thung, S. Ji, D. Shen, and J. Li, “A robust deep model for improved classification of AD/MCI patients,” *IEEE J. Biomed. Heal. informatics*, vol. 19, no. 5, pp. 1610–1616, 2015.

[58] H.-I. Suk, S.-W. Lee, D. Shen, and A. D. N. Initiative, “Latent feature representation

with stacked auto-encoder for AD/MCI diagnosis," *Brain Struct. Funct.*, vol. 220, no. 2, pp. 841–859, 2015.

[59] W. Zhang *et al.*, "Deep convolutional neural networks for multi-modality isointense infant brain image segmentation," *Neuroimage*, vol. 108, pp. 214–224, 2015.

[60] H.-C. Shin, M. R. Orton, D. J. Collins, S. J. Doran, and M. O. Leach, "Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1930–1943, 2013.

[61] H. Chen, Q. Dou, X. Wang, J. Qin, and P.-A. Heng, "Mitosis Detection in Breast Cancer Histology Images via Deep Cascaded Networks.," in *AAAI*, 2016, pp. 1160–1166.

[62] J. Arevalo, A. Cruz-Roa, and F. A. González, "Hybrid image representation learning model with invariant features for basal cell carcinoma detection," in *IX international seminar on medical information processing and analysis*, 2013, vol. 8922, p. 89220M.

[63] A. A. Cruz-Roa, J. E. A. Ovalle, A. Madabhushi, and F. A. G. Osorio, "A deep learning architecture for image representation, visual interpretability and automated basal-cell carcinoma cancer detection," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2013, pp. 403–410.

[64] A. Prasoon, K. Petersen, C. Igel, F. Lauze, E. Dam, and M. Nielsen, "Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network," in *International conference on medical image computing and computer-assisted intervention*, 2013, pp. 246–253.

[65] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[66] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2009, pp. 248–255.

[67] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning.," in *AAAI*, 2017, vol. 4, p. 12.

[68] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *arXiv Prepr.*, 2017.

[69] S. Zheng *et al.*, "Conditional random fields as recurrent neural networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1529–1537.

[70] N. Tajbakhsh *et al.*, "Convolutional neural networks for medical image analysis: Full training or fine tuning?," *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1299–1312, 2016.

[71] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," 2014.

[72] C. Szegedy *et al.*, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[73] D. Meng, L. Zhang, G. Cao, W. Cao, G. Zhang, and B. Hu, “Liver fibrosis classification based on transfer learning and FCNet for ultrasound images,” *Ieee Access*, vol. 5, pp. 5804–5810, 2017.

[74] X. Liu, J. L. Song, S. H. Wang, J. W. Zhao, and Y. Q. Chen, “Learning to diagnose cirrhosis with liver capsule guided ultrasound image classification,” *Sensors*, vol. 17, no. 1, p. 149, 2017.

[75] T. M. Hassan, M. Elmogy, and E.-S. Sallam, “Diagnosis of focal liver diseases based on deep learning technique for ultrasound images,” *Arab. J. Sci. Eng.*, vol. 42, no. 8, pp. 3127–3140, 2017.

[76] J. Donahue *et al.*, “Decaf: A deep convolutional activation feature for generic visual recognition,” in *International conference on machine learning*, 2014, pp. 647–655.

[77] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “CNN features off-the-shelf: an astounding baseline for recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 806–813.

[78] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[79] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in *European Conference on Computer Vision*, 2016, pp. 694–711.

[80] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2414–2423.

[81] T. Sercu *et al.*, “Network architectures for multilingual speech representation learning,” in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, 2017, pp. 5295–5299.

[82] W. Hartmann, R. Hsiao, T. Ng, J. Ma, F. Keith, and M.-H. Siu, “Improved Single System Conversational Telephone Speech Recognition with VGG Bottleneck Features,” *Proc. Interspeech 2017*, pp. 112–116, 2017.

[83] K. Jarrett, K. Kavukcuoglu, and Y. LeCun, “What is the best multi-stage architecture for object recognition?,” in *Computer Vision, 2009 IEEE 12th International Conference on*, 2009, pp. 2146–2153.

[84] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in *Proc. icml*, 2013, vol. 30, no. 1, p. 3.

[85] Y.-T. Zhou and R. Chellappa, “Computation of optical flow using a neural network,” in *IEEE International Conference on Neural Networks*, 1988, vol. 1998, pp. 71–78.

[86] H. Robbins and S. Monro, “ªA Stochastic Approximation Method, º Annals Math,”

Statistics (Ber)., vol. 22, pp. 400–407, 1951.

[87] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv Prepr. arXiv1412.6980*, 2014.

[88] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *J. Mach. Learn. Res.*, vol. 15, 2014.

[89] R. Caruana, “Multitask learning,” *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.

[90] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *European conference on computer vision*, 2014, pp. 818–833.

[91] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, “Factors of transferability for a generic convnet representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1790–1802, 2016.

[92] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[93] A. Mahendran and A. Vedaldi, “Understanding deep image representations by inverting them,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5188–5196.

[94] L. Gatys, A. S. Ecker, and M. Bethge, “Texture synthesis using convolutional neural networks,” in *Advances in Neural Information Processing Systems*, 2015, pp. 262–270.

[95] F. S. Helmli and S. Scherer, “Adaptive shape from focus with an error estimation in light microscopy,” in *Image and Signal Processing and Analysis, 2001. ISPA 2001. Proceedings of the 2nd International Symposium on*, 2001, pp. 188–193.

[96] R. G. Barr *et al.*, “WFUMB Guidelines and Recommendations for Clinical Use of Ultrasound Elastography: Part 2: Breast,” *Ultrasound Med. Biol.*, vol. 41, no. 5, pp. 1148–1160, May 2015.

[97] D. O. Watermann *et al.*, “Three-dimensional ultrasound for the assessment of breast lesions,” *Ultrasound Obstet. Gynecol. Off. J. Int. Soc. Ultrasound Obstet. Gynecol.*, vol. 25, no. 6, pp. 592–598, 2005.

[98] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv Prepr. arXiv1409.1556*, 2014.

[99] 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*, 2016, pp. 770–778.

[100] M. Lin, Q. Chen, and S. Yan, “Network in network,” *arXiv Prepr. arXiv1312.4400*, 2013.

[101] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” in *European conference on computer vision*, 2014, pp. 346–361.

[102] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1. MIT press Cambridge, 2016.

[103] M. U. Guide, "The mathworks," *Inc., Natick, MA*, vol. 5, p. 333, 1998.

[104] M. Abadi *et al.*, "Tensorflow: a system for large-scale machine learning," in *OSDI*, 2016, vol. 16, pp. 265–283.

[105] F. Chollet, "Keras." 2015.

[106] S. Chetlur *et al.*, "cudnn: Efficient primitives for deep learning," *arXiv Prepr. arXiv1410.0759*, 2014.

[107] G. Tse, P. H. Tan, and F. Schmitt, "Fine needle aspiration cytology of the breast," *Verlag Berlin Heidelb. Springer*, 2013.