A Computer Vision Pipeline for Fully Automated Echocardiogram Interpretation

Elisabeth Sarah Lane

School of Computing and Engineering

University of West London

Submitted in partial satisfaction of the requirements for the Degree of Doctor of Philosophy in Computer Science

Supervisor: Prof. Massoud Zolgharni

Second Supervisor: Dr. Nasser Matoorian

October 2022

Acknowledgements

I have received invaluable support from the University of West London and several important people who, along my PhD journey, have been instrumental to it's completion.

Thank you, Professor Massoud Zolgharni, for imparting upon me your knowledge and leading me through frequent challenges. I am incredibly fortunate to have been given this opportunity and grateful to have received your regular criticism, accompanied by your unwavering belief that I could do this.

Special thanks to my second supervisor, Nasser Mattoorian, who has always been a constant source of support and encouragement.

I've had the pleasure of working alongside, learning from and forging friendships with, wonderful colleagues over the past three years. I wish to specifically thank Jamie Pordoy, Jevgeni Jevsikov, Eman Alajrami, Preshen Naidoo and Neda Azarmehr.

With abiding gratitude I acknowledge the friendship, love, support and escapism provided to me by Qingwei Li, Kathryn Seastron, Lianne Stileman and Debbie Cremore.

My never-ending appreciation goes to Carlos Azuaga for the many years of unfaltering love and friendship you have shown me.

To Bryan and Kathleen Wood, I am sincerely blessed to be able to call you my parents. Your constant love, encouragement, patience and understanding have carried me thus far. And to Minnie Lane, my daughter and best friend. I dedicate this PhD thesis, and all my future work, to you.

Abstract

Cardiovascular disease is the leading cause of global mortality and continues to place a significant burden, in economic and resource terms, upon health services. A 2-dimensional transthoracic echocardiogram captures high spatial and temporal images and videos of the heart and is the modality of choice for the rapid assessment of heart function and structure due to it's non-invasive nature and lack of ionising radiation.

The challenging process of analysing echocardiographic images is currently manually performed by trained experts, though this process is vulnerable to intra- and inter-observer variability and is highly time-consuming. Additionally, echocardiographic images suffer from varying degrees of noise and vary drastically in terms of image quality.

Exponential advancements in the fields of artificial intelligence, deep learning and computer vision have enabled the rapid development of automated systems capable of high-precision tasks, often out-performing human experts. This thesis aims to investigate the applicability of applying deep learning methods to automate key processes in the modern echocardiographic laboratory. Namely, view classification, quality assessment, cardiac phase detection, segmentation of the left ventricle and keypoint detection on tissue Doppler imaging strips.

State-of-the-art deep learning architectures were applied to each task, and evaluated against ground-truth annotations provided by trained experts. The datasets used throughout each Chapter are diverse and, in some cases, have been made public for the benefit of the research community. To encourage transparency and openness, all code and model weights have been published.

Should automated deep learning systems, both online (in terms of providing real-time feed-

back) and offline (behind the scenes), become integrated within clinical practice, there is great potential for improved accuracy and efficiency, thus improving patient outcomes. Furthermore, health services could save valuable resources such as time and money.

Contributors and Funding Sources

Contributors

The project collaborator for real-world patient datasets, and the provision of specialist cardiologist annotations, is the School of Medicine, Imperial College, London. The School of Medicine uses state-of-the-art ultrasound equipment and the latest techniques to support clinical teams in the diagnosis and treatment of a full range of cardiac problems.

The School of Medicine provide access to their extensive database, including echocardiograms, that have been obtained from multiple vendor platforms and collected by numerous different clinical providers.

Funding Sources

This work was supported, in part, by the British Heart Foundation, United Kingdom (Grant no. PG/19/78/34733) and the Vice Chancellor's Scholarship at the University of West London.

Contents

List of Figures

List of Tables

Acronyms

1D One-Dimensional

2D Two-Dimensional

3D Three-Dimensional

A2C Apical Two-Chamber

A3C Apical Three-Chamber

A4C Apical Four-Chamber

A5C Apical Five-Chamber

AI Artificial Intelligence

ANNs Artificial Neural Networks

AP Average Precision

AUC Area Under the ROC Curve

BPTT Backpropagation Through Time

CNN Convolutional Neural Network

CT Computed Tomography

CV Computer Vision

CVD Cardiovascular Disease

DICOM Digital Imaging and Communications in Medicine

DL Deep Learning

ECG Electrocardiogram

Echo Echocardiogram

ED End-Diastole

ES End-Systole

FN False Negative

FP False Positive

FPS Frames Per Second

FR Frame Rate

GRU Gated Recurrent Networks

HRA Health Regulatory Agency

ICDs Implantable Cardioverter Defibrillator

ILSVRC ImageNet Large Scale Visual Recognition Challenge

IoU Intersection Over Union

LTSM Long Short-Term Memory

LV Left Ventricle

mAP Mean Average Precision

ML Machine Learning

MRI Magnetic Resonance Imaging

NAS Neural Architecture Search

NHS National Health Service

PACS Picture Archiving and Communication Systems

PLAX Parasternal Long-Axis

PR Curve Precision-Recall Curve

PRF Pulse Repetition Frequency

PSAX Parasternal Short-Axis

R-CNN Region-Based Convolutional Neural Networks

ReLU Rectified Linear Unit Activation Function

RNN Recurrent Neural Network

ROC Receiver Operating Characteristic Curve

ROI Pooling Region of Interest Pooling Layer

RPN Regional Proposal Network

RV Right Ventricle

S4C Subcostal Four-Chamber

SSN Suprasternal Notch

SVM Support Vector Machine

TDI Tissue Doppler Imaging

TN True Negative

TNR True Negative Rate

TP True Positive

TPR True Positive Rate

TTE Transthoracic Echocardiogram

Chapter 1

Introduction

Population growth, coupled with modern lifestyle choices and an aging demographic, have accelerated an increase in Cardiovascular Disease (CVD) [1]. Between 1990 and 2013, global CVD mortality rose by 41% and remains the leading cause of death in high-income countries [2]. Lifestyle choice is commonly attributed to obesity, hypertension and diabetes; all precursors to serious cardiovascular conditions [3].

Heart disease worsens over time, often without noticeable symptoms, and can be difficult, if not impossible, to reverse. Regular cardiac imaging provides vital physiological insights, resulting in improved patient outcomes. However, the cost of doing so (in monetary and resource terms) is currently prohibitive to health services [4].

Accounting for 1,422,968 deaths in the UK between 2014 and 2020, CVD places a significant economic burden upon the National Health Service (NHS). Estimated to be £15.7 billion in 2004 [5], a recent study reports a 2014 cost of £17.1 billion: 1.4% of the national GDP [6]. This figure is expected to rise to £30 billion by 2030.

Performed by trained operators, a Two-Dimensional (2D) Transthoracic Echocardiogram (TTE) uses ultrasound to capture high spatial and temporal resolution images and videos of the heart. An Echocardiogram (Echo) is the imaging modality of choice in cardiovascular medicine for the rapid assessment of heart function and structure due to its non-invasive nature and lack of ionising radiation [7]. Figure 1.1 provides an illustration of a typical Echo examination.

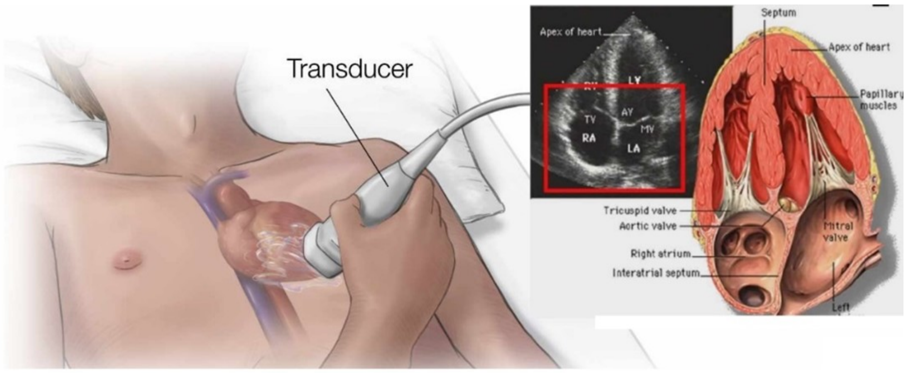


Figure 1.1: A diagram showing a typical echocardiographic exam [8]

Assessment of Left Ventricle (LV) function is of principal importance during an echo examination and is crucial for accurate patient evaluation. Echocardiography continues to be the most common technique in clinical practice for the quantification of LV function markers, such as ejection fraction (EF) and global longitudinal strain (GLS) [9].

The organisation, annotation and interpretation of echo images and videos remains a manual, human-led process. Figure 1.2 presents a high-level graphical overview of the modern clinical cardiac laboratory. First, the examination is performed and multiple images and videos are captured. Second, manual analysis of the captured frames is performed by human experts, such as such as tracing the boundaries of the heart or selecting fiducial points on the images. Finally, measuring and interpreting the clinical markers is performed to determine the presence of CVD.

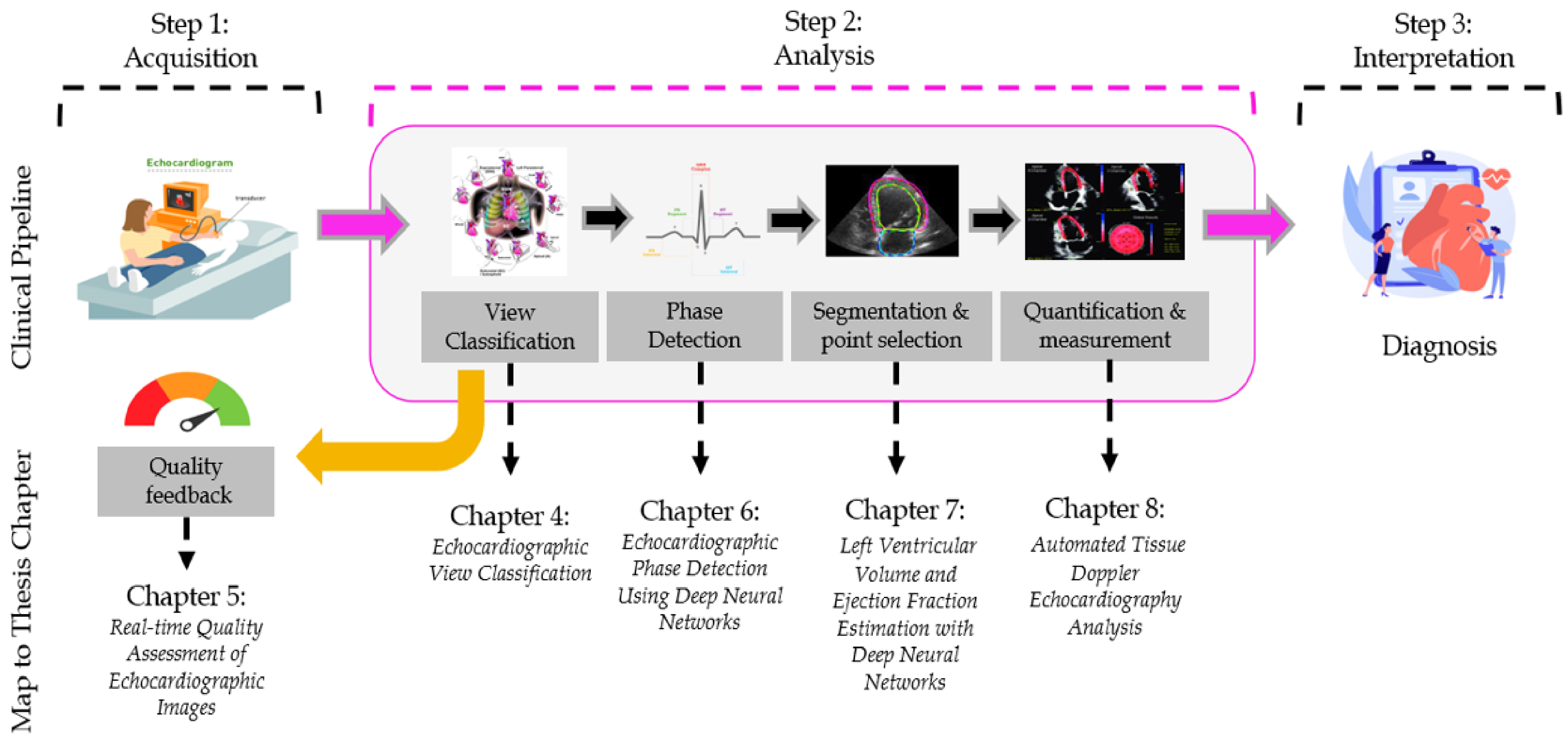


Figure 1.2: A high-level overview of the three steps included a typical, human-led echocardiogram examination pipeline, mapped to the chapters in this thesis

Rapid technological advancements in the fields of Machine Learning (ML) and Deep Learning (DL) have accelerated research into the application of Computer Vision (CV) algorithms for accurate automation of human-led tasks. Such as:

- classification of echocardiographic views [10–16] (Chapter 4)
- quality assessment of echo images [12, 17–22] (Chapter 5)
- detection of key phases in the cardiac cycle [16, 23–25] (Chapter 6)
- segmentation of the LV and automation of vital diagnostic calculations, such as EF [9, 26–34] (Chapter 7)
- measuring peak velocitites from Tissue Doppler Imaging (TDI) images [35–49] (Chapter 8)

Automated systems, without a reliance upon Electrocardiogram (ECG), would allow new protocols to be developed in clinical practice, thus reducing undesirable operator variability which can lead to diagnostic errors. Not only would such systems save valuable resources for health services, but they have potential to lead to improved patient outcomes. Without a reliance upon manual visual detection, specialists' time can be better spent acquiring more high-quality beats, reducing subjectivity and cost.

Furthermore, due to the increasing popularity of portable, lightweight echocardiographic scanners, focused studies can be performed in a variety of settings, including emergency care, lasting just a few minutes [50]. The need for automated echo analysis, independent from the ECG signal, is prevalent when implementing such automated technology in handheld devices [23, 51].

In this chapter, a problem statement for each automated echocardiographic laboratory tasks (listed above) is presented. Additionally, the motivation, main aim and objectives, research questions and contributions of this work are provided alongside an outline of the thesis structure and publications.

1.1 Problem Statement

Echocardiography constitutes the foremost prognostic and diagnostic imaging modality in modern clinical cardiology due to being non-invasive, low cost and extensively available. Traditionally, echo images are manually analysed by trained specialists, however such processes are both time-consuming and error prone. Novel DL networks increasingly demonstrate their ability to automate complex, manual processing and annotation tasks and have potential to transform clinical practices.

The focus of this thesis is on the application of state-of-the-art DL algorithms to leverage high resolution spatial and temporal data within echo images and video recordings to automate crucial processes in the echocardiographic laboratory with reproducability, speed and efficiency outperforming that of trained human experts.

Here follows a problem statement for each study comprising this research project:

1.1.1 Echocardiographic View Classification

The first step in interpreting echocardiographic data (step 2 in Figure 1.2) is manual classification of images and videos into their corresponding views. Standard practice dictates images are captured from several acquisition angles for full assessment of the complex cardiac structure. This is achieved by placing the ultrasound probe in different positions, and angles, on the patient's chest, as illustrated in Figure 1.3.

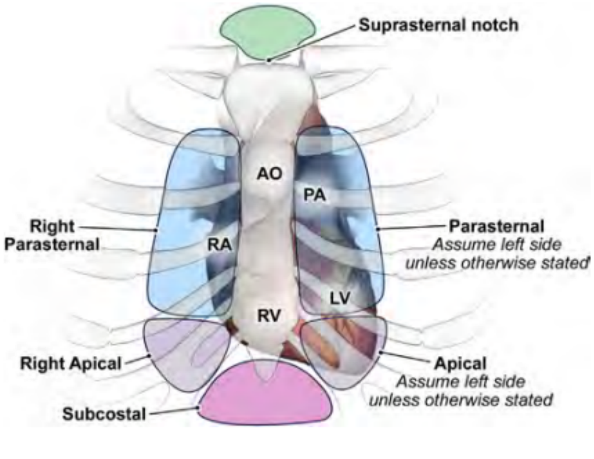


Figure 1.3: Echocardiographic windows used to obtain images from several standard views [52]

Each standard view can be considered a class, with the angle and rotation of the probe determining the view plane; equivalent to sub-classes of the main views. A comprehensive TTE examination can involve capturing up to 27 views, with 15 - 20 key views considered standard [53]. Examples of four standard echocardiographic views can be observed in Figure 1.4.

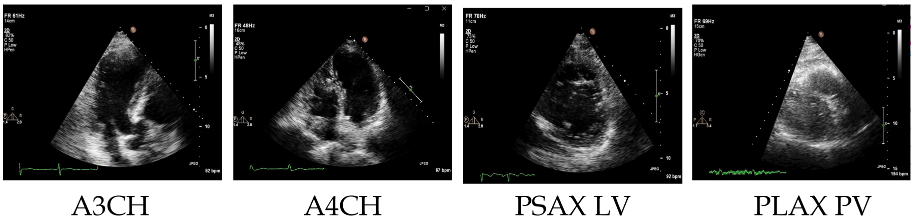


Figure 1.4: An example of standard echocardiographic views: Apical 3-chamber, Apical 4-chamber, Parasternal short-axis left ventricle and Parasternal long-axis pulminary valve

Echo data is sometimes incorrectly classified due to subtle differences in image properties, ambiguous to the human eye. When assessing important patient characteristics, such as regional wall motion abnormalities, crucial information from up to 7 views (apical two - four chamber, parasternal short-axis at the mitral valve, papillary muscles and apex and parasternal long-axis) should be combined to make an informed judgement of the LV [53]. Hence, accurate echocardiographic view classification is essential as some anatomical abnormalities are only detected by distinct views.

Ordinarily, examination data from different imaging modalities is stored in a Picture Archiving and Communication Systems (PACS) database. Software exists to partially automate the analysis and diagnosis steps, such as EchoPAC (GE Healthcare) and QLAB (Philips) [16], however these applications still rely upon varying degrees of human interaction. Considering the manual annotation of echo views is challenging to human experts and error prone due to subtle image differences and background noise, an accurate, automated system for classification of a large number of core views could have significant impact in clinical practice and save valuable resources for health services.

The automation of echocardiographic view classification is explored in more detail in Chapter 4.

1.1.2 Real-time Quality Assessment of Echocardiographic Images

A recent report suggests that up to 10% to 15% of routine echo images suffer from poor quality [54]. Currently, quality is assessed by visual on-screen inspection by human experts. Image quality significantly impacts the reliability of crucial diagnostic measurements, in addition to intra- and inter-observer variability [54–56] due to inadequate visualisation of the LV endocardial border.

The manufacturers of ultrasound devices are continuously striving to improve the quality of images captured with their machines. However, not all medical centres have access to the most modern devices and vendors vary. Subsequently, assessment of quality is inconsistent [57] and largely subjective. Furthermore, technological advancements over the past two decades have given rise to the emergence of hand-held echo devices used at the point of care, often by non-specialists.

Therefore, a novel system for echo image quality assessment, considering their performance during common DL tasks (such as keypoint detection and segmentation) as a metric for image quality, as opposed to visual human opinion, could provide valuable real-time feedback to the sonographer, using any ultrasound device, at the point of care and would be highly desirable.

A novel method of assessing image quality using DL approaches and providing real-time feedback via a web application is discussed in Chapter 5.

1.1.3 Echocardiographic Phase Detection Using Deep Neural Networks

A full cardiac cycle consists of every event occurring as the heart beats. It includes two important phases: diastole, as the heart relaxes, and systole, subsequent contraction. Measurements relate to time points, most notably End-Diastole (ED) and End-Systole (ES). Therefore, accurate detection of the end of the LV systole and diastole phases constitutes a critical step in the interpretation of echo data.

Manual frame identification from echocardiographic videos (cine loops) is performed by trained clinicians via on-screen visual selection and is subject to uncertainty, affecting crucial clinical measurements. Due to subtle frame-on-frame spatial differences, and complex temporal relationships virtually invisible to the human eye, manual detection presents a significant barrier to consistent diagnosis due to intra- and inter-observer variability lacking reproducibility and precision [58].

An ECG signal can be collected from the patient in parallel to an echo examination. Despite providing some useful data for the computation of clinically relevant parameters, ECG examinations require the connection of multiple cables which is not only time consuming but often inconvenient. Therefore, the ECG trace in the Doppler images could be missing or unreliable.

Research has shown an error of just two to three frames in detecting the ES phase elicits an approximate 10% difference in crucial diagnostic calculations [59]. The consequence of misidentification of ED and ES frames can be extensive; impairing concordance between observers in both research and clinical practice [60]. Therefore, automated methods for the resolution of accurate ED and ES phase detection, without reliance upon an ECG signal, could greatly contribute to improving the consistency of echocardiographic quantification.

Automated echocardiographic phase detection is discussed in more detail in Chapter 6.

1.1.4 Left Ventricular Volume and Ejection Fraction Estimation With Deep Neural Networks

Echo images are routinely applied for the assessment of LV ejection fraction (LVEF) and the quantification of LV systolic function [57]. In cardiac diagnostics, LV volume and ejection fraction (EF) are essential metrics. Both require manual tracing of the endocardial border by trained human experts from echocardiographic video frames representing the ED and ES frames.

This process is time consuming and suffers from high levels of observer variability. Thus, fundamentally compromising the accuracy of routine LVEF assessment in clinical practice [57, 61, 62]. Therefore, automated algorithms are desired for accurate, objective, and efficient EF measurements.

Automated segmentation of the LV and calculation of volume and EF is presented in Chapter 7.

1.1.5 Automated Tissue Doppler Echocardiography Analysis

TDI is a crucial echocardiographic technique; acquisition and interpretation is performed by human experts who visually localise pixels representing S', E' and A' peak velocities. This is an extremely subjective manual process, suffering from inter- and intra-observer variability vastly hindering detection of accurate measurements [63].

Current clinical guidelines recommend averaging peak velocity measurements over a minimum of three consecutive heartbeats [64–66]. However, this is highly disruptive to workflow. Accurate analysis of multiple beats from long, uninterrupted tissue Doppler recordings, without the need for an ECG signal, would allow new protocols to be developed in clinical practice, thus reducing undesirable operator variability which can lead to diagnostic errors. It also has potential to lead to improved patient outcomes by averaging peak tissue Doppler measurements over a greater number of heartbeats.

Automated tissue Doppler analysis is discussed in Chapter 8.

1.2 Motivation

The reliance upon manual, human analysis and interpretation of echocardiographic imaging data has limited it's full potential for precision medicine. Recent advancements in DL algorithms have demonstrated their applicability to develop automated systems capable of accurate interpretation [67] but have not yet been extensively applied to echocardiograms. In large part, due to the complex multi-view nature of echo data.

Echo view classification is a requisite process during the interpretation of post-acquisition videos and still images. This is a time-consuming, manual process performed by trained operators and is prone to error due to inter- and intra-observer variability and very subtle differences between classes and sub-classes of view types [68].

Image quality score is a subjective process based upon the opinion and level of experience of the human expert providing visual on-screen assessment, and limited by the ultrasound device used. It has been demonstrated the quality of an echo image, in addition to inter- and intra-observer variability, significantly impacts the reliability of crucial diagnostic calculations [54]

Current clinical recommendations state measurements should be averaged over three or more consecutive beats for patients in sinus rhythm. For those with an irregular rhythm, at least five consecutive beats should be considered [69].

Identification of ED and ES frames from echo cine loops is a user-dependent method performed by manual on-screen selection, or visual detection, by highly trained clinicians. Due to subtle frame-on-frame spatial differences and complex temporal relationships virtually invisible to the human eye, manual frame detection presents a significant barrier to consistent diagnosis due to intra- and inter-observer variability lacking reproducibility and precision [58]. Research has shown the medial disagreement between accredited and experienced cardiology experts is 3 frames [51]. An error of just 2-3 frames can result in approximately 10% difference in measurements causing potential misdiagnosis and the provision of ineffective medical intervention.

LVEF quantifies the fraction of chamber volume ejected in the systole phase of the cardiac cycle, relative to the volume of blood in the ventricle at the end of the diastole phase [70]. The Simpson's method for calculating EF is widely used and recommended by the American Society of Echocardiography and the European Association of Cardiovascular Imaging [71, 72]. However, the method requires manual delineation the endocardium border of the LV. Not only is this process time consuming, but it suffers from observer variability, fundamentally compromising the accuracy of routine LVEF assessment in clinical practice [57, 61, 62].

TDI measurements are a useful prognostic and diagnostic tool obtained during a standard TTE, particularly when assessing the diastolic and systolic LV, right ventricular and atrial function. Trained experts are required to manually annotate pixels representing S', E' and A' peak velocities along the doppler trace [73, 74]. This subjective process is impacted by inter- and intra-observer variability which has been proven as a major source of error in recording peak Doppler velocity measurements [63]. Therefore, echocardiographers tend to select one heartbeat they consider an average representative sample which may contribute significantly to test-retest variability [75].

1.3 Research Questions

1. What are the state-of-the-art DL techniques for image and video classification?
2. Can such techniques be applied to automate identification of a large number of echocardiographic images and video?
3. Can echocardiographic image quality be assessed by using performance of images for common DL tasks as the ground-truth?
4. Is it possible to increase the inference speed of a DL model enough to provide real-time feedback via a web application?
5. Can a DL network be developed for automated ED and ES phase detection from cine loops of arbitrary length, taking into consideration both spatial and temporal information?
6. Are DL models generalisable when tested on diverse, multi-centre data?
7. Are well-established evaluation metrics for the evaluation of LVEF adequate indicators of accuracy, or could a new approach for assessing volume be more effective?
8. Is it possible to reconstruct Tissue Doppler images into long, interrupted strips and perform automated keypoint detection of vital velocity measurements using DL methods?

1.4 Research Aims and Objectives

The aim of this research project is to develop several DL models capable of reliably automating key processes in the modern echocardiographic laboratory. Specifically, view classification, image quality assessment, detection of cardiac phases from cine loops, segmenting the LV and calculating crucial diagnostic metrics and automatic detection of velocity measurements from tissue Doppler images.

Online analysis of echocardiographic images and videos is important and requires automated models be quick in terms of processing speed, particularly when providing real-time operator feedback. However, offline analysis performed post-examination is equally important and, if implemented in a care setting, has potential to transform routine clinical practice.

The main research objectives are as follows:

1. to evaluate the current state-of-the-art DL architectures for video and image classification and develop an automated algorithm for accurate identification of 20 echocardiographic views

2. to understand the complexity of echocardiographic image quality assessment with regard to DL task performance and develop a multi-task learning network capable of providing meaningful and accurate feedback via a real-time web application, to be used in conjunction with various device manufacturers

3. to develop an automated DL algorithm for detecting ED and ES frames from cine loops of arbitrary length and evaluate the accuracy and generalisability when running inference on multi-centre datasets

4. to leverage the performance of the U-Net architecture for automated segmentation of the LV and investigate the impact of common evaluation metrics upon the accuracy of crucial diagnostic measurements, when compared with a novel volume calculation algorithm

5. to detect multiple heartbeats from long TDI recordings and subsequently identify Cartesian coordinates representing important velocity markers

6. to evaluate the performance of all developed DL models and algorithms by conducting extensive experiments and statistical analyses against the accompanying expert ground-truth annotations

7. to pre-process numerous echocardiographic datasets and accompanying expert annotations and make them available for the benefit of the research community and benchmarking of published results, thus promoting transparency and collaboration

1.5 Overview of Completed Work and Thesis Structure

This thesis is comprised of eight chapters, the structure is as follows:

- **Chapter 2: Clinical Background** - provides an overview of the clinical background to this thesis project, including an introduction to the cardiovascular system, the anatomy and physiology of the heart and cardiovascular diseases. Additionally, an outline of cardiovascular tests and procedures is provided along with a detailed introduction to echocardiography, display modes and views

- **Chapter 3: Technical Background** - the technical background chapter serves as an introduction to AI, ML and DL. Convolutional Neural Networks (CNNs) are discussed, with each of the key layers, or 'building blocks', explained. State-of-the-art classification, object detection and segmentation, and Recurrent Neural Networks (RNNs) are summarised alongside parameters, hyper-parameters, regularisation techniques and evaluation metrics. DL programming languages, frameworks and packages pertinent to this study are also discussed

- **Chapter 4: Echocardiographic View Classification** - this chapter provides detailed information regarding the complex task of echocardiographic view classification alongside a description of CNN and CNN-RNN models developed for image and video classification. The dataset, evaluation metrics and results are also discussed

- **Chapter 5: Real-time Quality Assessment of Echocardiographic Images** - a novel method of assessing echocardiographic image quality is presented using performance against common DL tasks. A web-based application for real-time operator feedback is also presented along with future work and further development

- **Chapter 6: Echocardiographic Phase Detection Using Deep Neural Networks**

- throughout this chapter, a CNN-RNN model for the detection of multiple ED and ES frames in Apical Four-Chamber (A4C) echocardiographic videos of arbitrary length length is described. Generalisability of the developed network is tested against multi-centre data, not used during training and validation of the models. The results are evaluated and a conclusion, including suggestions for future work, is provided

- **Chapter 7: Left Ventricular Volume and Ejection Fraction Estimation With Deep Neural Networks** - throughout this chapter, the task of automatically segmenting the LV is discussed alongside results from the popular U-Net architecture and the phase detection network presented in Chapter 6. A novel method for calculating the volume of the LV at ED and ES stages, following the Simpson's method, is also proposed. Future work and limitations are discussed

- **Chapter 8: Automated Tissue Doppler Echocardiography Analysis** - A project applying object detection and landmark localisation of Cartesian coordinates representing peak velocity measurements on reconstructed, long TDI strips is explained throughout this chapter. The dataset is detailed as well as the results and an in-depth discussion

1.6 Contributions

1.6.1 Contributions to Knowledge

The main contributions to knowledge for each of the studies comprising this research project can be summarised as follows:

Echocardiographic View Classification

This chapter investigates the feasibility of classifying a large number of TTE views from echo video sequences and still Doppler images using a CNN-RNN algorithm, leveraging both spatial and temporal information. The main contributions of this research can be summarised as:

- Classifying the largest number of echo views (20), when compared with recently published studies, including 15 videos and 5 Doppler images

- Proposing a CNN-RNN architecture to successfully classify sub-classes of standard TTE views, with each view considered a separate class
- Comparing the accuracy of four time-distributed state-of-the-art CNNs for feature extraction using, with the addition of LSTM and GRU layers, achieving 92.6% accuracy for Doppler image classification and 98.5% overall accuracy for videos (higher than any other previously published study)
- Investigates an efficient technique for addressing imbalanced video datasets and demonstrates the approach does not impact upon model accuracy

Real-time Quality Assessment of Echocardiographic Images

This chapter presents a multi-task DL model for echocardiographic image quality assessment using a novel labelling methodology and outlines the architecture of a web-based application to provide real-time feedback to operators. The main contributions are:

- Developing a novel, objective image quality assessment method based on the suitability of each image for downstream automated DL clinical measurements
- Investigating the applicability of the proposed method using segmentation and landmark detection tasks
- Investigating the feasibility of providing a web-based application for real-time image quality assessment feedback

Echocardiographic Phase Detection Using Deep Neural Networks

This Chapter outlines a CNN-RNN model for the accurate prediction of ED and ES frames in arbitrary length cine loops. The main contributions of this research project can be summarised as being the first study of its kind to:

- investigate the feasibility of using a deep learning framework to detect ED and ES frames in echocardiographic videos of arbitrary length, containing several heartbeats
- demonstrate the applicability of the developed framework by including several patient

datasets from various clinical centres, where one dataset was used for model development and the others used for testing

- use annotations (ground-truth) from several cardiologist experts, allowing for the examination of inter- and intra-observer variability

- include performance reports on a publicly available dataset, thereby providing a benchmark for future studies

Left Ventricular Volume and Ejection Fraction Estimation With Deep Neural Networks

This chapter details the application of three independent datasets (two public and one private) for segmenting the LV endocardial border and proposes a novel method of volume estimation based on the Simpson's rule. The main contributions are as follows:

- comparing the results of three, large segmentation datasets for assessing LVEF using commonly applied evaluation methods and proposing a novel method of calculating volume as a more informative metric, based upon the Simpson's rule

- investigating the effect upon EF calculations based upon the selected ED and ES frames, comparing expert ground-truth annotations with the phase detection network detailed in Chapter 6

- identifying the importance of data pre-processing strategy in improving model performance, paying particular attention to generalisability

Automated Tissue Doppler Echocardiography Analysis

This study presents a DL pipeline for the automated detection of peak velocity measurements from TDI strips of arbitrary length, containing varying numbers of beats. The main contributions are:

- acquired, prepared, and made publically available a dataset of tissue Doppler images, each annotated by three accredited and experienced cardiology experts, to be used for deep learning developments

- investigated the feasibility of using convolutional neural networks (CNN) to isolate complete heartbeats from TDI strips of arbitrary length, independently from the ECG information

- achieved accurate landmark localisation, to the pixel, for S', E' and A' peak velocities for each isolated heartbeat

1.6.2 Open-Source Contributions

Each of the sub-projects included in this research project represent an open-source contribution with regard to code, model weights and datasets. A summary of key open-source contributions is as follows:

Code and trained model weights

The code and trained model weights for for all DL models is available to access on GitHub. A description of all projects can be viewed on the IntSaV research group website.

Datasets

To encourage transparency, for benchmarking of results and for the general benefit of the research community, the following datasets have been prepared alongside accompanying expert ground-truth annotations. Access can be requested via the appropriate page on the IntSav research group website:

- **Echocardiographic Phase Detection Using Deep Neural Networks:** The "Multibeat-dataset"

- **Automated Tissue Doppler Echocardiography Analysis:** The long, reconstructed TDI strips dataset

1.7 Publications

1.7.1 Journal Publications

Lane, E., Jevsikov, J., Naidoo, P., Shun-shin, M., Francis, D. and Zolgharni, M., 2022. Automated Echocardiographic View Classification Using Deep Neural Networks. *tbc* [in preparation]

Lane, E., Jevsikov, J., Dhutia, N., Shun-shin, M., Francis, D. and Zolgharni, M., 2022. Automated Multibeat Tissue Doppler Echocardiography Analysis Using Deep Neural Networks. *Medical & Biological Engineering & Computing.*

Lane, E., Azarmehr, N., Jevsikov, J., Howard, J., Shun-shin, M., Cole, G., Francis, D. and Zolgharni, M., 2021. Multibeat echocardiographic phase detection using deep neural networks. *Computers in Biology and Medicine*, 133, p.104373.

Azarmehr, N., Zolgharni, M., Ye, X., Howard, J., Lane, E., Labs, R., Shun-Shin, M., Cole, G., Bidaut, L. and Francis, D., 2021. Neural architecture search of echocardiography view classifiers. *Journal of Medical Imaging*, 8(03).

1.7.2 Conference Proceedings

Lane, E., Azarmehr, N., Jevsikov, J., Howard, J., Shun-shin, M., Cole, G., Francis, D. and Zolgharni, M., 2021. Echocardiographic Phase Detection Using Neural Networks. In: *Medical Imaging with Deep Learning.*

Lane, E., Jevsikov, J., Dhutia, N., Shun-shin, M., Francis, D. and Zolgharni, M., 2022. Automated Multibeat Tissue Doppler Echocardiography Analysis Using Deep Neural Networks. In: *Medical Imaging with Deep Learning.*

Jevsikov, J., Lane, E., Stowell, C., Shun-shin, M., Francis, D. and Zolgharni, M., 2022. Automated Analysis of Mitral Inflow Doppler using Convolutional Neural Networks. In: *Medical Imaging with Deep Learning.*

Naidoo, P., Alajrami, E., Lane, E., Jevsikov, J., Shun-shin, M., Francis, D. and Zolgharni, M.,

2022. Influence of Loss Function on Left Ventricular Volume and Ejection Fraction Estimation in Deep Neural Networks. In: *Medical Imaging with Deep Learning*.

Chapter 2

Clinical Background

2.1 Introduction

Cardiology is a branch of internal medicine related to the study and treatment of the heart and associated blood vessels. If a person experiences symptoms of cardiovascular disease (CVD), they will be referred to visit a cardiologist. CVD is the leading cause of mortality in high-income countries and represents a significant resource and monetary cost to health services.

Recent advancements in the invent of non-invasive medical imaging tools have broadened access to life-saving interventions and led to the improved quality of anatomical visualisations. Throughout this chapter an overview of cardiology will be presented, including a brief introduction to the cardiovascular system, anatomy and physiology of the heart and a short summary of common CVD. Several cardiovascular tests and procedures are detailed, with an emphasis upon echocardiographic imaging modes, which is the focus of this study.

2.2 Overview of Cardiology

2.2.1 The Cardiovascular System

The main components of the cardiovascular system (also called the circulatory system) are the heart, blood vessels and blood. Materials transported and delivered throughout the cardiovascular system include nutrients from the digestive system, oxygen from the lungs, hormones from the endocrine system and waste from cells [76].

There are, in fact, two interconnected circulatory systems within the human body: Systemic (providing oxygen rich blood to organs, tissues and cells) and Pulmonary (where the fresh oxygen we breathe enters the blood stream), as illustrated in Figure 2.1

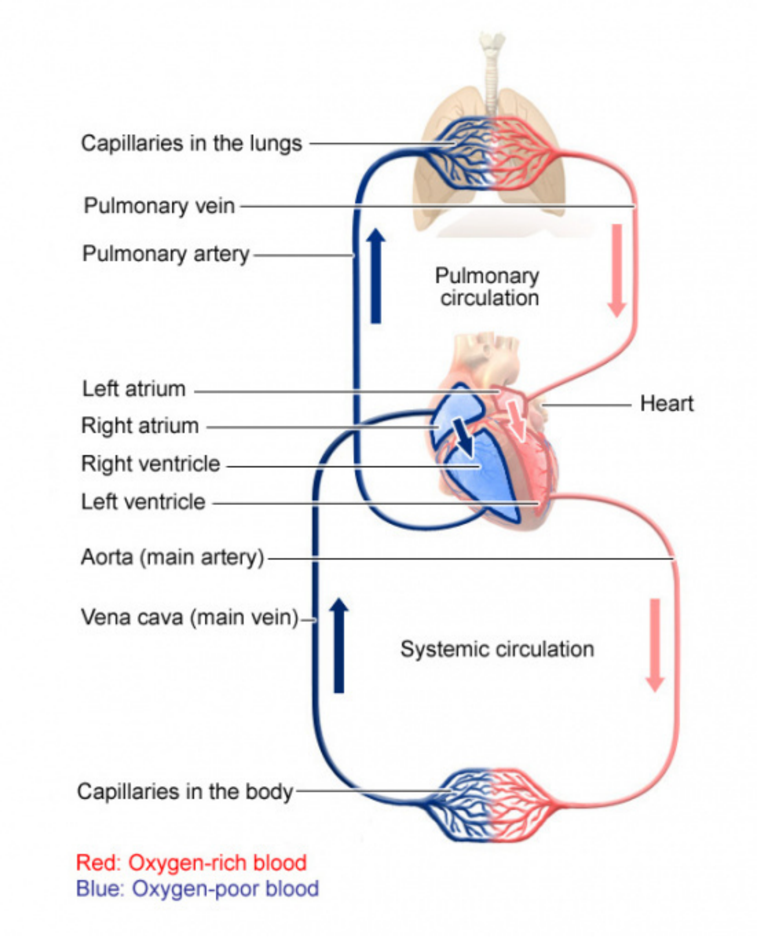


Figure 2.1: An overview of the circulatory system [77]

The cardiac cycle occurs as the heart beats and consists of three stages (as illustrated in Figure 2.2):

1. **Atrial and Ventricular diastole:** the period of relaxation during which the heart's chambers fill with blood

2. **Atrial systole:** during this phase, the atria contract and any blood remaining is pushed out into the ventricles

3. **Ventricular systole:** once filled with blood, the ventricles fully contract and push oxygenated blood back our through the aorta and pulmonary artery into the body

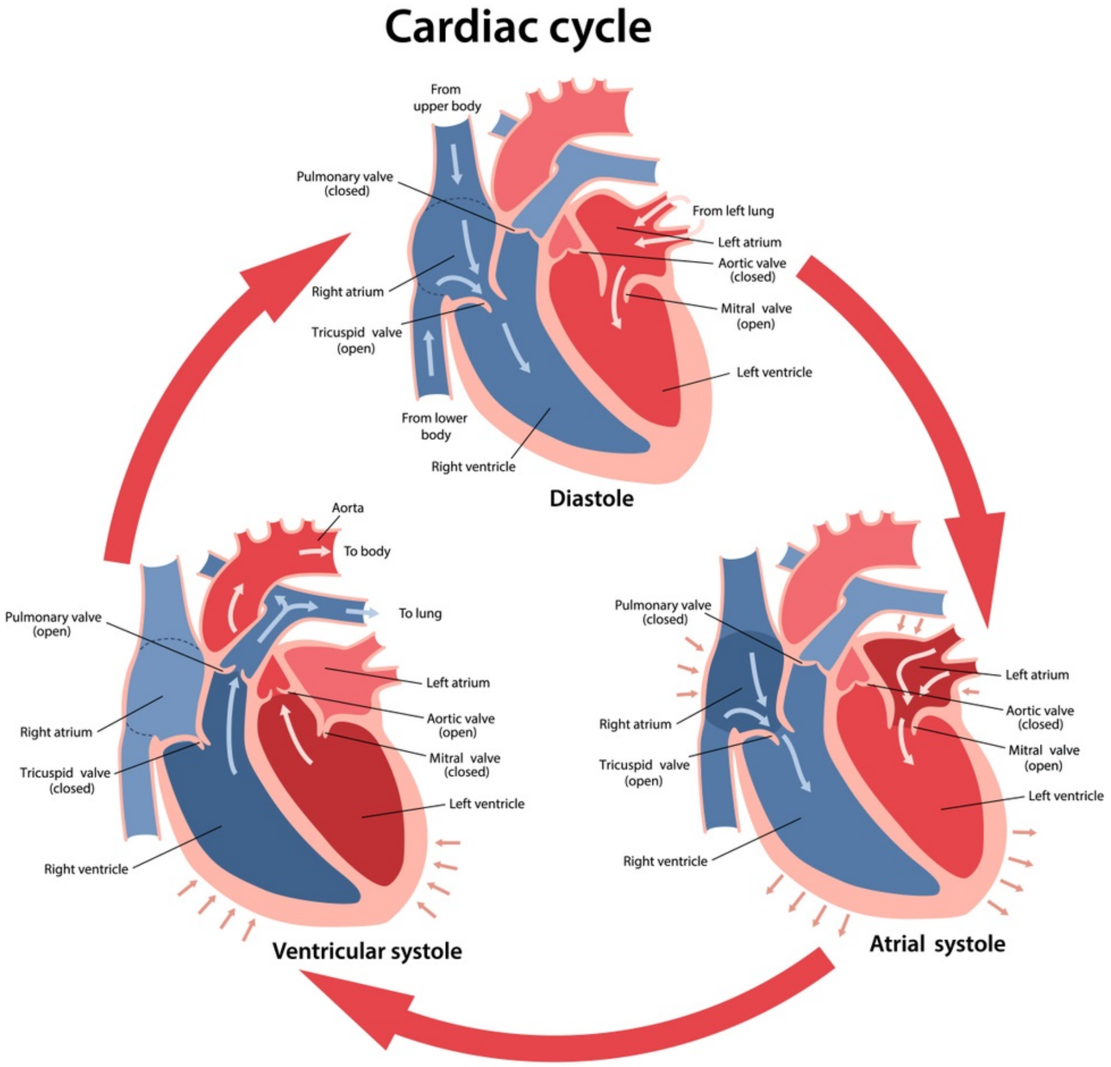


Figure 2.2: A diagram of the cardiac cycle [78]

The circulation of blood throughout the cardiovascular system begins in the atrial and ventricular diastole relaxation phase, between two heartbeats. During which, blood flows from both atria (the upper two chambers of the heart) into the ventricles (the lower two chambers) as they expand. The following phase, as both ventricles pump blood back into the larger arteries, is called the ejection period.

Systemic circulation occurs as the left ventricle (LV) pumps oxygenated blood into the aorta (the main artery of the heart) after which it travels to bigger and smaller arteries and the capillary network. This is when the blood delivers vital oxygen and nutrients across the body, and collects waste materials and carbon dioxide.

Subsequently, pulmonary circulation begins as the right ventricle (RV) contracts and pumps low-oxygen blood into the pulmonary artery. This artery branches off into smaller arteries and capillaries; an intricate network surrounding the pulmonary vesicles. Here, carbon dioxide is secreted from the blood into the air inside the pulmonary vesicles to allow the entry of fresh oxygen into the blood stream [77].

There are three types of blood vessels: arteries, veins and capillaries.

Arteries are central to the function and operation of the cardiovascular system as they distribute oxygenated blood and nutrients to each organ, and around the entire body. Specific arteries serve blood to vital organs, such as coronary (heart), carotid (brain, head, face and neck), vertebral (brain and spine) and femoral (legs).

In contrast, veins transport blood back to the heart, but is not under pressure of the heart beating so is moved along by the squeezing of skeletal muscles; such as when walking or breathing. Anatomically, veins and arteries are similar. However, veins are not as strong or thick and, unlike arteries, veins contain valves to ensure blood flow is one-directional. The largest veins are the superior vena cava (transporting blood from the upper body to the right atrium) and the inferior vena cava (carrying blood from the lower body to the right atrium) [76]

Capillaries are the connection between arteries and veins. The exchange of oxygen-rich blood from the heart occurs between the arteries and capillaries.

2.2.2 The Anatomy and Physiology of the Heart

The heart is the central organ of the cardiovascular system. It is complex; an intricate system of rhythmically and autonomously contracting muscle layers, chambers, valves and nodes [79] approximately the size of a clenched fist. The heart is located in the mediastinal cavity of the thorax, just behind and slightly to the left of the breastbone (sternum). For protection and to prevent over-expansion, the heart is enclosed within a double-layered membrane sac called the pericardium. It comprises a middle muscular layer called the myocardium, made up of cardiac muscle cells and an inner lining called the endocardium.

The cavity within the centre of the heart is divided into four chambers, as shown in Figure 2.2. These are:

- **The Right atrium**: takes non-oxygenated blood from the superior and inferior vena cava and pumps it through the tricuspid valve and into the Right Ventricle (RV)
- **The Right ventricle**: pumps blood via the pulmonary valve into the lungs where it becomes oxygenated
- **The Left Atrium**: receives the oxygenated blood from the lungs and, subsequently, pumps it back through the mitral valve to the LV
- **The Left ventricle**: Finally, the oxygen-rich blood is pumped through the aortic valve to the aorta and into the rest of the body

There are four valves within the heart to regulate the flow of blood [80], these are:

- **The Tricuspid Valve**: between the RV and LV
- **The Pulmonary Valve**: controls blood flow from the RV into the pulmonary arteries while carrying blood to the lungs for oxygenation
- **The Mitral Valve**: allows oxygenated blood to travel from the lungs and the left atrium into the LV
- **The Aortic Valve**: provides passage for the oxygen-rich blood from the LV into the aorta (the largest artery)

2.2.3 Cardiovascular Diseases

CVD is a general term attributed to conditions affecting the heart and/or blood vessels. CVD is the leading cause of mortality in high-income countries [2] due to precursors such as obesity, hypertension and diabetes; commonly attributed to unhealthy lifestyle choices [3].

The NHS Long Term Plan [81] identifies CVD as a clinical priority. The Plan states preventative measures for CVD hold the greatest potential for the Heath Service to save lives. Following it's publication in 2019, the NHS aim to help prevent over 150,000 heart attacks, strokes and dementia cases over the subsequent 10 years. Amongst a raft of initiatives and interventions, practical priorities will drive a digital transformation and explicitly state the use of artificial intelligence (AI) in assisting clinicians to apply best practice, eliminate variation in levels of care provided and support patients in their management of health conditions.

CVD is a broad term encompassing many conditions, Below is a summary of the most common [82, 83]:

- **Coronary artery disease**: the narrowing of arteries supplying blood to the heart due to the build up of cholesterol plaques

- **Angina pectoris**: stable angina pectoris is when narrowed coronary arteries cause predictable chest pain or discomfort with exertion and typically, symptoms improve with rest. Unstable angina pectoris means the pain and discomfort in the chest is worsening or occurs during periods of rest. This emergency condition could precede a heart attack, abnormal heart rhythm or cardiac arrest

- **Myocardial infarction**: is the term attributed to a heart attack. The coronary artery is suddenly blocked, the heart is starved of oxygen and part of the heart muscle dies

- **Congestive heart failure**: occurs when the heart is too weak or stiff to efficiently or effectively pump blood into the body

- **Arrhythmia**: abnormal heart rhythm due to changes in the conduction of electrical impulses

- **Cardiomyopathy**: the heart is abnormally enlarged, thickened and/or stiffened

- **Pericarditis**: inflammation of the lining of the heart (pericardium)

- **Pericardial effusion**: fluid between the lining of the heart

- **Atrial fibrilation**: abnormal electrical impulses in the atria causing an irregular heartbeat

- **Pulmonary embolism**: when a blood clot from the lungs travels to the heart

- **Heart valve disease**: disease of the four heart valves

- **Heart murmur**: an absnormal sound heard when listening to the heart with a stethescope

- **Endocarditis**: inflammation of the lining or heart valves

- **Mitral valve prolapse**: the mitral valve is forced backwards after blood has passed through

- **Sudden cardiac death**: death caused by a sudden loss of heart function (cardiac arrest)

2.3 Cardiovascular Tests and Procedures

Exponential technological advancements have permeated the medical field, particularly with the invent of non-invasive imaging examinations. Such procedures have widened access to life-saving interventions and led to improved quality of anatomical visualisations; saving time, improving diagnostic accuracy and considerable costs for health services [84].

Cardiac imaging modalities are central to the modern clinic for diagnosis and the management of various diseases. These modalities include:

- **Computed Tomography (CT) Coronary Angiogram:** visualises the heart and it's blood vessels and can be used to diagnose a variety of CVD, such as narrowed or blocked arteries. It is non-invasive and does not require recovery time. However, a CT does elicit radiation and patients undergoing an examination can be exposed, although the amount fluctuates depending on the type of machine being used. Subsequently, pregnant women are unable to have a CT angiogram to prevent potential harm to the fetus.

  Often the examination requires a contrast (dye) to be injected into the patient to help blood vessels show clearly on the images. On occasion, an allergic reaction can occur [85].

- **Magnetic Resonance Imaging (MRI):** ike the coronary CT, an MRI is a non-invasive examination that uses a magnetic field in conjunction with radio frequency waves to generate detailed, cross-sectional (2D and Three-Dimensional (3D)) images of the heart and surrounding structures. Unlike the CT, an MRI does not use ionising radiation. MRI techniques can also be used to measure heart function or how much blood the LV can pump out into the body.

  Generally, an MRI is safe, however it is not suitable for patients with any type of metal device in the body, such as a pacemaker or Implantable Cardioverter Defibrillator (ICDs) [86].

- **Echocardiogram:** an ultrasound scan used to visualise the heart and surrounding blood vessels. A small probe is used to send out high-frequency sound waves that create echos when they reflect off anatomical structures. Such echos are then captured by the probe and transformed into a moving image on a monitor as the test is carried out.

  There are multiple types of echocardiogram: transthoracic, transoesophageal, stress echocardiogram and contrast echocardiogram.

This study focuses on automated analysis of transthoracic echocardiogram (TTE) images using DL techniques, as such this modality will be discussed in more detail in the subsequent section.

2.3.1 Echocardiography

Echocardiography is an essential, non-invasive diagnostic tool for a range of cardiac pathology. 2D echocardiography was revealed in the late 1950's, however the development of pulsed Doppler towards the end of the 1960's widened opportunities for clinical adaptation and innovation. Advancements in ultrasound technology have continued and, in recent years, have seen the introduction of 3D applications [87].

Ultrasound waves are generated by piezoelectric crystals at the front of the transducer placed on the patients chest (see Figure 2.3). As an electrical current is sent through the piezoelectric crystals they vibrate, generating ultrasound waves with frequencies between 1.5 and 8MHz. The electrical signal is sent back, via the crystals, to transform electrical oscillations into an image. Each transmitting and receiving period lasts approximately one millisecond [88].

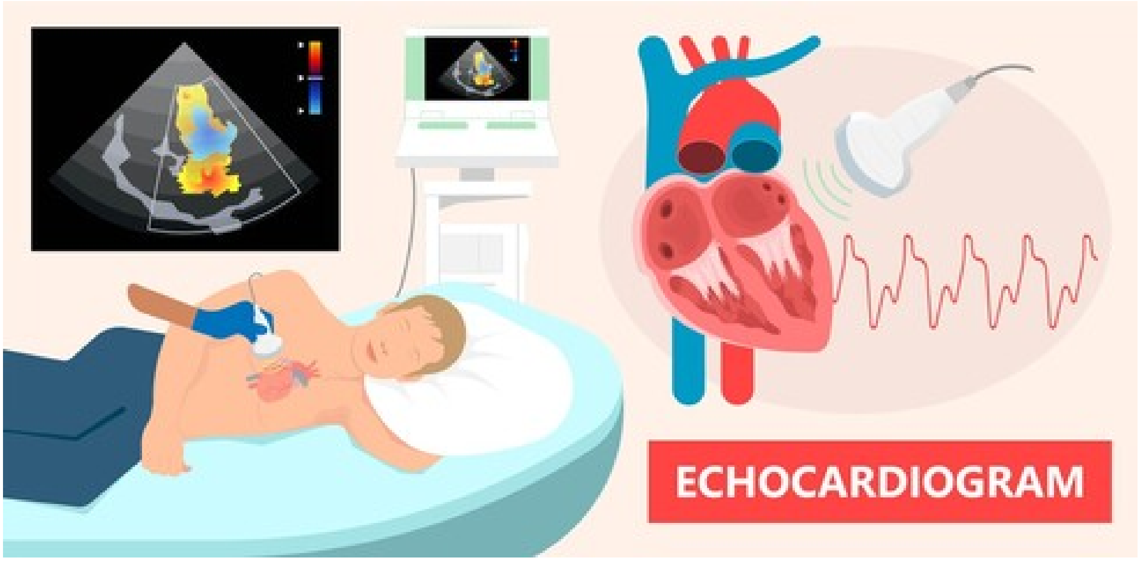


Figure 2.3: A typical Echo examination [89]

2D ultrasound is the most commonly used modality in echocardiography. The two dimensions are width (x-axis) and depth (y axis). The standard ultrasound transducer for 2D echo is the phased array transducer, creating a sector shaped ultrasound field, as seen in Figure 2.4.

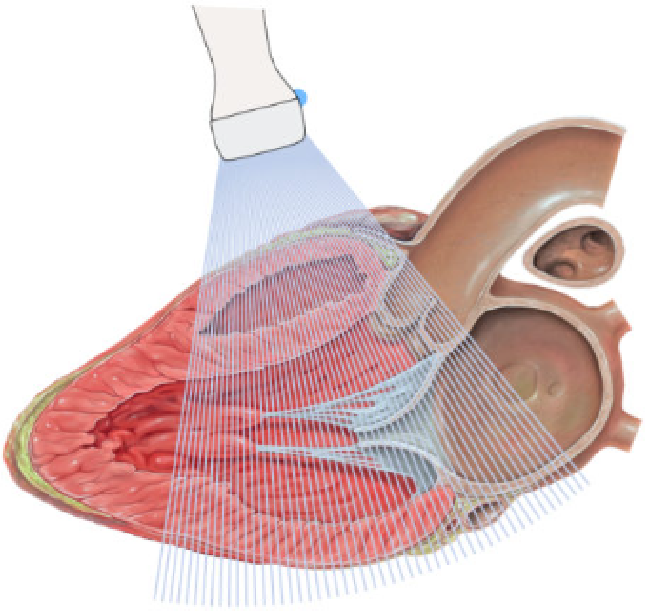


Figure 2.4: The phased array transducer, creating a sector shaped ultrasound field [90]

Echocardiographic images contain important diagnostic parameters. Often, an ECG signal is found at the bottom and can be used to identify key phases in the cardiac cycle, such as diastole and systole. Multiple images are captured per second, forming a detailed video of the heart from multiple angles (views). Temporal resolution refers to the video Frame Rate (FR). The human eye can only see approximately 25 Frames Per Second (FPS). The temporal resolution of the ultrasound capture can be improved by increasing the sweep speed of the ultrasound beam and the FPS [87].

Echocardiographic Display Modes

There are several ultrasound modes: A-mode, B-mode, M-mode, pulsed wave Doppler, continuous wave Doppler, color Doppler and tissue Doppler. A brief summary of each is as follows and continues in the next sub-section:

A-Mode (amplitude-mode): a vertical reflection of a point on an anatomic boundary that, when displayed, looks like spikes of varying amplitudes. A-mode ultrasound imaging is now obsolete in medical practice.

B-Mode: when the ultrasound signal is used to produce various points and the brightness depends on amplitude instead of spiking vertical movements (as in A-mode). Different types of B-mode images are: 2D, gray scale and real-time mode.

M-Mode (motion-mode): previously the dominating modality in echocardiography. Although, for the most part, M-mode has been replaced by 2D echocardiography. M-mode provides a One-Dimensional (1D) view of all structures reflecting ultrasound waves along one ultrasound line. A single beam in an ultrasound scan is used to produce the one-dimensional M-mode picture, where movement of a structure, such as a heart valve, can be depicted in a wave-like manner. An example of an M-mode image can be seen in Figure 2.5.

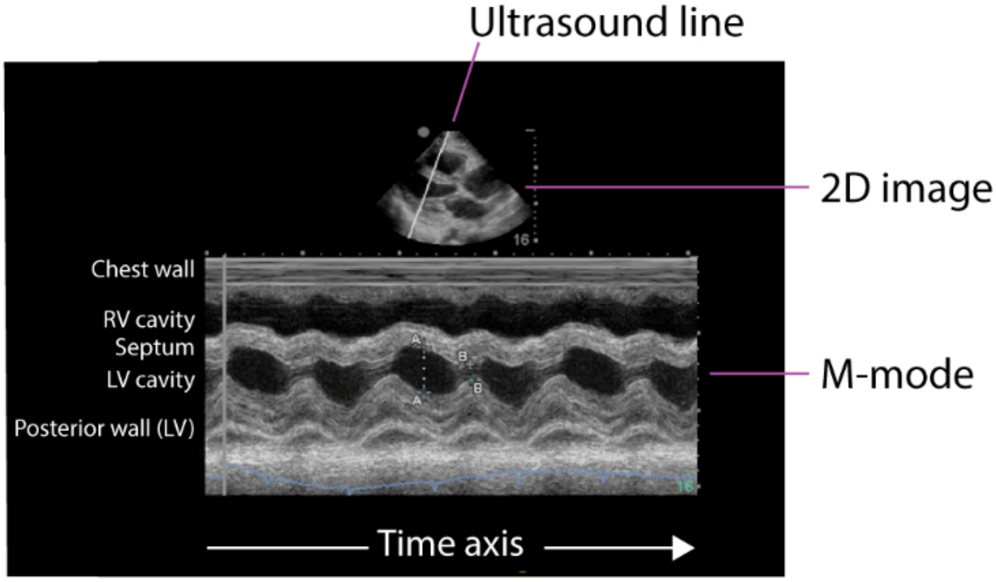


Figure 2.5: Example M-mode image [90]

Two-Dimensional (2-D) Echocardiography

An electrical current can be transmitted in “continuous” or “pulsed” current format, as observed in Figure 2.6.

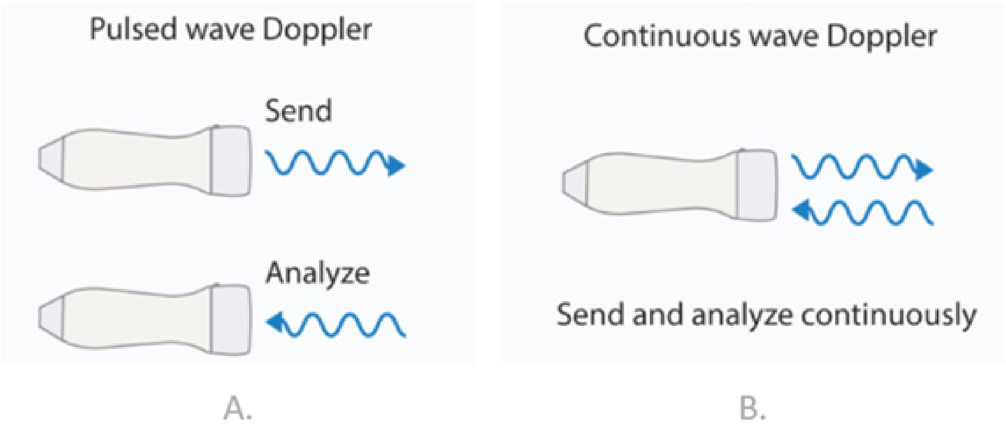


Figure 2.6: An illustration of A. Pulsed wave Doppler where there is a sending phase followed by analysis of the signal, and B. Continuous wave Doppler as the signal and current are analysed concurrently.

Continuous wave Doppler, as demonstrated in Figure 2.6, emits a continuous ultrasound beam while reflected waves are analysed concurrently. This is the key difference between Continuous wave Doppler and Pulsed wave Doppler. Continuous wave has the advantage of measuring far higher velocities. However, the disadvantage is that specific location information is not available.

Pulsed wave Doppler emits brief pulses of ultrasound and analyses the reflected wave before sending the next pulse. Sounds can be analysed from a specific location, the primary benefit of this mode. Therefore, it is possible to determine the location of a measured velocity but acquisition time is longer. The number of pulses emitted per second is called the Pulse Repetition Frequency (PRF).

Colour Doppler can be presented for velocities recorded as a pulsed wave. Blue is traditionally used to imply movement (velocities) away from the transducer, and red towards. The brighter the colour the intensity, the higher the velocity. Figure 2.7 shows a colour Doppler sector superimposed on top of the image for the interpretation of the velocity signals.

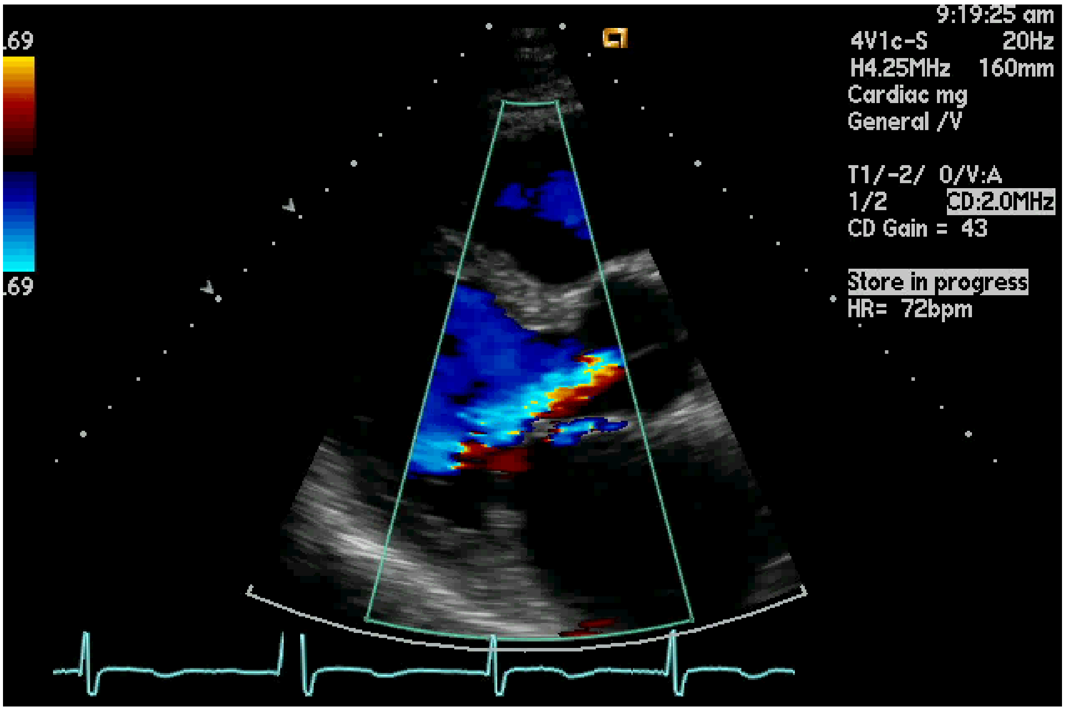
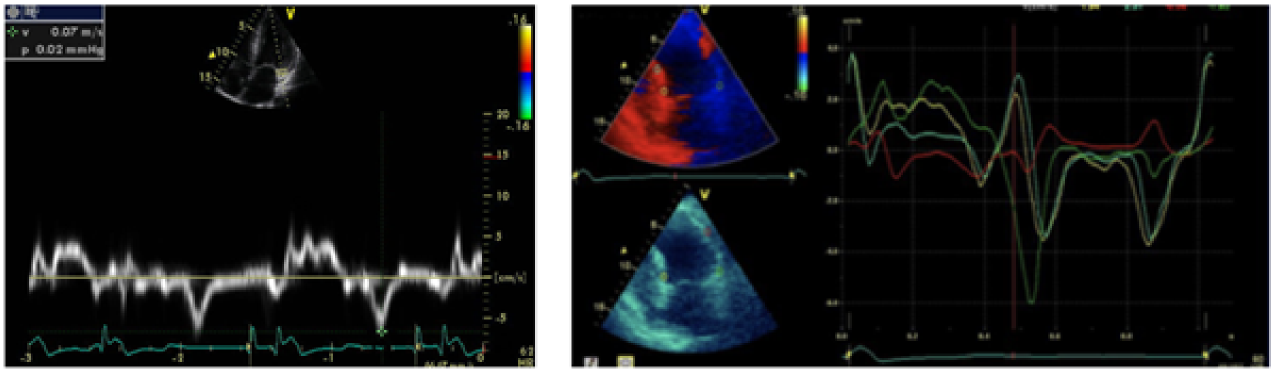


Figure 2.7: An example of Colour Doppler [90]

Tissue Doppler mode measures myocardial (the muscles of the heart) motion, during diastole and systole phases, as opposed to blood flow as with the previous modes. Tissue Doppler is captured by filtering out all other ultrasound wave reflections other than the myocardium. Figure 2.8 demonstrates a pulsed tissue Doppler image (A) and colour Doppler with four colour points (B).



A. Pulsed Tissue Doppler in the mitral valve plane

B. Colour Tissue Doppler

Figure 2.8: An example of A. Pulsed Tissue Doppler and B. Colour Tissue Doppler images [90]

Echocardiographic Tomographic Views

A thorough TTE requires imaging the heart from several windows, or views. Views are differentiated by the position of the transducer (parasternal, apical, subcostal or suprasternal) and the angle of the tomographic (imaging using a penetrating wave) plane (long axis, short axis, two- to five-chamber) [91].

Here follows a brief summary of each view:

Parasternal

Parasternal Long-Axis (PLAX) and Parasternal Short-Axis (PSAX): the parasternal window visualises the tricuspid valve, RV, pulmonic valve, right ventricular outflow tract, LV and the pericardium and is commonly used for quick assessment of patients in emergency scenarios. The Parasternal Short Axis provides an excellent view of overall function, pericardial disease, LV volume assessment [87]. Examples of PLAX and PSAX images are illustrated in Figure 2.9.

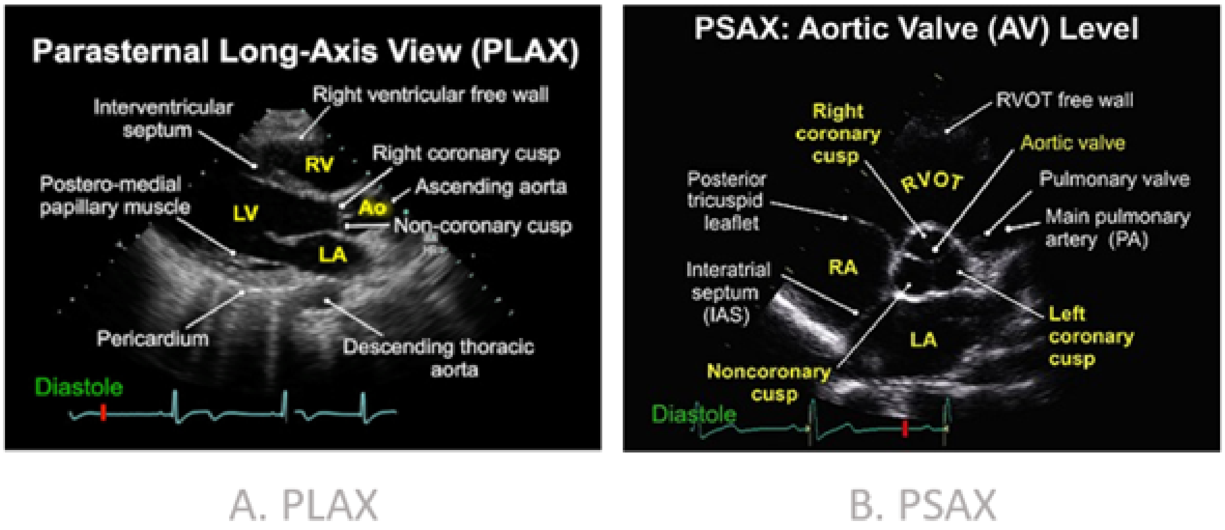


Figure 2.9: Examples of A. Parasternal Long-Axis (PLAX) and B. Parasternal Short-Axis (PSAX) tomographic views [92]

Apical

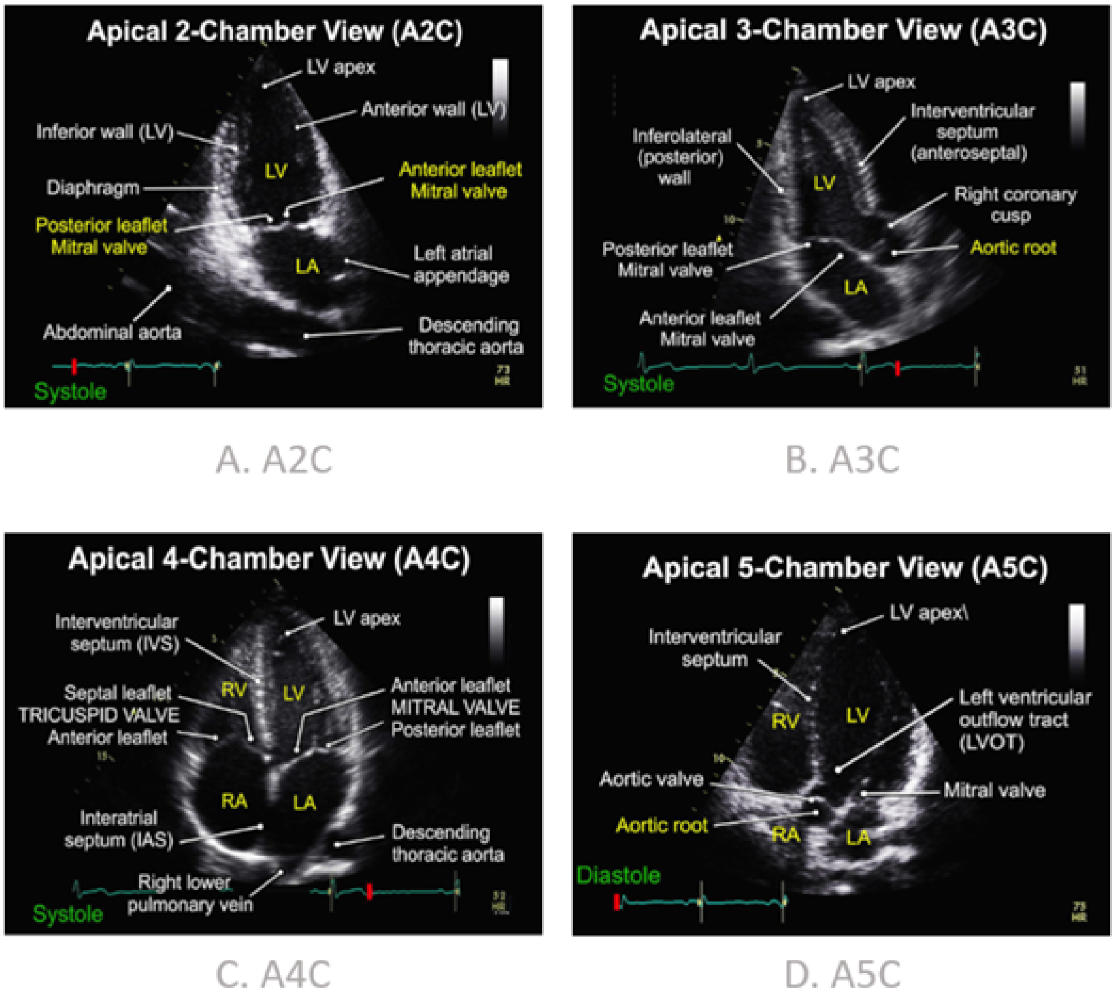
There are four views associated with the apical window (see Figure 2.10 for examples):

A4C: the first of the apical views obtained during an examination. It provides an image of the entire cardiac borders of the left and right atria and ventricles.

Apical Five-Chamber (A5C): easily acquired from the A4C view, A5C images the LV outflow tract and aortic valve to determine LV cardiac output.

Apical Three-Chamber (A3C): used to assess the walls of the LV, along with motion of the mitral valve, LV outflow tract, and aortic valve.

Apical Two-Chamber (A2C): the walls of the LV are visualised along with the left atrium . The right atrium and RV fall out of view.


Apical 2-Chamber View (A2C)
LV apex
Anterior wall (LV)
Inferior wall (LV)
LV
Anterior leaflet Mitral valve
Diaphragm
Posterior leaflet Mitral valve
LA
Left atrial appendage
Abdominal aorta
Descending thoracic aorta
Systole
Apical 3-Chamber View (A3C)
LV apex
Interventricular septum (anteroseptal)
Inferolateral (posterior) wall
LV
Right coronary cusp
Posterior leaflet Mitral valve
Aortic root
LA
Anterior leaflet Mitral valve
Systole
Apical 4-Chamber View (A4C)
Interventricular septum (IVS)
LV apex
Septal leaflet
TRICUSPID VALVE
Anterior leaflet
RV
LV
Anterior leaflet
MITRAL VALVE
Posterior leaflet
RA
LA
Interatrial septum (IAS)
Descending thoracic aorta
Right lower pulmonary vein
Systole
Apical 5-Chamber View (A5C)
LV apex\
Interventricular septum
RV
LV
Left ventricular outflow tract (LVOT)
Aortic valve
Mitral valve
Aortic root
RA
LA
Diastole

A. A2C

B. A3C

C. A4C

D. A5C

Figure 2.10: Examples of A. Apical Two-Chamber (A2C), B. Apical Three-Chamber (A3C), C. Apical Four-Chamber (A4C) and D. Apical Five-Chamber (A5C) tomographic views [92]

Subcostal

Subcostal Four-Chamber (S4C): images the right and left atrium, RV and LV. S4C is particularly effective in rescue echocardiography of clinically unstable patients. An example image can be seen in Figure 2.12.

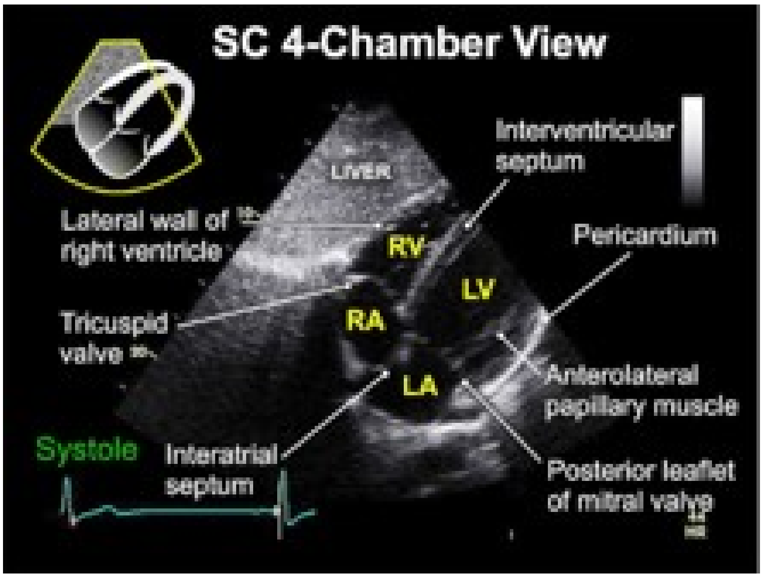


Figure 2.11: Example of the Subcostal Four-Chamber (S4C) tomographic view [92]

Suprasternal

Suprasternal Notch (SSN): allows for imaging of the aorta and great vessels coming off of the aortic arch, as demonstrated in Figure 2.12.

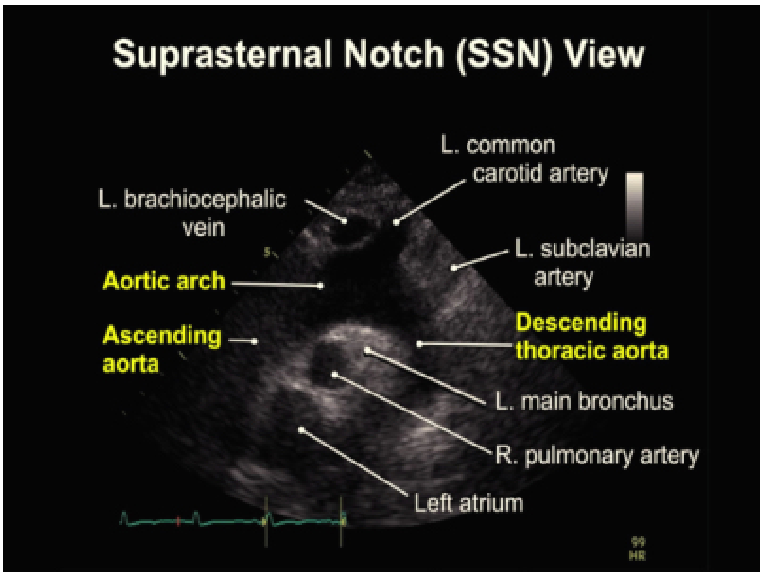


Figure 2.12: Example of the Suprasternal Notch (SSN) tomographic view [92]

Three-Dimensional Imaging

Since the 1980's echocardiography has evolved from single-beam to 3D techniques. However, it suffers from a considerable reduction in FR and image quality, hindering adoption into routine clinical practice [93]. Studies have shown 3D TTE was feasible in as few as 77% of patients due to the wide range of CVD encountered [94]. When such technological challenges in 3D ultrasound have been resolved, it would be possible to explore applying DL algorithms for automated interpretation. However, in the meantime, 2D echocardiography remains the gold-standard, especially when a high FR is required.

2.4 Conclusion

As CVD continues to lead mortality in high-income nations, the focus of health services remains upon improving access to high-quality medical interventions for improved diagnosis and treatment of several CVDs. 2D TTE remains the gold-standard for rapid, non-invasive visualisation of the heart and it's surrounding structure. In this chapter, an overview of clinical cardiography was provided, including a brief discussion of several conditions affecting the heart. Imaging modalities were discussed, with a focus upon echocardiography and the imaging modes used throughout this study.

Chapter 3

Technical Background

3.1 Introduction

Artificial Intelligence (AI) and deep learning (DL) technology has advanced exponentially over recent years [67]. The field of computer vision (CV) has experienced rapid growth, generating significant interest in it's application for automating medical image analysis; often outperforming state-of-the-art machine learning (ML) techniques [4, 95, 96].

Whilst the terms AI, ML and DL are often used interchangeably in the media and some literature, there is a distinct difference. As Figure 3.1 illustrates, AI is a broad term encapsulating ML and, it's sub-field, DL.

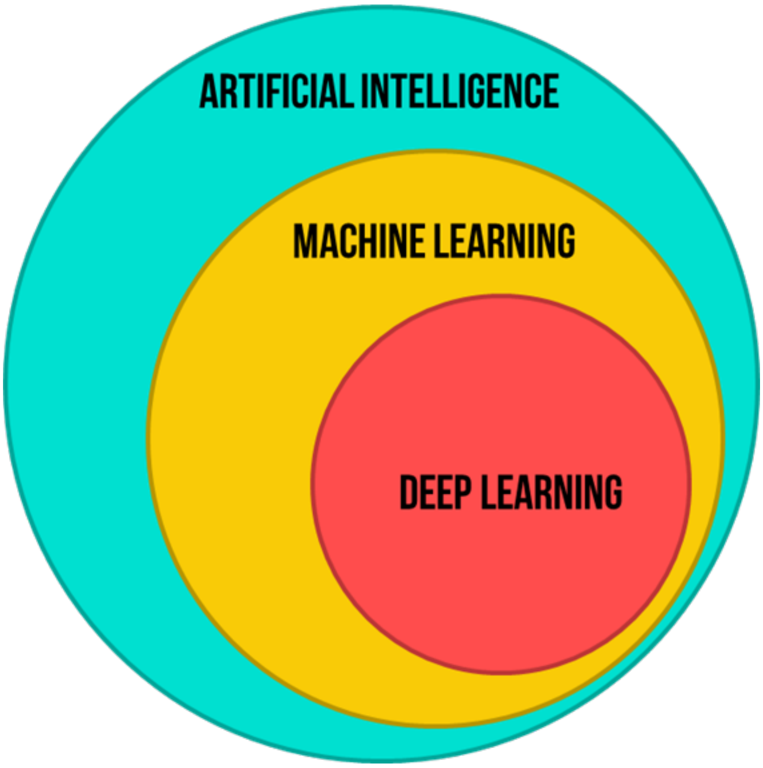


Figure 3.1: An illustration of the difference between AI, ML and DL [97]

When applying the Internet search the term “artificial intelligence” it is likely a multitude of definitions will return; many of which with varying degrees of similarity. For example, the Oxford English Dictionary refers to AI as “The theory and development of computer systems able to perform tasks normally requiring human intelligence” [98]. Whereas the American dictionary, Merriam-Webster, defines AI as “a branch of computer science dealing with the simulation of intelligent behavior in computers” and “the capability of a machine to imitate intelligent human behavior” [99]. Finally, in his book titled “Deep Learning with Python”, François Chollet (creator of the Keras deep learning framework), provides a more simplistic and, possibly more relatable, definition: “the effort to automate intellectual tasks normally performed by humans” [100].

Again, as depicted in Figure 3.1, ML is a subset of AI. However, it is important to note AI is not part of ML. ML is concerned with the automation of learning for tasks which computers have not been specifically programmed to perform. Arthur Samuel, the American computing pioneer, is often credited with coining the term “machine learning”; first appearing in his 1959 paper “Some studies in machine learning using the game of checkers” [101]. Multiple quotes defining ML are attributed to Samuel, however the closest matching in his seminal paper are "A computer can be programmed so that it will learn to play a better game of checkers than can be played by the person who wrote the program" and "Programming computers to learn from experience should eventually eliminate the need for much of this detailed programming effort". In essence, the core difference between AI and ML is that ML allows for systems to learn from experiences and data in a supervised or unsupervised manner, as opposed to being explicitly programmed.

ML algorithms can be broadly classified into three categories Supervised, Unsupervised and Reinforcement learning.

Supervised Learning: an algorithm learns to map information from the input variables (X) to an output variable (Y), this is a known set of data (for example, images of cats and dogs) with a corresponding ground-truth label (i.e. the class each image belongs to). During training, the algorithm learns from the data and accompanying labels to gradually produce predictions with an increased degree of accuracy.

Unsupervised Learning: in contrast, during unsupervised learning there are no accompanying ground-truth labels for the training data. Training focuses on extracting and inferring patterns in the data without any reference to a pre-defined output. Hence the term 'unsupervised'; the algorithm is allowed freedom to group the unsorted data by finding similarities, differences and patterns.

Reinforcement Learning: is a type of ML algorithm where an 'agent' learns through constant interaction with an interactive environment through a 'trial and error' approach with positive reinforcement by way of feedback, rewards and punishments based on the agent's previous actions and experiences.

Finally, as previously observed in Figure 3.1, DL is a sub-set of ML. The DL network architecture is loosely inspired by the way neurons filter information in the human brain. A neural network is considered 'deep' when comprised of >3 'hidden' neuron layers, each used to process and classify data. A layer is considered hidden if it receives an input from a previous layer, and outputs to a subsequent layer.

The Oxford English Dictionary defines DL as "a type of ML based on Artificial Neural Networks (ANNs) in which multiple layers of processing are used to extract progressively higher level features from data". Whereas, IBM define DL as follows: "Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain—albeit far from matching its ability—allowing it to "learn" from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy." [102] as illustrated in Figure 3.2.

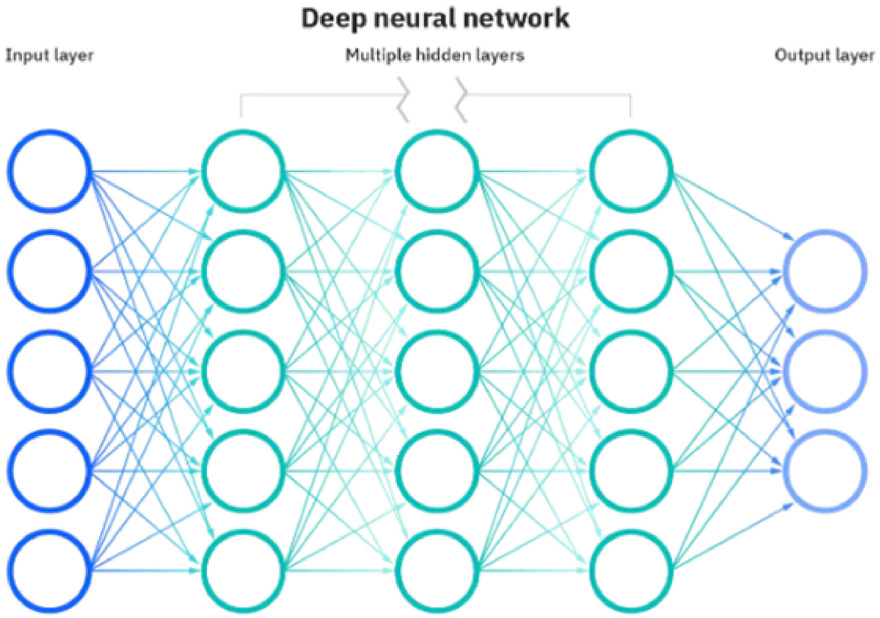


Figure 3.2: A high-level overview of a deep neural network [103]

There are several DL architectures, those pertinent to this study are detailed in the subsequent sub-sections.

3.2 Convolutional Neural Networks (CNNs)

3.2.1 An Introduction to CNNs

A Convolutional Neural Network (CNN), also referred to as a ConvNet, is a neural network architecture specialising in processing 2D data with a grid-like topology, such as images, and is commonly utilised for CV tasks. CV is a field of AI concerned with enabling computers to learn and decipher spatial and temporal information from images and infer predictions based on the given task. The name “convolutional neural network” indicates that the network employs a mathematical operation called convolution [104].

The CNN architecture is analogous to the connectivity pattern of neurons in the human brain. As such, convolutional operations were inspired by the operation and function of the visual cortex within which individual neurons respond to stimuli in a restricted region of the visual field, termed the receptive field. Many receptive fields are processed and combined in an overlapped fashion to encompass the entire visual area.

CNNs are designed to adaptively learn spatial hierarchies of image features, from low to high-level patterns, via backpropagation. Backpropagation refers to the process of fine-tuning weights based upon the loss (error rate) encountered after the most recent epoch (feed-forward training iteration). The effective training of a network’s weights ensures higher accuracy and reliability, thus increasing generalisability.

A neural network is initialised with a random set of weights. These weights are then updated and optimised during training to indicate the importance of various features. Bias is a constant vector (it does not change during network training) and is added to the product of the input and weight values. The purpose of a bias is to offset the result of each neuron and shift the activation function towards either a positive or negative value. Bias reduces variance (i.e. the sensitivity of the network) and introduces flexibility with regard to generalisation.

The activation function is a mathematical operation that normalises the input and produces an output which is then passed forward to neurons in subsequent layers. More detailed information about activation functions can be found in subsequent sections of this chapter.

A typical CNN is comprised of three types of layers, or ’building blocks’: convolutional, pooling and fully-connected. Additionally, a CNN consists of multiple ’hidden’ layers which process an input (image) into an output prediction. The layers are arranged to initially detect simple patterns, such as lines and curves, and subsequently complex patterns, such as faces and objects.

CNNs can be trained to perform several tasks, such as image classification, segmentation and object detection. The first two types of layer, convolution and pooling, perform feature extraction, whereas the fully connected layer maps the extracted features into a final output.

When defining the architecture of a CNN, there are multiple parameters and hyper-parameters to consider. Model parameters are internal configuration variables and are required to make predictions. The parameter values help determine the suitability and accuracy of the trained network. Parameters are usually learned over time during the training process and are not manually defined. Parameters are part of the configuration when a trained model is saved.

In contrast, hyper-parameters are external to the model; the values are not learned by the training data. Hyper-parameters are manually specified prior to training (before optimising the weights and bias) and can be configured using heuristics. Throughout training and experimentation, hyper-parameters are often fine-tuned to achieve maximum performance of the final network. Figure 3.3 depicts a typical CNN architecture with multiple hidden convolutional layers.

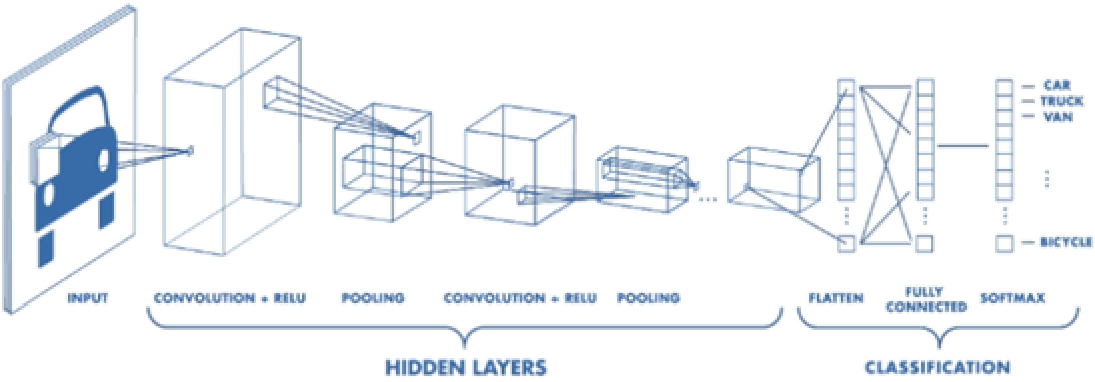
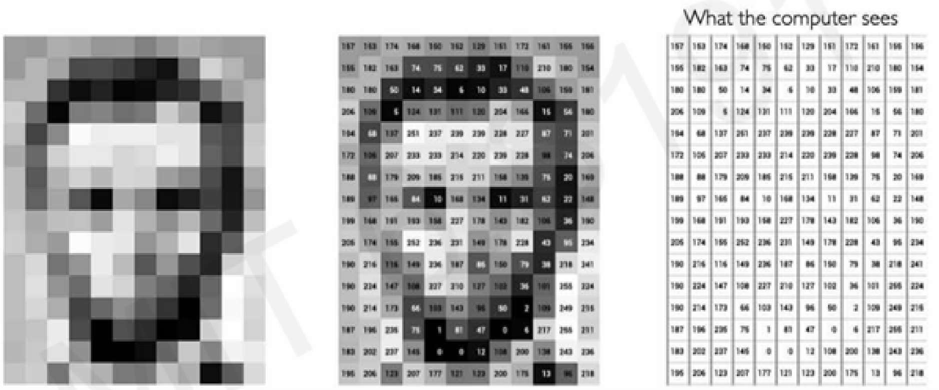


Figure 3.3: A typical CNN architecture [105]

3.2.2 Convolutional Layers

The convolutional layer is fundamental to the CNN architecture and performs the majority of the network's computational load. The term 'convolution' simply refers to the repeated application of a Kernel (also referred to as a filter) across the input (image) resulting in an activation.

CNNs benefit from sparse interactions (also called sparse connectivity or sparse weights) meaning the kernel is intentionally smaller, in terms of pixel dimensions, than the input. Images are comprised of a matrix of pixels which are represented by numbers, see Figure 3.4 for an example.



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Figure 3.4: An image is a matrix of pixels each represented by a number [106]

The input image may comprise thousands, or even millions, of pixels, but as a kernel convolves (or moves) across the breadth and depth it can gradually detect small, meaningful features such as edges. The advantage being we need to store fewer parameters, reducing memory requirements and improving statistical efficiency [104].

A CNN is able to successfully capture the spatial and temporal information from within an image by iteratively applying multiple relevant filters. By applying the same filter to the input multiple times a map of activations is produced called a feature map. This map indicates the location and prominence of detected features, such as lines and edges. Each filter is applied systematically to the input, overlapping as necessary: left to right, top to bottom.

During the process of convolution, summed dot product element-wise multiplication is performed between the input data matrix and a two-dimensional array of weights. The output from multiplying the kernel with the input once is a single value, as illustrated in Figure 3.5

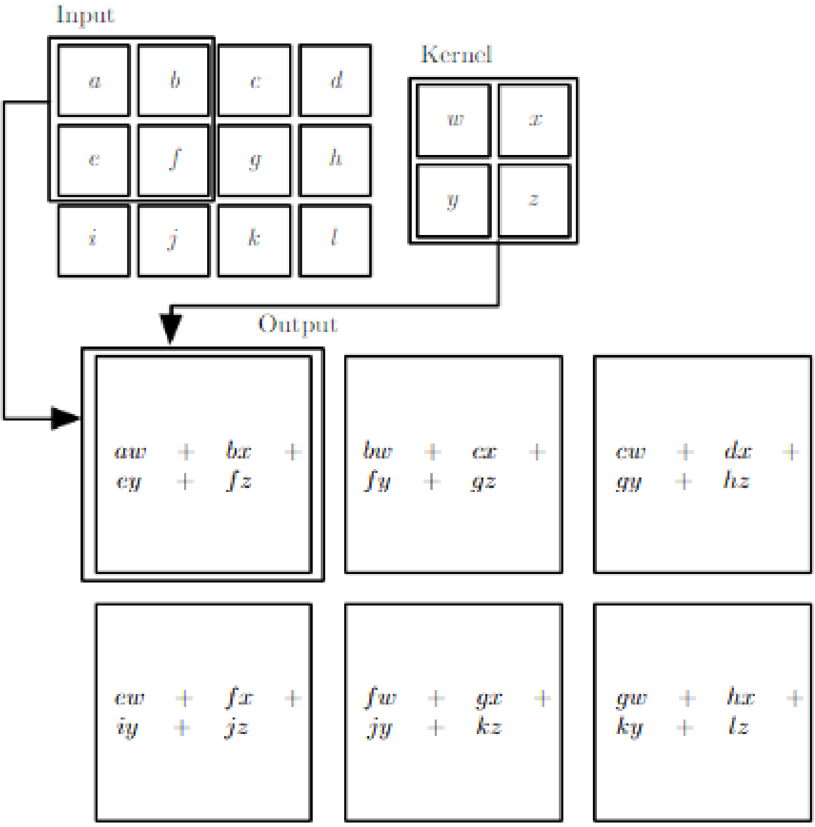


Figure 3.5: An example of 2D convolution [104]

Kernel weight values are progressively learned and adjusted during training. CNNs do not learn from a single kernel, instead they compare multiple features in parallel. When defining the hyper-parameters of a convolutional layer, the number of filters is specified. This can be anything from 16 to 512, all working in parallel. Therefore, the model has between 16 and 512 different methods to detect features. Examples of low to high level features are shown in Figure 3.6.

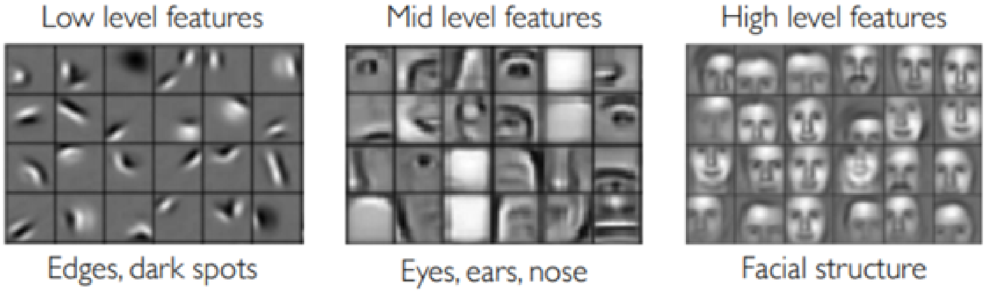


Figure 3.6: Examples of features extracted by low-level to high-level convolutional layers [106]

Colour images have multiple channels, typically three, meaning a single image input to the CNN is in fact three images stacked on top of one another. Therefore, the kernel must have the same number of channels as the input, also called "depth", as illustrated in Figure 3.7.

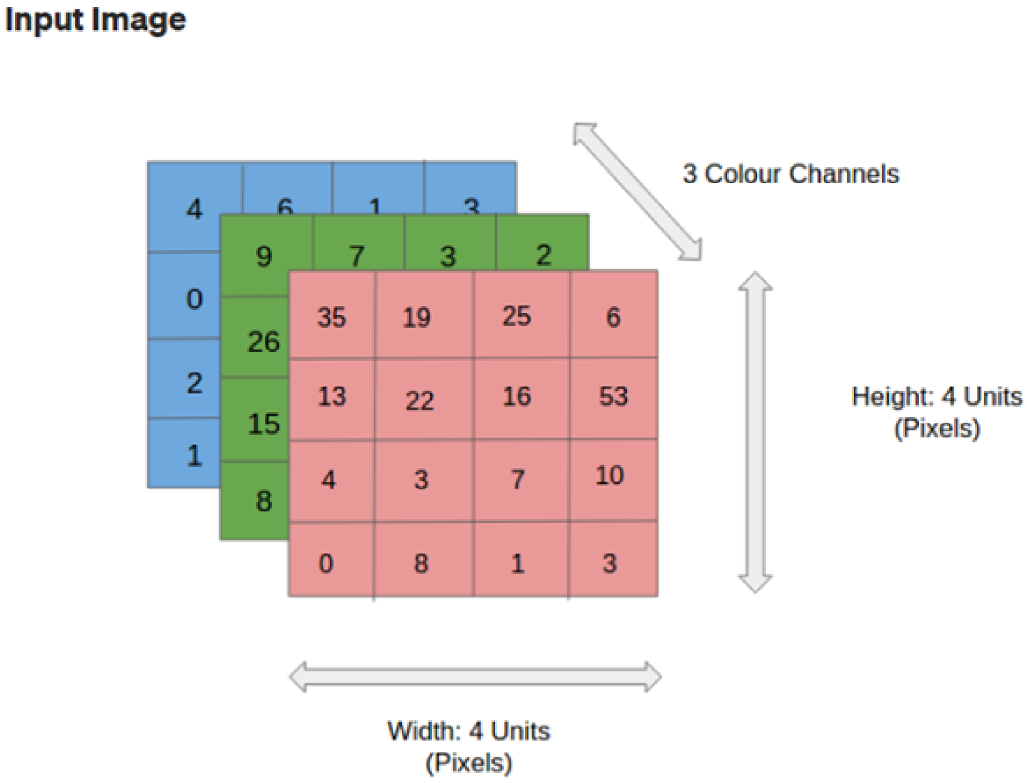


Figure 3.7: An RGB image (4x4x3) which has been separated by it's three color planes: Red, Green, and Blue [107]

As an example, if an input image has 3 channels (e.g. a depth of 3), then a filter applied to that image must also have 3 channels. In this case, a 3×3 filter would in fact be 3x3x3, for rows, columns, and depth. Meaning that if a convolutional layer has 32 filters, these 32 filters are not just 2D for the 2D image input, but are also 3D, having specific filter weights for each of the three channels. Yet, each filter results in a single feature map.

Convolutional Layer Parameters

There is one parameter in the convolutional layer, the kernel. Weights are randomly allocated at the start of training and gradually updated until the network converges.

Convolutional Layer Hyper-parameters

There are several hyper-parameters that need to be defined prior to training a CNN. These are:

Number of neurons: the number of neurons in each hidden layer, this can either be the same or different and should be adjusted to the complexity of the problem domain. The range of neurons is typically between 10 and 100.

Kernel size: the height and width of the kernel, in pixels, to convolve across the input (image).

Number of kernels: the number of kernels applied (16 to 64 is common).

Stride: determines the step size (number of pixels) the kernel will move each time it convolves across the input.

Padding: to avoid missing or ignoring pixels from the perimeter of the input, adding pixels with a zero value along each edge is a common strategy. Padding ensures the kernel can move across the image matrix uniformly and the resulting feature map will have the desired dimensions.

Activation function: the activation function defines how the weighted sum of the input is transformed into an output of each node (or neuron) that serves as an input to the next layer. This value is referred to as the summed activation of the node.

The choice of activation function determines how the network learns from the training data and has a significant impact on the performance of the network.

To handle complex data scenarios, there are a number of activation functions. Those most common are listed below:

Linear activation function: the simplest activation function; no transform is applied. A network comprised of only linear activation functions is very easy to train, but cannot learn complex mapping functions because the input is simply scaled by a factor; implying there is a linear relationship between the input and output.

Sigmoid activation function: guarantees the output will be between 0.0 and 1.0. The Sigmoid activation function is "S" shaped and can add non-linearity to the output by returning a binary value. It can be referred to as a "squashing" function as any large negative, or positive, values will be restricted to a range of 0 - 1.

If we use a linear activation function in a neural network, then this model can only learn linearly separable problems. However, with the addition of just one hidden layer and a sigmoid activation function, the neural network can easily learn a non-linearly separable problem. Using a non-linear function produces non-linear boundaries and hence, the sigmoid function can be used in neural networks for learning complex decision functions.

Tanh activation function: Tanh is an extension of the Sigmoid activation function, however the output is within the range of -1.0 to 1.0.

A general limitation of both the Sigmoid and Tanh activation functions is their tendency to saturate, meaning large values are snapped to 1.0 and negative values to 0 or -1.0 for Signoid and Tanh, respectively. This also means the functions are more susceptible to changes around their mid-range, such as 0.5 for Sigmoid and 0.0 for Tanh. Once saturated, it becomes challenging for the learning algorithm to continue to adapt the weights to improve the performance of the model.

Rectified Linear Unit Activation Function (ReLU): one of the most widely used activation functions in hidden layers. In a similar way to Sigmoid and Tanh, ReLU adds non-linearity to the output, however, the result can range from 0 to infinity. If positive, the output will remain the same, however if negative, the value will be set to zero.

Because rectified linear units are nearly linear, they preserve many of the properties that make linear models easy to optimise with gradient-based methods. They also preserve many of the properties that make linear models generalise well.

Softmax activation function: also an extension of the Sigmoid activation function. Softmax adds non-linearity to the output, however it is mainly used for classification tasks, where multiple classes of results can be computed. Softmax is commonly observed in the final fully-connected layer of a CNN because it returns a probability likelihood for each class with a sum for all values of 1.0.

3.2.3 Pooling Layers

As previously stated, a typical CNN consists of three stages. First, several convolutions are performed in parallel to produce a set of linear activations, each of which is run through an activation function. Next, a pooling function is applied to the feature map to perform dimensionality reduction.

The pooling process ensures the final representation is invariant to small translations of the feature maps output from the activation function. Such invariance is useful when the presence of a feature (or not) is more relevant than where the feature resides spatially. For example, when classifying whether an image contains a face, we are more concerned with whether there is an eye present on either side, as opposed to requiring pixel-perfect accuracy of each feature [104].

The convolutional process alone increases the risk of overfitting. Overfitting occurs when the network is too familiar with the training data and does not generalise well to previously unseen images. Pooling assists in generalising the features in the feature map. The problem of overfitting is explained in more detail in the Regularisation Techniques section of this chapter.

The size of an output feature map can be relatively large, therefore it is necessary to perform dimensionality reduction while preserving spacial information. The two most common pooling methods are max and average. Max abstracts the most prominent features within the feature map, thus sharpening the key features and rejecting irrelevant proportions of the data.

In contrast, as the name suggests, average pooling smoothly extracts all features and retains the majority of the original data by averaging values across the entire feature map. The choice of pooling method is very much dependent on the expectations from the layer and the overall CNN objective.

An example of max and average pooling can be seen in Figure 3.8.

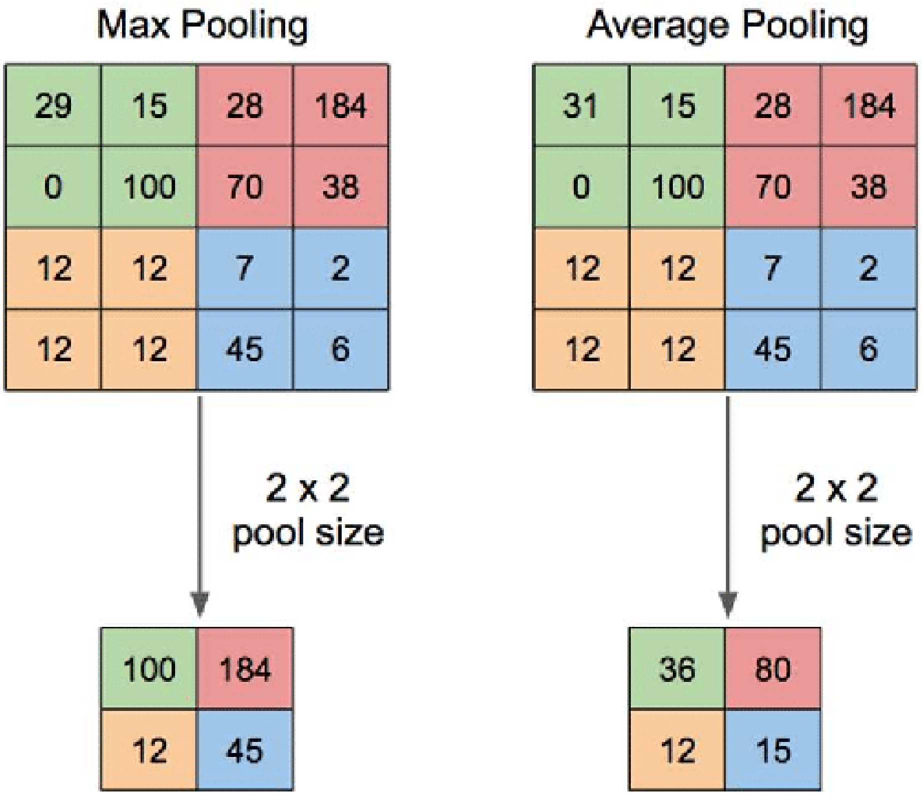


Figure 3.8: An example of max and average pooling operations with a 2x2 pixel filter from a 4x4 pixel input [108]

The effect of several pooling methods can be observed on a sample image in Figure 3.9, below.

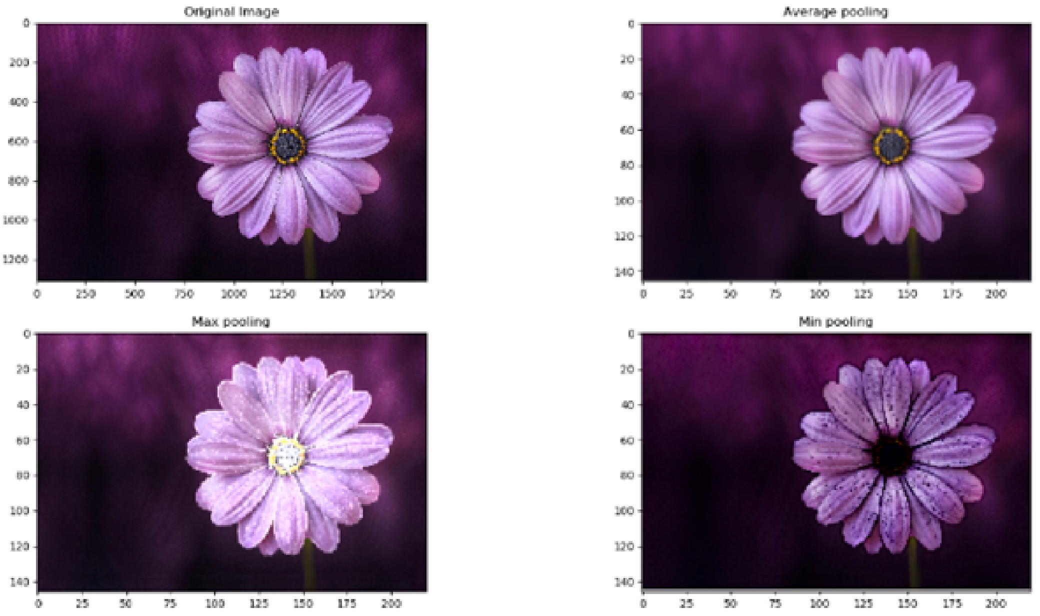


Figure 3.9: Average, Maximum and Minimum pooling with size 9x9 pixels on a sample image [109]

Pooling Layer Hyper-parameters

While there are no parameters in the pooling layer, there are several hyper-parameters. These are:

Pooling method: Average, max and in some cases the min pooling method.

Filter size: the height and width of the pooling filter, in pixels.

Stride: the step the filter will move after completing each pooling process.

Padding: padding by adding zero pixels across all sides (the same as seen in the convolutional layer).

3.2.4 Fully-Connected Layer

The fully connected layer is the final layer in the CNN. It derives the final classification decision and is a single-layer perceptron; consisting of only a single layer of output nodes. The output of the previous convolutional or pooling layer is flattened and fed directly to the outputs via a series of weights and the addition of a bias vector.

The “flattening” layer is one-dimensional and refers to the process of “unrolling” the input, 3D matrix, into a 1D vector, as illustrated in Figure 3.10.

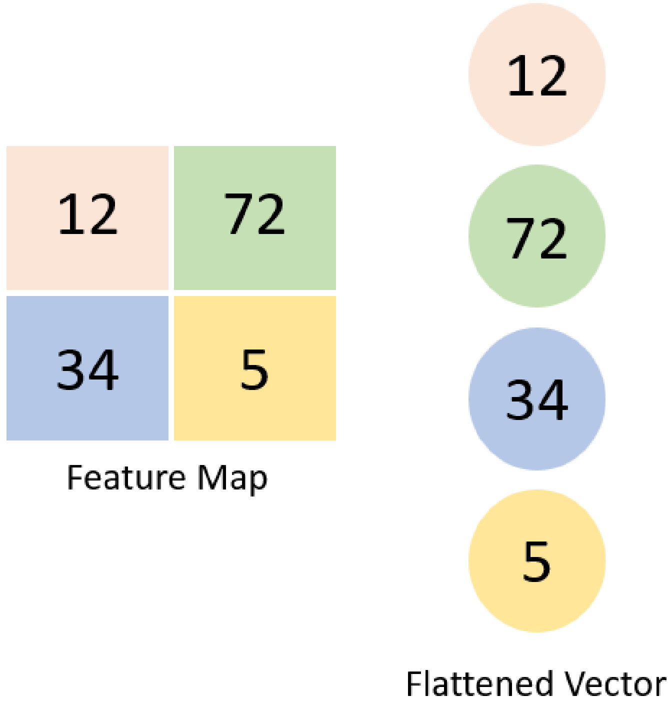


Figure 3.10: A visual representation of the feature map flattening process into a 1D vector

Neurons in a fully connected layer have full connections to all activations in the previous layer, as in regular Neural Networks. Hence, their activations can be computed with matrix multiplication followed by a bias offset.

Fully-connected Layer Parameters

There is only one parameter in the fully connected layer, the weights that are fine-tuned during backpropagation.

Fully-connected Layer Hyper-parameters

There are two hyper-parameters for the fully-connected layer, these are:

Number of outputs: a positive integer representing the dimensionality of the output prediction.

Activation function: the choice of activation function applied to the output of the layer.

3.2.5 Regularisation Techniques

Regularisation techniques are strategies employed in ML with the aim of reducing generalisation error. Hence, the model works well on a subset of validation data from the original dataset, but not on new, unseen data. Poor generalisation of a network is also known as underfitting. The purpose of regularisation techniques is to reduce overfitting, thus decreasing generalisation error, but maintaining the lowest possible training error [104].

Bias error (underfitting) occurs when wrong assumptions about the data are inferred by the learning algorithm. If the bias is too high, the network could miss important relationships between detected features and intended outputs. The term variance refers to an error in the network from sensitivity to small changes (fluctuations) in the training set. As previously discussed, overfitting occurs when the model learns statistical noise in the training data and results in poor performance when evaluated against previously unseen data, as demonstrated in Figure 8.11

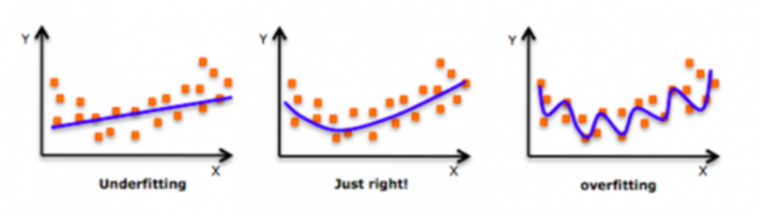


Figure 3.11: Examples of trained networks (blue line) where it has underfit (left), fit well (centre) and overfit (right) to the data points (red dots) [110]

The bias-variance trade off is used to describe the fact that when we reduce the variance in a model we, in turn, increase the bias. To achieve generalisation, we can employ regularisation techniques, which seek to simultaneously minimise errors in variance and bias. Common regularisation techniques are summarised below:

L1 and L2 Regularisation

L1 and L2 regularisation were popular techniques even before the DL era, for problems such as linear and logistic regression.

Both L1 and L2 are “constraints” which must be adhered to when minimising the loss function, between the ground-truth y and the prediction $\hat{y}$. The addition of an L1 or L2 regularisation term reduces the value of the weights matrix by assuming that a neural network having less important weights makes for a simpler model, thus assisting in reducing overfitting.

Mean Squared Error (MSE) is a simple and common loss function in ML and will be used for illustration purposes. MSE is calculated as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{N}(\hat{y} - y)^2$$

L1 regularisation, also referred to as Lasso (an acronym for “least absolute shrinkage and selection operator”) is preferable in networks with a high number of features as it provides sparsity (fewer variables). This is because L1 helps to gain a computational advantage by removing features with zero coefficients.

$$MSE = \frac{1}{n}\sum_{i=1}^{N}(\hat{y} - y)^2 + \lambda\sum_{i=1}^{N}|w_i|$$

As illustrated, L1 is the sum of the absolute value of all the weights multiplied by the lambda (λ) term. The λ value is manually tuned, hence the larger the value, the larger the penalty imposed. Thus, as the error increases, the weights value reduces. As the weight value approaches zero, the number of features reduces because as the variable importance declines it can be removed altogether. Consequently, L1 regularisation can be effective for feature selection.

L2 regularisation, also referred to as ridge regression, adds the squared magnitude of the weights vector, multiplied by the λ penalty term, as illustrated below:

$$MSE = \frac{1}{n}\sum_{i=1}^{N}(\hat{y}-y)^2 + \lambda\sum_{i=1}^{N} w_i^2$$

As with L1 regularisation, the higher the λ value, the higher the MSE will become, resulting in a greater penalty and forcing the weight value to reduce. However, because L2 takes the square of all the weights, the value will not reach zero, so multicollinearity (where several independent variables in a network are correlated) is addressed because all variables are retained.

L1 and L2 regularisation both have advantages and disadvantages, depending on the nature of the project and the neural network architecture, therefore it is common to experiment to observe which technique achieves better results.

Data Augmentation

The performance of a DL model generally improves with the size of the training dataset. However, in practice, data is often limited. To artificially increase data, augmentation techniques can be applied. The purpose of augmentation is to generate a new variant of each training image. Thus, forcing the model to be more tolerant to small changes in spacial features and reducing overfitting while improving model accuracy and generalisation.

There are two main methods of applying data augmentation:

1. Increase the size of the training set by generating and saving new, augmented images as files

2. 'On the fly', during training an image data generator class can be used to augment images in real time; they are not saved. This is by far the most popular technique employed, examples are the Keras ImageDataGenerator class [111] and 'imgaug' [112] library for custom augmentation pipelines.

A summary of popular data augmentation techniques is as follows:

- Random rotations (within a % range)

- Random shifts (width and height)

- Random flips (horizontal or vertical)
- Adjusting image brightness
- Image whitening (to sharpen features)
- Zooming into the image

Early Stopping

Early stopping is a commonly applied regularisation strategy, it is both simple yet very effective. The general premise is that training stops when the error is no longer reducing, yet the validation error is seen to start increasing.

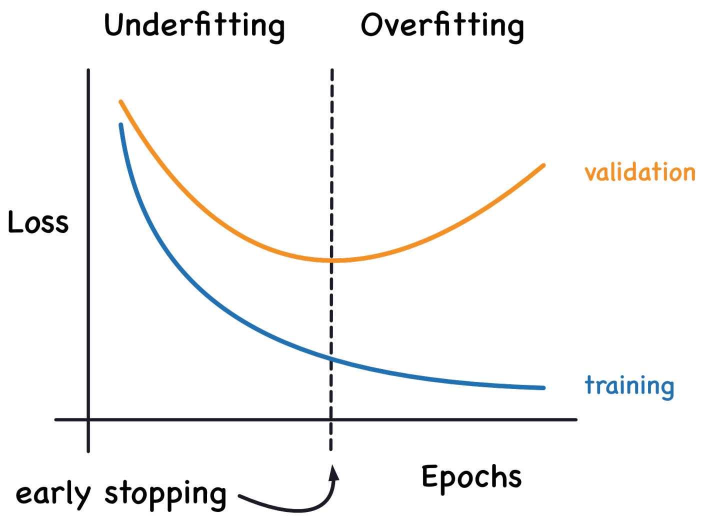


Figure 3.12: An example of early stopping where the trained model is saved when the validation loss is at the minimum [113]

In addition to preventing overfitting by training for too long, early stopping can help prevent underfitting by not training for enough epochs. A good strategy is to set the number of epochs to a high figure and set a "patience", referring to the number of epochs that can complete without seeing an improvement in training loss. Early stopping can be easily implemented using the Keras Callbacks API [114].

Parameter Sharing

As opposed to penalising model parameters, such as weights, parameter sharing encourages groups to have equal value. The most common approach is convolution in a CNN where a filter, or kernel, with a fixed width and height moves across the input and pools features. Thus, taking advantage of spacial information. Figure 3.13 demonstrates parameter sharing via the convolutional process during which each kernel focuses on a block of the image at each time, whilst the weights are shared and pooled. The filters, acting upon the input data, generate the output of a convolutional layer called the feature map.

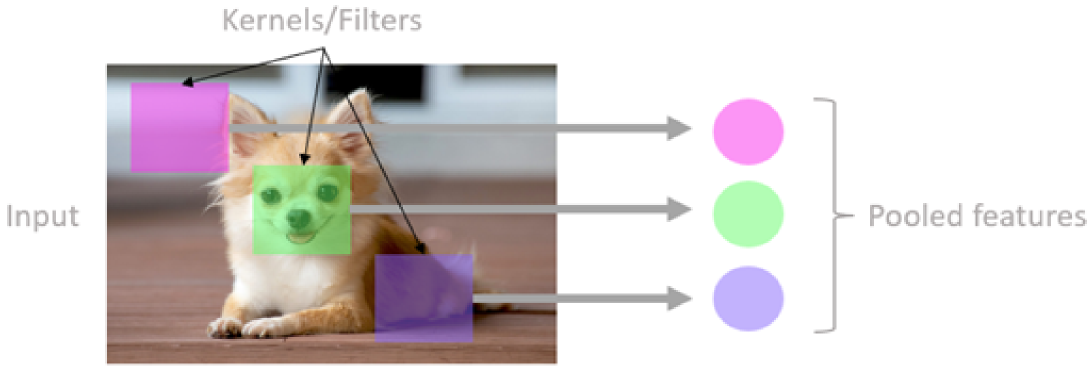


Figure 3.13: An example of parameter sharing during the convolutional process

Dropout

Dropout is a commonly used regularisation technique to prevent overfitting. As the name suggests, the dropout layer randomly drops a certain number of neurons in a layer as defined in a pre-set dropout rate. Generally, a small dropout rate of 20%-50% of neurons is preferred. A value too high can result in the network underfitting. A high-level example of a CNN including dropout regularisation can be seen below in Figure 3.14.

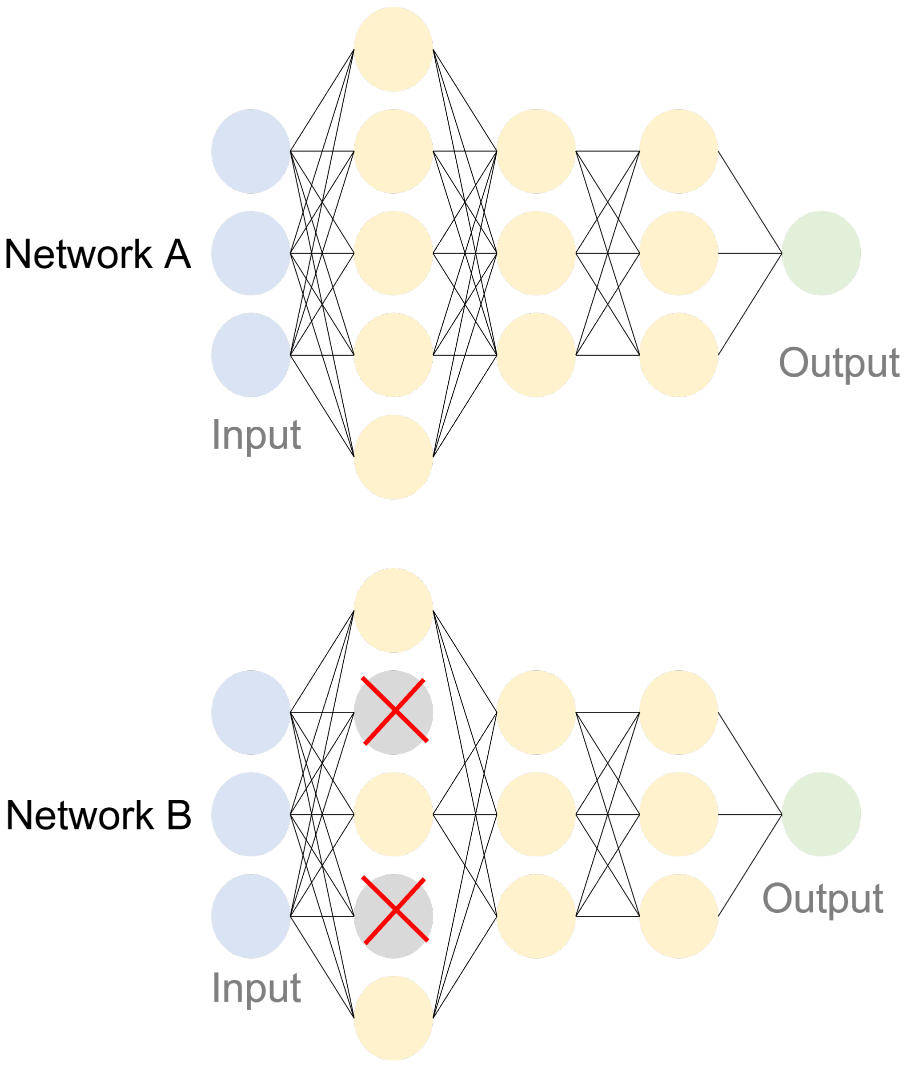


Figure 3.14: A visualisation of using dropout for regularisation in a typical neural network. Network A shows the network without dropout, where as Network B represents dropout where 2 nodes are no longer connected to the subsequent hidden layer.

During dropout, neurons are temporarily removed from the network for the current forward pass, in addition to all it's incoming and outgoing connections. Hence, the connectivity of one CNN layer to the next will alter, forcing it to search for alternative routes to convey the same information onto the subsequent layer.

Batch Normalisation

Training a DL model with multiple layers can be challenging due to sensitivity to initial, random, weight value allocations. Additionally, the order of the inputs to deep layers might change after each mini-batch as weights are updated, causing the network to chase a "moving target".

This change in the distribution of inputs to layers is termed “internal covariate shift”.

Batch normalisation can be applied at various stages during the definition of a CNN, but is often placed after the convolution and pooling layers. Batch Normalisation can improve both convergence and generalisation in training a neural network [115] by performing normalisation after each mini-batch. Hence, allowing for higher learning rates and, in some cases, negating the need for dropout [116].

3.2.6 General CNN Hyper-parameters

Model architecture: refers to the stack of distinct layers that comprise the network and transform the input into a predicted output.

Optimiser: the optimiser is required when compiling a model for training and is responsible for adjusting the learning rate and neuron weights with the objective of reaching the minimum loss. The optimiser is very important and often requires experimenting with multiple alternatives to reach the highest accuracy. Various optimisers will be discussed throughout the subsequent chapters.

Learning Rate: the learning rate is a hyper-parameter applied to the optimiser. It controls the step size for the model to reach the minimum loss function. A high learning rate will result in faster learning, but may result in missing the minimum loss. A low learning rate provides a greater chance of reaching the minimum loss, however it requires more epochs. A low learning rate slows down the learning process though converges smoothly. Usually a gradually decaying Learning rate is preferred.

Loss Function: the purpose of the loss function is to compute the quantity a model should seek to minimise during training. As with the optimiser, there are multiple loss functions for different training objectives and those relevant to this study will be explained in detail in forthcoming chapters.

Batch size: or mini-batch size refers to the number of sub-samples from the training dataset given to the network during training after which the parameter fine-tuning, or update, happens. A commonly used default batch size is 32, however it largely depends on the dataset and capacity of computational resources used for training. The smaller the batch size the faster the learning process, but the variance of the validation dataset accuracy is higher. A bigger batch size has a slower learning process, but the validation dataset accuracy will result in a lower variance.

Epochs: the number of times the whole training dataset is passed to the network during training. One epoch means that the training dataset is passed forward and backward through the network once. If the epoch size is too small this could result in the network underfitting. Conversely, too many epochs can lead to overfitting. A strategy to prevent under and overfitting is to monitor the accuracy of the network whilst training and stop once it ceases to improve (converges), saving only model with the highest accuracy.

Weight Initialisation: defines the way the initial neuron weights of the layers are set. There are a number of options, however the most common is random initialisation.

Dataset Splitting: the entire dataset should be split into 3 sub-sections: training, testing and validation sets. A common example is to retain 20% of the overall data as a validation set, completely unseen by the network and used for inference on the trained model to test accuracy. Of the remaining dataset, a split of 70%-80% and 20%-30% for training and testing, respectively, is standard.

3.2.7 Loss Functions

DL neural networks learn to map an input(s) to an output(s) from examples. The choice of loss function used to minimise the network error must match the predictive modeling problem, for example classification or regression, and the configuration of the output layer.

Optimisation refers to the process of minimising or maximising a function, called the objective function or criterion, $f(x)$ by altering x. When minimising, this optimisation function is referred to as a cost, loss or error function and can be used interchangeably [104].

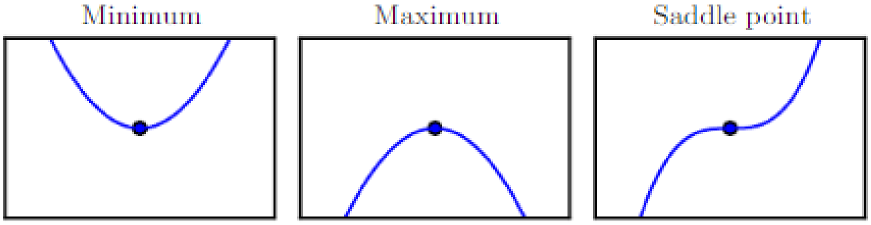


Figure 3.15: A critical point refers to a point with zero slope and can either be local minimum (left) a local maximum (centre) or a saddle point with neither a maxima or minima (right) [104]

The derivative of the cost function is denoted as $f'(x)$ and provides the slope of $f(x)$ at point x. This is useful for minimising a function because it instructs how to modify x. As observed in Figure 3.15, the local minimum refers to the point at which $f(x)$ is lower than all neighbouring data points, the local maximum occurs when $f(x)$ is higher than all other points and those with neither a maxima or minima are referred to as saddle points. It is possible to have more than one minima in the context of DL thus rendering optimisation a difficult task; particularly when the algorithm involves complex inputs. In practice, an $f(x)$ value as close to the minima as possible is often settled upon.

Many functions exist to estimate the minimum error in the neural network weights. It is typical to seek to minimise the error, hence the value calculated by the loss function is commonly referred to as "loss". Loss functions are explained with regard to specific DL tasks in subsequent chapters of this thesis.

3.2.8 Optimisers

Optimisers are algorithms used to minimise the loss function. During each training epoch, the model's weights are updated to minimise the loss and the optimiser adaops the network's parameters, such as learning rate and weights. Therefore, the optimiser assists in improving the accuracy of a trained network. However, there are multiple optimisers and the choice of best fit can be daunting but is generally related to the problem domain. Several optimisers are discussed in the subsequent chapters of this thesis.

3.2.9 Network Training

Once a CNN architecture, initial parameters and hyper-parameters have been set, there are three main ways to train the network. These are:

Training From Scratch

This method is highly accurate though it is also the most challenging as it requires significant computational resources and a large, labelled, dataset.

Transfer Learning

Based on the premise you can use knowledge of solving one type of problem to solve another. For example, you could use a pre-trained network capable of categorising dogs and cats to initialise and train a new model that differentiates between two types of flower. This method requires less data and fewer computational resources than training from scratch.

Feature Extraction

Finally, it is possible to use a pre-trained model (transfer learning) to extract features for training a model. For example, features extracted from one of the hidden layers may relate to finding the edge of an image and, as such, can be applied to any computer vision task. This approach requires the least amount of training data and computational resources.

3.3 Classification

Several state-of-the-art CNN architectures for image classification exist. Indeed, available within the Keras framework there are eighteen pre-trained networks in the Applications module freely downloadable to import into a new project. For example, for transfer learning or feature extraction purposes [117]. It would be beyond the scope of this project to detail each, however a brief history is considered along with the most important, award winning architectures: AlexNet, VGG, Inception/GooLeNet and ResNet.

The design of a CNN for image classification has, in essence, been described in the previous sections of this chapter. However, the challenge is in constructing an accurate and efficient network using these basic building blocks (convolution, pooling and fully-connected layers) and a combination of hyper-parameter choices and tuning.

Widely recognised as the first successful application of a CNN architecture applied to classification is LeNet-5, proposed by Lecun, et al. in their 1998 paper titled "Gradient-Based Learning Applied to Document Recognition" [118]. The network was trained on the publicly available MNIST hand-written digit dataset [119] containing $\sim$ 70,000 grayscale images (size 32x32) and achieved an impressive 99.2% accuracy.

The LeNet-5 network consists of a pattern of convolutional layers followed by an average pool layer (referred to in the paper as a subsampling layer). This pattern is repeated two and a half times before output feature maps are flattened and processed by two fully-connected layers and, finally, a prediction layer. A high-level diagram depicting the LeNet-5 architecture can be observed in Figure 3.16, below.

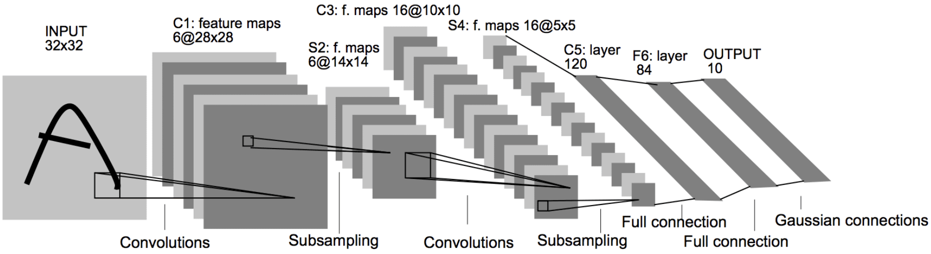


Figure 3.16: High-level overview of the LeNet-5 CNN architecture for classification of hand-written digits [118]

In the first convolutional hidden layer, six kernels are used with a dimension of 5x5 pixels. After the average pooling process there is another convolutional layer with sixteen filters of the same size, again followed by average pooling. The trend of increasing the number of filters as the architecture deepens is one seen in modern architectures. However, the number of filters used in LeNet-5 is considered comparatively small when compared with today's standards.

The pattern of blocks of convolutional layers and pooling layers grouped together and repeated remains common today. Similarly, flattening feature maps is a continued approach, as is the extraction of features by the final fully-connected layer.

To analyse the effectiveness of a classification network architecture, the successful entrants to the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [120] can be explored. The ILSVRC project was instrumental in advancing the field of computer vision and classification tasks due to rapid research and innovations in competition entries (between 2012 and 2016, when it ceased to run). ImageNet is an image classification and localisation dataset comprising 1,500,000 images from 1,000 classes and remains free for researchers to download and use in personal projects.

3.3.1 State-of-the-Art Classification Networks

AlexNet

In their 2012 paper titled "ImageNet Classification with Deep Convolutional Neural Networks", Alex Krizhevsky, et al. [121] propose a CNN architecture called AlexNet as an entry to the ILSVRC competition. The paper innovation showed it is possible to develop a deep and effective end-to-end model for challenging problems.

The AlexNet architecture utilises the ReLU activation runction for non-linearity after each convolutional layer. This was considered an innovation because "S-shaped" functions such, such as Tanh, were common at the time. The activation function for the final output layer is Softmax; now commonly applied to multi-class classification problems.

The architecture of AlexNet builds upon some of the patterns established with LeNet-5, though max pooling is applied instead of the average method. The newly proposed dropout regulation technique was also applied to the fully-connected layers to prevent overfitting and improve generalisation. The AlexNet architecture is summarised in Figure 3.16 below (note there are two pipelines due to training on GPU hardware).

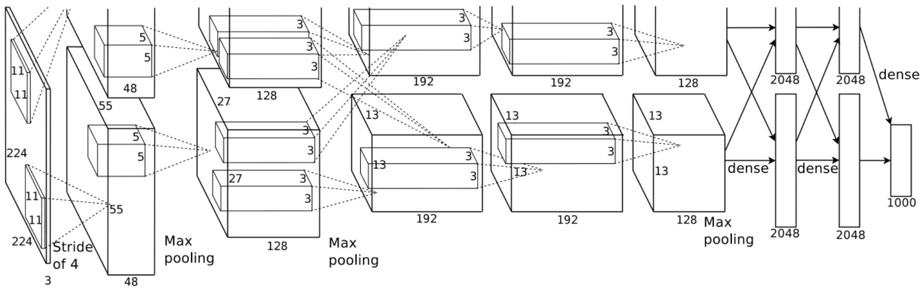


Figure 3.17: An illustration of the AlexNet CNN architecture for multi-class classification of the ImageNet dataset [121]

A summary of the AlexNet model architecture is as follows:

- Image dimension of 224 x 224 pixels with three colour channels
- Five convolutional layers in the feature extractor
- Increasing filter numbers as the network deepens (96, 256, 384, 384, and 256)
- Kernel size decreases as network deepens (11x11, then 5x5 and, finally, 3x3)
- Three fully connected layers in the classifier
- Data augmentation was used to artificially increase the size of the training dataset

VGG

2014 saw the publication of a paper titled “Very Deep Convolutional Networks for Large-Scale Image Recognition” by Karen Simonyan and Andrew Zisserman in which they propose an architecture commonly referred to as VGG (after their lab, the Visual Geometry Group at the University of Oxford). Their model was an entry to the ILSVRC in the same year [122].

An important distinction between VGG and earlier architectures, and a technique still commonly applied today, is a large number of small filters with a small stride. Additionally, max pooling layers follow most convolutional layers (similar to AlexNet), however, all pooling is applied with a kernel size of 2x2 and a fixed stride. Again, the number of filters increases with the depth of the network, though the volume is far greater (64 to 512).

There are several variants of the VGG network, most commonly VGG-16 and VGG-19 (containing 16 and 19 learned layers, respectively). The VGG work was among the first to release their model weights under a permissive license, leading to a trend among CV researchers. This, in turn, has led to the heavy use of pre-trained models, like VGG, in transfer learning as a starting point on new CV tasks. An overview of the standard VGG-19 network architecture can be found in Figure 3.18

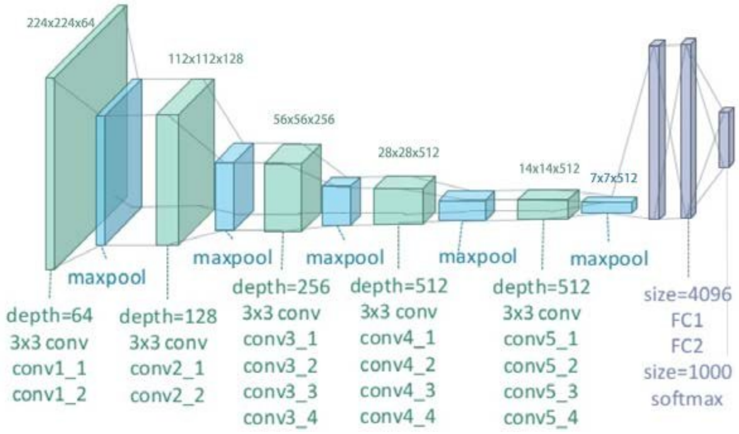


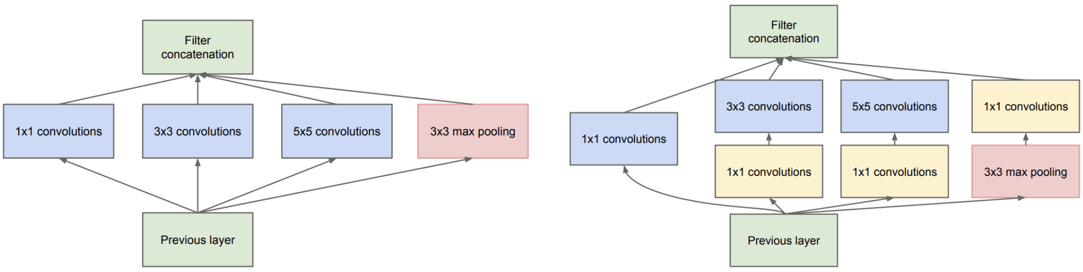
Figure 3.18: An overview of the standard VGG-19 network architecture [123]

Inception and GoogLeNet

By 2016, Szegedy, et al. published their paper titled “Going Deeper with Convolutions”, complementing important innovations in the field of DL for classification at the time [124]. Most notably, Inception is proposed in addition to GoogLeNet that also achieved top results in the 2014 ILSVRC challenge (the same year as VGG).

There have since been multiple iterations, central to all is the inception module; a block of parallel convolutional with filters of differing sizes (1×1, 3×3, 5×5, etc.) followed by a max pooling layer. The resulting feature maps are subsequently concatenated.

However, in the naive implementation of Inception, when multiple inception modules are stacked, the number of filters rapidly increases and is computationally expensive. Consequently, 1x1 convolutional layers were implemented to reduce the number of filters, before the 3x3 and 5x5 convolutaional layers and pooling. An illustration of the dimensionality reduction in the Inception model is illustrated in Figure 3.19 below:



(a) Inception module, naïve version

(b) Inception module with dimension reductions

Figure 3.19: An illustration of a naive inception module (left) and the improved architecture (right) designed to perform dimensionality reduction [124]

A second important design decision in the Inception model was connecting the output at different points in the model. This was achieved by creating small off-shoot output networks from the main network that were trained to make a prediction. The intent was to provide an additional error signal from the classification task at different points of the deep model in order to address the vanishing gradients problem. These small output networks were then removed after training.

Figure 3.20 illustrates a rotated network architecture (left-to-right) for the Inception V1 model with the addition of two auxiliary classifiers; an important design feature. These small, off-shoot output networks are trained to make a prediction with the intention of providing an additional error feedback for the classification task at various stages of the deep neural network, in an attempt to address the vanishing gradient problem. The auxiliary classifiers are removed after training is complete. Overlapping max pooling was used and a large average pooling operation is applied at the end of the feature extraction part of the model prior to the classifier part of the model with global average pooling for the output of the model.

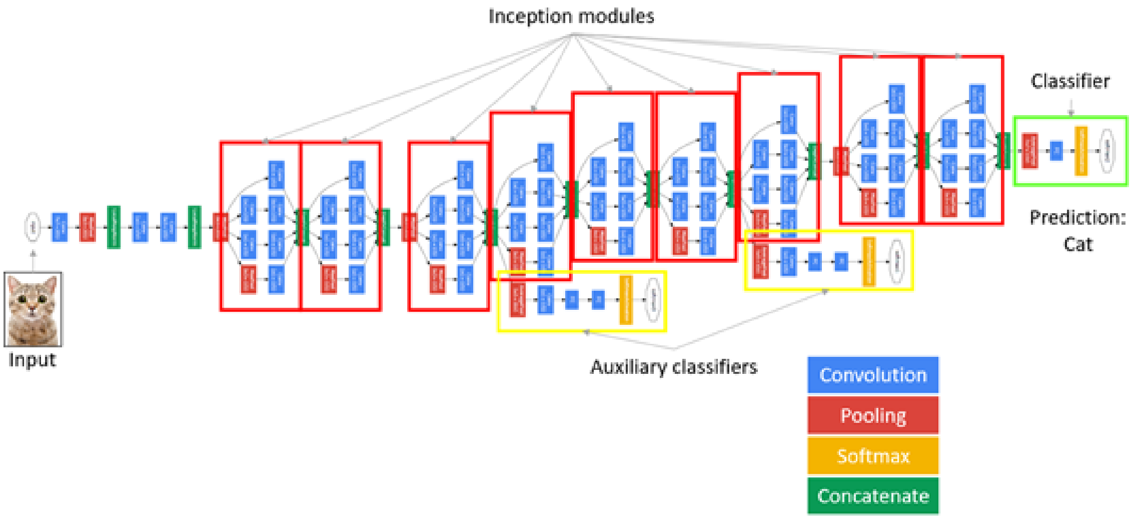


Figure 3.20: Rotated (left to right) illustration of the Inception V1 architecture

ResNet

In the same year as Inception (2016), another important innovation in convolutional neural networks for classification, called a Residual Network (ResNet, for short) was proposed by He, et al. in their paper titled "Deep Residual Learning for Image Recognition" [125]. Again, ResNet competed in the ILSVRC challenge (2015) and achieved great success.

The ResNet network is very deep (152 layers) and central to it's design are residual blocks utilising shortcut connections. These shortcuts simply take the input (with no weights applied) and connect it to a deeper layer, in essence skipping the intermediate layer entirely. The residual block is comprised of a pattern of two convolutional layers (with ReLU activation), the output of the block is then combined with the input to the block (the shortcut, or skip, connection). The shape of the input for the shortcut connection is the same size as the output of the residual block.

From left to right, Figure 3.21 compares the architecture of a VGG model, a plain convolutional model, and a version of the plain convolutional with residual modules, called a residual network.

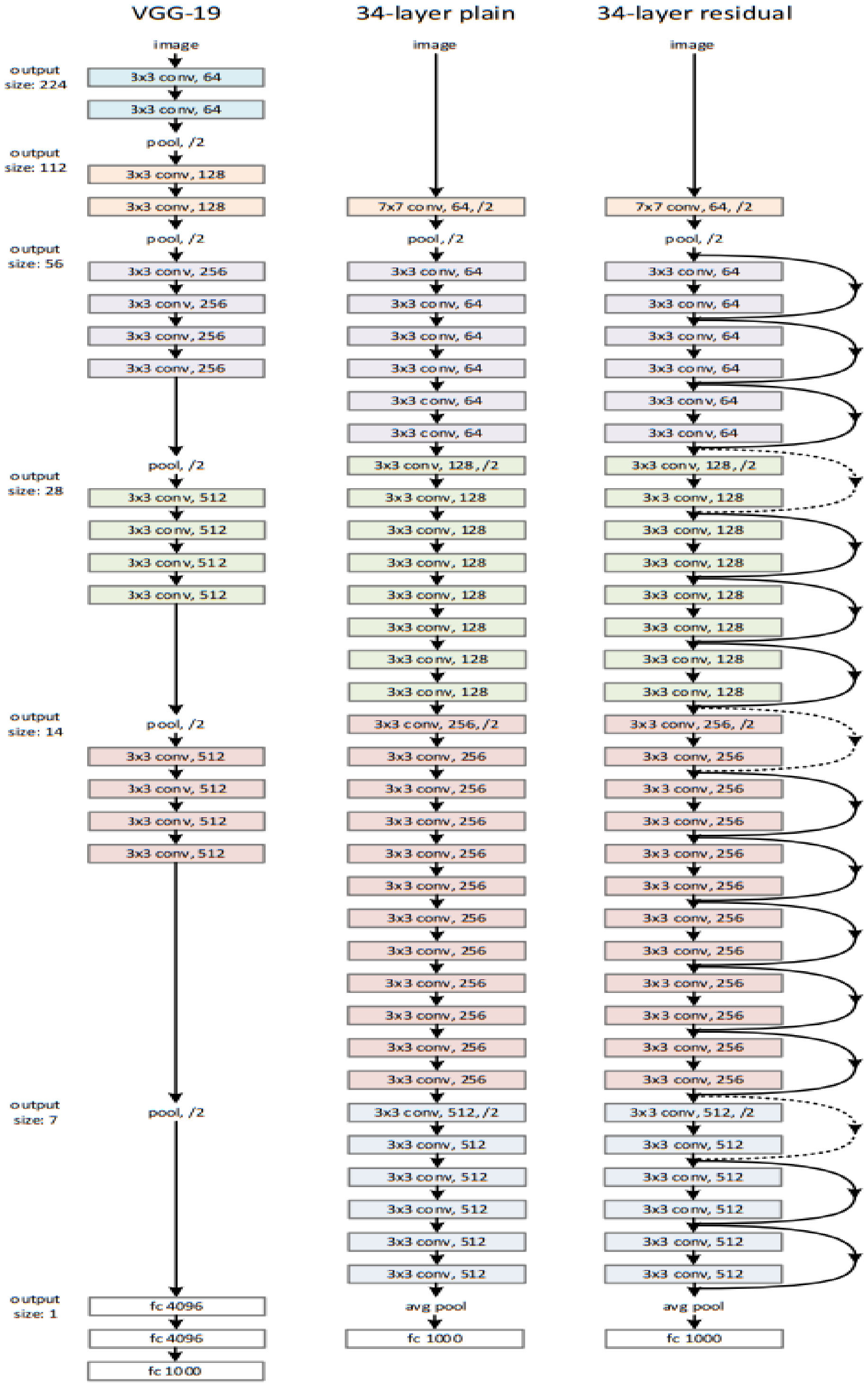


Figure 3.21: Architecture of a residual network with skip connections (right) compared to a standard network (centre) and VGG-19 (left) [125]

3.3.2 Evaluation Metrics

Evaluation metrics are employed to estimate how well a trained model will generalise to previously unseen data. A summary of general classification evaluation metrics are as follows:

Accuracy

A calculated percentage describing how well the model performs across all classes (particularly useful when all classes are of equal importance)

Precision

Precision measures the model's accuracy when classifying a sample as positive. The ratio of positive samples correctly classified to the total number of samples classified as positive (either correctly or incorrectly) is compared.

Recall

Recall measures the model's ability to detect positive samples. The higher the recall, the more positive samples were detected. To calculate recall, the ratio between the number of positive samples correctly classified to the total number of positive samples is compared.

Sensitivity

A measure of how well a model can detect positive instances, also known as the True Positive Rate (TPR).

Specificity

A measure of the model's ability to predict a true negative of each classification category, also known as the True Negative Rate (TNR).

Confusion Matrix

A confusion matrix helps to visualise the performance of a model for either binary or multi-class classification. The example in Figure 3.22 below shows a 2x2 confusion matrix for a binary classification problem.

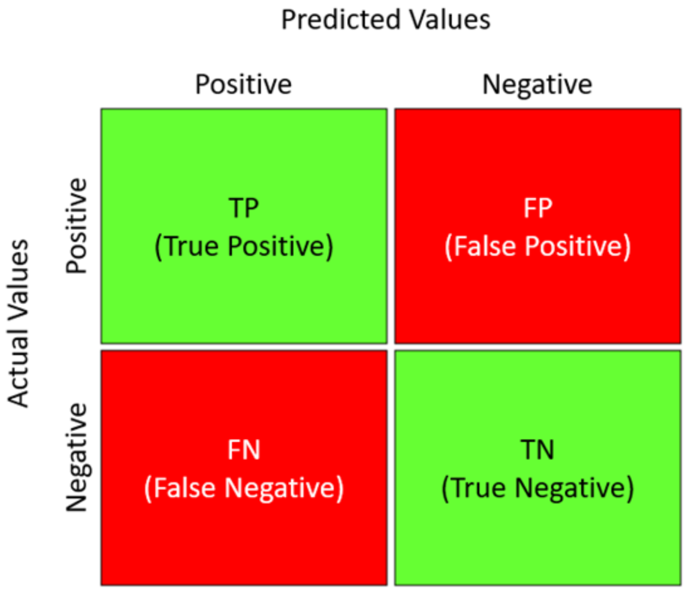


Figure 3.22: Simple example of a binary classification confusion matrix

True Positive (TP): The predicted valuse is positive and the ground-truth is positive

False Positive (FP): The prediction is positive, but the ground-truth is negative

False Negative (FN): The prediction is negative, but the ground truth is positive

True Negative (TN): The prediction is negative and the ground-truth is negative

Accuracy, sensitivity and specificity are calculated using the following equations:

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

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

K-fold Cross-Validation

K-fold cross validation is a simple evaluation technique to test a model's performance on a limited, unseen data sample. "K" is a parameter referring to the number of groups the data sample will be split into. K-fold cross-validation is popular because it is simple and results in a less biased (less optimistic) estimation of the model's capabilities.

Below is a general overview of the K-fold cross validation process:

1. Randomly shuffle the dataset

2. Split the dataset into "K" groups, or folds, (for example, 10)

3. Iteratively, for each group, keep aside and train on the remaining 9 groups. After each training iteration, test on the retained group then disregard the model. Make a note of the accuracy for each of the "K" data groups

4. The accuracy (or skill) of the model is a summary of all "K" accuracies

Receiver Operating Characteristic Curve (ROC)

The ROC curve plots the performance of a classification model at different classification thresholds. The curve represents two values: TP and FP. Figure 3.23 shows a typical ROC curve.

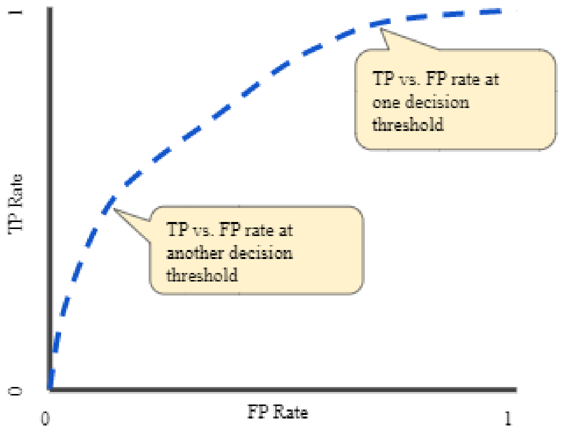


Figure 3.23: Sample ROC curve showing TP Vs FP rate at different classification thresholds [126]

Area Under the ROC Curve (AUC) measures the 2D depth below the curve, as demonstrated in Figure 3.24.

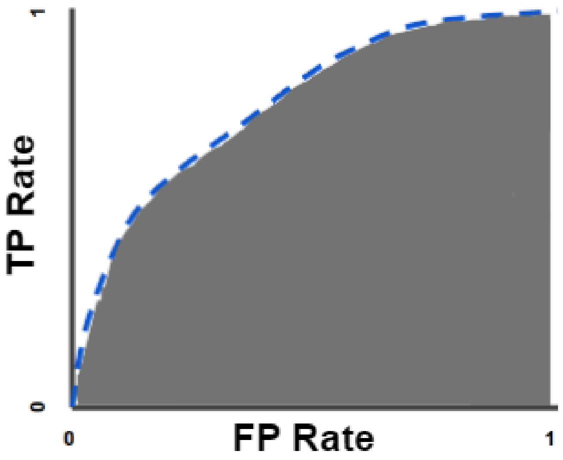


Figure 3.24: Sample ROC curve showing TP Vs FP rate at different classification thresholds [126]

The AUC provides a measure of performance across all classification thresholds and the objective is to increase the area as much as possible, thus demonstrating increased performance.

F1-score

The F1-Score is a method of combining precision and recall. However, it is possible to adjust the preference to give more importance to precision over recall, or vice-versa. The higher the F1 score, the more accurate the model.

The mathematical equation to calculate the F1 score is as follows:

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

3.4 Object Detection and Segmentation

Object detection (or localisation) and segmentation are heavily-researched topics in the computer vision domain and can be considered an extension of classification. Image classification assigns a class label to an image, whereas object localisation not only classifies one or more instances of an object occurrence, it pinpoints exactly where the specific object(s) occur by drawing a bounding box or outlining its boundary. Hence, object detection and segmentation are more complex tasks than classification alone, as illustrated in Figure 3.25.

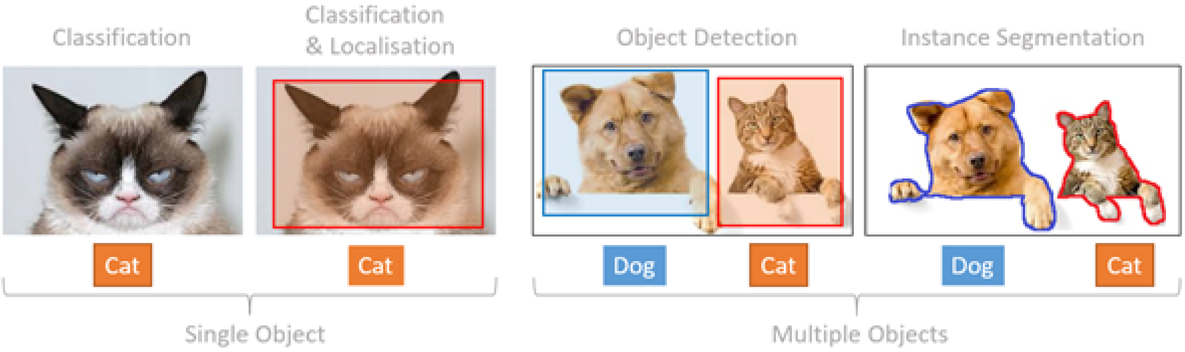


Figure 3.25: An illustration differentiating single and multiple classification, localisation and instance segmentation

Most object detection and segmentation models consist of an encoder-decoder architecture, as opposed to a single encoder as in classification. Supervised training requires a corresponding map for each image, as highlighted in Figure 3.26, which can be time-consuming and costly. Additionally, use cases such as medical imaging and autonomous vehicles require high precision annotations.

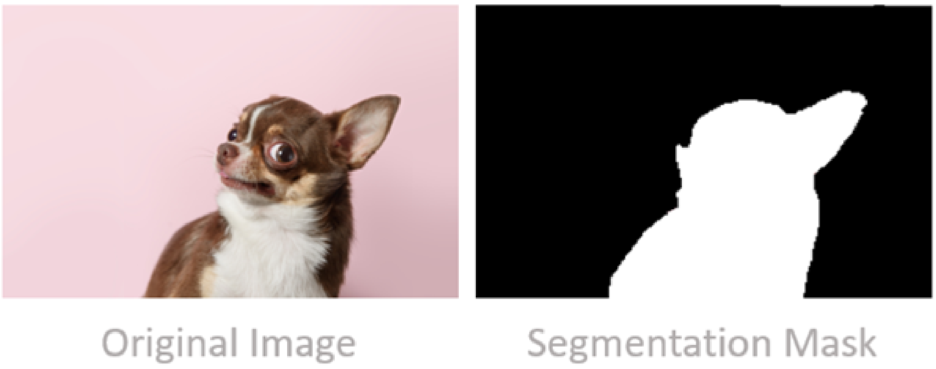


Figure 3.26: An example digital image (left) with corresponding segmentation mask (right)

Segmentation masks are n-channeled with n being the number of object classes to segment. Each channel is binary in nature with object pixels represented by ones and empty regions consisting of zeros (as in Figure 3.26). The output of the network is a predicted n-channel mask, similarly in a binary format, also known as a two-dimensional one-hot encoded representation of each prediction.

Segmentation tasks can be classified into three groups: semantic, instance and panoptic.

Semantic segmentation singles out a broad boundary of objects belonging to a particular class, while instance provides a 'map' of the object boundary (without a class prediction). Panoptic segmentation provides the most information with a combination of semantic (the class) and instance (the object shape and location). Examples of which can be observed in Figure 3.27.

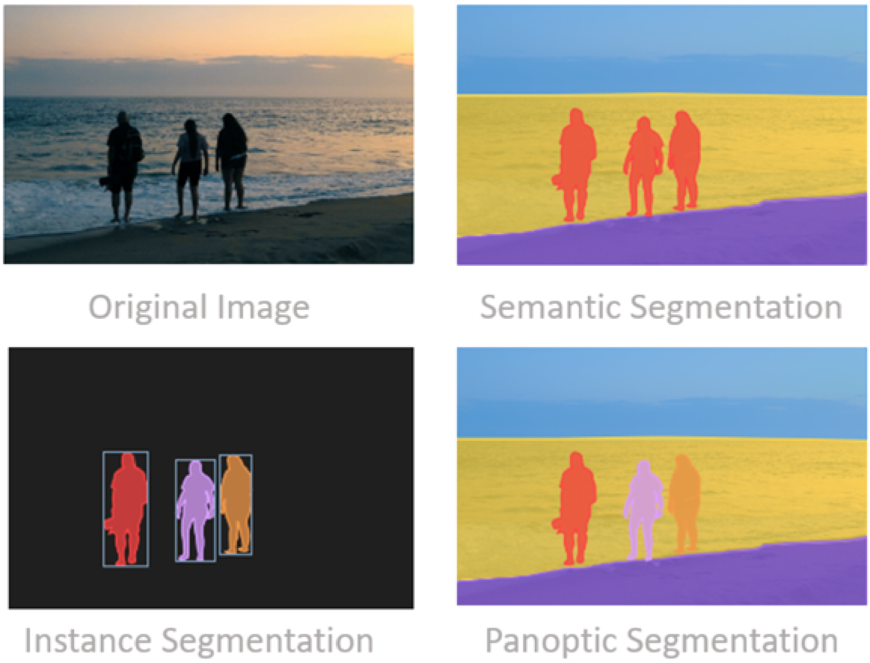


Figure 3.27: An original digital image (top left) with examples of semantic segmentation (top right), instance segmentation (bottom left) and panoptic segmentation (bottom right)

3.4.1 State-of-the-Art Object Detection and Segmentation Networks

The most recent breakthroughs and innovations in the field of image recognition problems have come from participation in the ILSVRC, running from 2010 to the present day [127].

In 2015, Badrinarayanan et al. proposed the concept of an encoder-decoder architecture for semantic segmentation, named SegNet [24]. SegNet uses a combination of convolutional and pooling layers to create a bottleneck, forming a condensed representation of the input; called the encoder. The decoder reconstructs the image to formulate a segment map. Within which, regions of interest are highlighted and grouped under their corresponding classes. The final stage of the network decoder is a Sigmoid activation function to squeeze the output prediction pixel values between a range of 0 - 1.

At the same time as SegNet, another segmentation project, called U-Net, proposed by Ronneberger et. al. was the first to introduce skip connections in DL with the aim of solving the loss of spacial information during downsampling layers [128]. Skip connections refer to connections from the encoder directly to the decoder, via concatenation of feature maps, without passing through the bottleneck. They have proven extremely popular, especially in the domain of computer vision techniques for medical imaging. U-Net, in particular, has demonstrated state-of-the-art results in multiple anatomical segmentation tasks. High-level overviews of both SegNet and U-Net architectures can be found in Figure 3.28.

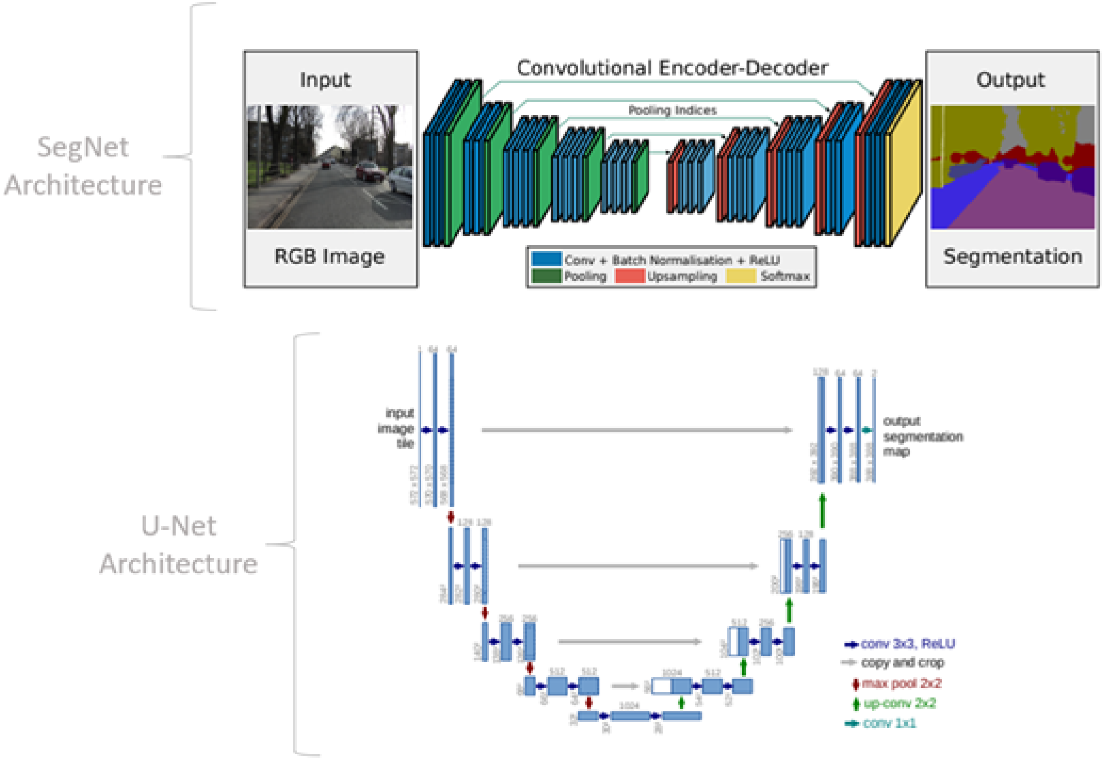


Figure 3.28: High-level illustrations of both SegNet (top) and U-Net (bottom) architectures [24, 128]

Region-Based Convolutional Neural Networks (R-CNN), are a collection of architectures designed to address object detection and recognition tasks, optimised for model performance. The success of previous segmentation networks paved the way for further innovation.

Notably, DeepLab [129] proposed Atrous convolutions, replacing traditional pooling layers while preventing information loss during the downsampling process. Furthermore, DeepLab introduced multi-scale feature extraction consisting of Atrous Spatial Pyramid Pooling to enable the network to segment objects regardless of size. Accurately detecting boundary information of objects is crucial to both semantic and instance segmentation. DeepLab utilised fully connected Conditional Random Fields (CRFs) for fine-grained localisation resulting in their methods to outperform early proposed fully-connected networks and SegNet by a significant margin.

Work such as SegNet, U-Net, and DeepLab paved the way for future innovations, such as the architectures detailed below:

R-CNN Models

R-CNN

R-CNNs for object localisation and recognition were first proposed by Girshick, et al. in their 2014 paper titled “Rich feature hierarchies for accurate object detection and semantic segmentation” [130].

As illustrated in 3.29, the R-CNN model architecture is comprised of three stages:

1. **Region Proposal:** for the generation of object region proposals and creation of bounding boxes

2. **Feature Extractor:** extract features for each region using a deep convolutional neural network (AlexNet)

3. **Classifier:** categorise feature candidates (via linear SVMs) as one of the known classes

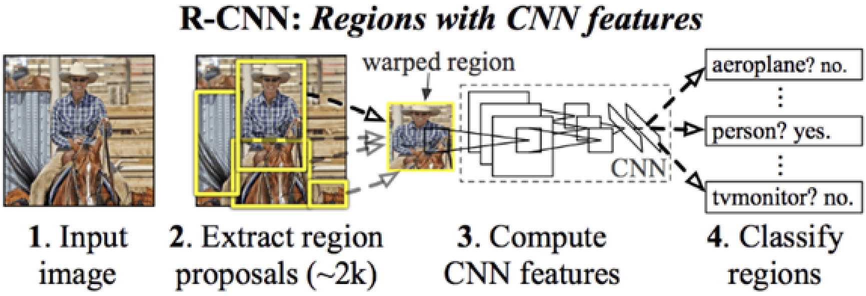


Figure 3.29: Summary of the R-CNN model architecture [130]

This initial R-CNN proposal was simple and effective for the problem of object localisation and recognition. However, with approximately 2,000 proposed regions per image it was computationally slow. Thus, several updates have been proposed to improve the accuracy and efficiency of the R-CNN architecture, as detailed below.

Fast R-CNN

In 2015, Girshick proposed "Fast R-CNN" [131] to resolve the limitations of R-CNN by using a single model as opposed to the previous method of a pipeline comprising multiple trained networks. A summary of the Fast R-CNN architecture can be found in Figure 3.30. However, while the network is significantly faster than it's predecessor, it still requires multiple region candidates to be proposed for each input.

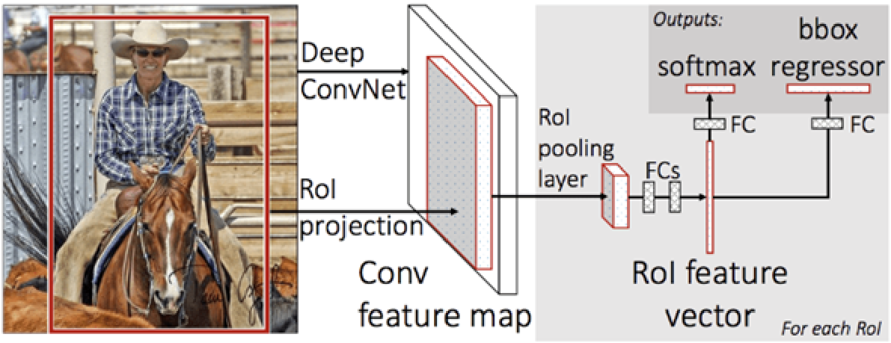


Figure 3.30: Summary of the Fast R-CNN model architecture [131]

The key difference between R-CNN and Fast R-CNN is a custom layer, after feature extraction, at the end of the deep CNN (such as VGG-16), called a Region of Interest Pooling Layer (ROI Pooling). The purpose of the ROI Pooling is to extract features specific to each proposed region.

The output of the network is interpreted by a final, fully-connected layer and bifurcated into an output class prediction (via a Softmax layer) and a linear output for the bounding box coordinates.

Faster R-CNN

Faster R-CNN, proposed by Shaoqing Ren, et al. in 2016 [132], is the latest iteration of the R-CNN family and, to date, achieves state-of-the-art accuracy for object recognition tasks. Faster R-CNN achieved top results on both ILSVRC 2015 and MS COCO-2015 object recognition and detection competition tasks.

Faster R-CNN builds upon the efficiency and accuracy of it's predecessors with the addition of a Regional Proposal Network (RPN). The RPN passes a small network over the feature map output from a pre-trained deep CNN (such as VGG-16), resulting in multiple regional proposals and classifications for each. The class prediction is binary; indicating the presence of an object or not, termed "objectness". These predictions are then used in conjunction with Fast R-CNN to reduce inference time to near 'real-time'.

Although one unified network with shared weights, the Faster R-CNN architecture is comprised of two modules (illustrated in Figure 3.31:

1. **Region Proposal Network:** a CNN, acting as an attention mechanism, for proposing regions and what type of object could be considered within each region

2. **Fast R-CNN:** also a CNN for feature extraction from the proposed regions of interest and subsequently outputting the bounding box and class labels

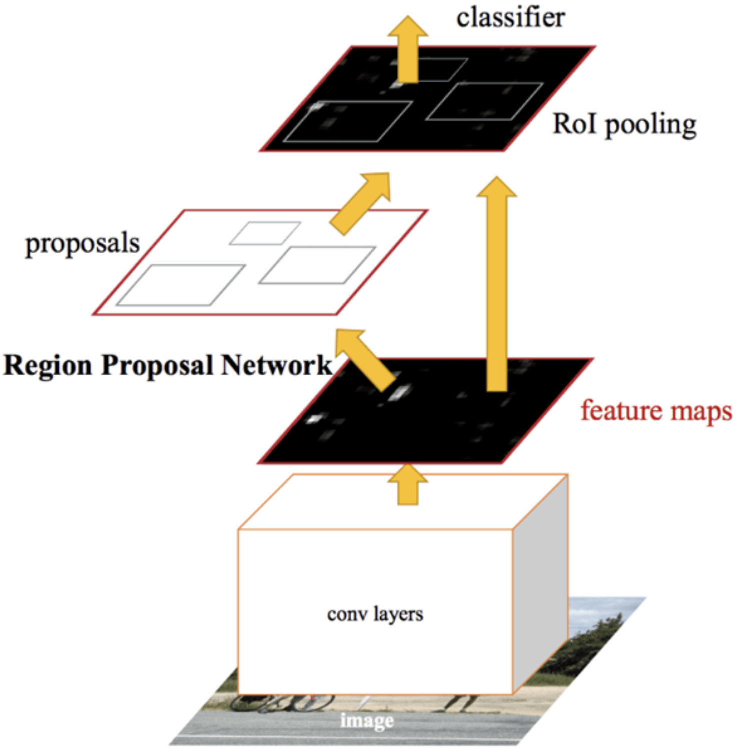


Figure 3.31: Summary of the Faster R-CNN model architecture [132]

Faster R-CNN has since been further extended to include instance segmentation in the 2017 paper titled "Mask R-CNN" proposed by Kaiming He, et al. [125]. More information about the Mask R-CNN implementation can be found in chapter 8.

YOLO Models

YOLO, or "You Only Look Once", is another popular collection of recognition models proposed by Joseph Redmon, et al. in their 2015 paper titled "You Only Look Once: Unified, Real-Time Object Detection" [133]. Whilst R-CNN networks are accurate but comparatively slow, YOLO models are much faster but where they gain the speed advantage, accuracy is compromised.

The YOLO architecture splits the input image into a grid of cells. Should a cell contain the centre of a predicted bounding box, it is responsible for producing coordinate predictions along with a confidence score. Each cell is also given a class prediction, as depicted in Figure 3.32.

For example, if an input image were divided into a 7x7 grid, each cell may predict 2 bounding boxes which would result in 94 proposed bounding box predictions. All class probabilities and bounding box confidence scores are combined to produce a final set of bounding boxes and class labels.

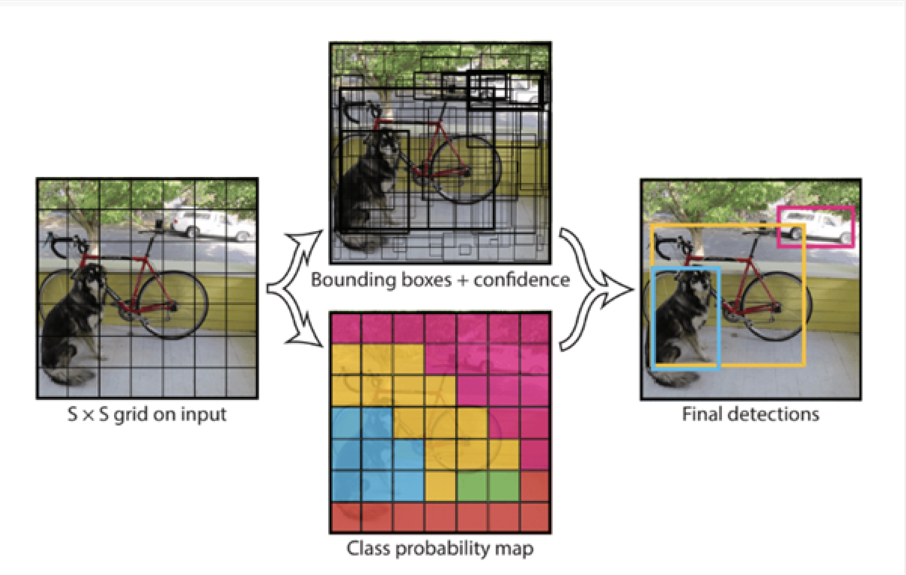


Figure 3.32: A visual outline of the YOLO model architecture [133]

Since the publication of the seminal YOLO paper in 2015, there have been two updates. Firstly, Joseph Redmon and Ali Farhadi proposed YOLO v2, trained on two object recognition datasets in paralel along with a number of training and architectural updates. Such as including the use of batch normalisation and high-resolution training images [134]. More recently, in 2018 Joseph Redmon and Ali Farhadi again proposed some minor improvements, naming the latest model YOLO v3. This iteration includes a deeper feature extractor though other changes are relatively minor [135].

3.4.2 Segmentation Evaluation Metrics

Intersection over Union (IOU) and Dice Score (DC)

Intersection over Union (IOU), also known as the Jaccard Index, and Dice Coefficient (DC), also referred to as the F score, are the most commonly applied evaluation metrics for image segmentation. Both metrics are positively correlated, meaning that if the performance of model A is better than model B under IoU, it will also evaluate as better with DC. Both metrics range from 0 to 1, with 1 signifying a perfect match and 0 being completely disjoint.

IOU measures the intersection (overlap) of pixels in the prediction and ground truth divided by the area of union; the total number of pixels when both the ground truth and prediction are combined, as illustrated in Figure 3.33. The overlap can also be referred to as the True Positive (TP) rate.

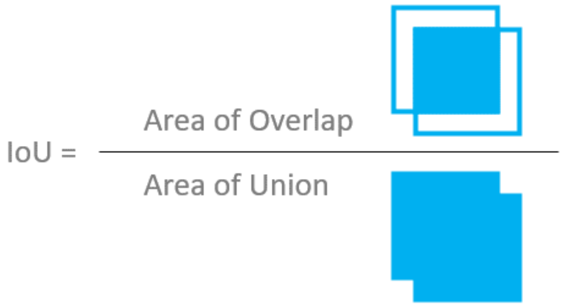


Figure 3.33: The calculation of the Intersection over Union (IoU) metric

The False Positive (FP) rate refers to the number of pixels predicted outside of the ground truth and the False Negative (FN) rate refers to the number of pixels in the ground truth that the model failed to predict. Figure 3.34 provides a visual illustration of the TP, FP and FN rate.

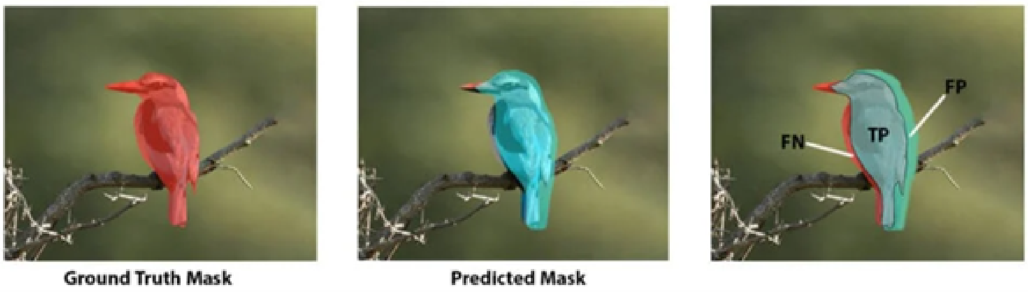


Figure 3.34: A visual illustration of the TP, FP and FN rate for semantic image segmentation [136]

The IoU is the ratio of the overlapping pixels to the combined area of the prediction and ground truth mask and can be expressed in terms of TP, FP and FN regions as follows:

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

The DC is calculated by multiplying the area of overlap by two and dividing by the sum of pixels, as illustrated in Figure 3.35.

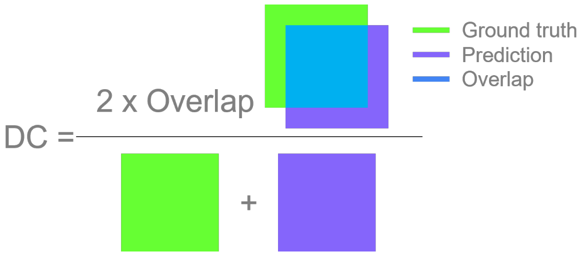


Figure 3.35: The calculation of the Dice Score (DC) metric

The DC can also be expressed, in terms of TP, FP and FN regions as follows:

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

A common assumption is that both IoU and DC are functionally equivalent and thus, it is necessary to evaluate a trained model using only one or the other. However, the difference occurs when taking average scores from a set of inferences and determining how much worse one model performs over another, as opposed to calculating the score of a single inference.

Generally, IoU quantitatively penalises a single instance of a poor prediction greater than DC, even when both metrics indicate an instance of a prediction is poor. IoU can be considered to have a squaring effect, thus giving more weight to larger differences. Hence, the DC can be considered as a measure of the average performance, whereas when taking the mean over a set of inferences the IoU generates a measure closer to the worst-case performance.

3.4.3 Object Detection Evaluation Metrics

Precision-Recall Curve (PR Curve)

Precision refers to the degree of accuracy the model achieves when identifying relevant objects. Recall is a measure of a model's ability to detect all ground-truths. Ideally, a model will have high precision and high recall, thus a perfect model would have no FNs or FPs.

The PR Curve summarises the trade-off between the TP rate (x-axis) and the network precision on the y-axis (also called the positive predictive value) at various thresholds.

Average Precision (AP)

AP is calculated by obtaining the AUC of the PR Curve.

Mean Average Precision (mAP)

The mAP score is calculated by averaging all AP over multiple classes and/or Intersection Over Union (IoU) thresholds.

3.5 Recurrent Neural Networks (RNNs)

Recurrent Neural Network (RNN)s are capable of achieving state-of-the-art accuracy on tasks such as language modeling, speech recognition, and machine translation. [137]. RNNs are designed to solve sequence based tasks with the addition of a memory state for learning valuable temporal information in conjunction with spatial features. Cycles within the neural network graph, enable an internal state allowing for the evaluation of sequence information. However, there are two salient considerations when utilising RNNs:

- How to train the RNN using backpropagation
- The vanishing (or exploding) gradient problem

3.5.1 Backpropagation Through Time (BPTT)

Backpropagation is essential for the updating of weights after each training epoch in a standard feed-forward neural network. However, due to the cycles, or loops, in RNNs standard backpropagation is ineffective. BPTT is an amended backpropagation technique designed to tackle this problem.

The basic premise of BPTT is the “unrolling” of the RNN network; essentially generating copies of the neurons which have recurrent connections. Therefore, the cyclic nature of the network is assembled into a graph-like feedforward network for which backpropagation can be applied.

RNNs contain cycles that feed the activations from a previous time step as inputs to the current by way of influencing predictions. These activations are stored in the so called “internal state”; acting as memory and holding long-term temporal information. Thus allowing RNNs the opportunity to append a contextual “window” over the input sequence [138].

3.5.2 Vanishing Gradient Problem

A common problem when training very deep neural networks using backpropagation is vanishing, or exploding, gradients. This is because the weights can oscillate and become very large (exploding) or very small (vanishing) and the network becomes unstable.

In modern RNNs this problem has been addressed with the introduction of the commonly used Long Short-Term Memory (LTSM) and Gated Recurrent Networks (GRU).

3.5.3 Overview of Long Short-Term Memory (LSTM)

The LTSM network was developed with the specific aim of overcoming the vanishing gradient problem. LTSMs differ from traditional recurrent neural networks because of their “feedback connections”. These connections allow LTSMs to process sequences of data (also referred to as a “time series”, with each instance being a “time step”), without treating each point in the sequence separately. Rather, useful temporal relationship information is retained about previous data in the sequence intended to assist with the interpretation of subsequent data points. LTSM networks are particularly effective in processing sequences such as text, speech and video time-series data.

Instead of neurons, LTSMs comprise memory ’blocks’ which are connected into network layers. Each block contains a memory component to keep hold of important information about the recently processed part of a sequence. The block also contains gates that control the block’s current state and output, via a combination of Sigmoid and Tanh activation functions. Thus, the change of state and addition of information flowing through the unit is conditional.

At the most basic level, the output of one cell, or memory block, depends upon three factors:

1. The current cell state, which refers to the long-term memory of the network having stored information about previous time steps

2. The output of the previous cell, known as the hidden state, which acts as input to the current cell

3. The input data at the current time step

Figure 3.36 provides an illustration of a single LTSM cell. Each cell comprises three gates: forget, input and output. Each gate has it's own weights that are learned during training.

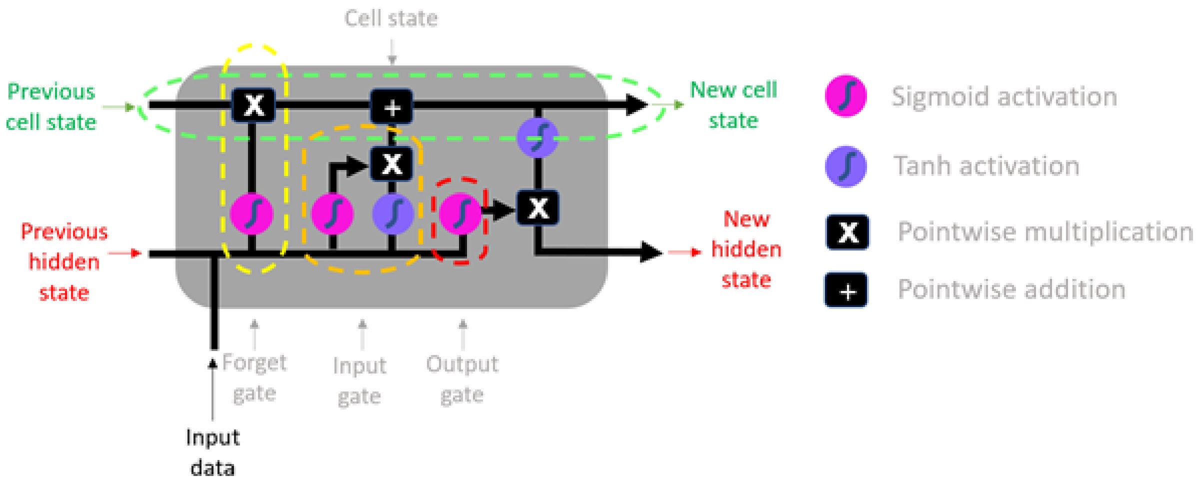


Figure 3.36: An illustration of an LTSM cell

Step 1: The Forget Gate

The forget gate decides what parts of the current cell state (long-term memory) are useful when considering both the previous hidden state and the new input data from the current time step. Both the new input and the hidden state are fed to a neural network that generates a vector within which each element is in the interval of [0, 1]. This is due to the application of the Sigmoid activation function. For each vector element, the output will be close to 0 if the information is deemed irrelevant and closer to 1 if it is considered of high importance. The output vector is then pointwise multiplied with the previous cell state, as illustrated in Figure

3.37

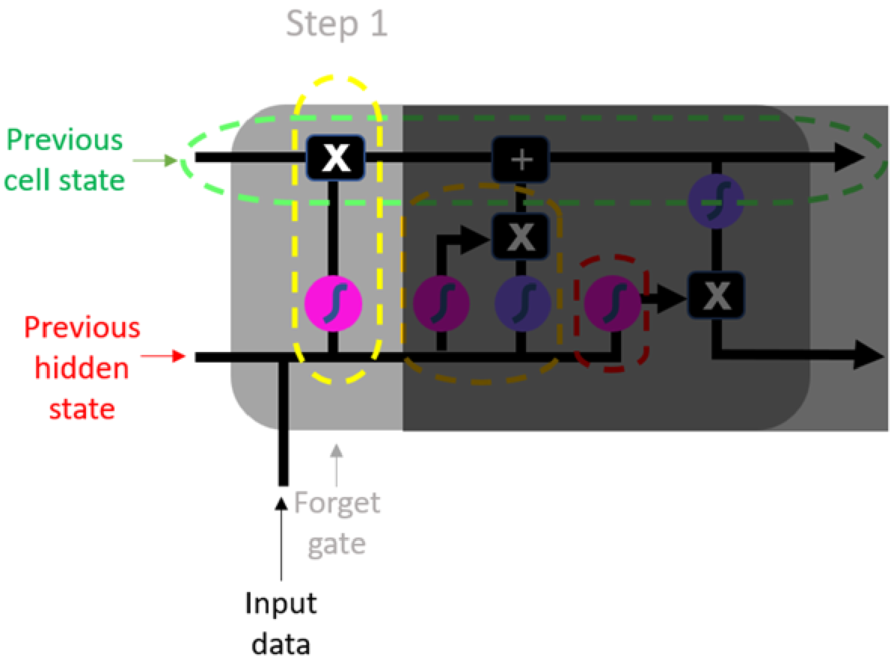


Figure 3.37: Step 1: The Forget Gate

Step 2: The Input Gate

The inputs to the input gate are the same as those to the forget gate: the new time step input and the previous hidden state. The goal of the input gate is to determine which new information should be added to the long-term memory (cell state).

The new memory from the previous hidden state is fed into a neural network with a Tahn activation function; ensuring the vector values are between [-1, 1]. The input gate is a Sigmoid activated network, so the input vector elements are filtered and normalised between [0, 1]. Both vectors are then filtered through pointwise multiplication, similar to the forget gate. Finally, the resulting vector is added to the cell state, thus updating the long-term memory. Step 2 can be seen in Figure 3.38.

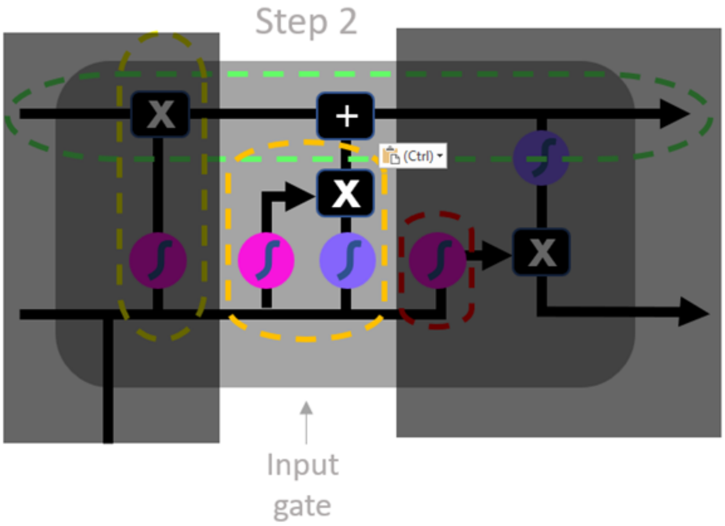


Figure 3.38: Step 2: The Input Gate

Step 3: The Output Gate

Once the updates to the cell state have been applied, the final step involves the output gate and deciding upon the new hidden state. For this, the newly updated long-term memory, the previous hidden state and the new input data are analysed.

The output gate acts as a filter, to ensure only relevant information is saved. Before this, the cell state is processed through a neural network with a Tanh activation function (to squish the values between [-1, 1]). Next, the previous hidden state and the current input vector are processed by the Sigmoid activation function, the output being the filtered vector. Finally the output (the new hidden state) is the result of pointwise multiplication between the cell state and the filtered input vector. Step 3 is highlighted in Figure 3.39

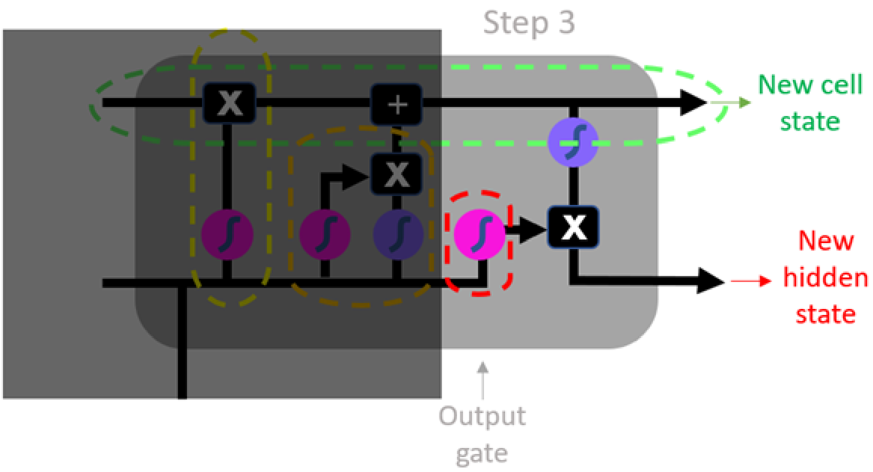


Figure 3.39: Step 3: The Output Gate

In practice, the above steps are repeated multiple times. Should the input contain 20 data points the process will be repeated 20 times; once for each time step.

LTSMs are extremely effective at capturing long-term temporal dependencies without the drawback of traditional RNNs and have been applied to advance the state-of-the-art in several challenging tasks. For example, language translation and modelling, analysis of audio and video data, handwriting recognition, and many more [139].

Bi-Directional LTSMs

The performance of an LTSM network can, in some cases, be improved with the addition of a bi-directional layer. The input is fed to a standard LTSM and the second takes a reversed copy. The concept is to divide the neurons so one LTSM layer is responsible for extracting positive temporal information (the forward cell states) and one is responsible for the negative time direction (the backward cell states) [140].

In a bi-directional LTSM network, input sequence time steps are processed sequentially, in both directions concurrently, as illustrated in Figure 3.40.

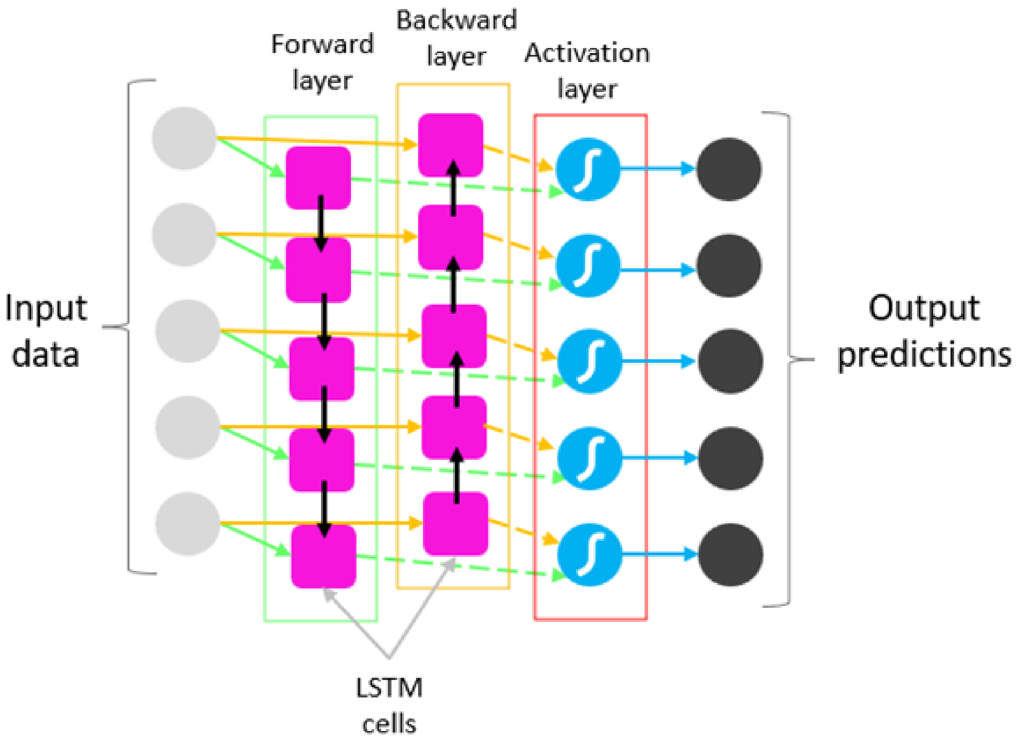


Figure 3.40: A high-level overview of bi-directional LTSM layers

3.5.4 Overview of the Gated Recurrent Unit (GRU)

The GRU is a newer iteration of the LTSM cell. Both are similar, with the key exception of the current cell state. GRUs contain two gates: reset and update. Meaning long-term memory is transferred via the hidden state. GRUs offer an advantage over LTSMs as they are simpler, meaning they can be quicker. A high-level illustration of a GRU cell can be found in Figure 3.41 and a breakdown of the two gates is included below.

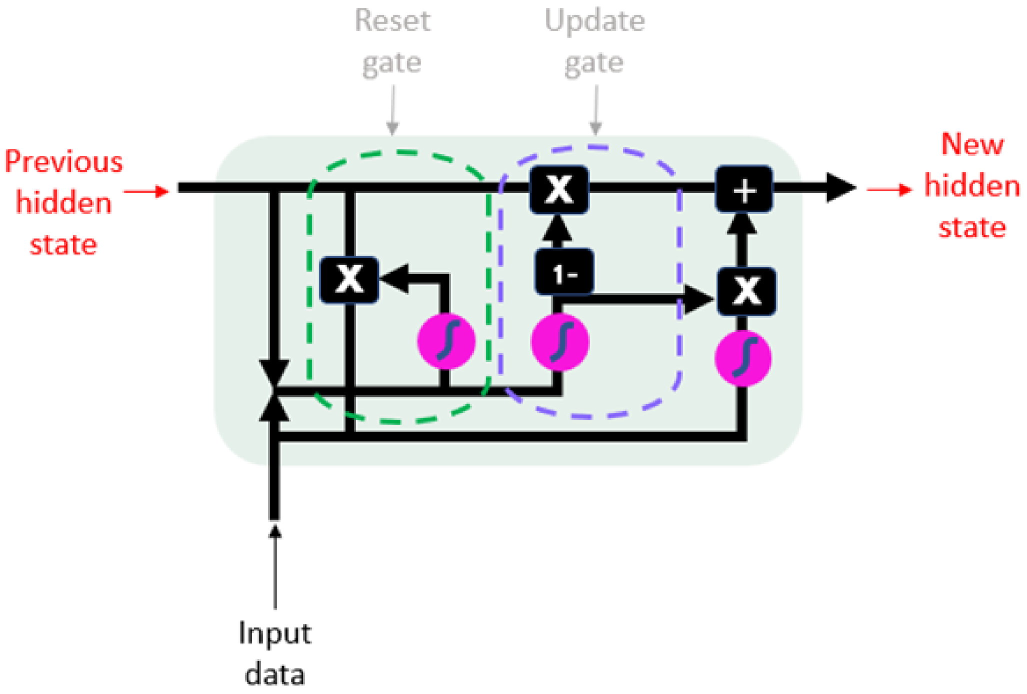


Figure 3.41: A high-level overview of the GRU cell

Step 1: The Reset Gate

The reset gate is responsible for the hidden state of the network (also referred to as the memory component) and determines how much previously learned information about the data sequence should be retained.

Step 2: The Update Gate

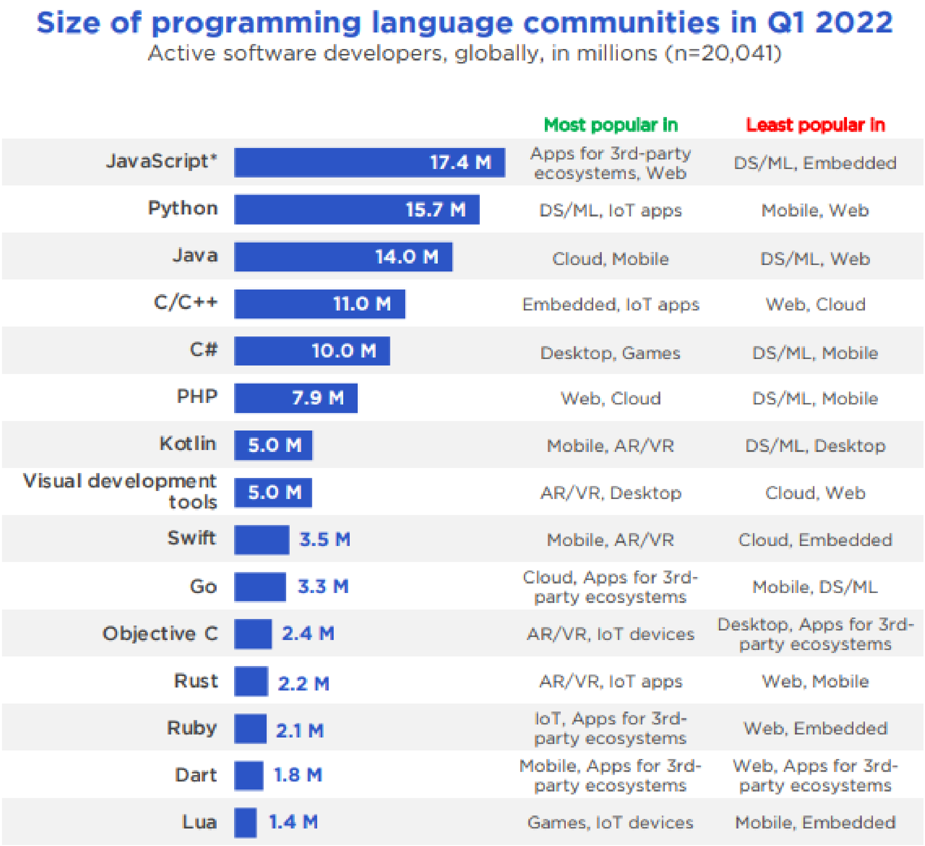
The update gate acts in a similar fashion to the forget gate in the LTSM cell and is a filter, deciding which information from the previous hidden state and the current time step input should be retained. The output vector elements are squeezed between the range of [0, 1] with the aid of the Sigmoid activation function.

Commonly, developers will run experiments using both LTSM and GRU layers, separately, to establish which delivers the best results for their use case.

3.6 Overview of Programming Languages and Frameworks

3.6.1 Programming Languages

A raft of programming languages exist. While there is no clear “best” language of choice, Figure 3.42 summarises the popularity of the top fifteen programming languages in 2022.



Size of programming language communities in Q1 2022

Active software developers, globally, in millions (n=20,041)

		Most popular in	Least popular in
JavaScript*	17.4 M	Apps for 3rd-party ecosystems, Web	DS/ML, Embedded
Python	15.7 M	DS/ML, IoT apps	Mobile, Web
Java	14.0 M	Cloud, Mobile	DS/ML, Web
C/C++	11.0 M	Embedded, IoT apps	Web, Cloud
C#	10.0 M	Desktop, Games	DS/ML, Mobile
PHP	7.9 M	Web, Cloud	DS/ML, Mobile
Kotlin	5.0 M	Mobile, AR/VR	DS/ML, Desktop
Visual development tools	5.0 M	AR/VR, Desktop	Cloud, Web
Swift	3.5 M	Mobile, AR/VR	Cloud, Embedded
Go	3.3 M	Cloud, Apps for 3rd-party ecosystems	Mobile, DS/ML
Objective C	2.4 M	AR/VR, IoT devices	Desktop, Apps for 3rd-party ecosystems
Rust	2.2 M	AR/VR, IoT apps	Web, Mobile
Ruby	2.1 M	IoT, Apps for 3rd-party ecosystems	Web, Embedded
Dart	1.8 M	Mobile, Apps for 3rd-party ecosystems	Web, Apps for 3rd-party ecosystems
Lua	1.4 M	Games, IoT devices	Mobile, Embedded

Figure 3.42: Summary of the most popular programming languages in 2022 [141]

A recent study has shown that in 2022 70% of Machine Learning developers use Python, whereas 17% use R (another data science language) [141].

3.6.2 Statistical and Image Processing Packages

The programming language used throughout this project for cleaning and preparing datasets, network construction and training, inference of trained models and statistical analysis of results is Python 3 [142].

As previously stated, Python is preferred by 70% of Machine Learning developers for a multitude of reasons. Predominantly because Python includes core packages and modules for data pre- and post-processing and statistical analysis.

Such packages used in this study include:

- **Numpy**: a Python library used for working with arrays
- **Scikit-learn**: a Python library providing access to many unsupervised and supervised learning algorithms
- **OpenCV**: a Python library used for computer vision
- **Pillow**: one of the important modules for image processing in Python
- **Pandas**: an open-source data analysis and manipulation library
- **Matplotlib**: a comprehensive library for creating static, animated, and interactive visualisations in Python

3.6.3 Deep Learning Frameworks

As with programming languages, there are multiple frameworks to choose from when undertaking a Deep Learning project. As Python was the chosen language for this PhD project, a description of some of the available deep learning frameworks follows.

Tensorflow: developed by the Google Brain team, Tensorflow is an open-source machine learning platform designed to make constructing, training and testing neural networks as user-friendly as possible. Tensorflow was initially used internally, however Google released the library under Apache License 2.0 in November 2015. Tensorflow uses Python or JavaScript to provide a convenient front-end API for building applications, while executing those applications in high-performance C++ [143].

Keras: is also an open-source neural network library written in Python that can run on top of TensorFlow. Designed by François Chollet (a Google engineer), Keras is designed to enable fast experimentation with neural networks and is considered very user-friendly. In 2017, Google announced it would support Keras in it's core library. With the release of TensorFlow 2.0, the Keras API became integrated; making it easier to create neural networks in conjunction with the Tensorflow framework [144].

Pytorch: is also an open-source neural network library, released in 2016, and maintained by Facebook's AI Research Lab (FAIR). FAIR built PyTorch on top of the Torch library, another open-source machine learning library, a scientific computing framework, and a scripting language based on Lua. Since PyTorch is developed by Facebook and offers an easy-to-use interface, its popularity has gained momentum in recent years, particularly in academia. PyTorch is the main competitor of TensorFlow [145].

TensorFlow has maintained its popularity within the deep learning developer community due to it's increased support, performance and online documentation. TensorFlow 2.0 has introduced remedies to the shortcomings of TensorFlow 1.x. and, along with the Keras implementation, is the framework of choice for this project.

3.7 Conclusion

Recent advancements in AI and DL have paved the way for several cutting-edge CV tasks, each lending themselves well to automated medical image interpretation. Throughout this chapter, state-of-the-art architectures are discussed, including the history of CNNs, segmentation and object detection networks and RNNs for sequence analysis. A thorough description of network parameters, hyper-parameters and regularisation techniques for optimisation are also included. Furthermore, an analysis of current trends in programming languages and DL frameworks, libraries and packages is provided.

Chapter 4

Echocardiographic View Classification

4.1 Introduction

An echocardiogram (echo) is the modality of choice for assessing several cardiovascular diseases (CVDs). Throughout the examination, it is standard practice to capture videos, still images and Doppler recordings from several acquisition angles to observe the complex cardiac anatomy. This is achieved by placing the Doppler probe in different positions on the patient's chest. The standard views are parasternal (right and left), apical, subcostal, and suprasternal, as illustrated in Figure 4.1.

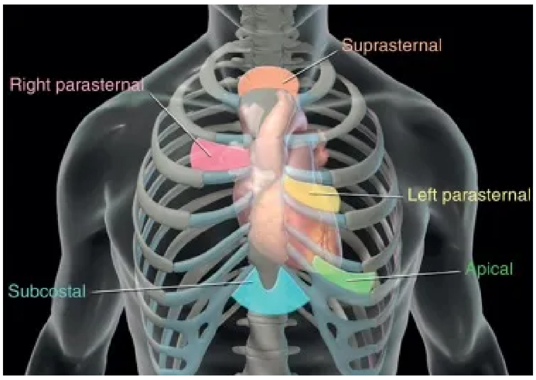


Figure 4.1: An illustration of the standard views: parasternal (right and left), apical, subcostal, and suprasternal [146]

Each standard view can be considered a class, with the angle and rotation of the Doppler probe determining the view plane; equivalent to sub-classes of the main views. Each view and associated imaging plane, as demonstrated in Figure 4.2, provides a unique vantage point and, when analysed in combination, give a complete overview of the cardiac structure [146].

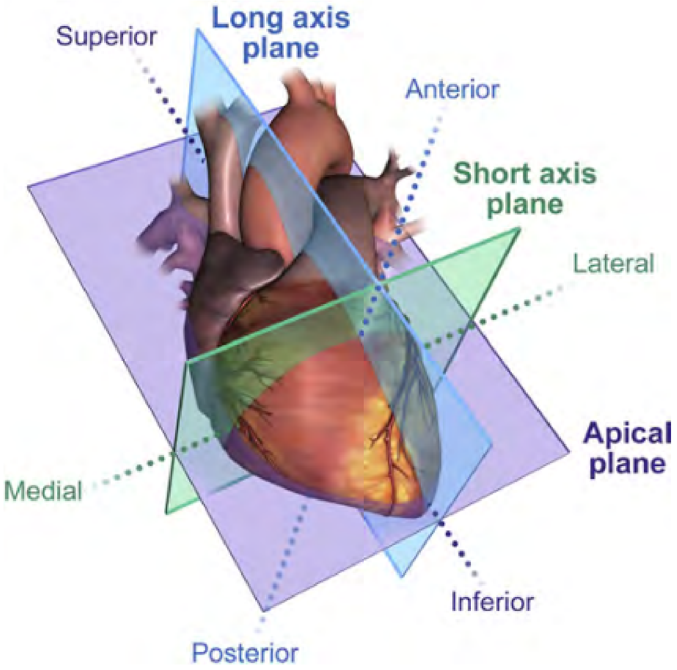


Figure 4.2: An example of cardiac imaging planes resulting from the angle and rotation of the Doppler probe at each standard position: the long-axis, the short-axis, and 4-chamber plane [52]

More detailed information about CVD, the transthoracic echocardiogram (TTE) examination and different image views can be found in Chapter 2, Clinical Background.

Routine clinical echo analysis remains, for the most part, human-led. Figure 4.3 represents a high-level overview of the typical 3-step analysis and interpretation procedure. Beginning with the TTE examination (image acquisition) , followed by manual image analysis and annotation and, finally, interpretation and measurements. The final step also includes calculating vital parameters to determine if a patient is suffering from CVD.

Whilst steps 1 and 3 remain manual, artificial intelligence (AI) and deep learning (DL) networks have shown great promise for future clinical adoption due to rapid, reliable and reproductive automation of several tasks in step 2 [16, 23–25].

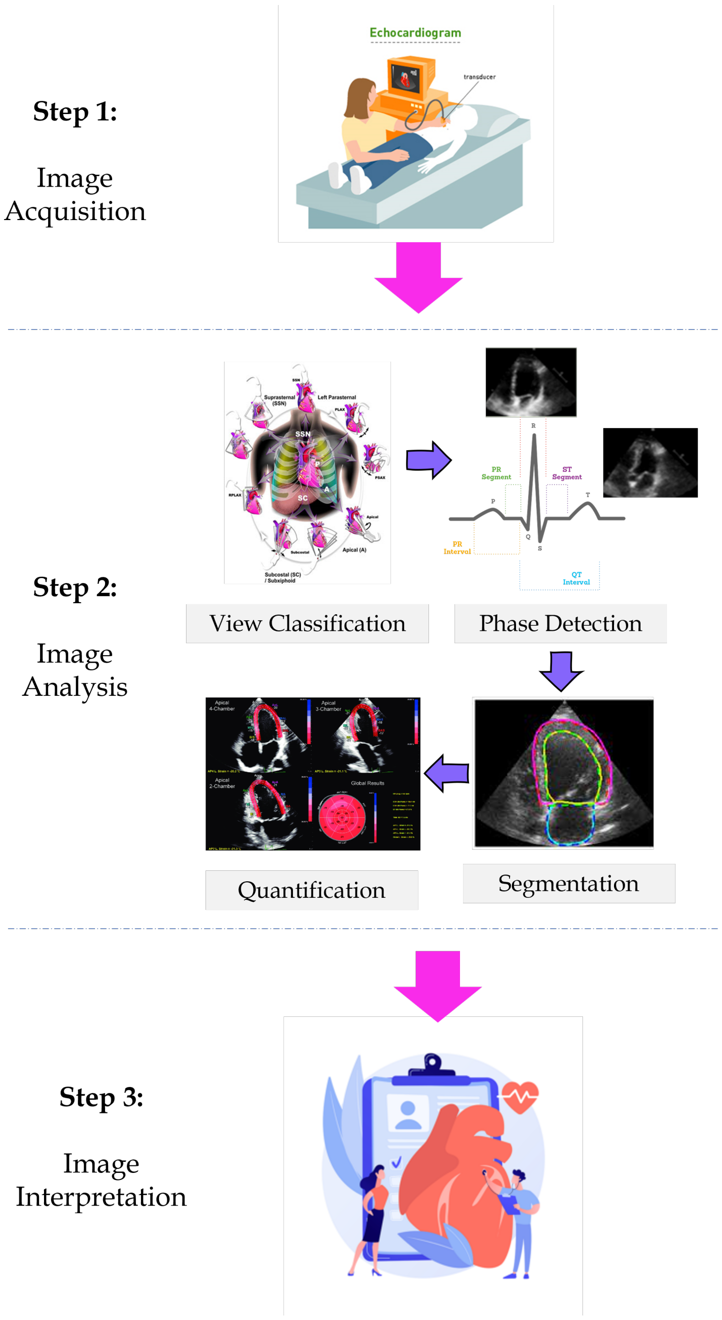


Figure 4.3: A high-level overview of the three steps included a typical, human-led TTE acquisition and interpretation clinical pipeline

Crucially, the first task in the interpretation of echo images and videos is view classification; a manual process performed by trained operators. This process is time-consuming and prone to error due to inter- and intra-observer variability and very subtle differences between classes and sub-classes of view types, rendering the task particularly challenging to the human eye [68].

Ordinarily, examination data from different imaging modalities is stored in a PACS database. Software exists to partially automate the analysis and diagnosis steps, such as EchoPAC (GE Healthcare) and QLAB (Philips) [16], however these applications still rely upon varying degrees of human interaction.

Recent advancements in ML [147–150] and DL [10–16] algorithms, have enabled significant gains to be made in automating human tasks, such as video and image processing. Accurate, rapid and reliable classification of standard echo views by trained DL networks has great potential in the modern cardiovascular clinic. Such an innovation could improve workflow, help guide trainees and inexperienced users, improve accuracy and speed up diagnoses.

4.2 Related Work

Echo view classification is a requisite process during the interpretation of post-acquisition videos and still images. Previous studies apply traditional machine learning (ML) techniques for task automation, however these approaches still require time-consuming, laborious, manual annotations and are limited to a small number of views. Furthermore, such algorithms require images to be of high-quality and do not generalise well when tested on images from multiple vendors. These limitations deem traditional ML approaches unsuitable for application in routine clinical practice [151].

DL algorithms, particularly CNNs, have proven successful when learning high and low-level features from both the spatial and temporal image domains. Consequently, CNNs are applied to analyse multifarious medical image modalities with varying degrees of success [152]. Early studies attempt echocardiographic (echo) view classification with Markov Random Field networks to spatially locate each heart chamber, building a relational graph that was later verified by using a Support Vector Machine (SVM). This method was not without it's limitations, however. It was particularly susceptible to image noise and transformations, hindering generalisation and the feasibility of clinical deployment [153]. Comparable studies apply several ML architectures, though all suffer from distinct limitations; e.g. necessitating human intervention, a small fraction of the overall number of echo views, and low accuracy [147–150].

DL has been applied to chamber segmentation [25, 154, 155] along with U-Net style CNNs [12, 30] and transfer learning with well-established CNN architectures [13–15]. However, the accuracy has been unpredictable as all experienced varying degrees of difficulty in differentiating between similar sub-class views, such as apical two chamber (A2C) and apical three chamber (A3C). Typically, previous studies have extracted a sample of singular frames from video sequences, classified them and taken the consensus as the class prediction. Thus, excluding information about crucial temporal relationships between frames. Alternative approaches segment the heart chambers to ascertain the view [156–159]. More recent studies adopt DL techniques [10], demonstrating improved performance in both accuracy and performance over traditional ML methods.

A previous study performed by our research group applied Neural Architecture Search (NAS) to find optimal CNN 'cells' capable of accurately classifying videos into 14 classes. NAS, such as Google Brain's Efficient Neural Architecture Search (ENAS) [160] and Differentiable Architecture Search (DARTS) [161] have demonstrated their capabilities in streamlining popular CNN architectures that frequently repeat the same sequentially stacked convolutional building blocks. By focussing attention on small, optimal computational cells, NAS finds a notably reduced search space, thus recommending cells with less layers than entire network architectures. Azarmehr, et. al. demonstrate impressive echo view accuracy when extracting frames from video sequences with DARTS cells containing significantly fewer parameters than well-established CNN architectures commonly utilised for transfer learning tasks [16]. However, this study was limited in the fact it extracted single frames from echo video sequences (not utilising the temporal dimension) and did not include Doppler recordings.

Supervised learning techniques, including occlusion testing and saliency mapping, to combine spatial and temporal information have outperformed CNNs where, typically, only spatial information is analysed [11, 151]. 3D CNNs have been compared with time-distributed networks showing the latter is more efficient when classifying 14 views from echocardiographic videos [68]. This study provided valuable information in the form of saliency maps, providing clues as to the temporal details CNNs consider important when performing video classification. In particular, findings suggested the classification decision was influenced by anatomical borders of major structures such as the pulmonary artery and valve leaflets and the left ventricle. Therefore, it is crucial the network is able to track the movement of anatomical structures when ascertaining important information for view classification tasks.

4.3 Main Contributions

This chapter investigates the feasibility of classifying a large number of TTE views from echo video sequences and still Doppler images using a CNN-RNN algorithm, leveraging both spatial and temporal information. The main contributions of this research can be summarised as:

- Classifying the largest number of echo views (20), when compared with recently published studies, including 15 videos and 5 Doppler images

- Proposing a CNN-RNN architecture to successfully classify sub-classes of standard TTE views, with each view considered a separate class

- Comparing the accuracy of four time-distributed state-of-the-art CNNs for feature extraction using, with the addition of LSTM and GRU layers, achieving 92.6% accuracy for Doppler image classification and 98.5% overall accuracy for videos (higher than any other previously published study)

- Investigates an efficient technique for addressing imbalanced video datasets and demonstrates the approach does not impact upon model accuracy

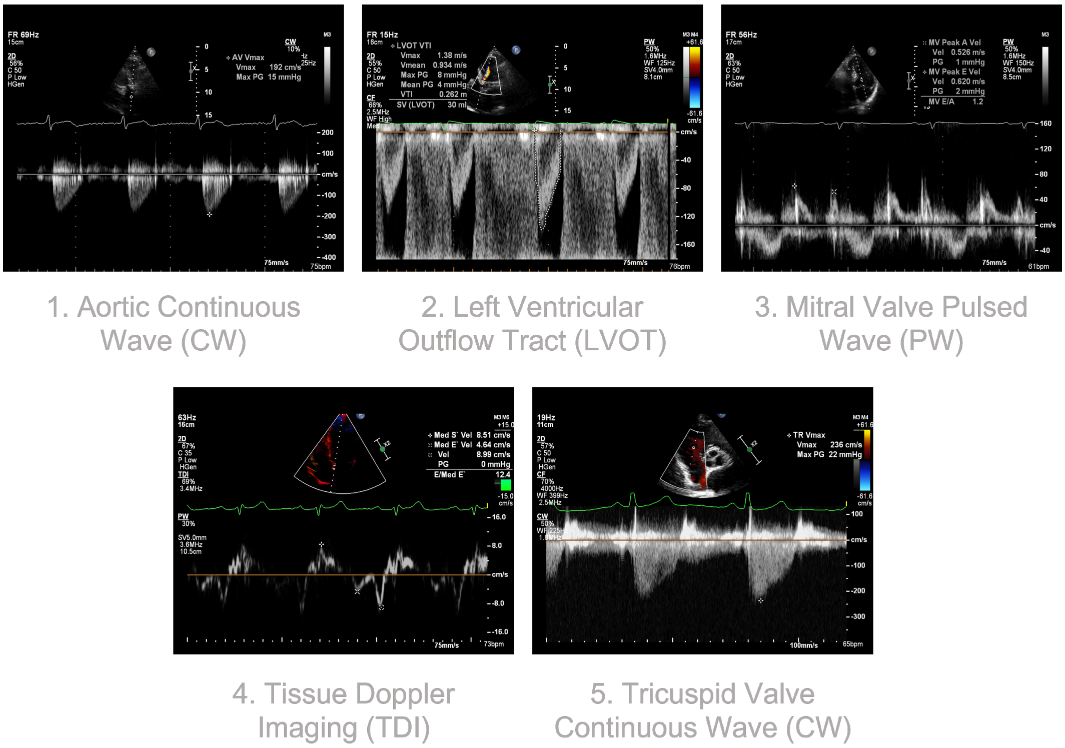
4.4 Methodology

4.4.1 Dataset, Ethics and Expert Annotations

The dataset used in this study was acquired by extracting a random sample of studies from Imperial College Healthcare NHS Trust's echocardiogram database, in Digital Imaging and Communications in Medicine (DICOM) format. Ethical approval was granted from the Health Regulatory Agency (HRA) (identifier 243023). Only full patient studies without intravenous contrast administration were included. All images and videos were fully anonymised to remove patient information and identifiers. Ground-truth annotations were provided by an expert human who classified each view into its corresponding class and sub-class, pertaining to the views providing anatomical information.

Howard et. al. [68] utilised the same echocardiographic video dataset in their study comparing several state-of-the-art CNNs (both 2D and 3D time-distributed networks). However, 14 views were included (the two subcostal views were combined). A second human expert classified their test dataset, blind to the annotations of the first, to test for inter-observer variability. Due to differences in data pre-processing, we cannot directly compare the same test set, however we can assess the accuracy of the trained networks against inter-observer variability.

This study classifies 20 echocardiographic views as separate classes; more than any in previously published work. Figure 4.4 illustrates the 5 still Doppler image views, whilst Figure 4.5 shows the 15 video classes and and sub-classes.



1. Aortic Continuous Wave (CW)

2. Left Ventricular Outflow Tract (LVOT)

3. Mitral Valve Pulsed Wave (PW)

4. Tissue Doppler Imaging (TDI)

5. Tricuspid Valve Continuous Wave (CW)

Figure 4.4: Echocardiographic Doppler image views and corresponding classes: 1. Aortic Continuous Wave (CW), 2. Left Ventricular Outflow Tract (LVOT), 3. Mitral Valve Pulsed Wave, 4. Tissue Doppler Imaging (TDI) and 6. Tricuspid Valve Continuous Wave (CW)

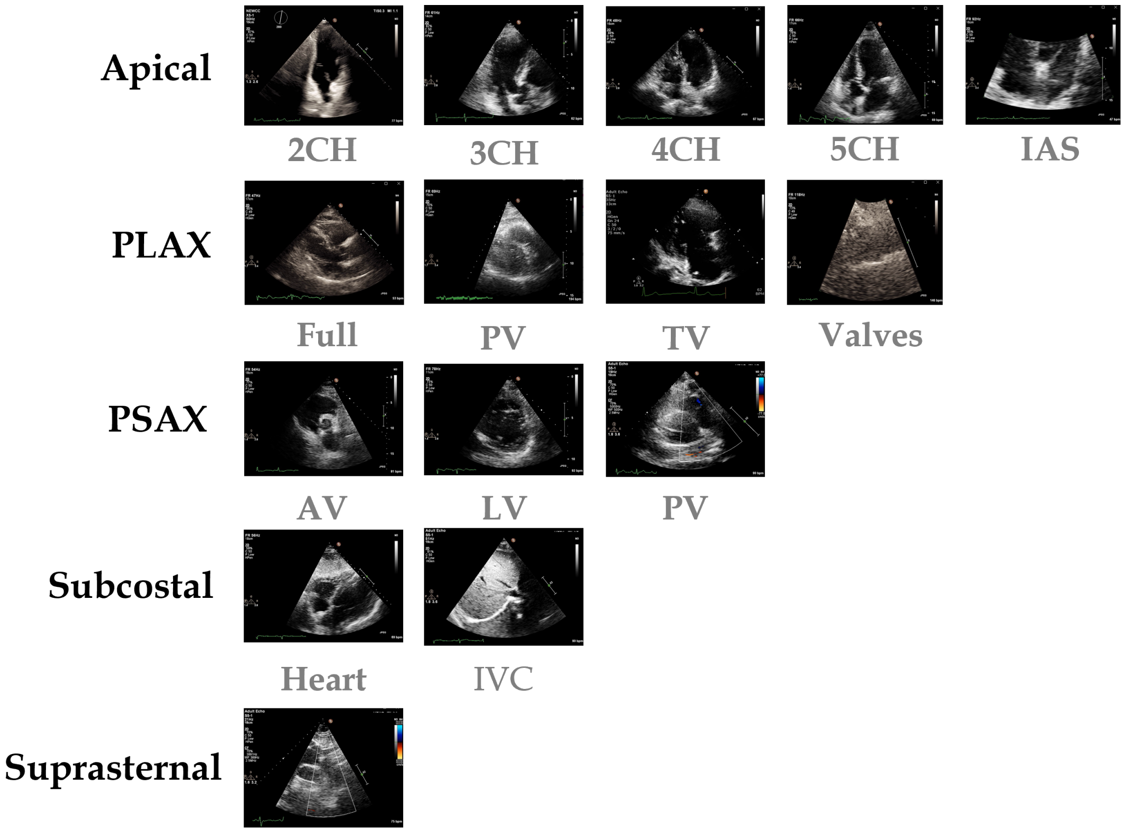


Figure 4.5: Echocardiographic video view classes and corresponding sub-classes:

Apical: 2-Chamber, 3-Chamber, 4-Chamber, 5-Chamber and inter-atrial septum (IAS)

Parasternal long axis (PLAX): Full, pulmonic valve (PV), tricuspid valve (TV) and valves

Parasternal short axis (PSAX): aortic valve (AV), left ventricle (LV) and pulmonic valve (PV)

Subcostal: heart and inferior vena cava (IVC)

Suprasternal

4.4.2 Data Pre-Processing

For still Doppler image view classification, 4,800 (the number of images in the smallest class) images for each class (24,000 images in total) were randomly split by 20%/20%/60% for test, validation and training datasets, respectively. Images were resized to 224 x 224 pixels using bilinear interpolation and 3 channels.

The 15 video classes were originally imbalanced, with the least represented class being PLAX PV, containing only 161 videos. The most highly represented class was PSAX AV with a total of 2,007 videos. When considering all videos from all classes, the total number of frames ranged from 2 to 601.

To ensure each class was balanced before splitting the data and training the network, all videos were pre-processed into chunks of 20 frames. During the process of video "chunking" a 20-frame wide window moved across the frames with differing strides: as low as a stride of 1 frame for each movement in under-represented classes, to a stride of 10 where a class contained a greater number of videos. Figure 4.6 provides a visual illustration of this process.

For each view, Table 4.1 contains information regarding the original number of videos, the stride used for chunking and the resulting number of 20-frame videos after the process was complete. If the final video chunk contained fewer than 20 frames, blank frames were appended.

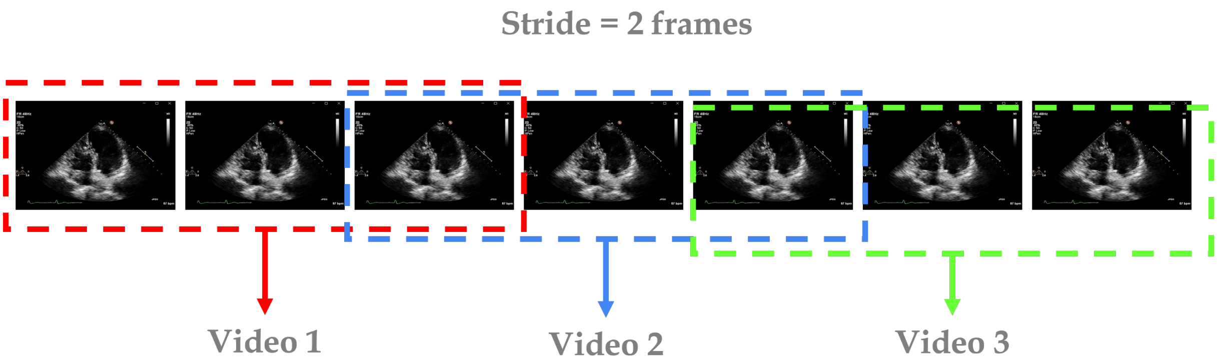


Figure 4.6: An example of how a window of fixed frame width (in this study this was 20 frames, for illustration purposed the width is 3 frames) moves across the original video with a fixed stride (number of frame steps) saving "chunked" videos

Table 4.1: A full list of echocardiographic video view classes with the original number of videos, the stride used for chunking into 20 frame clips and the resulting number of videos

View	Original # videos	Stride	Total # videos
A2CH	1,021	10	6,407
A3CH	1,120	10	6,196
A4CH	1,606	10	8,623
A5CH	808	5	7,140
IAS	461	4	6,791
PLAX Full	1,526	10	8,650
PLAX Valves	938	10	6,706
PLAX PV	161	1	7,041
PLAX TV	490	3	7,535
PSAX AV	2,007	10	12,011
PSAX LV	1,495	10	10,991
PSAX PV	269	1	10,109
Subcostal Heart	914	5	8,590
Subcostal IVC	357	2	8,177
Suprasternal	473	2	8,523

In all, 6,195 videos were randomly selected for each class from the pre-processed 20 frame chunks (a total of 1,858,500 images for 15 anatomical views). Frames were resized to 224 x 224 pixels with 3 channels. The data was randomly split by 20%/20%/60% for test, validation and training datasets, respectively. Each dataset contained frames from separate echo studies (i.e. different patients) to maintain sample independence.

To ensure comparability, the same images and videos were used for the data split for all experiments.

It is assumed the data pre-processing strategy aimed at balancing the video classes will improve the overall accuracy of the trained network. However, an experiment will also be conducted on the original, imbalanced dataset to test this hypothesis.

4.4.3 Neural Network Architecture

The entire pipeline for the neural network architecture is two-fold, as illustrated in Figure 4.7. First, the input type (whether still image or video) is determined and directed along the appropriate path. Doppler images are processed through an established, pre-trained CNN before features are extracted from the final fully-connected layer and fed to a dense layer with Softmax activation, predicting a probability for each of the five classes.

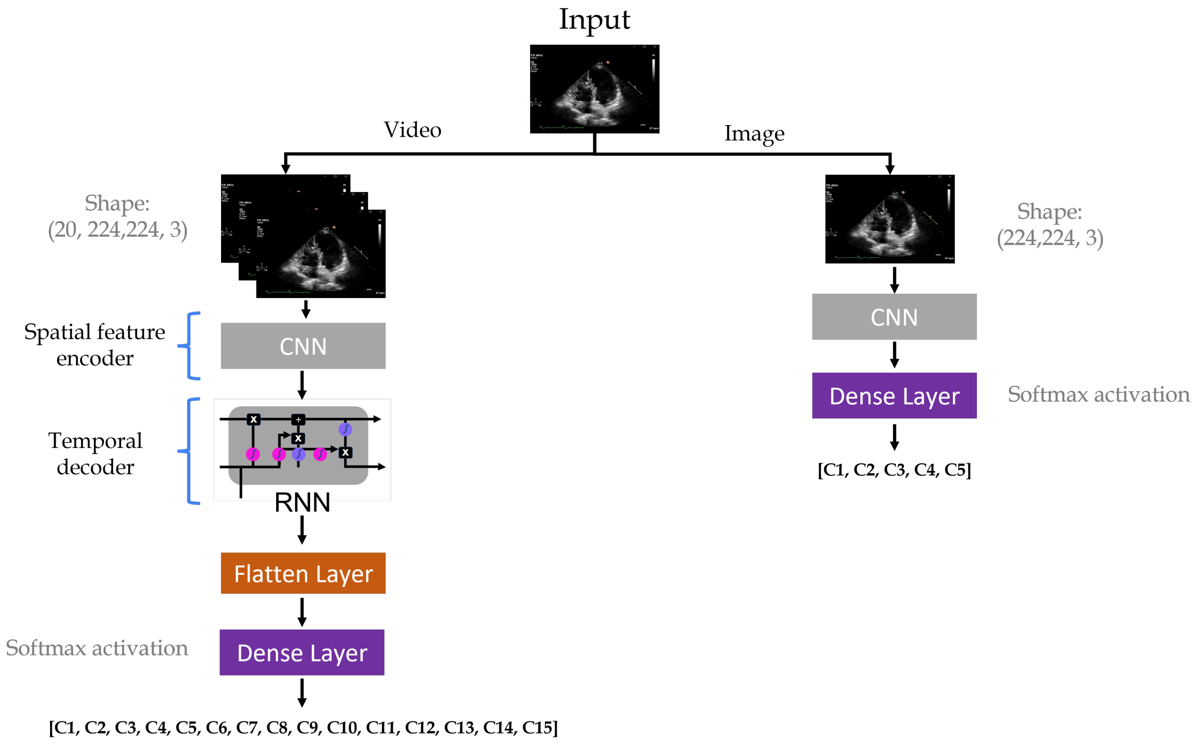


Figure 4.7: High-level illustration of the two-fold CNN/CNN-RNN neural network pipeline

Video sequences are processed by a time-distributed, pre-trained state-of-the-art CNN for the encoding of spatial features. The CNN utilises 2D convolutions, however the video clips have three dimensions. Therefore, the Keras "TimeDistributed" layer serves an important role in applying additional dimensions to the input sequence; allowing 2D convolutions to be applied to each time step (frames within the video sequence). Every input to a time-distributed layer should have a minimum of three dimensions. Consider the input for a video sequence of 20 frames, each with a dimension of 224 x 224 x 3 (with channels last format) and a batch size of

64, the input shape would be (64, 20, 224, 224, 3) relating to (batch size, time steps, height, width, channels).

After the spatial features for each time step in each video are encoded, the feature maps are fed to a temporal decoder consisting of one or more RNN layers (LSTM or GRU). The output of the RNN is flattened and processed by a dense layer with a Softmax activation to produce percentage probabilities for each of the 15 video classes.

For the extraction of spatial feature vectors from the still Doppler images and each frame in the video sequence, a series of state-of-the-art architectures were employed. These include ResNet50, InceptionV3, DenseNet201 and VGG19. All layers in the established CNNs were set to non-trainable, except the additional dense layer which was trained. Details of each architecture can be found in the relevant resources [122, 124, 125, 162] and in section 3.3 of Chapter 3 (Technical Background).

4.4.4 Implementation Details

The models were implemented using Tensorflow 2.0 [143] and Keras [144] DL frameworks. Training and inference was conducted on a server containing four NVIDIA GeForce RTX 3090 GPUs.

To prevent overfitting, random, on-the-fly augmentation was applied with a rotation of between $-10°$ and $10°$ and spatial cropping between 0% and 0.1% pixels along each axis. Early stopping, monitoring validation loss, with a patience of between 10 and 50, was also implemented. For both networks, the optimiser was Adam [163] with a learning rate of $1e-5$. The loss function for both networks was sparse categorical crossentropy.

The still Doppler image classification network was trained using a batch size of 128 and conducted over a minimum of 245 epochs with the maximum being 831 (depending on the CNN architecture).

The video classification CNN-RNN network was trained using a batch size of 4 and training was conducted over a minimum of 11 and maximum of 27 epochs.

Each pre-trained CNN (ResNet50, InceptionV3, DenseNet201 or VGG19) was loaded with ImageNet weights and average pooling. For the RNN component, experiments were conducted using 1xLSTM, 2xLSTM, 1xGRU and 2xGRU.

4.4.5 Evaluation metrics

Each network was evaluated using accuracy, precision, recall and F1 score. The accuracy was calculated as the fraction of correctly classified samples from the overall number of samples. Scikit Learn model evaluation methods [164] were used.

Inference time for both trained networks was estimated by running predictions on the GPU using the same 10 images/videos and calculating the average.

4.5 Results and Discussion

The entire classification pipeline comprises two neural network architectures: one for classifying still Doppler images into 5 views and the other for classifying videos into 15 views. Due to the distinct difference in the network architectures and datasets, the following results and discussion will be separate for each of the networks to allow for independent analysis and discussion.

4.5.1 Doppler Image Classification

Four well established, state-of-the-art CNNs (ResNet50, InceptionV3, DenseNet201 and VGG19) were used to extract spatial feature vectors from Doppler images with the addition of a dense layer and Softmax activation function to produce a prediction for each of the five classes.

Of the 24,000 images in the original dataset, 4,800 were retained at random as an unseen test set, representing 960 images from each class. As with the training and validation sets, test images were resized to 224 x 224 pixels with 3 channels.

Evaluation metrics for the four separate networks can be seen in Table 4.2. The results show the most accurate architecture is ResNet50. Accuracy, precision, recall and F1 score were all 92.6%, however, when measuring inference time ResNet50 took the longest to process predictions at

49ms, whereas the quickest was DenseNet201 (though yielding the lowest accuracy). One possible explanation could be the reduction in parameters, with ResNet50 containing 23,587,712 and DenseNet201 having 18,321,984. Due to the nature of the classification task and similarities between the Doppler image views, it can be assumed greater learnable parameters was advantageous to the task, despite taking slightly longer to produce predictions.

Table 4.2: Percentage accuracy, precision, recall and F1 Score for each of the state-of-the-art CNNs used for still image Doppler view classification with corresponding inference time (in milliseconds)

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Inference Time	Parameters
ResNet50	**92.6**	**92.6**	**92.6**	**92.6**	49ms	**23,587,712**
InceptionV3	78.6	78.6	78.6	78.4	31ms	21,802,784
VGG19	85.9	86.9	85.9	85.6	38ms	20,024,384
DenseNet201	64.9	74.9	64.9	61.7	29ms	18,321,984

ResNet50 is assumed the optimal model architecture for Doppler view classification. The results are visualised in the form of a confusion matrix in Figure 4.8.

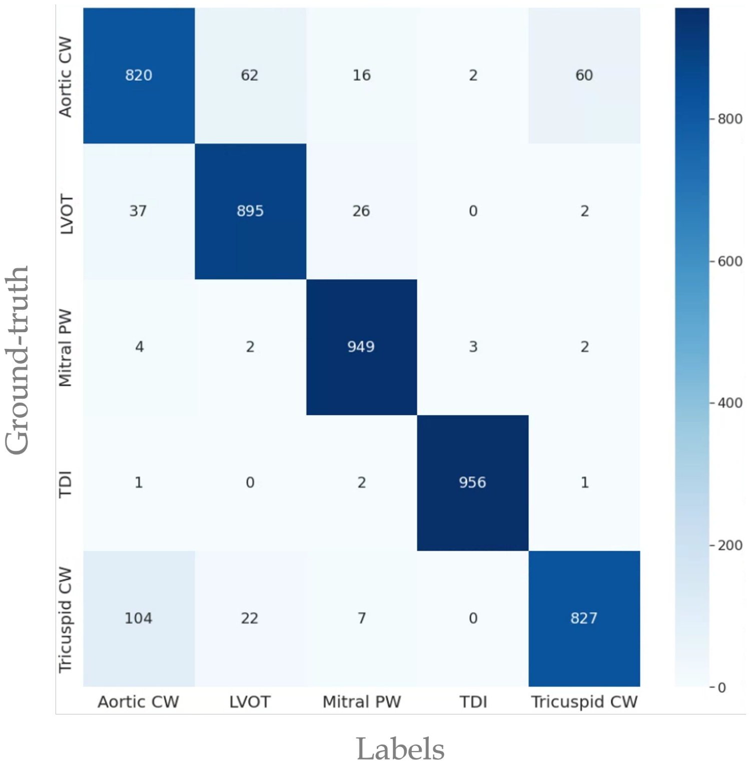


Figure 4.8: Confusion matrix for ResNet50 showing the **total number** of test set images classified correctly and misclassified

Upon inspection, it is clear the majority of misclassifications are Tricuspid CW wrongly predicted as Aortic CW (11%) and in reverse, Aortic CW predicted as Tricuspid CW (7%). This observation is further reinforced with the breakdown of classification results in Table 4.3.

Table 4.3: A breakdown of evaluation metrics for each class using ResNet50 for spatial feature vector extraction

Class	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Aortic CW	85	85	85	85
LVOT	93	91	93	92
Mitral Valve PW	99	95	99	97
TDI	99	99	100	100
Tricuspid Valve CW	86	93	86	89

Figure 4.9 shows two typical examples of aortic CW (on the left) with two tricuspid vale CW

images (to the right). It is easy to observe the similarities between the two views.

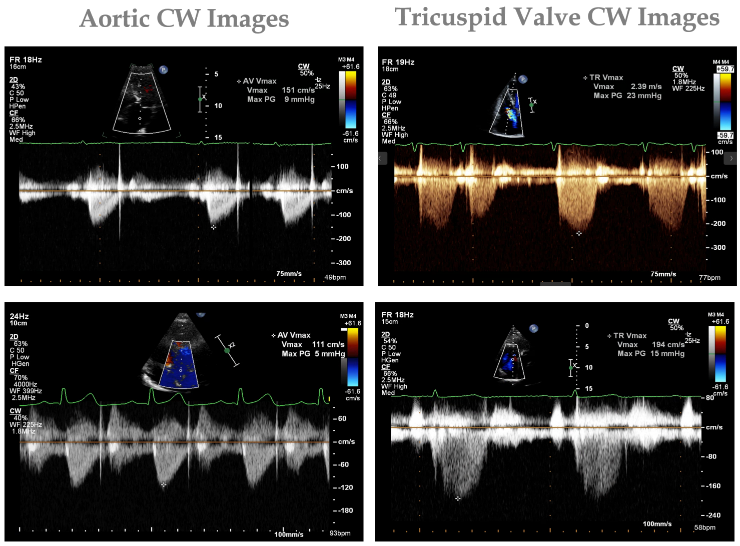


Figure 4.9: Examples of Aortic CW and Tricuspid Valve CW images

Figure 4.10 highlights two examples of aortic CW images misclassified as tricuspid valve CW images on the left, and vice versa on the right.

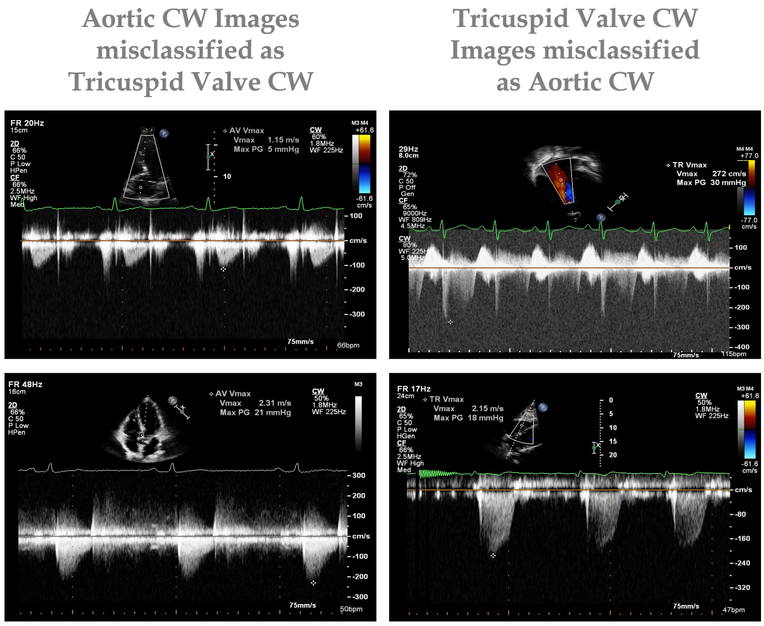


Figure 4.10: Examples of **misclassified** Aortic CW and Tricuspid Valve CW images

4.5.2 Video Classification

To leverage both spatial and temporal information retained within the videos, a CNN-RNN architecture was developed. The CNN component acts as a spatial feature encoder, subsequently feeding the extracted feature vector to an RNN to regress predictions for each of the 15 video view classes, taking into consideration important data regarding the relationship between sequential frames and anatomical structure changes.

For the CNN, the same four well established, state-of-the-art networks (ResNet50, InceptionV3, DenseNet201 and VGG19) were employed. One and two stacked LSTM and GRU layers were used to assess the effect upon test set accuracy.

Of the 92,925 videos in the entire dataset (6,195 videos from each class), 18,585 were randomly selected and set aside as an unseen test set. The test set constitutes a total of 371,700 images (20 frames per video). Video frames were resized to uniform dimensions of 224 x 224 x 3.

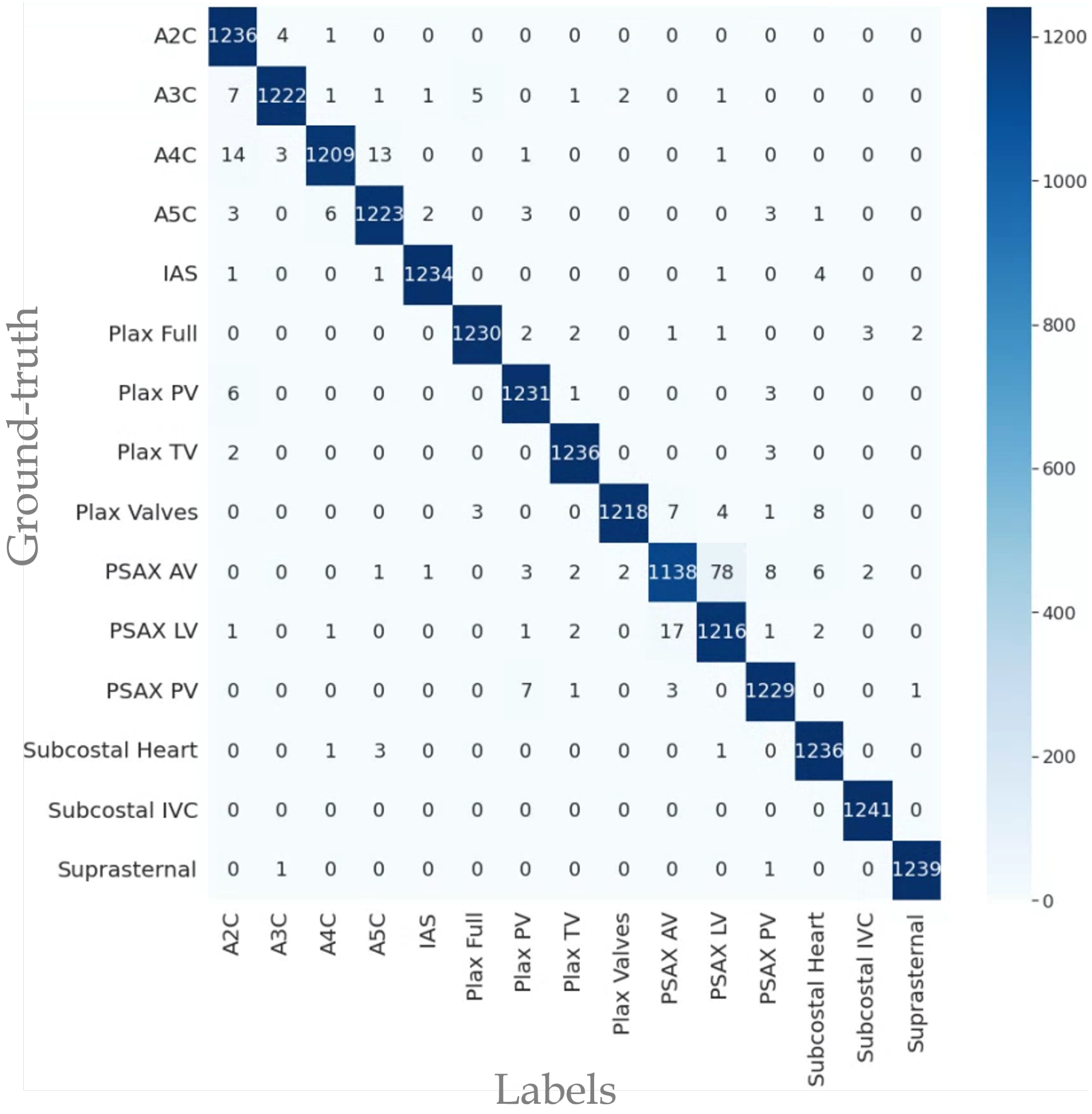
Table 4.4 highlights the results for each of the architectures with one LSTM layer. All CNN feature extractors achieve similar accuracy. However, interestingly, DenseNet201 generates the best results; in contrast with the image classification results, above.

Table 4.4 details video classification results for 1xLSTM layer.

Table 4.4: Results for state-of-the-art CNNs for spatial feature encoding with the addition of 1xLSTM layer

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Inference Time
ResNet50 + 1x-LSTM	97.7	97.7	97.8	97.8	46ms
InceptionV3 + 1x-LSTM	98.3	98.3	98.3	98.3	44ms
VGG19 + 1x-LSTM	97.5	97.6	97.5	97.5	64ms
DenseNet201 + 1x-LSTM	**98.5**	**98.5**	**98.5**	**98.5**	**58ms**

Assuming Densenet201+1x-LSTM as the optimal architecture, the confusion matrix in Figure 4.10 provides a breakdown of the misclassified videos. The most difficult for the model to differentiate between were PSAX AV and PSAX PV. This stands to reason as the views are distinctly similar, as highlighted in Figure 4.12.



	A2C	A3C	A4C	A5C	IAS	Plax Full	Plax PV	Plax TV	Plax Valves	PSAX AV	PSAX LV	PSAX PV	Subcostal Heart	Subcostal IVC	Suprasternal
A2C	1236	4	1	0	0	0	0	0	0	0	0	0	0	0	0
A3C	7	1222	1	1	1	5	0	1	2	0	1	0	0	0	0
A4C	14	3	1209	13	0	0	1	0	0	0	1	0	0	0	0
A5C	3	0	6	1223	2	0	3	0	0	0	0	3	1	0	0
IAS	1	0	0	1	1234	0	0	0	0	0	1	0	4	0	0
Plax Full	0	0	0	0	0	1230	2	2	0	1	1	0	0	3	2
Plax PV	6	0	0	0	0	0	1231	1	0	0	0	3	0	0	0
Plax TV	2	0	0	0	0	0	0	1236	0	0	0	3	0	0	0
Plax Valves	0	0	0	0	0	3	0	0	1218	7	4	1	8	0	0
PSAX AV	0	0	0	1	1	0	3	2	2	1138	78	8	6	2	0
PSAX LV	1	0	1	0	0	0	1	2	0	17	1216	1	2	0	0
PSAX PV	0	0	0	0	0	0	7	1	0	3	0	1229	0	0	1
Subcostal Heart	0	0	1	3	0	0	0	0	0	0	1	0	1236	0	0
Subcostal IVC	0	0	0	0	0	0	0	0	0	0	0	0	0	1241	0
Suprasternal	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1239

Figure 4.11: Confusion matrix for DenseNet201 + 1x-LSTM showing the **total number** of test set images classified and misclassified

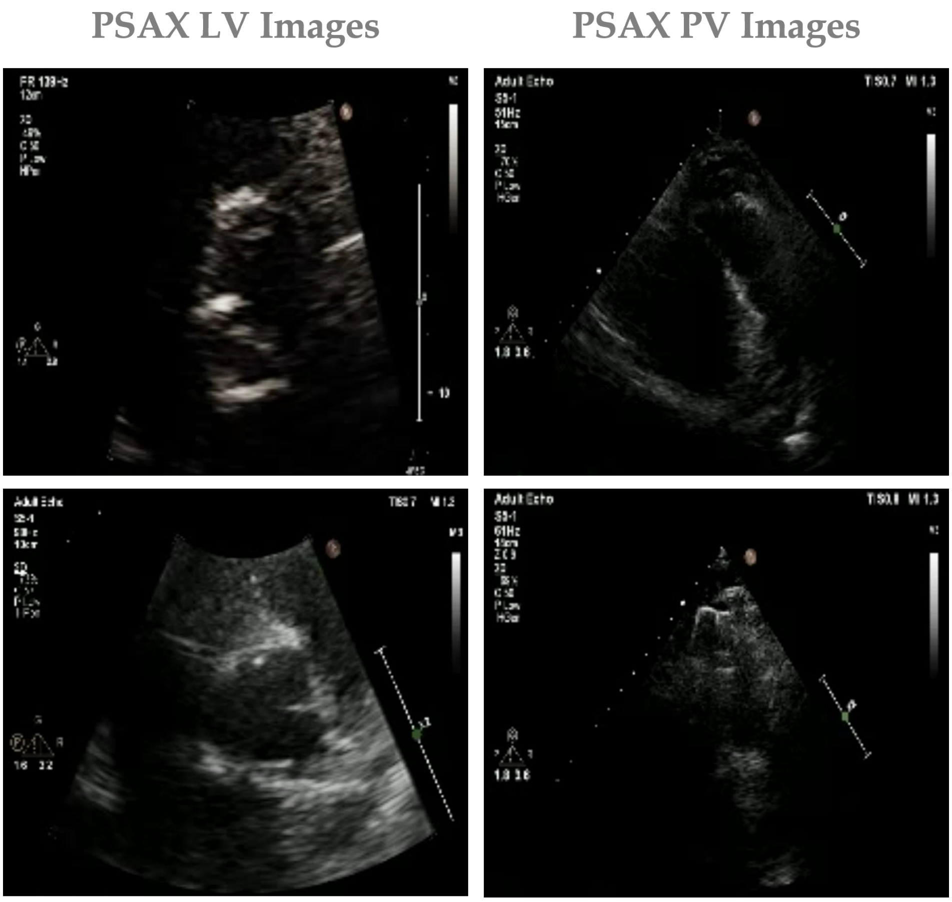


Figure 4.12: Examples of PSAX LV images (left) and PSAX PV images (right)

Table 4.8 shows the evaluation metrics for a breakdown of each video class using the DenseNet201 + 1x-LSTM model.

Table 4.5: Class breakdown using DenseNet201

Class	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
A2CH	100	97	100	98
A3CH	98	99	98	99
A4CH	97	99	97	99
A5CH	99	98	99	99
IAS	99	100	99	100
PLAX Full	99	99	99	99
PLAX PV	99	99	99	99
PLAX TV	100	99	100	99
PLAX Valves	98	100	98	99
PSAX AV	91	98	92	95
PSAX LV	98	93	98	96
PSAX PV	99	98	99	99
Subcostal Heart	100	98	100	99
Subcostal IVC	100	100	100	100
Suprasternal	100	100	100	100

In their study, Howard et. al. [68] report the greatest accuracy for their time-distributed CNN network when compared with standard classical state-of-the-art 2D CNNs and a 3D convolutional neural network. Their test set comprised 2,140 videos, when comparing intra-observer variability, 74 videos (3.5%) were misclassified by the two human experts. The most challenging distinctions were between A5C and A4C, and A3c and A2c. The time-distributed network achieved 96.1% accuracy (3.9% error), while the second expert agreed with the first in 96.4% of cases (3.6% error).

Comparing this study with Howard et. al., the addition of an RNN module to the proposed time-distributed CNN leverages 98.5% accuracy, an improvement of 2.4% against their automated network and 2.1% against the inter-observer variability of two expert humans.

For experimental purposes, Table 4.7 highlights the addition of a second LSTM layer did not significantly improve the results.

Table 4.6: Results for video classification with the same CNN architectures, but with two stacked LSTM layers

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Inference Time
ResNet50 + 2x-LSTM	97.5	97.5	97.5	97.4	46ms
InceptionV3 + 2x-LSTM	**98.5**	**98.5**	**98.5**	**98.4**	**43ms**
VGG19 + 2x-LSTM	98.3	98.3	98.3	98.3	66ms
DenseNet201 + 2x-LSTM	98.3	98.3	98.3	98.3	60ms

For comparison purposes, the time-distributed DenseNet201 CNN was paired with one and two GRU layers, the results can be seen in Table 4.7. There were no discernible gains in accuracy (or other evaluation metrics) or inference time. More detailed descriptions of the function and operation of LSTM and GRU layers can be found in section 3.5, Recurrent Neural Networks (RNNs).

Table 4.7: Results for video classification with one and two stacked GRU layers

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Inference Time
DenseNet201 + 1x-GRU	98.5	98.5	98.5	98.5	60ms
DenseNet201 + 2x-GRU	97.8	97.8	97.8	97.8	61ms

As previously explained, the video classification dataset was originally imbalanced, as illustrated in Figure 4.13.

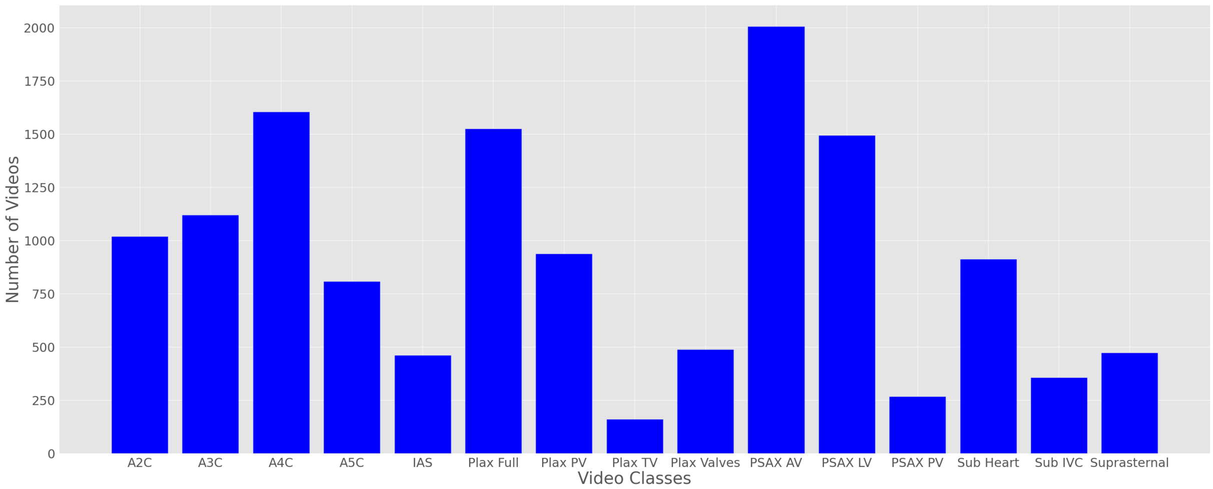


Figure 4.13: A bar chart detailing each class and corresponding number of videos contained within the original dataset

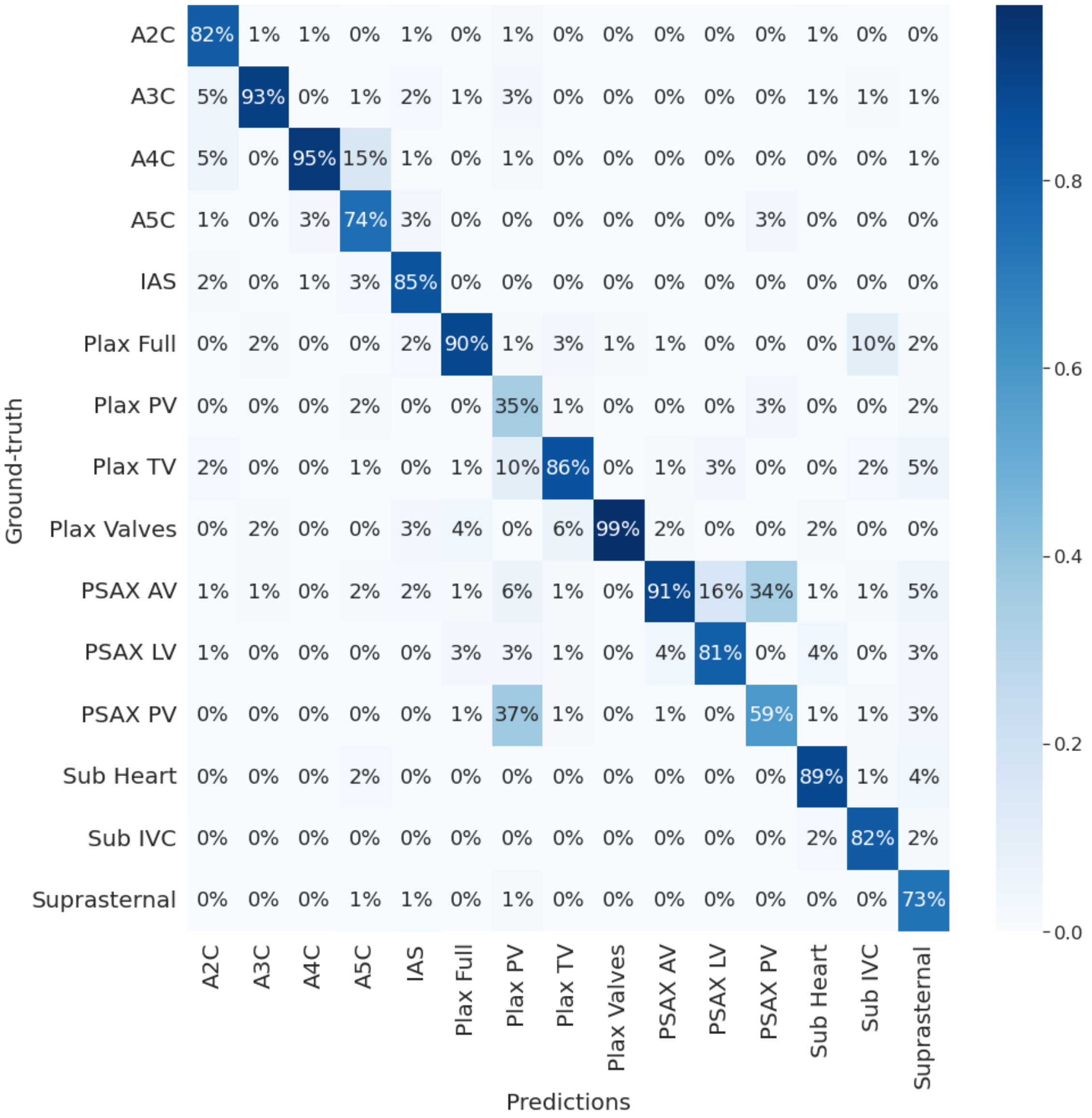
It is hypothesised the method of pre-processing, and balancing, the video classes would not worsen the overall accuracy of the trained network. To test this assumption, an experiment was conducted using 20 frames from each of the original, imbalanced, dataset to train the optimal network architecture (Densenet201+1x-LSTM), retaining the original 60%/20%/20% data split.

As illustrated in Table 4.8, and reinforced in the confusion matrix at Figure 4.14, a correlation between underrepresented classes and a significant reduction in accuracy is clear.

PLAX TV and PSAX PV constitute the least represented classes, achieving 61% and 31% accuracy, respectively. When compared, the results in Table 4.8, utilising the pre-processing strategy for balancing the classes, PLAX TV and PSAX PV achieved 100% and 99% accuracy, respectively.

Table 4.8: Class breakdown using DenseNet201 and 1xLSTM with imbalanced classes

Class	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
A2CH	96	82	96	88
A3CH	89	93	89	91
A4CH	85	95	85	90
A5CH	90	74	90	82
IAS	88	85	88	86
PLAX Full	92	90	92	91
PLAX PV	73	35	73	48
PLAX TV	61	86	62	72
PLAX Valves	81	99	81	89
PSAX AV	77	91	77	84
PSAX LV	87	81	87	84
PSAX PV	31	59	31	41
Subcostal Heart	93	89	93	91
Subcostal IVC	92	82	92	87
Suprasternal	96	73	96	83



	A2C	A3C	A4C	A5C	IAS	Plax Full	Plax PV	Plax TV	Plax Valves	PSAX AV	PSAX LV	PSAX PV	Sub Heart	Sub IVC	Suprasternal
A2C	82%	1%	1%	0%	1%	0%	1%	0%	0%	0%	0%	0%	1%	0%	0%
A3C	5%	93%	0%	1%	2%	1%	3%	0%	0%	0%	0%	0%	1%	1%	1%
A4C	5%	0%	95%	15%	1%	0%	1%	0%	0%	0%	0%	0%	0%	0%	1%
A5C	1%	0%	3%	74%	3%	0%	0%	0%	0%	0%	0%	3%	0%	0%	0%
IAS	2%	0%	1%	3%	85%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Plax Full	0%	2%	0%	0%	2%	90%	1%	3%	1%	1%	0%	0%	0%	10%	2%
Plax PV	0%	0%	0%	2%	0%	0%	35%	1%	0%	0%	0%	3%	0%	0%	2%
Plax TV	2%	0%	0%	1%	0%	1%	10%	86%	0%	1%	3%	0%	0%	2%	5%
Plax Valves	0%	2%	0%	0%	3%	4%	0%	6%	99%	2%	0%	0%	2%	0%	0%
PSAX AV	1%	1%	0%	2%	2%	1%	6%	1%	0%	91%	16%	34%	1%	1%	5%
PSAX LV	1%	0%	0%	0%	0%	3%	3%	1%	0%	4%	81%	0%	4%	0%	3%
PSAX PV	0%	0%	0%	0%	0%	1%	37%	1%	0%	1%	0%	59%	1%	1%	3%
Sub Heart	0%	0%	0%	2%	0%	0%	0%	0%	0%	0%	0%	0%	89%	1%	4%
Sub IVC	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	2%	82%	2%
Suprasternal	0%	0%	0%	1%	1%	0%	1%	0%	0%	0%	0%	0%	0%	0%	73%

Figure 4.14: Confusion matrix for the original, imbalanced dataset, when trained and tested using the optimal architecture (DenseNet201 + 1x-LSTM)

4.6 Conclusion and Future Work

4.6.1 Conclusion

This chapter sought to establish a two-stage DL network for accurate classification of complex echocardiographic views in both video and still image data. A method of balancing video classes using a 20 frame length window and varying strides proved effective and demonstrates sequence data can be artificially increased in volume without negatively impacting upon model performance.

Four state-of-the-art CNNs were compared for accuracy, precision, recall and F1 score when encoding spatial feature vectors from the final fully connected layer. ResNet50 achieved an accuracy of 92.6% across five view classes with an average inference time of 49 milliseconds. The most challenging views to classify were tricuspid CW and aortic CW predominantly due of their similarity.

A time-distributed DenseNet201 CNN with the addition of 1x LSTM layer produced 98.5% accuracy when classifying 15 complex video views. The most challenging being PSAX AV and PSAX LV, again due to similarities between views. Interestingly, the proposed model compares well when classifying apical views that proved challenging to both DL architectures and human experts in previous studies [68]. These results represent the highest classification accuracy, from the largest unseen test dataset, across the greatest number of classes of any of the previously published studies.

4.6.2 Publications

Lane, E., Jevsikov, J., Naidoo, P., Shun-shin, M., Francis, D. and Zolgharni, M., 2022. Automated Echocardiographic View Classification Using Deep Neural Networks. *tbc* [in preparation]

Azarmehr, N., Zolgharni, M., Ye, X., Howard, J., Lane, E., Labs, R., Shun-Shin, M., Cole, G., Bidaut, L. and Francis, D., 2021. Neural architecture search of echocardiography view classifiers. *Journal of Medical Imaging*, 8(03).

4.6.3 Study Limitations and Future work

Interpretation of the results outlined in this study against previously published literature is challenging due to the use of different datasets, varying modalities, image quality and patient characteristics. For example a study reported 84% accuracy using standard CNNs [165] whereas another published 97.8% accuracy [151] for comparative classification tasks. There are several possible explanations for differing results, most notably the more views the more challenging the classification task.

This chapter investigates view classification using a 2D echocardigraphic dataset as this is the clinically relevant modality. 3D echocardiography currently suffers from considerable reduction in frame rates and image quality, rendering it's adoption in clinical practice limited [93]. When such issues are resolved, classification of echocardigraphic views could be applied to 3D modalities, until this time, 2D echocardiography remains unrivalled.

This study used a dataset comprised of videos and images captured using ultrasound equipment from GE and Philips manufacturers. To be considered vendor-neutral, data captured using a wide range of manufacturers from multiple medical centres should be included.

Future work could centre around the collection and annotation of a large-scale, diverse echocardiographic view dataset from multiple vendors across several sites. For the classification results to be truly comparable, a dataset should be made public for evaluation of reported studies.

Chapter 5

Real-time Quality Assessment of Echocardiographic Images

5.1 Introduction

It has been demonstrated that the quality of a two-dimensional (2D) echocardiographic (echo) image has a significant impact upon the reliability of crucial diagnostic measurements, in addition to intra- and inter-observer variability [54–56] due to inadequate visualisation of the left ventricle (LV) endocardial border. A recent report suggests that up to 10% to 15% of routine echo images suffer from poor quality [54].

Ultrasound device manufacturers are continuously striving to improve image quality, meaning images obtained during an examination using a modern machine may be superior. However, this is inconsistent, as medical centres use different machines from various vendors [57]. Additionally, technological advancements over the past two decades have given rise to the emergence of hand-held echo devices used at the point of care, often by non-specialists. Therefore, any automated method for the assessment of image quality should generalise across all modalities.

The main purpose of this chapter is to develop a novel method of assessing echo image quality via a prototype for a real-time deep learning (DL) web application. Whilst there are no formal standards for the evaluation of echo image quality, several studies are reviewed proposing point-based methods, focusing on the percentage of LV endococardial border visible to the human eye. A novel method of utilising image performance on DL tasks (segmentation of the LV and mitral valve (MV) hinge point localisation) is proposed. This method of image quality labelling captures important spatial information, not visible to the human eye, and provides deeper context as to information DL models require when making accurate predictions.

5.2 Related Work

DL methods, specifically convolutional neural networks (CNNs) [17, 18] and long short-term memory (LSTM) cells for regression [12, 18–22], have been applied to automate echocardiographic image quality assessment, with varying degrees of success.

Almost all employ a similar strategy for label generation. Generally, one experienced operator is asked to label the images in an echo video (cine loop) into one of n point-based classes where numerical percentages refer to image quality. For example, a four point system could be:

- Excellent: >75% - 100% quality
- Good: 50-74% quality
- Fair: 25-49% quality
- Poor: <25% quality

This approach often centres around the observers perception of how clearly visible the LV endocardial border appears in each image [54]. Although this approach is relatively quick and easy, when compared to specifying fine-grained labels [166], it is purely subjective and bias to the opinion of the singular expert who labels the image.

One study deviates from the practice of applying a percentage point-based methodology for label generation, and instead uses a bespoke classification confidence (cc) metric. Their approach consists of a CNN for classification of 10 standard echocardiographic views. Based upon the confidence of the classification for each image, a quality label is derived. Though relatively novel, this approach was limited in scope due to a small dataset and only considering the straightforward view classification accuracy as an overall metric for image quality [162].

As computer vision (CV) algorithms yield considerable knowledge of pixel-level details through the convolutional process, an approach to determining quality utilising this spatial context, virtually invisible to the human eye, has great potential. In essence, allowing DL models to determine which images would generate accurate predictions based upon a specific task, as opposed to visual assessment and human opinion.

5.2.1 Deep Learning (DL) Based Approach to Quality Assessment

This chapter details the implementation of a U-Net model for LV segmentation and a heatmap-based landmark localisation network for predicting anterior and posterior MV hinge points (pixel coordinates).

An apical 4-chamber (A4C) echo dataset (including the same split into train/validate and test sets) was used for each model. Once trained, inference was run on the entire dataset and established metrics (HD, DC, IoU for segmentation and Euclidean distance for landmark detection) give an indication of how well each image performs for each DL task. These metrics then form the quality assessment score. A more detailed explanation of label generation is discussed in subsequent sections.

5.2.2 Clinical Deployability

Echocardiography has established itself as the primary imaging modality for diagnosing heart conditions. Indeed, over the past 20 years, exponential technological advancements have enabled the emergence of compact, battery operated hand-held echocardiography meaning the examination can be performed at the point of care with acceptable image quality. Examples of such devices can be found in Figure 5.1.

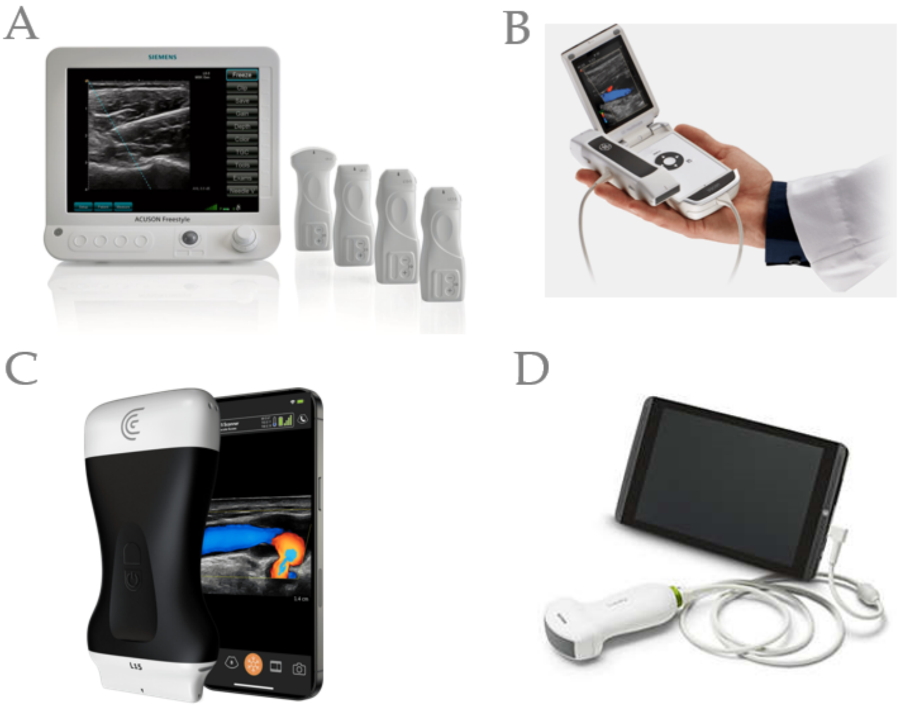


Figure 5.1: Examples of handheld echocardiogram devices: **A:** ACUSON Freestyle Series Ultrasound Systems (wireless), Siemens Healthineers [167], **B:** Vscan Extend Handheld Ultrasound, GE Healthcare [168], **C:** HD3 Ultra-portable Ultrasound, Clarius [169], **D:** Iviz [170]

Easy to use, portable and low-cost, these devices have increased in popularity for use during crucial decision making processes [171]. AI enabled applications have potential to widen accessibility to lifesaving intervention by guiding the operator with indications of optimal views, probe angles and image quality. Allowing non-specialists to perform image acquisition and image analysis and interpretation to occur remotely, practically anywhere in the world.

Two real-time point of care applications have been published in the relevant literature, one uses a binary classification network to divide images into a “high” or “low” quality group [20] while the other provides real-time operator feedback on view classification (14 views) as well as an indication of quality using an application on an Android mobile phone [172]. Both of these approaches utilise the point-based human assessment of image quality as previously described.

5.3 Main Contributions

This chapter presents a multi-task DL model for echocardiographic image quality assessment using a novel labelling methodology and outlines the architecture of a web-based application to provide real-time feedback to operators. The main contributions are:

- Developing a novel, objective image quality assessment method based on the suitability of each image for downstream automated DL clinical measurements
- Investigating the applicability of the proposed method using segmentation and landmark detection tasks
- Investigating the feasibility of providing a web-based application for real-time image quality assessment feedback

5.4 Methodology

5.4.1 Dataset, Ethics and Expert Annotations

The dataset used in this chapter, called Unity, consists of 1,224 A4C cine loops, acquired using ultrasound equipment from GE and Philips manufacturers between 2015 and 2016 from Imperial College Healthcare NHS Trust's echo database. Examinations were performed by experienced echocardiographers following standard protocols. Ethical approval was obtained from the Health Regulatory Agency (Integrated Research Application System identifier 243023).

In total, 4,372 still images were extracted from the videos at various points in the cardiac cycle. The entire dataset was split 60%/20%/20% for train, validation and test sets, respectively. To maintain consistency, the same images remained in each set for all experiments.

5.4.2 Ground-Truth Definition

MV keypoint Localisation Network

Images were labelled by one human expert, from a pool of available experts, using the proprietary Unity Imaging labelling application [173]. Cartesian coordinates for the pixel location of the anterior and posterior MV hinge points were provided. Where the operator judged an image to be of low quality, and thus could not locate the hinge points, an annotation was not provided.

LV Segmentation Network

The ground-truth definition for the segmentation of the LV border were provided as annotations by trained experts, again using the Unity Imaging application. The manually labelled endocardium was provided in the form of control points which represented a B-Spline curve that constitutes the left ventricular (LV) cavity boundary. As with the MV hinge points, if the image was considered poor quality by the human operator, no ground-truth was provided.

Quality Assessment Network

To generate the ground-truth to reflect the quality of each image, they were first passed through the trained MV keypoint localisation network (from hereon out referred to as the ("Keypoint model") and LV segmentation network (referred to as "Segmentation model"). The results of the evaluation metrics (Euclidian distance for the Keypoint model and DC, HD and IoU for the Segmentation model) were used as the ground truth. If no ground-truth was provided by the human expert who originally annotated the image, it was considered poor quality and given a quality score of zero.

All metrics were normalised between [0,1], inversely normalised in the case of HD and IoU, to make sure all labels were uniform and suitable for regression tasks.

An experimental 10-point classification scale was also established by taking the mean of the evaluation metric labels:

- 10: .9 - 1 average metric
- 9: >.8 - .89
- 8: >.7 - .79
- 7: >.6 - .69

- 6: >.5 - .59
- 5: >.4 - .49
- 4: >.3 - .39
- 3: >.2 - .29
- 2: >.1 - .19
- 1: >.0 - .9

5.4.3 Neural Network Architecture

MV keypoint Localisation Network

MV anterior and posterior hinge points are detected using a multi-Stage heatmap regression network. An example A4C image from the Unity dataset, with corresponding annotations, can be seen in Figure 5.2.

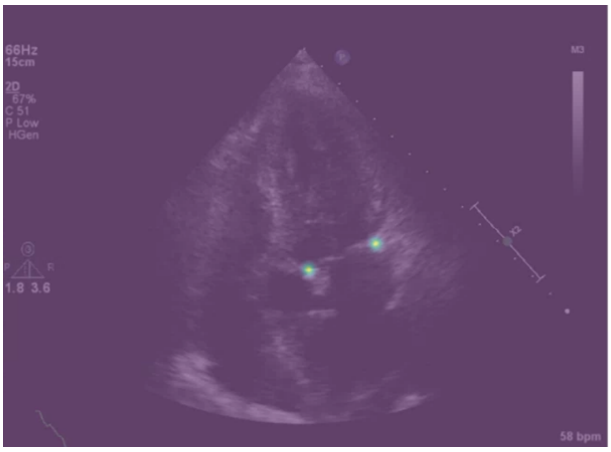


Figure 5.2: An example A4C image from the Unity dataset with heatmap overlay for mitral valve anterior and posterior hinge point annotations

Instead of predicting Cartesian coordinates, the Keypoint model predicts a different Gaussian response heatmap, or belief map, for every keypoint of interest (2, in this case). A heatmap is simply an image indicating the likelihood of a specific keypoint residing at that pixel. Subsequently, keypoints are obtained by finding the local maxima. Indirect inference through a predicted heatmap offers several advantages over a direct prediction [174]. A high-level overview of the Keypoint model is illustrated in Figure 5.3.

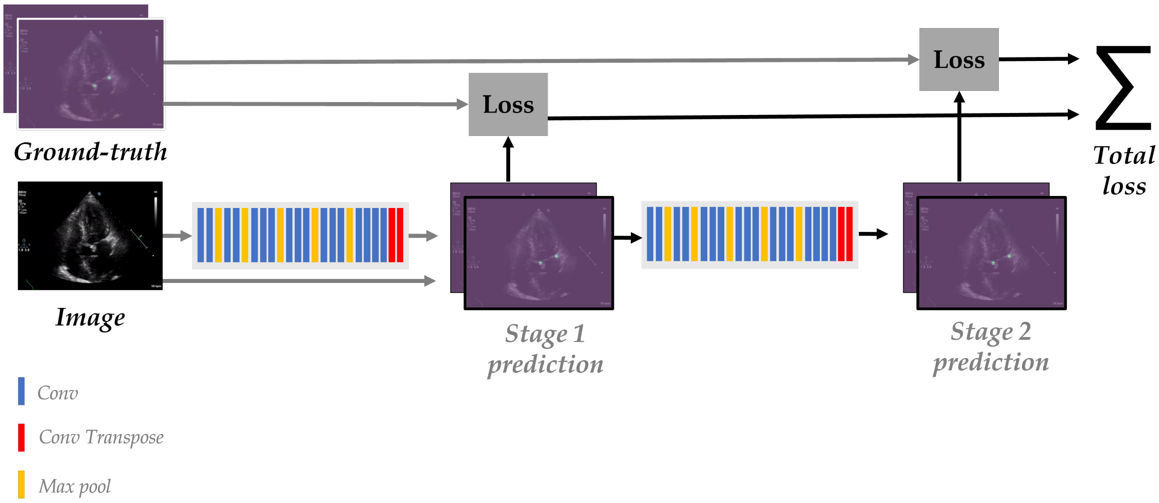


Figure 5.3: A high-level overview of the Keypoint model

The Keypoint model consists of a two-stage sequence of CNNs that produce 2D heatmaps for the location of the keypoint. At the second stage, image features and the heatmaps produced by the first stage, are used as the input. The heatmaps provided at the second stage are an expressive, non-parametric encoding of the spatial uncertainty for each keypoint location, allowing the network to learn rich image-dependent spatial models of the relationships between keypoints.

In order to consolidate the images global and local features, a fully convolutional hourglass module network was adopted as the CNN feature extractor at each of the two stages [174]. This networks acts as an encoder-decoded by first passing the image through five convolutional blocks, followed by max pooling layers to extract a feature vector. The Keypoint model architecture is similar to the proposed model in Chapter 8, within which a more detailed explanation about the implementation is included,

LV Segmentation Network

The U-Net architecture, also discussed in Chapter 3 and Chapter 7, was adopted for semantic segmentation of the LV due to it's high performance in CV tasks using biomedical imaging datasets. It is also described as an encoder-decoder architecture, as illustrated in Figure 5.4.

The left portion of the diagram represents the encoder, during which convolutional blocks, followed by max pooling downsampling layers, encode the image into a feature vector representation.

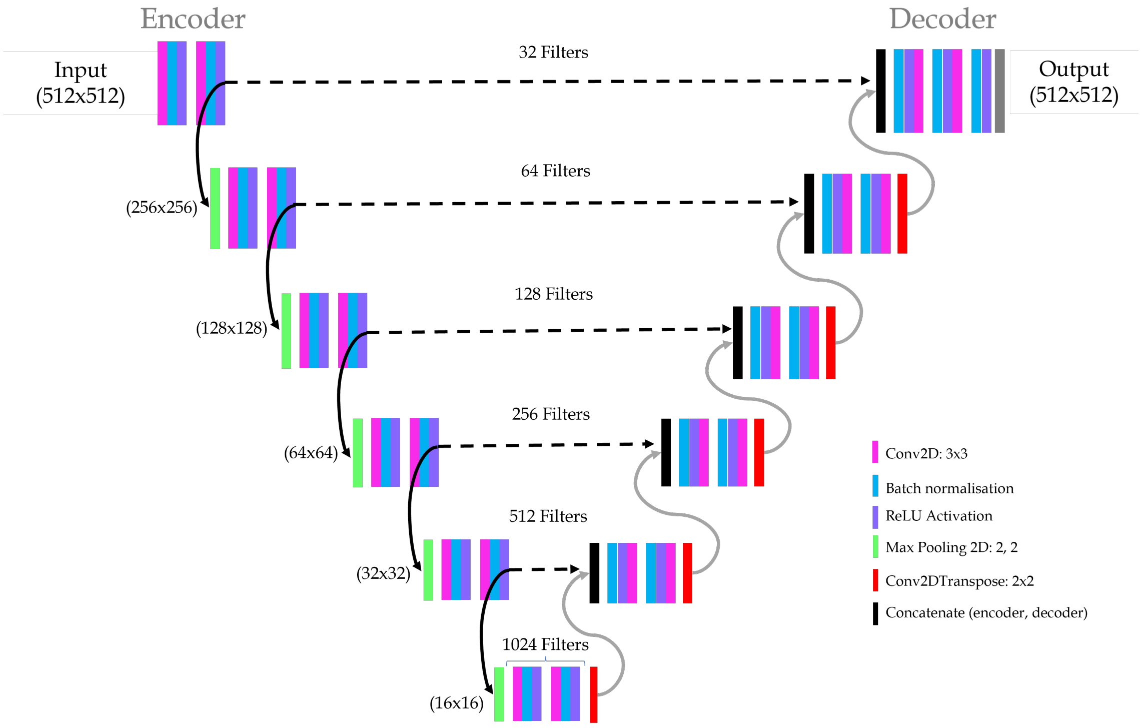


Figure 5.4: High-level illustration of the encoder (left) and decoder (right) U-Net architecture

During the decoder phase, on the right of Figure 5.4, the input is upsampled to restore the condensed feature vectors back to the original image dimension. Skip connections concatenate higher resolution feature vectors from the corresponding encoder stage with upsampled features to better learn feature representations.

Quality Assessment Network

The quality assessment network is a multi-task learning model with a ResNet50 backbone for feature extraction, a classification branch and regression branch. A high-level illustration of the network is provided in Figure 5.5.

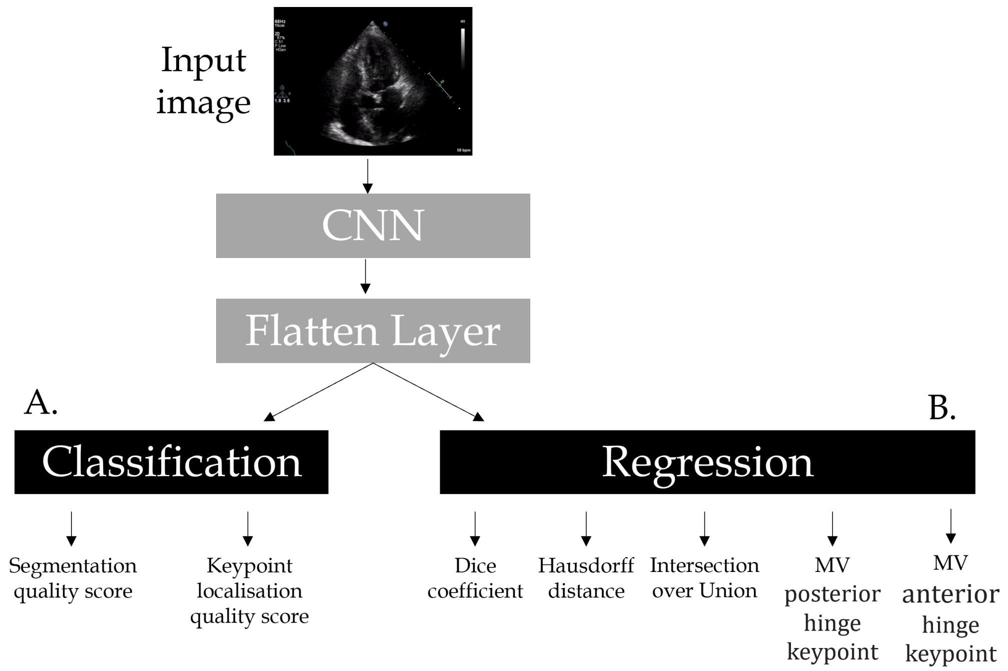

Input image
CNN
Flatten Layer
A.
B.
Classification
Regression
Segmentation quality score
Keypoint localisation quality score
Dice coefficient
Hausdorff distance
Intersection over Union
MV posterior hinge keypoint
MV anterior hinge keypoint

Figure 5.5: A high-level overview of the echo image quality assessment network: A. a 10-point classification branch for segmentation and keypoint localisation tasks and B. regression output for each evaluation metric for segmentation and keypoint localisation networks

Branch A. provides a 10-point classification score for image quality related to both segmentation and keypoint localisation tasks. The classification branches of the multi-task network comprise a dense layer with Softmax activation and sparse categorical cross entropy loss. The regression branch outputs a normalised value for each evaluation metric (as illustrated in Figure 5.5 and comprises three dense layers with ReLU activation and mean squared error (MSE) loss.

5.4.4 Implementation Details

All models were implemented using Tensorflow 2.0 [143] and Keras [144] DL frameworks. Training and inference was conducted on a server containing four NVIDIA GeForce RTX 3090 GPUs.

MV Keypoint Localisation Network

The Keypoint model was trained using the Adam optimiser and an learning rate of $1e-5$ and MSE loss function. Images were resized to 192x192 pixels with one channel. The batch size was 64 and early stopping with a patience of 30, monitoring the validation loss, was used.

LV Segmentation Network

To train the LV segmentation network, the images were resized to 512x512 with one channel. The loss function was Adam with a learning rate of $1e-5$ and the loss function was binary cross entropy. Early stopping was used with a patience of 100 monitoring the validation loss.

Quality Assessment Network

To fit the ResNet50 backbone of the multitask learning network, the gray scale images were resized to 224x224 and stacked to assemble 3 channels. The optimiser was Adam with a learning rate of $1e-4$, a batch size of 32 and early stopping monitoring validation loss with a patience of 100.

The regression branch of the multitask network was trained using MSE loss and the classification branch with sparse categorical cross entropy loss.

Web Application Framework

The application for real-time echo image quality assessment was programmed using Python 3 and the Flask 2.2 [175] micro web framework, incorporating Jinja2 3.1 [176] templating engine and Werkzeug WSGI (Web Server Gateway Interface) [177].

Echo images are acquired using an Epiphan frame grabber [178] connected to the ultrasound machine. The images are then input to the DL multitask model and inference is run via an NVIDIA Quadro RTX 3000 GPU integrated within a Microsoft Surface Book 3 running the Windows 11 Pro operating system (OS). Figure 5.6 provides a diagram illustrating the hardware configuration.

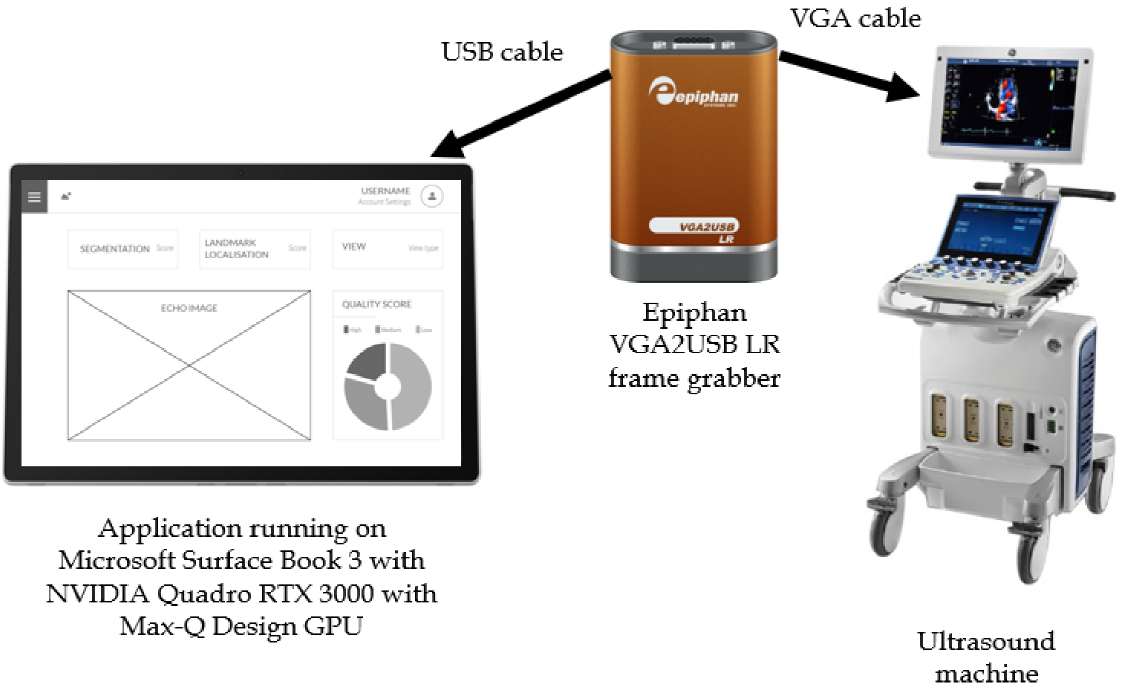

VGA cable
USB cable
USERNAME
Account Settings
SEGMENTATION Score
LANDMARK LOCALISATION Score
VIEW View type
ECHO IMAGE
QUALITY SCORE
High
Medium
Low
epiphan
VGA2USB LR
Epiphan VGA2USB LR frame grabber
Application running on Microsoft Surface Book 3 with NVIDIA Quadro RTX 3000 with Max-Q Design GPU
Ultrasound machine

Figure 5.6: An illustration of the hardware configuration for real-time echo image quality assessment

Echo images are captured by the frame grabber and rendered to the application UI using methods from the OpenCV [179] image and video processing library in conjunction with the video stream function from the "imutils" [180] integration package for OpenCV.

Updating the echo image and quality score components of the web page dynamically, without the user having to hit a button on the browser, presented a significant challenge. Turbo-Flask [181] is a Flask extension that integrates the turbo.js JavaScript library with the application itself. The library consists of a number of different features with the goal of making server generated web pages behave like single-page applications. The "Turbo Streams" method was employed to allow the server-side application to update parts of the web page by submitting HTML fragments to the client. Additionally, Python threading enables concurrent processing of the image captured by the frame grabber and DL model inference (to ensure correct score is dynamically rendered for each captured image in the video stream).

The open source Chart.js [182] community project for HTML 5 and JavaScript charts was used to render the image quality metric on the front end in the form of a donut chart.

The quality assessment network is web-based and, as such, the user interface was designed using HTML, Bootstrap 5 and CSS. A wireframe mockup of the interface can be seen in Figure 5.7

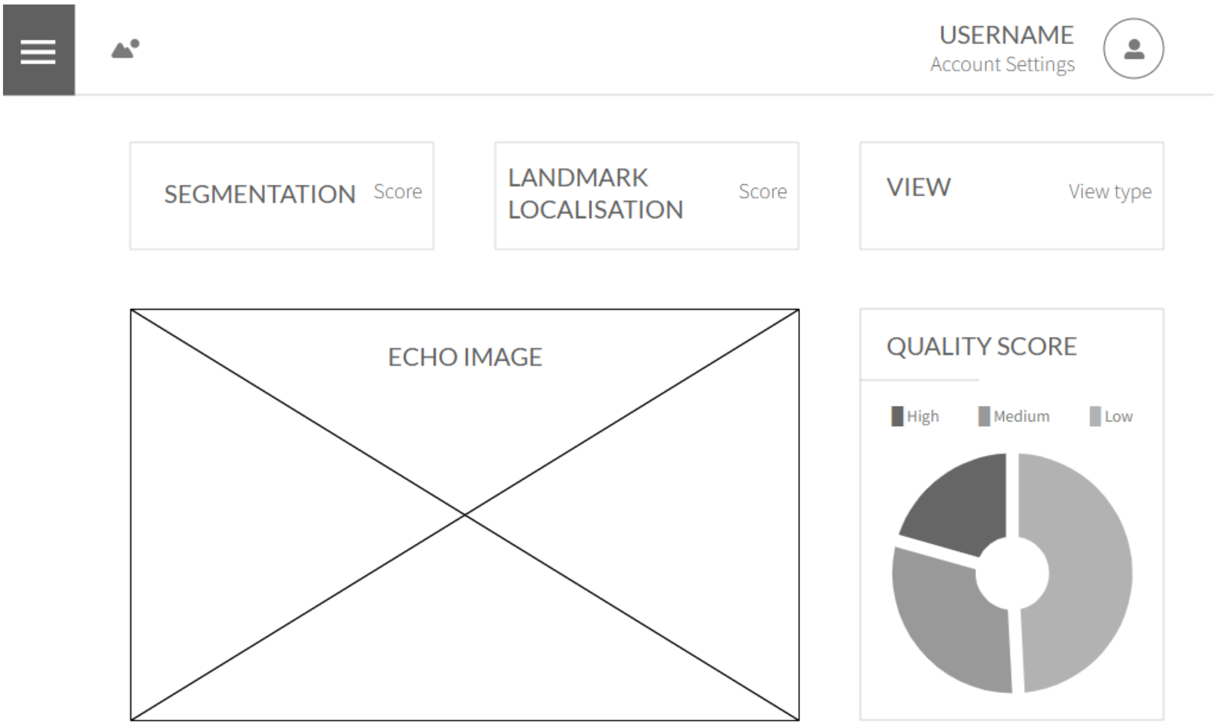


Figure 5.7: A wireframe mockup of the quality assessment web application user interface

5.4.5 Evaluation Metrics

MV Keypoint Localisation Network

For each MV keypoint (anterior or posterior) the Euclidean distance between the ground-truth annotation and prediction was calculated. Where X and Y are the ground-truth and x and y are network predictions,the equation for the Euclidean distance can be expressed as:

$$\sqrt{(x-X)^2+(y-Y)^2}$$

LV Segmentation Network

The accuracy of the segmentation model is assessed using well-established evaluation metrics: Dice Coefficient, Hausdorff Distance and Intersection over Union (IoU). More detailed information about these metrics can be found in Chapter 3 and Chapter 7.

Quality Assessment Network

Once the output of the Keypoint model and Segmentation model have been evaluated and normalised, the accuracy of the quality assessment network is calculated. The classification branch of the multi-task learning network is evaluated using precision, recall, F1 score and accuracy. For the regression branch, the MSE is calculated. More detailed information about these metrics can be found in Chapter 3 and Chapter 4.

5.5 Results and Discussion

5.5.1 MV Keypoint Localisation Network

The Keypoint model produced predictions on the unseen test dataset for anterior and posterior hinge points and accuracy was measured by the Euclidean distance between the ground-truth pixel location and the maxima of the predicted heatmap. The mean and standard deviation for the anterior MV hinge point was 23.49 ± 42.43 and 27.95 ± 48.84 for the posterior. The images with particularly high Euclidean distance are those where the network did not perform as expected and thus, the quality score will be close to 0. For those images where the prediction was an exact, or close, match to the ground-truth, the quality score would be closer to 1.

5.5.2 LV Segmentation Network

The segmentation model produced binary mask predictions for the delineation of the LV endocardium. Evaluation metrics are the average Dice coefficient (DC), Hausdorff Distance (HD) and Intersection over Union (IoU) are displayed in Table 5.1. As with the Keypoint model, those images which did not fair as well with regard to the accuracy of the predicted model, regarding any of the three metrics, would be considered worse quality and thus result in a label close to 0. Conversely, where a prediction is a close match, the score label would be close to 1.

Table 5.1: Evaluation metrics for the LV segmentation model

Model	Av. DC	Av. HD	Av. IoU
Segmentation model	0.91	22.33	0.83

5.5.3 Quality Assessment Network

The quality assessment multitask learning network was trained for classification and regression tasks with labels generated from the Keypoint and Segmentation models. All metrics were normalised between [0,1], and where necessary were inversely normalised.

Table 5.2 represents the MSE for the regression branch. The MSE is an estimator of the model's performance and decreases as the error is close to zero. With regard to the regression predictions for image quality, the MSE is very pleasing. The figures are close across all five metrics and demonstrate consistency. In this case, the error is relatively low.

Table 5.2: MSE for the regression branch evaluation metrics of the multitask learning network for echo image quality assessment

Model	DC	HD	IoU	MV Anterior	MV Posterior
Quality assessment network	0.13	0.13	0.12	0.14	0.14

However, Table 5.3 demonstrates the classification branch, with labels derived from a 10-point quality score system (as previously described) was not an effective measure of image quality. The accuracy, precision, recall and F1 score are all very low. Based on the figures from both tables, the regression score is significantly more accurate than a classification approach.

Table 5.3: Evaluation metrics for the classification branch of the multitask learning network for echo image quality assessment

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Quality assessment network	.50	.2	.19	.19

Figure 5.8 is a confusion matrix for the classification branch. There are no images which fall into the class 2 and class 3 bands; demonstrating the classes are imbalanced. Furthermore, inspection of the confusion matrix bolsters the conclusion classification is not the optimal method of assessing a quality score with a point-based system for this particular dataset.

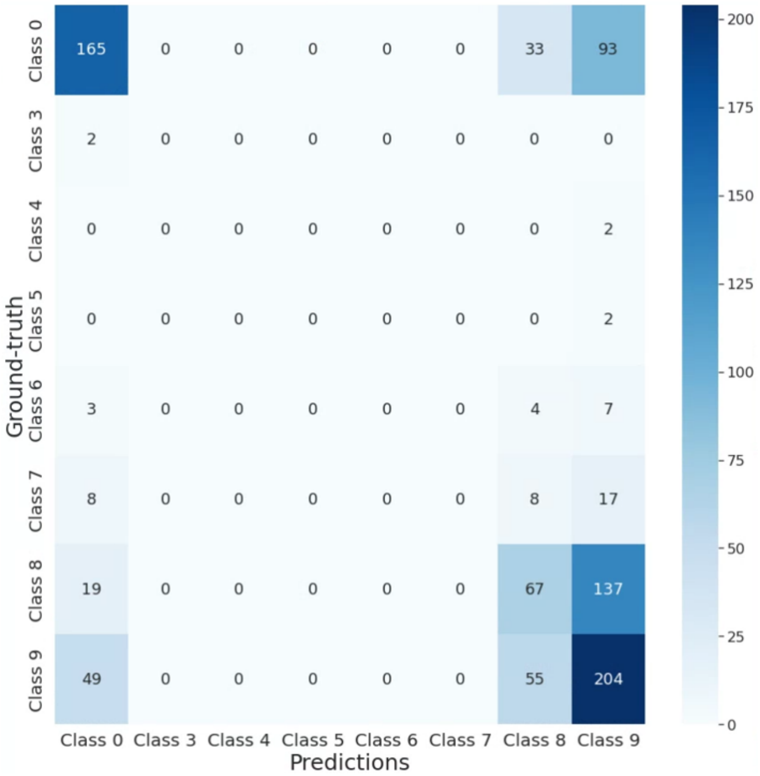


Figure 5.8: Confusion matrix for the classification branch of the multitask learning network for echo image quality assessment

5.6 Conclusion, Limitations and Future Work

5.6.1 Conclusion

The purpose of this chapter is to present a novel approach for echocardiographic image quality assessment using the performance of each image when inference is performed on a trained LV segmentation and MV hinge point localisation models. The proposed network is a multitask learning network with a ResNet50 backbone for feature extraction and subsequent classification and regression branches. Early experimental results demonstrate the accuracy of the regression branch significantly outperformed a 10-point classification approach. However, the Unity dataset used in this Chapter is imbalanced, due to the inclusion of a significant proportion of high quality images, some low quality and very few in between.

Additionally, a web-based Flask application is presented fora potentially real-time visualisation of image quality score at acquisition via connection to a frame grabber and ultrasound machine. The application is still in the very early stages of development and, due to inference time, is currently providing feedback for one image per second. In order for the application to be real-time, this should be resolved to a minimum of 20 - 25 frames per second.

5.6.2 Limitations

This project is still in the initial prototype and experimental stage, however there are a number of limitations which have become clear, these are:

- The Unity dataset is imbalanced, for improvement in the accuracy of the trained network, more poor to mid quality images should be included

- The novel method of producing image quality labels from the evaluation of predicted accuracies from segmentation and keypoint detection models is only as effective as the ground-truth annotations produced by experts. In some cases the network is confused by inconsistent annotations, as highlighted in Figure 5.9

- The web application is currently not real-time in terms of processing ultrasound frames and rendering the quality score to the front-end. Improvements in inference time will need to be made to increase the number of frames processed per second

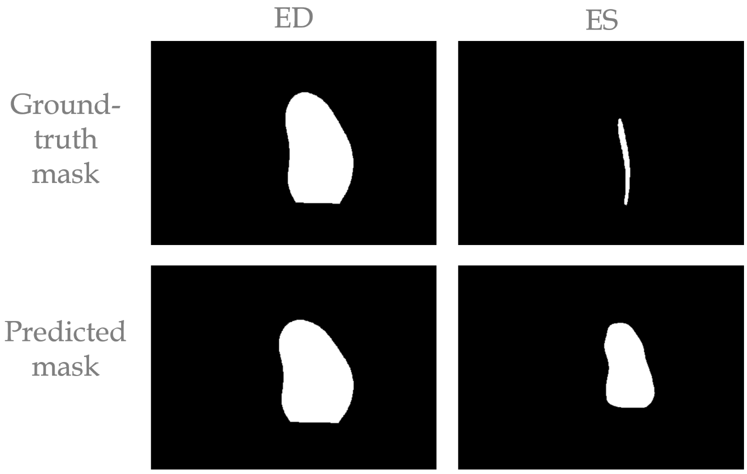


Figure 5.9: An example of inconsistent ground-truth labels for the ES phase LV segmentation contained within the Unity dataset

5.6.3 Future Work

A summary of the suggested future work is as follows:

- Gather more diverse dataset encompassing an improved range of image qualities
- Ensure all ground-truth annotations are accurate
- Expand the network to assess quality of more echocardiographic views, not limited to A4C
- Improve the inference time of the trained network to speed up the web application and render results in real-time. This could be done by removing the classification branch of the network as the results detailed in this Chapter have proved it is not an effective method of assessing image quality and convert the model to a Tensorflow Lite instance

Chapter 6

Echocardiographic Phase Detection Using Deep Neural Networks

6.1 Introduction

Cardiovascular disease is the leading cause of death in western counties. In the UK, despite a decline in cardiovascular disease mortality, hospital admissions for related conditions are rising. A 2D echocardiogram (ultrasound) is the modality of choice for a non-invasive examination of heart function. A full cardiac cycle consists of every event occurring as the heart beats. It includes two important phases: diastole, as the heart relaxes, and systole, subsequent contraction.

Accurate identification of end-diastolic and end-systolic frames in echocardiographic videos (cine loops) is important, yet challenging, for human experts. Manual frame selection is subject to uncertainty, affecting crucial clinical measurements, such as myocardial strain. Therefore, the ability to automatically detect frames of interest is highly desirable.

This research develops a deep neural network, trained and tested on multi-centre patient data, for the accurate identification of end-diastolic and end-systolic frames in apical four-chamber 2D multibeat cine loop recordings of arbitrary length. Seven experienced cardiologist experts independently labelled the frames of interest, thereby providing infallible annotations, allowing for observer variability measurements.

When compared with the ground-truth, the proposed model shows an average frame difference of -0.09±1.10 and 0.11±1.29 frames for end-diastolic and end-systolic phases, respectively. When applied to patient datasets from a different clinical site, to which the model was blind during its development, average frame differences of -1.34±3.27 and -0.31±3.37 frames were obtained for both frames of interest. All detection errors fall within the range of inter-observer variability.

The proposed automated model can identify multiple end-systolic and end-diastolic frames in echocardiographic videos of arbitrary length, with performance indistinguishable from that of human experts, but with significantly shorter processing time.

Assessment of left ventricular (LV) function is of principal importance during an echocardiographic examination and is crucial for accurate patient evaluation. Echocardiography continues to be the most common technique in clinical practice for the quantification of LV function markers, such as ejection fraction (EF) and global longitudinal strain (GLS) [9]. Measurements usually relate to time points, such as end-diastole (ED) and end-systole (ES). Therefore, accurate detection of the end of the LV systole and diastole phases constitutes a critical step in any echocardiographic exam. Figure 1.1 provides a visual overview of vital parameters throughout the duration of a full cardiac cycle using electrographic trace lines.

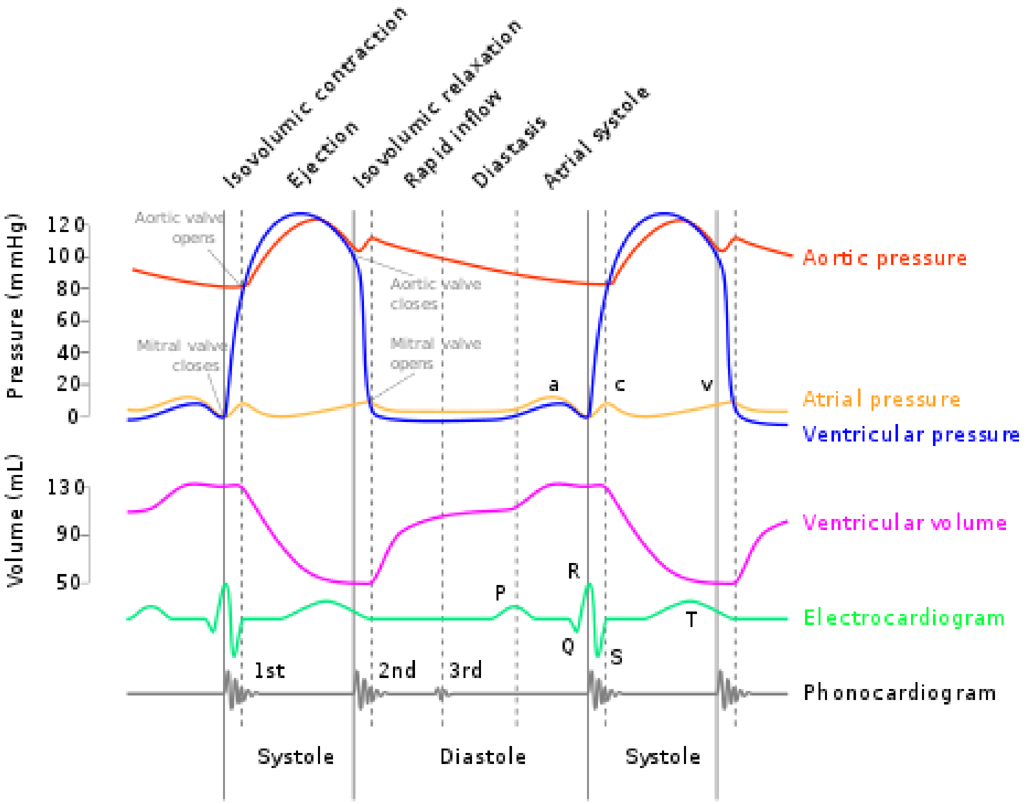


Figure 6.1: An illustration of the events during a complete cardiac cycle. From left to right, electrographic trace lines depict changes in each parameter as time elapses and throughout the duration of each phase: systole, diastole and systole again. [183]

6.1.1 The Need for Fully Automated Systems

The importance of accurate identification of ED and ES frames was recently demonstrated by Mada et al. [59]. As previously stated, an error of just two to three frames in detecting the ES phase elicits an approximate 10% difference in segmental ES strain. Furthermore, the sensitivity of frame selection is greater in relation to the left bundle branch block. As highlighted by Amundsen [60], the consequence of misidentification of ED and ES frames can be extensive; impairing concordance between observers in both research and clinical practice. Therefore, automated methods for the resolution of accurate ED and ES phase detection could greatly contribute to improving the consistency of echocardiographic quantification.

The process of identifying ED and ES frames in video data is manually performed by trained clinicians via on-screen visual selection. ED frames can be determined using cues such as mitral valve closure, ECG R-wave and maximum LV volume. Whereas ES frames are commonly defined by mitral valve opening, minimal LV volume, aortic valve closure, or the end of the ECG T-wave. However, due to subtle frame-on-frame spatial differences, and complex temporal relationships virtually invisible to the human eye, manual detection presents a significant barrier to consistent diagnosis due to intra- and inter-observer variability lacking reproducibility and precision [58].

Recent research previously identified the medial disagreement between accredited and experienced experts as 3 frames [51] when performing manual identification. Therefore, reliable and reproducible methods for ED and ES frame detection would allow for the development of fully automated techniques. Thus, meeting the objective of accurate quantification of LV function, in addition to automated calculation of EF and stroke volume, GLS and wall thickening.

6.1.2 Value of Independence From ECG

Often, cardiac timing is determined through analysis of an accompanying ECG signal during an echocardiogram exam. Despite providing information enabling the computation of some clinically important parameters, such as temporal intervals from the R-wave peaks, ECG recordings require the connection of multiple cables which is time-consuming and, at times, inconvenient. In an era when highly portable scanners can be used to undertake focused studies lasting just a few minutes [50], the capacity of detecting cardiac timing events, independent from the ECG signal, has potential. Such as integration with automated technology on handheld devices.

6.2 Related Work

Recent studies have attempted to address this problem. In the absence of an ECG signal, tissue Doppler data has been used to estimate cardiac cycle length [184] or detect ED frames [185]. Machine learning approaches have also been applied to automatically detect ED and ES frames from 2D echocardiography images (B-mode). This includes manifold learning [186], speckle tracking [187], correlation-based frame-to-frame deviation measures [188, 189], nonlinear filtering and boundary detection techniques [190].

More recently, studies have focused on deep learning approaches, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Deep residual recurrent neural networks were applied to phase detection in apical-4-chamber (A4C) echocardiograms [191]. A major limitation of this study is the proposed model only accepts videos with a fixed number of frames, containing just one cardiac cycle. Presumably, this approach necessitates preprocessing of the input image sequence to isolate a single heartbeat.

The same authors later reported on combining CNN and RNN modules to detect frames of interest [96]. Although varying length inputs (22-59 frames) were used, again their results indicate the videos contained just one cardiac cycle. It is assumed this variation in length was due to different frame and heart rates. A summary of all accuracies from previously reported studies, compared with those of the developed research models from this study, is provided in the results section.

Additionally, 3D CNNs have been applied for the extraction of spatial-temporal features from A4C and apical-2-chamber (A2C) echocardiographic videos [192]. While the study states the model was trained on variable length sequences, the feasibility of the model was demonstrated only on a pair of detected ED/ES frames in each video with the QRS-complex in the accompanying ECG signal being used to detect an additional ED frame for the videos starting in systole phase; thereby providing ground-truth for a full cardiac cycle (ED-ES-ED).

6.2.1 Value of Multibeat Analysis

In clinical practice, longer recordings would allow for probing of physiological reactions after intervention, where detecting a subtle change in the mean value of a clinical maker, amongst much larger background beat-to-beat variability, is essential.

As stated, recent studies have failed to target the application of automated phase detection in arbitrarily long, uninterrupted echocardiogram recordings containing several full heartbeats. Clinically, it is necessary to monitor changes in crucial markers, such as EF or strain, from one examination to the next. Measurements taken from only one heartbeat may result in test-retest variability. Therefore, it would be impossible to reliably conclude whether a patient's condition has deteriorated over time. Such variability and inaccuracy can be reduced by averaging measurements over several heartbeats, from the same acquisition. However, this is impractical when a proposed automated model is incapable of returning more than one single pair of ED/ES frame predictions.

As previously reported, the issue of beat-to-beat variability in echocardiography and potential bias due to using a single measurement from a single heartbeat exists [193][194][195]. When random variability between heartbeats is large, clinicians use "clinical judgement" to select which value to report, largely unaware of the devastating consequences for subsequent use.

The ability to acquire and automatically analyse many heartbeats within reasonable time constraints would permit clinical protocols to be developed for multi-beat measurements, hence reducing undesirable variability between clinical assessments. In such measurements, the exact time of ED and ES events for each heartbeat is required.

6.2.2 Clinical Deployability

Without exception, all previously reported studies related to echocardiographic phase detection have used 'private single-centre' clinical datasets for model developments, in both training and testing sets.

Experience shows the performance of models trained using a single dataset may reduce considerably when transferred from one clinical site to another and when applied to different equipment and protocols [196]. This limitation has proved prohibitive to the development of automated models becoming an acceptable mainstream methodology in daily clinical practice. Evaluating models on multi-centre clinical datasets naturally results in greater patient numbers, a wider range of groups, external validity and lower systematic bias. Thus, resulting in increased generalisability of the developed models in contrast to single-centre dataset studies.

Additionally, the effectiveness of previous approaches is difficult to measure. This is, in part, due to an absence of publicly available benchmarks. Therefore, accurate interpretation of previously reported results from the literature, encompassing a wide range of accuracies, is not feasible since a direct comparison of the frame detection accuracy would require access to the same patient dataset. To date, no study has used and reported accuracies on a publicly available echocardiography dataset.

6.3 Main Contributions

This Chapter outlines a CNN-RNN model for the accurate prediction of ED and ES frames in arbitrary length cine loops. The main contributions of this research project can be summarised as being the first study of its kind to:

- investigate the feasibility of using a deep learning framework to detect ED and ES frames in echocardiographic videos of arbitrary length, containing several heartbeats
- demonstrate the applicability of the developed framework by including several patient datasets from various clinical centres, where one dataset was used for model development and the others used for testing
- use annotations (ground-truth) from several cardiologist experts, allowing for the examination of inter- and intra-observer variability
- include performance reports on a publicly available dataset, thereby providing a benchmark for future studies

6.4 Methodology

6.4.1 Dataset, Ethics and Expert Annotations

Descriptions of the datasets used in this study is as follows, with a summary provided in Table 6.1.

PACS-dataset:

A large random sample of echocardiographic studies from different patients performed between 2010 and 2020 was extracted from Imperial College Healthcare NHS Trust's echocardiogram database. Ethical approval was obtained from the Health Regulatory Agency for the anonymised export of large quantities of imaging data. It was not necessary to approach patients individually for consent of data originally acquired for clinical purposes.

The images were acquired during examinations performed by experienced echocardiographers, according to the standard protocols for using ultrasound equipment from GE and Philips manufacturers. Only studies with full patient demographic data, and without intravenous contrast administration, were included. Automated anonymisation was performed to remove the patient-identifiable information. A detailed description, including patient characteristics, can be found in Howard et al. [68].

A CNN model, developed and presented in Chapter 4, to detect different echocardiographic views, was then used to identify and separate the apical four chamber (A4C) views. Figure 6.2 is a diagram showing a typical A4C view and echocardiographic image. A total of 1,000 videos from different patients of varying lengths, containing 1-3 heartbeats, were randomly selected.

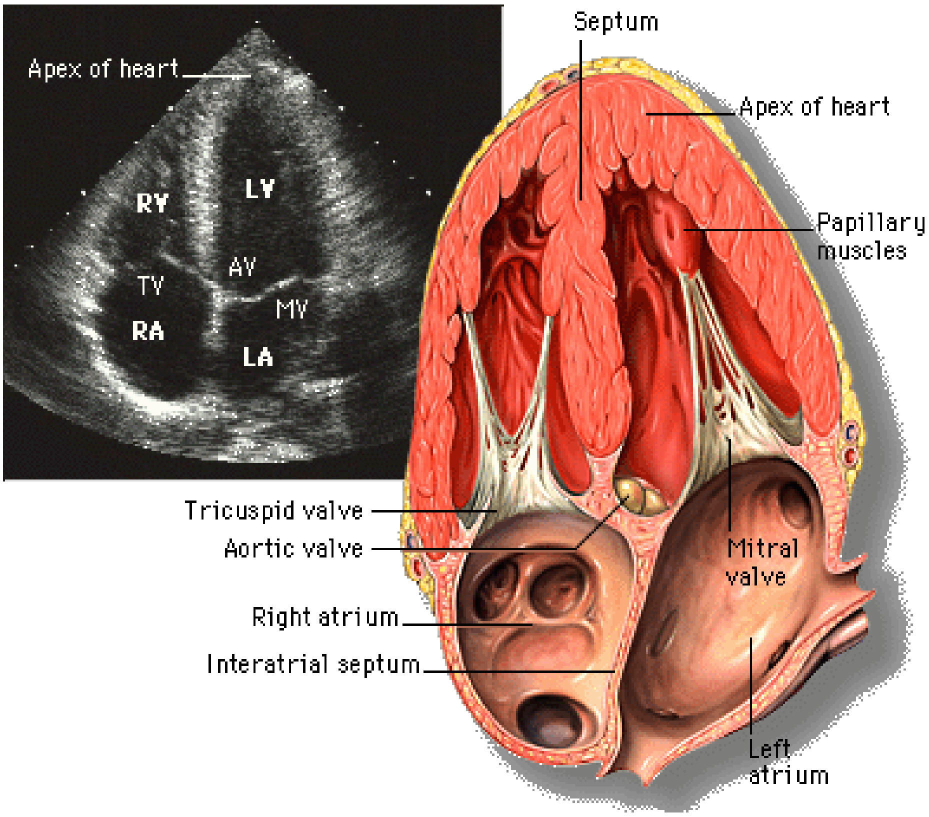


Figure 6.2: Typical A4C echocardiographic image view [197]

Two accredited and experienced cardiology experts manually selected ED and ES frames, each blinded to the judgment of the other. A custom-made program closely replicating the interface of clinical echocardiography hardware was developed for use in this study. Operators visually inspected the cine loops by controlled animation using a trackball, or arrow keys. The operators were asked to pick ED and ES frames in the A4C view, as they would in preparation for a Biplane Simpson's measurement in clinical practice. Selections were made in one or more sessions at their convenience and the time taken was recorded. Videos thought to show more than one view, or misclassified by the CNN as A4C, were excluded, resulting in 898 A4C videos which were then used to define the reference ground-truth ED and ES frames for model developments (both training and testing).

Finally, the original DICOM-formatted image sequences were down sampled by cubic interpolation into a standardised size of 112×112 pixels.

MultiBeat-dataset:

2D echocardiographic images were collected from 40 patients (18 males), with an age range of 27-80 years and a mean age of 59 years, who were referred for echocardiographic examination in the Echocardiography Department at St Mary's Hospital, London. There were no selection criteria, and all patients were in sinus rhythm. The study was approved by the local ethics committee and written- informed consent was obtained from all patients.

Standard transthoracic echocardiography was performed using a GE Vivid.i (GE Healthcare, London, United Kingdom) ultrasound machine equipped with a 1.5-3.6 MHz transducer (3S-RS). For each subject, an A4C view was obtained in left lateral decubitus position as per standard clinical guidelines [71].

The operators performing the exam were instructed not to change any machine setting (e.g. sector, gain, depth, etc.) and the probe position during the acquisition period to obtain consistent data. The acquisition period was 20 seconds to make sure at least 10 cardiac cycles were present in all videos. The images were stored digitally for subsequent offline analysis. The ECG trace was present on all echocardiographic recordings.

Using the same platform described for the PACS-dataset and in a similar process, five other accredited and experienced cardiology experts manually selected ED and ES frames, again each blinded to the judgment of the others. All videos were then renamed and provided to one operator in a random order for second analysis, no previous result was shown. Thus, the operator was blinded from their own previous frame selections. To maintain independence, the operators annotating the MultiBeat-dataset were different from those who labelled the PACS-dataset.

Where an operator judged a beat to be of low quality, they declared it invalid and did not make a selection. Therefore, since the operators were blinded to each other and their own previous selections, there were heartbeats that were delineated on one or two viewings only by each operator. Only the heartbeats which had 6 delineations (540 in total) were used for testing the models. The location of the typical frames identified by the operators is plotted as red circular markers in Figure 6.5. DICOM-formatted image sequences were again down sampled by cubic interpolation into a standardised size of 112×112 pixels.

EchoNet-dataset:

This publicly available dataset [198] contains 10,030 A4C echocardiography videos from individuals who underwent imaging between 2016 and 2018 as part of routine clinical care at Stanford University Hospital. Each video has been cropped and masked to remove text and information outside of the scanning sector.

The image sequences are provided with a dimension of 112×112 pixels. The videos are annotated by a registered sonographer. Although some videos may contain a couple of heartbeats, only one pair of ED/ES frames is labelled and were used as the reference ground-truth for testing the developed models (no training was performed using this dataset). A more detailed description of the EchoNet-dataset can be found in [199].

Table 6.1: A summary of the patient datasets used in this study.

Dataset Name	**PACS-dataset**	**MultiBeat-dataset**	**EchoNet-dataset**
Source	Private NHS Trust PACS Archives - Imperial College Heathcare	Private St Mary's Hospital Acquired for this study	Publicly available Stanford University Hospital echonet.github.io/dynamic
Ultrasound machine	Philips Healthcare (iE33 xMATRIX)	GE Healthcare (Vivid.i) and Philips Heathcare (iE33 xMATRIX)	Siemans Healthineers (Acuson SC2000) and Philips Healthcare (iE33, Epiq 5G, Epiq 7C)
Number of videos	1,000	40	10,030
Length of videos	1-3 heartbeats	≥ 10 heartbeats	1 heartbeat
Ground-truth	2 annotations by 2 experts	6 annotations by 5 experts (twice by one expert)	1 annotation
Original size (pixels)	(300-768)x(400-1024)	422x636	112x112
Frame rate (fps)	23-102	52-80	50
Format	DICOM	DICOM	AVI
Use	Training/Testing	Testing	Testing

6.4.2 Ground-Truth Definition

The target output, or ground-truth, was generated using reference annotations provided by experts and subsequently used to train the deep learning models.

Treating the definition of ground-truth as a classification task, with three classes for frames (ED, ES, trivial), would result in an imbalanced problem since the 'trivial' class would be greatly over-represented. A recent study put forth the argument of a binary classification approach for cardiac phase detection [192]. However, by allocating the same label to all frames in the diastole phase (1) and systole phase (0), one risks ignoring high-level spatial and temporally related markers, including crucial physiological differences throughout the entire cardiac cycle.

Therefore, the problem was formulated as a regression task. To label individual cardiac frames, it was assumed the predictions for a cardiac sequence should decrease during the systole phase and increase during the diastole phase. Given two consecutive ground truth labels y_t and $y_{(t-1)}$, we expect $y_{(t-1)} < y_t$ in systole, and vice versa. Assigning the target values of 1 and 0 to ED and ES time-points, respectively, and using a linear interpolation function, the target output was defined as:

$$aaFD = \begin{cases} \frac{f_t - f_{ED}}{f_{ES} - f_{ED}}, & \text{in systole phase} \\ \frac{f_t - f_{ES}}{f_{ED} - f_{ES}}, & \text{in diastole phase} \end{cases}$$

Here, y_t is the ground-truth label for frame f_t at time-point t and f_{ED} and f_{ES} are the frame numbers for ED and ES events, respectively. Due to varying video lengths, some contain a combination of singular or multiple events in the image sequence.

6.4.3 Neural Network Architecture

Figure 6.3 provides an overview of the network architecture. The model comprises (i) CNN unit for the encoding of spatial information for each frame of an echocardiographic video input, (ii) RNN (LSTM) units for the decoding of complex temporal information, and (iii) a regression unit for the prediction of the frames of interest.

Spatial feature extraction: First, a CNN unit is used to extract a spatial feature vector from every cardiac frame in the image sequence. A series of well-established, state-of-the-art architectures were employed for the CNN unit. These include ResNet50, InceptionV3, DenseNet and InceptionResNetV2, details of which can be found in the relevant resources [124, 162, 200, 201].

Temporal feature extraction: LSTM units are used to process the image features extracted from the entire image sequence by the CNN. Stacks of LSTM units (1-layer to 4-layers) were explored, where the output of each LSTM unit not in the final layer is treated as input to a unit in the next.

Regression unit: Finally, the output of the LSTM unit is regressed to predict the location of ED and ES frames. The model returns a prediction for each frame in the cardiac sequence (timestep).

6.4.4 Deep Learning Framework

For the model to be capable of processing a video input of arbitrary length, thus containing any number of heartbeats and events, a sliding window approach was adopted. As illustrated in Fig.5B., a sliding window with a fixed stride segments the cardiac image sequence into overlapping chunks of fixed length. Each segment is then fed into the neural network model, as described above, where a prediction vector p_k is returned. The final target output is computed as:

$$\hat{y}_t = \frac{1}{K}\sum_{k=1}^{K} P_{k,t}$$

Where $P_{k,t}$ is the prediction for frame t in the k^{th} segment, and K is the total number of predictions available for each frame, obtained from overlapping segments. A custom peak detection algorithm then searched for the local maxima and minima, representing the ED and ES frames, respectively.

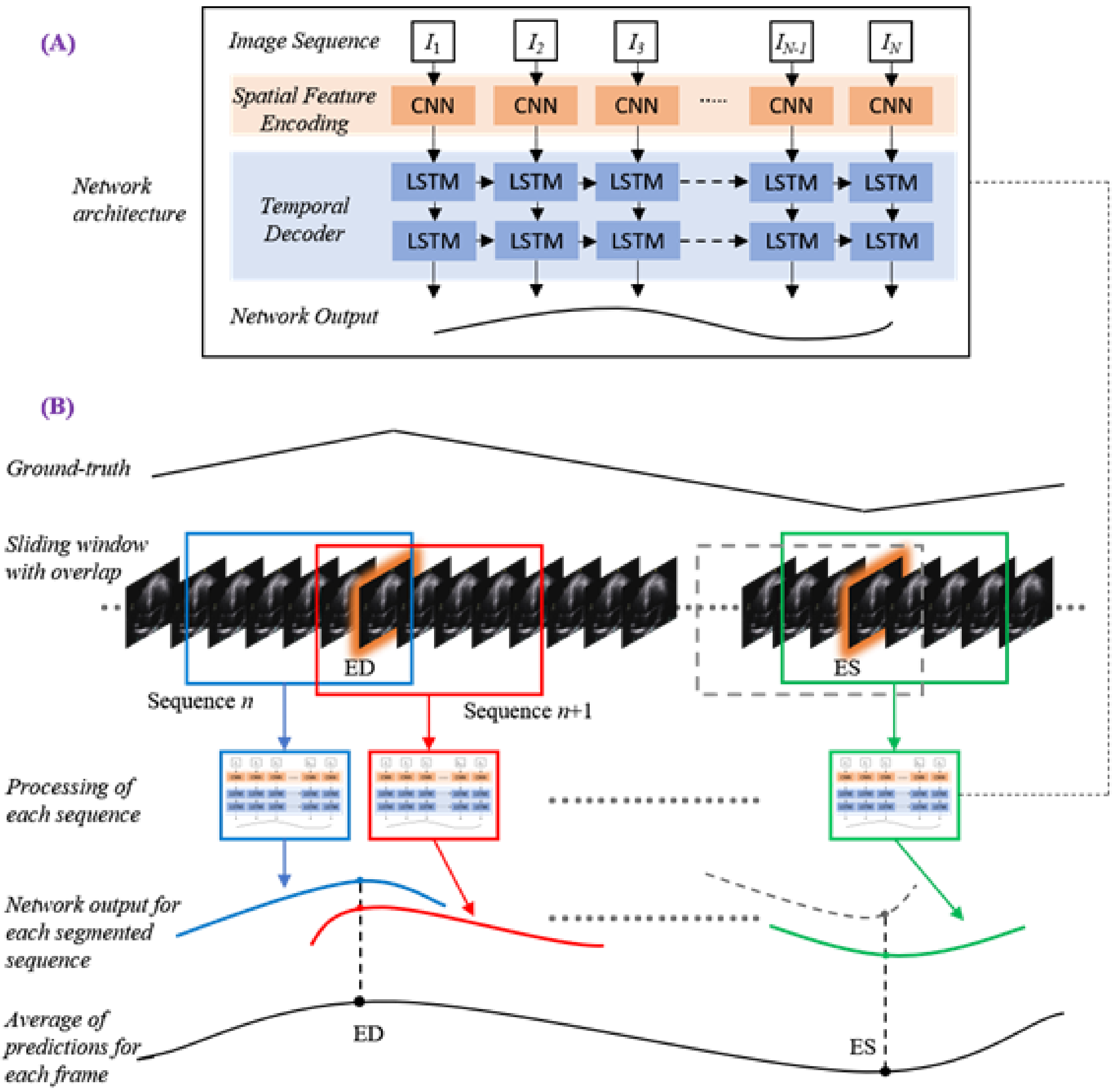


Figure 6.3: Detailed schematic of the proposed deep learning framework: (A) the network architecture combining a CNN unit for spatial feature extraction with RNN (LSTM) blocks for temporal analysis; (B) the sliding window method processing fixed, overlapped, chunked sequences, generating multiple predictions for each frame with the mean calculated for each.

As depicted in section A of Figure 6.3, the sequence is input to the CNN for spatial feature extraction. The shape of the input is (30, 112, 112, 3), representing (number of image frames/time steps, height, width, channels), respectively. The CNN is ResNet50 trained using ImageNet weights. To extract a feature vector for each time step, retaining the temporal sequence, ResNet50 is wrapped in the Keras Time Distributed layer. The Time Distributed wrapper applies the CNN to every temporal slice of the input, outputting a feature vector from the final average pooling layer with the shape (30, 2048) corresponding to the number of time steps and features, respectively.

Subsequently, the entire 30 frame sequence is input to an LSTM cell with 30 units; one for each time step. As output, the LSTM cell returns the full sequence to a dense layer with ReLU activation, reducing the dimensionality of the output space to 512 features. Next, a second, 30 unit, LSTM cell is applied to the sequence with a dropout of 0.5, similarly returning the full sequence as the output. The vector is subsequently flattened and input to a dense layer with an output dimensionality of 30; one for each of the original time step, retaining the temporal sequence relationships.

As illustrated in section B of Figure 6.3, inference is performed for each 30 frame "chunk", with a stride of 1 frame. Where necessary, blank frames are padded to the end of the final video chunk so as to create a uniform 30 frame sequence. Predictions for each chunk are appended, in sequence, to a matrix for which the mean for each time step in the entire video is calculated. Finally, the custom peak detection algorithm to detects frames (time steps) representing ED (maxima) and ES (minima) making full use of all spatial and temporal information.

6.4.5 Implementation Details

The models were implemented using the TensorFlow 2.0 deep learning framework [143] and trained using an NVIDIA GeForce GTX 1080 Ti GPU. Random, on the fly augmentation prevented overfitting, such as rotating between -10 and 10 degrees and spatial cropping between 0 and 10 pixels along each axis. The loss function was mean squared error (MSE) with Adam optimiser [163] initialised with a learning rate of 10^{-5}. Throughout the study, training was conducted over 70 epochs with a batch size of 2 for all models.

The PACS-dataset was used to train the models, with a data split of 60%, 20% and 20% for training, validation and testing, respectively. Early stopping was employed to avoid overfitting meaning training continued until the validation loss plateaued.

During testing, a sliding window of 30 frames in width with a stride of one was applied, allowing up to 30 predictions of differing temporal importance to be calculated for each timestep. Toward the end of each video, should a segment be fewer than 30 frames in length, it was zero-padded with the added frames removed after completion. Experimentation proved a stack of 2 LSTM layers was the optimum configuration across all models.

6.4.6 Evaluation Metrics

Evaluation of trained network predictions measures the difference between each labelled target y_t, either ED or ES, and the timestep prediction $\hat{y}_t$. Average Absolute Frame Difference (aaFD) notation is applied, where N is the number of events within the test dataset:

$$aaFD = \frac{1}{N}\sum_{t=1}^{N} |y^t - \hat{y^t}|,$$

The mean (μ) and standard deviation (σ) of the error (i.e. frame differences) were also calculated.

6.5 Results & Discussion

6.5.1 PACS-dataset

The average time (mean±SD) taken by the operators to manually annotate ED/ES frames was 26±11 seconds, per event. The equivalent time for our automated models, executed on the GPU, was less than 1.5 seconds; significantly faster than the human-led process.

Table 6.2 details the error in ED and ES frame detection for all videos in the PACS-dataset. The results indicate the level of disagreement between Operator-1 annotations, considered as the ground-truth, compared with automated predictions and those made by Operator-2.

Of all architectures explored, 'ResNet + 2x-LSTM' demonstrates the smallest discrepancy with Operator-1. The aaFD was less than one frame in both events, with a mean difference of -0.09±1.10 and 0.11±1.29 frames for ED and ES events, respectively.

The discrepancy between Operator-1 and Operator-2 indicates a level of inter-observer variability; with an average absolute (and mean) frame difference of 1.55 (-1.35±1.31) and 1.44 (-0.90±1.80) frames for ED and ES events, respectively. Therefore, suggesting the discrepancy between automated models and Operator-1 is within the range of disagreement observed between two trained human operators.

Table 6.2: Errors in ED and ES frame detection between Operator-1, the reference ground-truth, and predictions with Operator-2, for all testing videos in the PACS-dataset. Detection time is the average time it takes for the model (inference time) or the operator (annotation time) to identify an ED/ES event. The best performing architecture is highlighted.

Model/operator	ED			ES	Detection time (s)
	aaFD	$\mu \pm \sigma$	aaFD	$\mu \pm \sigma$	
ResNet50 + 2x-LSTM	**0.66**	**-0.09±1.10**	**0.81**	**0.11±1.29**	**0.776±0.33**
InceptionV3 + 2x LSTM	1.19	0.48±1.89	1.21	0.66±1.76	0.697±0.30
DenseNet + 2x LSTM	0.81	0.19±1.30	0.98	-0.01±1.53	1.379±0.59
InceptionResNetV2 + 2x LSTM	0.77	-0.02±1.38	0.83	0.23±1.29	1.07±0.46
Operator-2 (inter-observer)	1.55	-1.35±1.31	1.44	-0.90±1.80	26±11

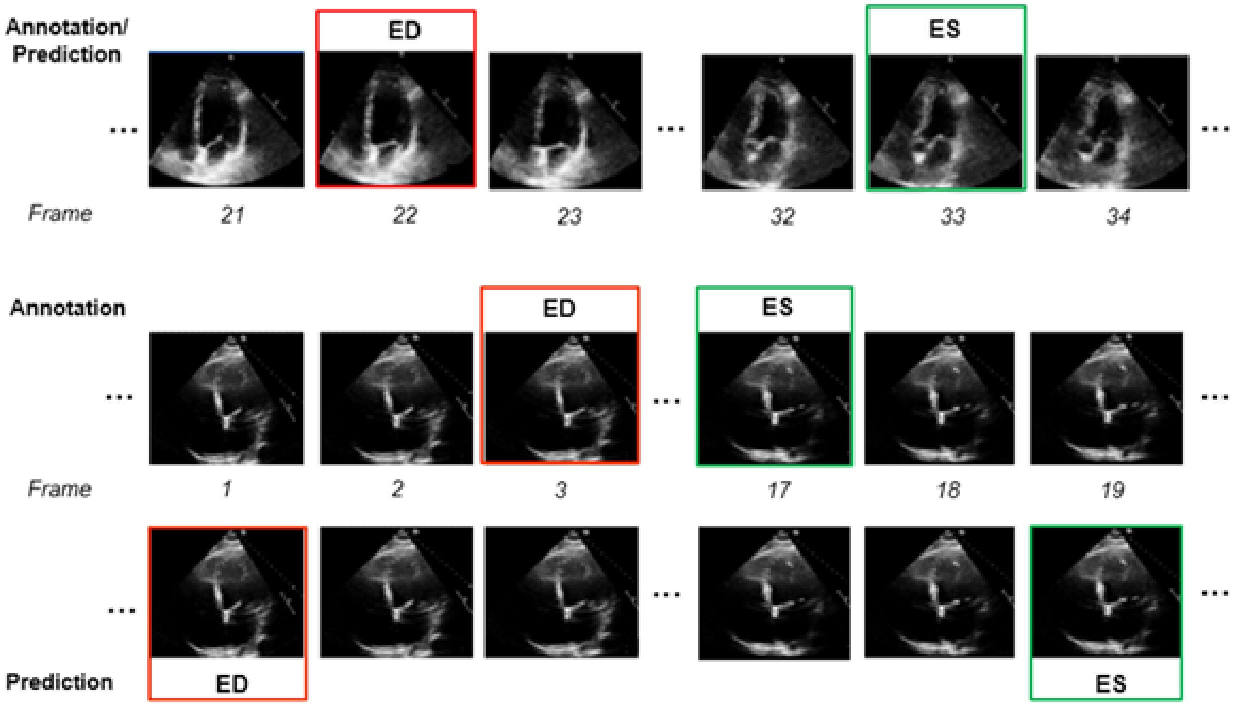


Figure 6.4: Illustrates model frame predictions and Operator-1 annotations for two arbitrary patients from the PACS-dataset test set and demonstrates typical examples where there is full agreement and conversely, when there is a mismatch.

Due to its lowest error and shortest inference time, the 'ResNet + 2x-LSTM' architecture (hereinafter, referred to as the model) was selected for further analysis using the additional MultiBeat and EchoNet datasets. Table 6.3 provides a comparison between the performance of the model and previously reported deep learning results. Figure 6.4 illustrates model frame predictions and Operator-1 annotations for two examples: one with full agreement and one with a mismatch.

The model outperforms almost all existing approaches, indicating smaller discrepancies with the ground-truth from which it has learnt. However, caution is necessary, as different studies have used different private patient datasets, presumably with various levels of image quality and experience of human experts for annotations. Therefore, a direct comparison between the reported accuracies may not be as informative as desired. However, the proposed model's removal of all pre-processing steps and its capacity to identify multiple heartbeats in one long video is, however, an indisputable advantage.

It is also observed that ES frame detection error is consistently higher in all models than that for ED. Potentially owing to minute differences in consecutive frames indicating the mitral valve opening as the onset of the diastole phase is less apparent in the images; thus, resulting in a more challenging detection task for the model.

Table 6.3: Comparison of the proposed model with previously reported deep learning architectures regarding aaFD in ED and ES event detection.

Model/Operator	aaFD ED	aaFD ES
ResNet50 + 2x-LSTM	0.66	0.81
ResNet + 2x-LSTM [191]	3.7	4.1
3D CNN + LSTM [192]	1.6	1.7
DenseNet + 2x-Bi-GRU [96]	1.6	1.7

6.5.2 Multibeat-dataset

An ECG signal was recorded simultaneously alongside image acquisition for the MultiBeat-dataset and appears as a transverse trace on the echo image sequence. The ECG was extracted using a combination of constraints where the trace was assumed to be (i) continuous, (ii) have a consistent colour profile, and (iii) distinct from the background. The extracted signal for a random patient is used in Figure 6.5 to plot the identified frames by the human operators (6 annotations) and the automated model.

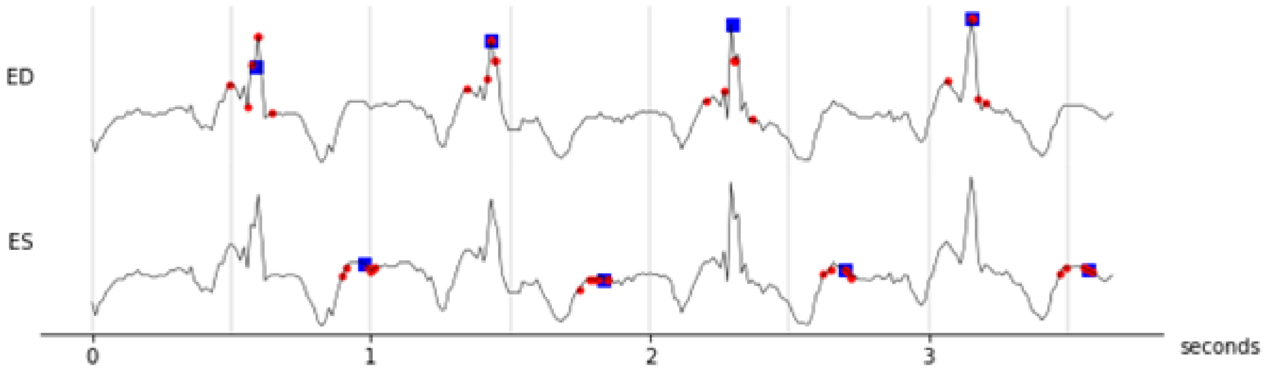


Figure 6.5: Extracted ECG trace spanning 4 heartbeats for a random patient, delineated showing the 6 annotations from 5 operators (red circles) and automatically identified (blue squares) ED and ES frames.

Table 6.4 details detection errors between Operator-1 and detections made by the model and other operators. The model disagrees with Operator-1, as do Operators 2-5. Indeed, Operator-1 disagreed with themselves on their second annotation attempt (denoted as Operator-1b). The smallest error was the discrepancy between the two annotations on separate occasions by the same operator (i.e. intra-observer variability), with a mean difference -0.22 ± 2.76 and 0.25 ± 3.75 for ED and ES events, respectively.

The range of mean difference between two different operators (i.e. inter-observer variability) was $[-0.87, -5.51] \pm [2.29, 4.26]$ and $[-0.97, -3.46] \pm [3.67, 4.68]$ for ED and ES events, respectively. The model discrepancy falls within the range of inter-observer variability. Clearly demonstrating the reliability of the model in frame detection, compared with the experienced human experts.

Significantly, both intra- and inter-observer variability measures suggest the experts' disagreement is greater when identifying ES frames. This is consistent with the model's performance, for which higher errors are observed when detecting ES frames.

Table 6.4: Errors in ED and ES frame detection between Operator-1a (considered as ground-truth) and predictions made by the other operators and the model for all testing videos in the MultiBeat-dataset. Operator-1b denotes the second set of annotations by the first human operator, indicating intra-observer variability.

Model/operator	ED		ES	
	aaFD	$\mu \pm \sigma$	aaFD	$\mu \pm \sigma$
Operator-1a vs Operator-1b	1.96	-0.22±2.76	1.90	0.25±3.75
Operator-1a vs Operator-2	2.65	-1.22±4.26	3.67	-2.25±4.68
Operator-1a vs Operator-3	5.82	-5.51±3.77	4.80	-4.46±3.77
Operator-1a vs Operator-4	1.72	-0.87±2.29	2.01	-0.97±3.48
Operator-1a vs Operator-5	3.27	-2.96±2.57	4.11	-3.64±3.67
Operator-1a vs model	2.62	-1.34±3.27	1.86	-0.31±3.37

To ensure fair comparison between model performance and operators, Figure 6.6 plots detection errors. Each human operator is compared with other 5, their consensus (mean) is considered as the reference annotation (red boxplots). The model is also compared with the consensus of the same 5 human annotations (blue boxplots). All 12 panels suggest performance of the model is similar, if not better, to that of an individual operator when using the other operators as a reference standard.

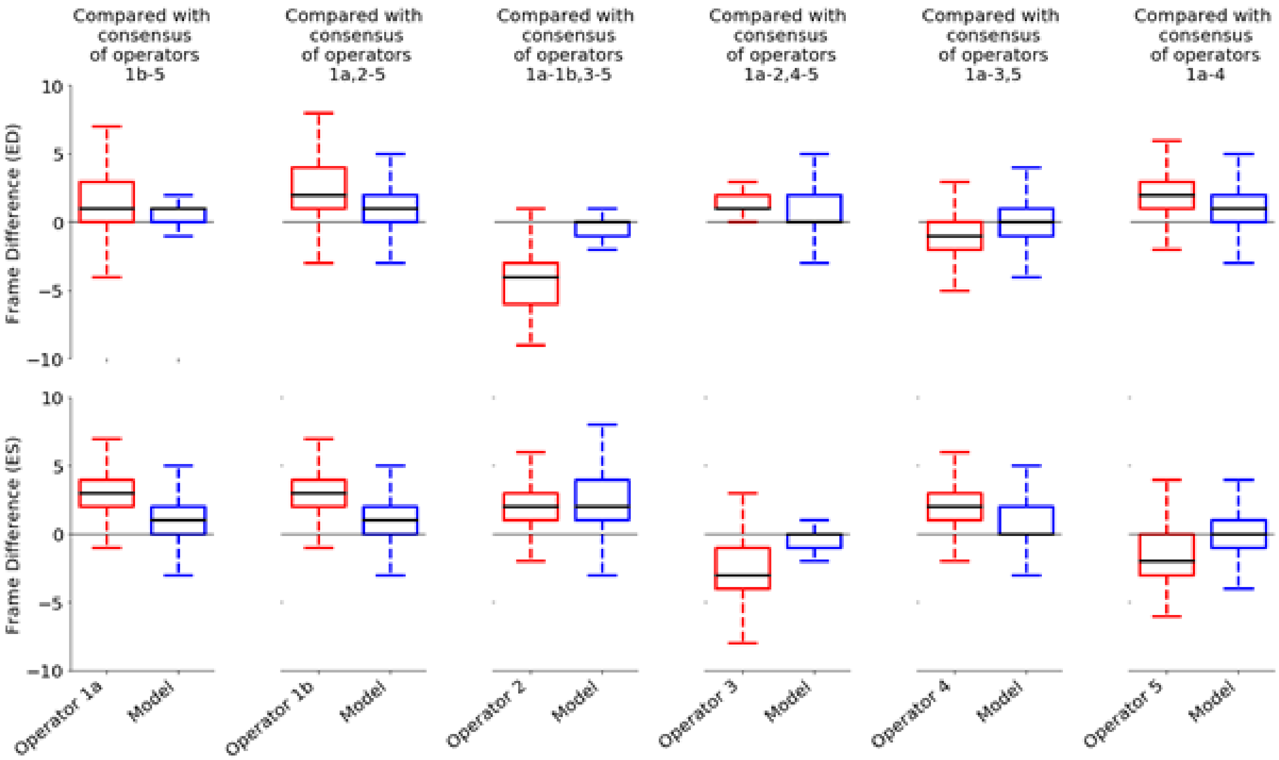


Figure 6.6: Errors in ED and ES frame identification by each operator, expressed relative to the consensus (mean) of all other 5 human annotations (red boxplots). In each case, alongside these errors, are those identified by the model expressed relative to the consensus of the same 5 annotations (blue boxplots). In the box-and-whisker plots, the thick line represents the median, the box represents the quartiles, and the whiskers represent the 2.5% and 97.5% percentiles.

Because different human experts make different judgments, it is not possible for any automated model to agree with all expert annotations all the time. However, it is desirable for automated models to have fewer discrepancies when compared with the performance of human judgment. Given the model was never exposed to this dataset (image sequences, and any of the corresponding annotations), its predictions in ED and ES frame detection can be treated as one of the independent assessors.

Hence, for each heartbeat, there were 7 assessments of the desired frame: 6 human and one automated. Therefore, for each assessor, 6 frame differences were calculated when compared to other human or automated assessors. The pool of these differences across all heartbeats and image sequences indicates the overall performance for each assessor and is shown as boxplots in Figure 6.7

Operator-4 demonstrates the smallest range of discrepancies in identification of ED frames (standard deviation of 3.47), but was consistently late, with a bias of −1.50 frames when compared to the consensus of other assessments.

The model had a relatively acceptable discrepancy from the consensus of the human operators, with a mean difference of 0.39 ± 3.97 and 1.54 ± 3.80 frames in ED and ES events, respectively. Indicating the model can be used to detect the frames of interest and that it is as reliable as the experienced human experts.

The range of human operator judgments for each heartbeat (i.e. difference between the earliest and latest manually identified frames) may be assumed as the uncertainty of the reference method and, therefore, the highest accuracy obtainable. The mean frame intervals among all heartbeats was 8.10 ± 3.84 and 7.01 ± 4.28 frames for ED and ES events, respectively.

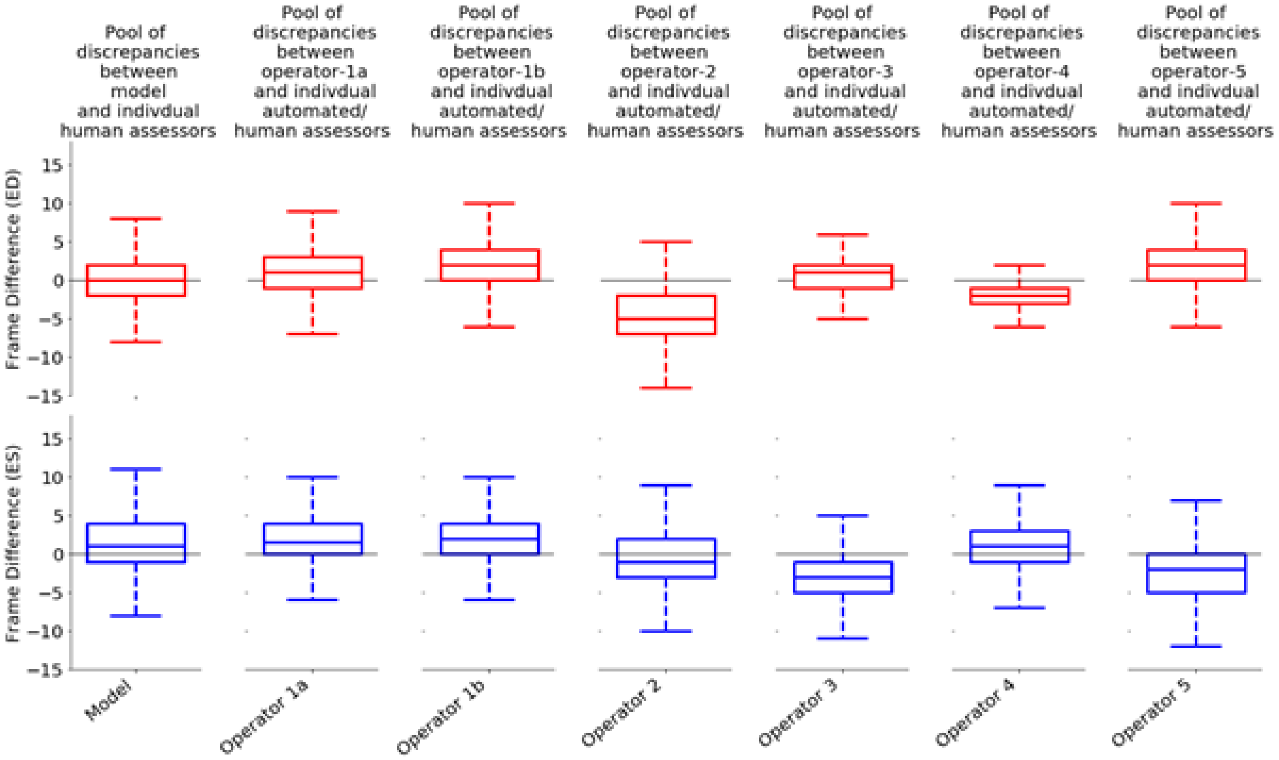


Figure 6.7: Errors in ED and ES frame identification by each of the assessors across all heartbeats and all patients. For each heartbeat there were 7 assessments (6 human and one automated). Errors are expressed as the pooled data from frame differences between each individual assessor and the 6 others. In the box-and-whisker plots, the thick line represents the median, the box represents the quartiles, and the whiskers represent the 2.5% and 97.5% percentiles.

6.5.3 EchoNet-dataset

The proposed model has been compared against previously reported approaches. However, each study used a different private dataset, making a direct comparison extremely difficult. Here, the proposed model is applied to the publicly available EchoNet-dataset, allowing for future studies to be benchmarked against it. Like the MultiBeat-dataset, no further training was carried out, and the dataset was used in its entirety for testing.

From the total number of videos (10,000), 810 were excluded owing to one of the ED or ES events occurring in the penultimate or final frame in the video, hence being unsuitable. EchoNet was made available for a challenge focused on segmentation of the left ventricular. Therefore, it was acceptable to have ED or ES events occurring in first or last frames. The retained 9,190 videos were fed into the model, when no resampling of the images was required as the dataset is provided with a resolution of 112×112 pixels; identical to the input size of the model.

An aaFD of 2.30 and 3.49 frames was obtained for ES and ES events, respectively and the mean frame difference was 0.16 ± 3.56 and 2.64 ± 3.59 for ED and ES; well within the range of inter-observer variability observed.

6.6 Conclusion, Publications and Future Work

6.6.1 Conclusion

This research project seeks to investigate the feasibility of fully automated identification of ED and ES frames derived from 2D echocardiographic images and independent from an accompanying ECG signal using deep neural networks. The performance of the proposed method was examined by comparisons to gold standard reference data, obtained from multiple cardiologist experts. It has been demonstrated that the performance of the proposed model is like that of human experts, with its detection error falling within the range of inter-observer variability and can therefore be used to reliably identify multiple ED and ES frames from videos of arbitrary length.

Furthermore, the performance of the automated model, measured as the processing time, is superior to that of human operators, where an improvement of ¿20 times was observed.

The proposed framework was tested on A4C views; however, it is believed that the utilised deep learning approaches could be applied to other echocardiographic views. This will be the subject of future work. As in previous studies, this research investigates 2D echocardiography as the clinically relevant modality. Currently, 3D echocardiography suffers from a considerable reduction in frame rate and image quality, hindering its adoption into routine practice [93]. When such issues are resolved, automatic frame detection in 3D images could be explored. Meanwhile, 2D echocardiography remains unrivalled, particularly when high frame rates are required.

Interpreting the results of the proposed model alongside other published architectures from the literature was not feasible. A direct comparison of detection accuracy would require access to the same patient dataset. At present, no echocardiography dataset, and corresponding annotations specifically prepared for cardiac phase detection, is publicly available. Additionally, representative multi-centre patient data, essential for ensuring any developed model would scale up well to other sites and environments, is currently scarce.

6.6.2 Publications

Journal

Lane, E., Azarmehr, N., Jevsikov, J., Howard, J.P., Shun-shin M. J., Cole, D. G., Francis, D.P. and Zolgharni, M., 2021. Multibeat Echocardiographic Phase Detection Using Deep Neural Networks. (under-review).

Conference

Lane, E., Azarmehr, N., Jevsikov, J., Howard, J., Shun-shin, M., Cole, G., Francis, D. and Zolgharni, M., 2021. Echocardiographic Phase Detection Using Neural Networks. In: *Medical Imaging with Deep Learning.*

6.6.3 Future Work

The main aim of this Chapter is to develop a deep learning model capable of automatically and reliably detecting end-diastolic and end-systolic frames in echocardiographic image sequences. The research conducted to date, and detailed in this report, fulfils this aim. However, there is a great deal more to accomplish in terms of developing, refining and improving the accuracy of the proposed deep learning model.

As previously stated, this research has focussed on one echocardiographic image view (A4C). However, in clinical practice many other views are considered then computing complex diagnostic markers relating to cardiac function. Figure 6.8 shows the difference between A4C and apical 2 chamber (A2C) images. In line with the research objectives outlined in section 1.3, specifically, to improve the accuracy and efficiency of the system after conducting initial experiments, future research will focus on expanding echocardiographic views.

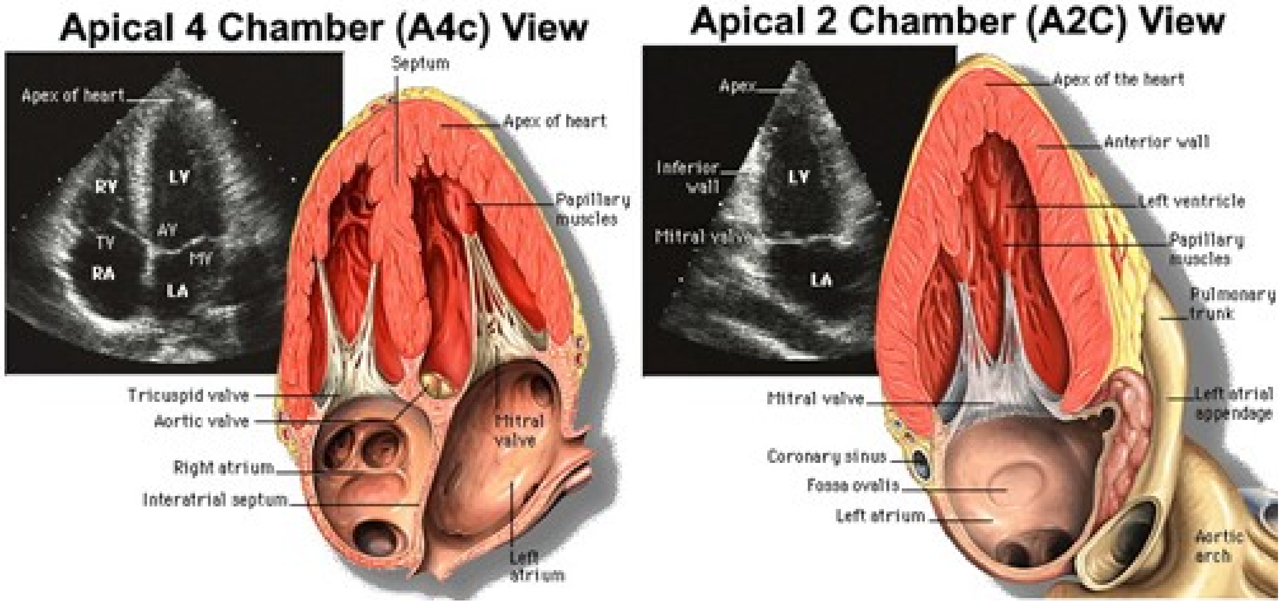


Figure 6.8: A comparison between A4C and A2C echocardiographic views [202]

The proposed model was trained upon data from one centre, but tested upon data from multiple centres, produced using a range of ultrasound machines. To improve the accuracy and clinical applicability of the network, it is proposed that an amalgamated dataset is formed being more representative of multi-centre data and thus more representative of current clinical practice. Additionally, the data is limited in that it was only collected from patients in sinus rhythm, therefore future data collection should include a more diverse selection of cardiovascular conditions to ensure accurate phase detection in a real-world clinical setting.

Chapter 7

Left Ventricular Volume and Ejection Fraction Estimation With Deep Neural Networks

7.1 Introduction

Two-dimensional (2D) echocardiography is routinely applied for the assessment of left ventricular (LV) ejection fraction (LVEF) and the measurement of LV systolic function [57]. LV volume and ejection fraction (EF) are essential metrics in cardiac diagnostics and require manual tracing of the endocardial border by trained experts, as illustrated in Figure 7.1. Alternative methods for the assessment of LVEF exist, however echocardiography remains the modality of choice due to its widespread availability and high temporal resolution. Furthermore, handheld ultrasound devices are increasingly being used to assess LVEF at the point of emergency care, by healthcare operators with varying degrees of expertise [159].

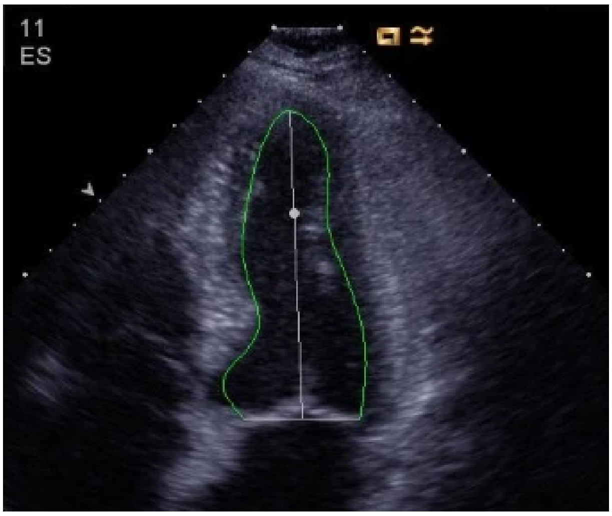


Figure 7.1: An example of manual tracing of the LV endocardial border by a trained expert [203]

LVEF quantifies the fraction of chamber volume ejected in the systole phase of the cardiac cycle, relative to the volume of blood in the ventricle at the end of the diastole phase. Stroke volume (SV) is calculated as the difference between end-diastolic volume (EDV) and end-systolic volume (ESV) [70]. The Simpson's method for calculating EF is widely used and recommended by the American Society of Echocardiography and the European Association of Cardiovascular Imaging [71, 72], though it relies on specialists to detect appropriate frames (end-diastole (ED) and end-systole (ES)) from echocardiographic videos and delineate the endocardium border of the LV. This process is time consuming and suffers from high levels of observer variability. Thus, fundamentally compromising the accuracy of routine LVEF assessment in clinical practice [57, 61, 62]. Therefore, automated algorithms are desired for accurate, objective, and efficient EF measurements.

7.2 Related Work

Automated and semi-automated segmentation of the endocardial border from 2D echocardiographic images has been the focus of multiple studies [204]. Many approaches have demonstrated their ability to out-perform the speed and precision of manual processes by human experts. Prior to the increasing popularity of Deep Learning (DL) methods, Machine Learning (ML) yielded results in close agreement with human operators [9, 26, 28, 29]. However, a major limitation of such approaches is the requirement of significant feature engineering, or prior knowledge, to achieve satisfactory accuracy[205]; thus, limiting potential application in clinical practice.

Gradually, DL approaches for segmentation tasks have outperformed state-of-the-art ML methods. Deep convolutional neural networks (CNNs) have shown great promise due to their speed and accuracy in tasks such as echocardiographic view classification (discussed in Chapter 4), image quality assessment (Chapter 5), ED and ES phase detection (Chapter 6) and when processing images with sub-optimal resolution [199, 206].

CNN approaches based upon the symmetrical, encoder-decoder U-Net architecture [30] have demonstrated high performance and accuracy without the need for manual intervention by a trained operator [61]. Current state-of-the-art methods leverage U-Net, or it's 3D equivalent [207], as a backbone [31–34]. U-Net is widely acknowledged as producing high accuracy in medical image segmentation tasks, due to skip connections passing spatial information between the encoder and decoder, ensuring no context is lost in the down-sampling process. Additionally, research has shown the combination of U-Net and long short-term memory (LSTM) improves segmentation accuracy and is robust to fluctuations in image quality throughout a sequence, as opposed to one single frame [208].

It is undeniable that studies reported in literature demonstrate neural networks have rapidly evolved in both sophistication and accuracy for the automation of computer vision tasks, especially in the medical imaging domain. A significant contributing factor being the publication of large echocardiographic datasets; allowing for transparent evaluation of results and comparison between studies. Two major clinical datasets for LV segmentation are: CAMUS (Cardiac Acquisitions for Multistructure Ultrasound Segmentation) [29] and EchoNet-Dynamic [199] from the Stanford University Center for Artificial Intelligence in Medicine & Imaging [209].

The results presented in this Chapter show volume is a more accurate measure of accuracy than the established Hausdorff distance and Dice coefficient metrics. Additionally, the automated phase detection network presented in Chapter 6 is used to predict ED and ES frames from the EchoNet-Dynamic dataset and demonstrates generalisability of the network due to close alignment with expert annotations, especially when comparing volume estimation.

7.3 Main Contributions

This chapter details the application of three independent datasets (two public and one private) for segmenting the LV endocardial border and proposes a novel method of volume estimation based on the Simpson's rule. The main contributions are as follows:

- comparing the results of three, large segmentation datasets for assessing LVEF using commonly applied evaluation methods and proposing a novel method of calculating volume as a more informative metric, based upon the Simpson's rule
- investigating the effect upon EF calculations based upon the selected ED and ES frames, comparing expert ground-truth annotations with the phase detection network detailed in Chapter 6
- identifying the importance of data pre-processing strategy in improving model performance, paying particular attention to generalisability

7.4 Methodology

7.4.1 Dataset, Ethics and Expert Annotations

Descriptions of the three datasets used in this study are as follows, with a summary provided in Table 7.1:

Unity Dataset:

The Unity dataset consists of 1,224 apical 4-chamber (A4C) view echocardiographic videos acquired using ultrasound equipment from GE and Philips manufacturers between 2015 and 2016 from Imperial College Healthcare NHS Trust's echocardiogram database. Examinations were performed by experienced echocardiographers following standard protocols. Ethical approval was obtained from the Health Regulatory Agency (Integrated Research Application System identifier 243023).

In total, 2,600 still images were extracted from the videos at various points in the cardiac cycle. Images were labelled by one human expert, from a pool of available experts, using the proprietary Unity Imaging labelling application [173]. The manually labelled endocardium was provided in the form of control points which represented a B-Spline curve that constitutes the LV cavity boundary.

A separate test dataset, comprising 100 videos, was obtained from consecutive echocardiographic examinations conducted over a 3 day period in 2019; over 3 years after the acquisition of the training and validation dataset. The echocardiographic phase detection network forming Chapter 6 of this thesis was used to identify ED and ES frames for each of the test videos - a total of 200 images (100 for ED and 100 for ES, respectively). These images were labelled by 11 independent human experts, again using Unity Imaging. Ground-truth labels were computed using the consensus of the experts' annotations.

All images were resized to 512 x 512 pixels with padding used to preserve the original aspect ratio.

CAMUS Dataset:

The publically available CAMUS dataset consists of apical 2-chamber(A2C) and A4C ED and ES frames extracted from 450 patient echocardiographic videos of varying pathologies. Examinations were conducted by trained experts using a GE Vivid E95 ultrasound machine and a wide variety of acquisition settings and image qualities were included to maintain clinical realism. This Chapter utilised the A4C images, not A2C, hence the entire dataset consisted of 900 images with a data split of 78%, 21%, 21% for train (700 images), validation and test datasets (100 images each), respectively.

The data is presented in Digital Imaging and Communications in Medicine (DICOM) format, individual frames were extracted and converted into .png format. Original image dimensions were between the range of $[388, 787]$ and $[778, 1297]$ pixels for height and width, respectively. All images were were resized to 512x512 pixels prior to training the deep neural network.

Annotations are provided for the LV endocardium, myocardium and left atrium. Annotations were performed by three trained experts, twice by one of the experts to asses intra-observer variability.

EchoNet-Dynamic Dataset:

The publically available EchoNet-Dynamic dataset consists of one A4C 2D gray-scale video extracted at random from 10,030 patients who underwent an echocardiogram at Stanford Health Centre between 2016 and 2018. Examinations were performed by skilled sonographers using iE33, Sonos, Acuson SC2000, Epiq 5G, or Epiq 7C ultrasound machines. Processed images were stored in a Philips Xcelera picture archiving and communication system. ED and ES frames were identified from each video by extracting the DICOM file linked to measurements of ventricular volume used to calculate the ejection fraction. All data was fully anonymised and unintended human labels were removed. The video frames ranged in size, either 600x600 or 768x768 pixels. All were down sampled by cubic interpolation using OpenCV into standardised 112x112 pixel videos. For each video 1 annotation for ED and another for ES is provided and performed by 1 trained human expert. For the purpose of this Chapter, and to be compatible with the U-Net implementation, the images were resized to 512x512 pixels.

The same exclusion criteria were applied to the dataset as in Chapter 4. Resulting in 8,950 videos with a data split of 60%, 20%, 20% for train, validation and test sets, respectively.

Table 7.1: A summary of the patient datasets used in this Chapter

Dataset Name	**Unity**	**CAMUS**	**EchoNet**
Source	**Private** NHS Trust PACS Archives - Imperial College Heathcare	**Public** University Hospital of St Etienne (France)	**Public** Stanford University Hospital
Ultrasound machine	Philips Healthcare (iE33, Affinity 70C, Epic 7C, Affinity 50G, CX50) and GE (Vivid: i, q, S70, S6, E9, 7)	GE Vivid E95	Siemans Healthineers (Acuson SC2000) and Philips Healthcare (iE33, Epiq 5G, Epiq 7C)
View	A4C	A2C/A4C	A4C
Number of videos/images	2587 images	500 images	10,030 videos
Length of videos	1-3 heartbeats	1 heartbeat	1 heartbeat
Ground-truth	2 annotations by 2 experts	4 annotations by 3 experts (twice by one expert)	1 annotation
Original size (pixels)	(422-636)x(768-1024)	549x778	112x112
Frame rate (fps)	N/A	N/A	50
Format	DICOM	DICOM	AVI

7.4.2 Neural Network Architectures

Echocardiographic Phase Detection Network

Detection of ED and ES frames from echocardiographic videos is a crucial first step, prior to the segmentation of the LV and subsequent volume calculations. A deep neural network for automated, multibeat ED and ES phase detection is presented in Chapter 6.

The phase detection model is comprised of a CNN unit for the encoding of spatial information for each frame of an echocardiographic video input, followed by an RNN, (LSTM units) for the decoding of complex temporal information. Finally, the output is regressed to predict the location of ED and ES frames. The model returns a prediction for each frame in the cardiac sequence (time step). The backbone for feature extraction is ResNet50 with ImageNet weights.

Segmentation Network

The U-Net architecture, further discussed in Chapter 3, was adopted for semantic segmentation tasks using biomedical imaging datasets. It is described as an encode-decoder architecture, as illustrated in Figure 7.2. The left portion of the diagram represents the encoder; a feature extractor, usually transfer learning using a pre-trained classification network such as a VGG or ResNet implementation. During encoding, convolutional blocks, followed by max pooling downsampling layers, encode the image into a feature vector representation at various stages.

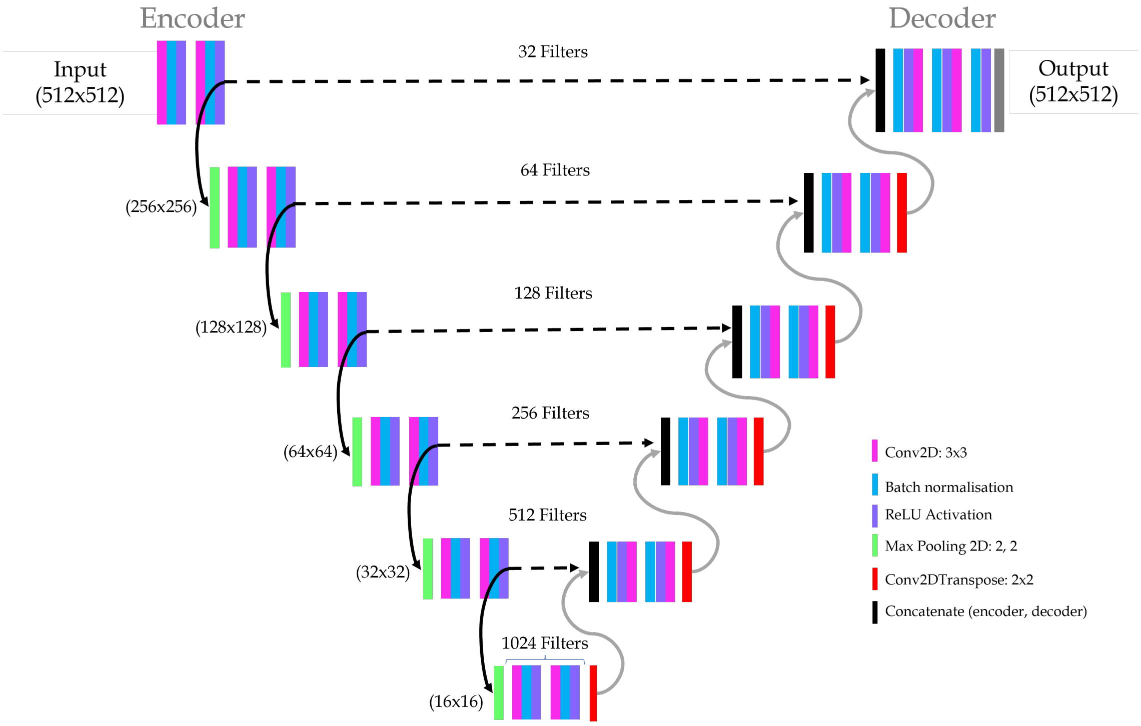


Figure 7.2: High-level illustration of the encoder (left) and decoder (right) U-Net architecture

During the decoder phase, on the right of Figure 7.2, the input is upsampled to restore the condensed feature vectors to the original image dimensions. Skip connections (represented by black horizontal dashed lines) concatenate higher resolution feature vectors from the corresponding encoder stage with upsampled features to better learn feature representations.

Unlike classification tasks, where the class predicted by the network is the end result, semantic segmentation discriminates between two distinct classes at pixel-level. Producing a predicted mask where the value of each pixel indicates whether the object of interest is located in that region or not.

The standard 2D U-Net implementation was applied for the segmentation of the LV in this chapter.

7.4.3 Implementation Details

All models were implemented using Tensorflow 2.0 [143] and Keras [144] DL frameworks. Training and inference was conducted on a server containing four NVIDIA GeForce RTX 3090 GPUs.

For the U-Net implementation the loss function was Tversky loss, the optimiser was Adam with a learning rate of $1e-5$ (monitoring the accuracy metric) and early stopping regularisation was applied with a patience of 10 epochs. The batch size was 8 with an image input size of 512x512 pixels and 1 channel.

Our recent study [27] investigates the influence of several popular loss functions for training the U-Net implementation for segmentation of the LV. Tversky loss [210] was optimal for the problem as it is designed to address the issue of class imbalance (in this case, binary classification of pixels either belonging to the region of interest, or not).

The echocardiographic phase detection network was also trained using the Adam optimiser, with a learning rate of $1e-5$ and the mean squared error (MSE) loss function. Again, early stopping was applied with a patience of 10 and a batch size of 4. The network converged after 259 epochs.

7.4.4 Ground-truth Definition

The echocardiographic phase detection model, forming Chapter 6, was trained using the PACS dataset. It was tested using EchoNet-Dynamic, to assess generalisabilty. Frame predictions for ED and ES were compared with the ground-truth to assess the effect different frames have upon the volume and EF metrics.

The phase detection model was then re-trained using the EchoNet-Dynamic dataset for comparison purposes. However, due to differences in the PACS and EchoNet-Dynamic datasets, some changes in the methodology and ground-truth definition were necessary.

In Chapter 6, a “sliding window” method for pre- and post-processing the dataset is described. This was possible for the PACS dataset due to annotations being provided for multiple consecutive beats in each video. The ground-truth annotations provided for the EchoNet-Dynamic dataset are for one beat, one ED/ES pair, only. Therefore, the sliding window approach was not possible. Instead, the videos were trimmed to 30 frames in length, encompassing both ED and ES frames. The method of generating labels for the frames of interest, and those in between, was a simple interpolation function, as described in Chapter 6.

The ground-truth definition for the segmentation of the LV border and calculation of EF were provided as annotations by trained experts for each dataset and were compared to the network predictions using several evaluation metrics, described below.

7.4.5 Evaluation Metrics

Echocardiographic Phase Detection

Evaluation of trained phase detection network predictions measures the difference between each labelled target y_t, either ED or ES, and the time step prediction $\hat{y}_t$. Average Absolute Frame Difference (aaFD) notation is applied, where N is the number of events within the test dataset:

$$aaFD = \frac{1}{N} \sum_{t=1}^{N} |y^t - \hat{y^t}|,$$

The mean (μ) and standard deviation (σ) of the error (i.e. frame differences) were also calculated.

Left Ventricular (LV) Segmentation

Quantitative evaluation metrics for the performance of automated segmentation algorithms generally fall into one of three categories:

- Volume-based: Dice coefficient, Jaccard similarity index
- Surface distance-based: Mean contour distance, Hausdorff distance
- Clinical performance: Ventricular volume and mass

This study reports the accuracy of segmentation methods using the Dice coefficient and Hausdorff distance metrics.

Dice coefficient is the measure of the ratio of overlap between the ground-truth and the predicted segmentation map and can be expressed as:

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

A complete mismatch is represented as 0, with a perfect match as 1. The closer to 1 the Dice coefficient score, the more accurate the prediction.

Average Hausdorff distance is a measure between the binary objects in two images. It is defined as the maximum surface distance between the two objects. More detailed information about segmentation evaluation metrics can be found in Chapter 3.

Volume and Ejection Fraction (EF)

A bespoke algorithm, based on Simpson's Rule, was created for measuring the volume of the LV from the ground-truth and predicted masks of the endocardium. The algorithm uses image processing techniques to draw a "cutting line" (extended in length) on the predicted mask in one colour and then fills the LV cavity with another. The intersection points are located by detecting the change in colour between the intersection line and the filled LV cavity using thresholding.

The algorithm accommodates irregularities in the shape and smoothness of the predicted mask. For example, when experts manually annotate the endocardial boundary, the bottom of the LV is a straight line, created by joining the start and end points (i.e. the two opposite sections of the mitral ring). The midpoint of this straight line is used to find the z-axis in order to compute the disks. However, for the predicted masks, the shape at the bottom of the LV is non-uniform and must be considered when finding the bottom midpoint, as shown in Figure 7.3 A and B.

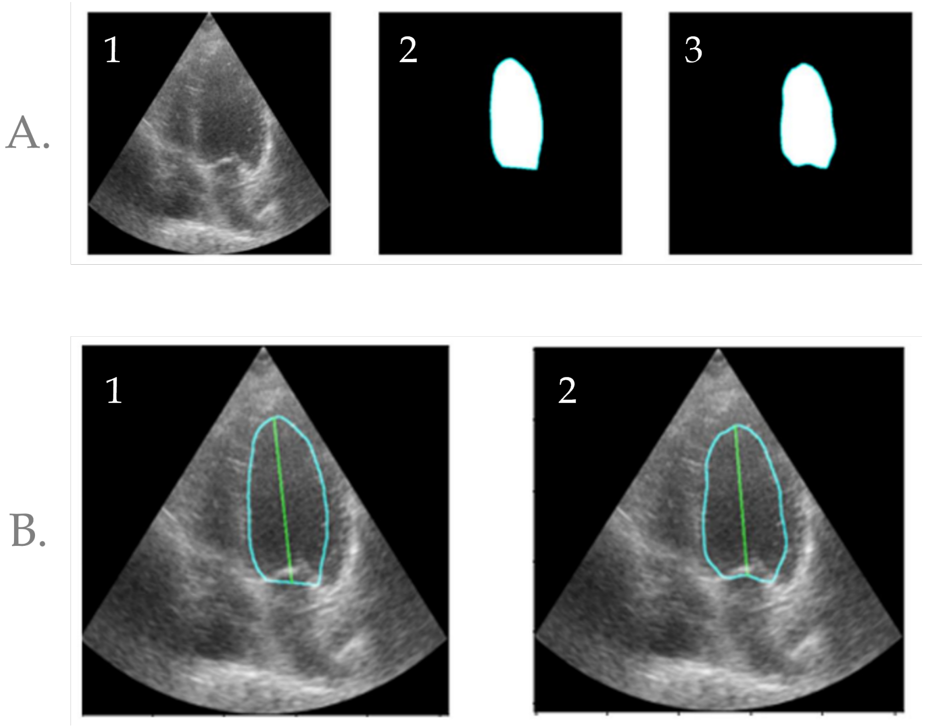


Figure 7.3: A. Automatically obtaining the endocardial borders: 1. the original scan, 2. ground-truth annotation mask and 3. predicted mask produced by the U-Net model. B. Automatic localisation of the bottom midpoint and centre line (z-axis). 1. ground-truth, 2. Prediction

Furthermore, the algorithm is capable of handling disconnected regions and any holes in erratically predicted masks, an example of which can be observed in Figure 7.4. The disconnected region is ignored in the volume computation and has no negative effect on the results.

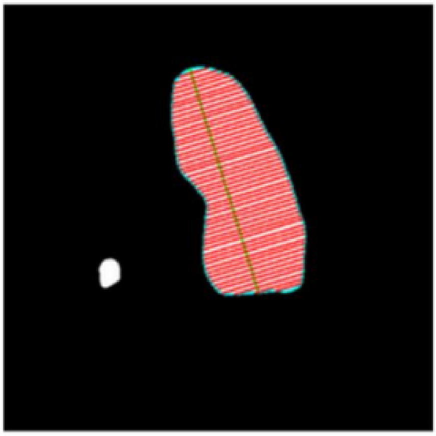


Figure 7.4: An example of handling disconnected regions from erratic predictions

The centre line, as in Figure 7.3 B, is defined as the line joining the bisector of the bottom line and the apical point of the LV contour, as recommended by the American Society of Echocardiography and the European Association of Cardiovascular Imaging [71, 72].

Given the nature of 2D echo images with regards to the potential high variability in image quality, it could be difficult for a DL model to distinguish the separation border between the left ventricle and left atrium. Separation border is defined as the straight line connecting to the two opposite sections of the mitral ring, spanning across the area where the MV is located, often the MV is not clearly visible in the echo image.

When echocardiographers manually delineate the endocardium, they use their discretion based on their expertise and experience to differentiate this separation border region when it is obscure. However, as shown in Figure 7.5, it is observed that this border region is unclear and could potentially result in erratic predictions in this portion of the predicted endocardium, since there is no clear distinct feature in the image that separates the LV and the left atrium.

Further observations from Figure 7.5 suggest that the other areas of the endocardium, besides the MV area between the LV and left atrium, are relatively more pronounced and could likely result in a more accurate prediction of the endocardium. Given an input echo image into the DL model for LV segmentation, the model will infer an output based on logic interpreted from the input. However, a DL model is unable to use human-like discretion when certain parts of the image are unclear; such as accurately defining the region where the left ventricle separates from the left atrium.

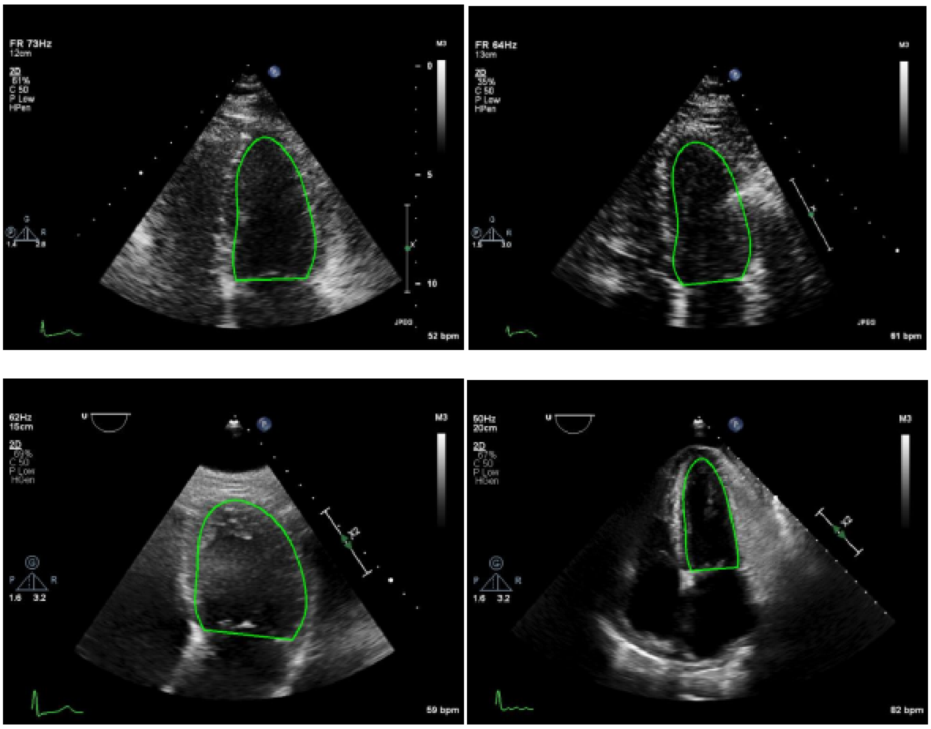


Figure 7.5: Four examples depicting the unclear separation between the left ventricle and left atrium. The average of 11 expert annotations are shown in green

This distortion could affect the performance results of DC and HD segmentation metrics.

As opposed to the recommended 20 discs, the volume is computed by dividing the LV cavity into K discs. This means that each cylinder has the same height which is computed by dividing the length of the centre line by K. For experimental purposes, $K = 50$, which produces a more accurate approximation, as illustrated in Figure 7.6.

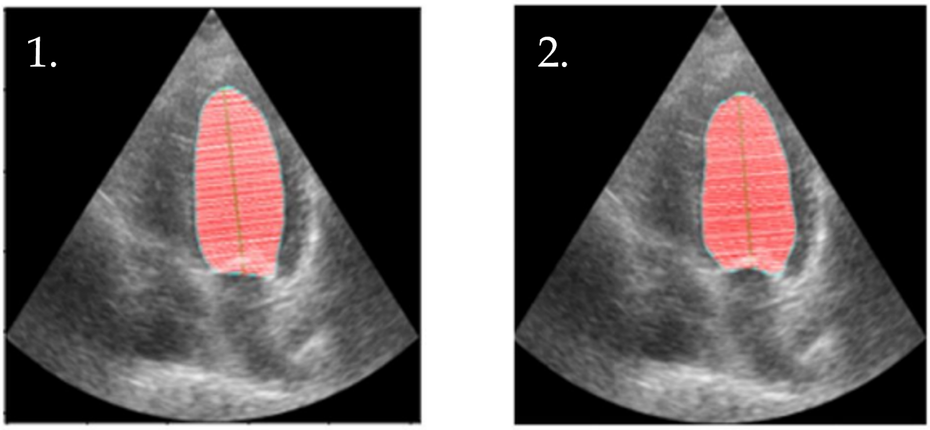


Figure 7.6: Automatic computation of disks with parameter $K = 50$. 1. ground-truth and 2. model prediction

EF typically refers to the left side of the heart and indicates the percentage of oxygen-rich blood pumped out of the LV during each cardiac cycle. LVEF helps to detect and determine the level of dysfunction on the left side of the heart.

The formula for EF is the amount of blood pumped out of the ventricle with each contraction (stroke volume (SV)) divided by the ED volume. EF was estimated by dividing the stroke volume (i.e. the difference between ED and ES volumes) by the ED volume:

$$EF(\%) = (\frac{EDV - ESV}{EDV})x100$$

The average EF error was computed by as the mean absolute error (MAE) between the ground-truth and predicted masks.

The average Volume Error is calculated using Cartesian pixel coordinates by computing the difference between the volume of the ground-truth and the predicted endocardial border and can be expressed as:

$$Vavg\epsilon = \frac{1}{m}\sum_{i=1}^{m} |Vgt_i - Vpred_i|,$$

7.5 Results & Discussion

7.5.1 Segmentation and LV Quantification

Table 7.2 shows the average results for LV segmentation using the U-Net implementation trained and tested on the three datasets: Unity, CAMUS and EchoNet-Dynamic. The Average volume error was computed using the bespoke algorithm previously described. The pixel spacing information was requested from Stanford University, however this information is confidential and could not be released. This information is necessary to convert volume from pixel coordinates to cm3/ml, without which it is impossible. Regardless, this has no effect on the EF measurements, since EF is a ratio and it is sufficient to use the raw pixel volume measurements for calculating EF.

Table 7.2: Average results for segmentation of the LV using the U-Net architecture trained and tested on each of the three datasets: Unity, CAMUS and EchoNet-Dynamic

Dataset	Avg EF Error	Avg DC	Avg HD
Unity	6.96	0.93	4.41
Camus	6.79	0.92	5.05
EchoNet-Dymanic	7.94	0.86	4.86

Figures 7.7 and 7.8 provide Bland-Altman plots for the Unity dataset expert ground-truth annotations and trained network predictions for the proposed volume estimation method and EF calculations.

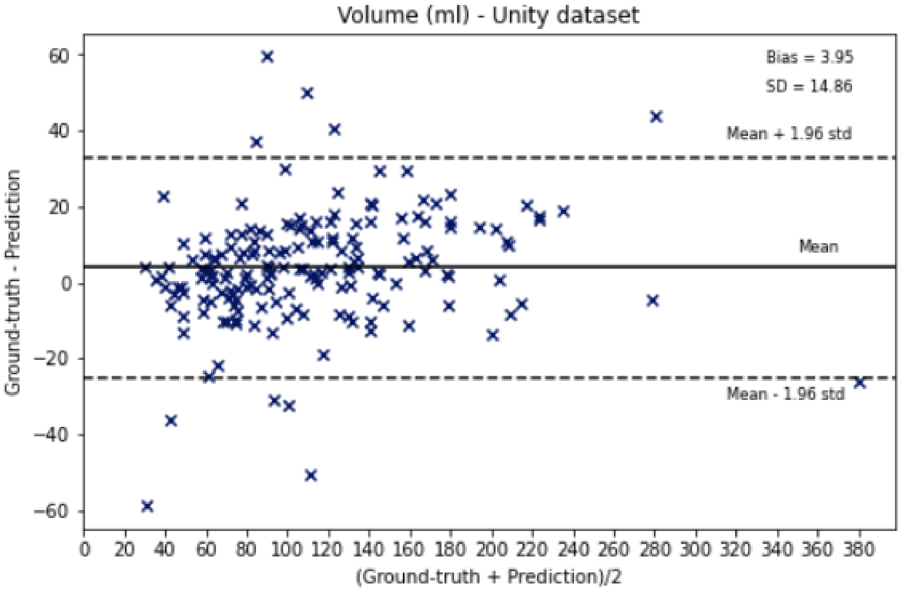


Figure 7.7: Bland-Altman plot for the Unity dataset expert ground-truth annotations and trained network predictions for Volume (ML) calculations using the bespoke method detailed in this Chapter

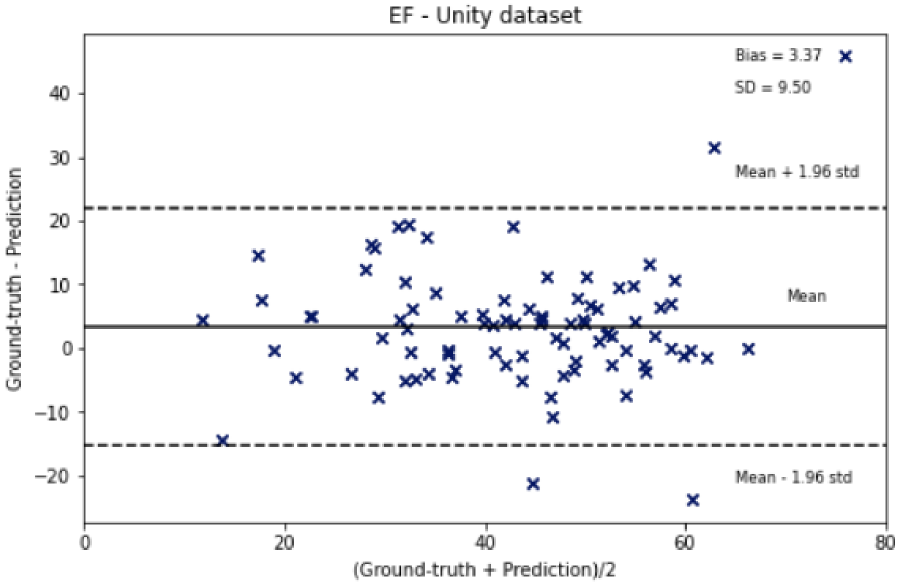
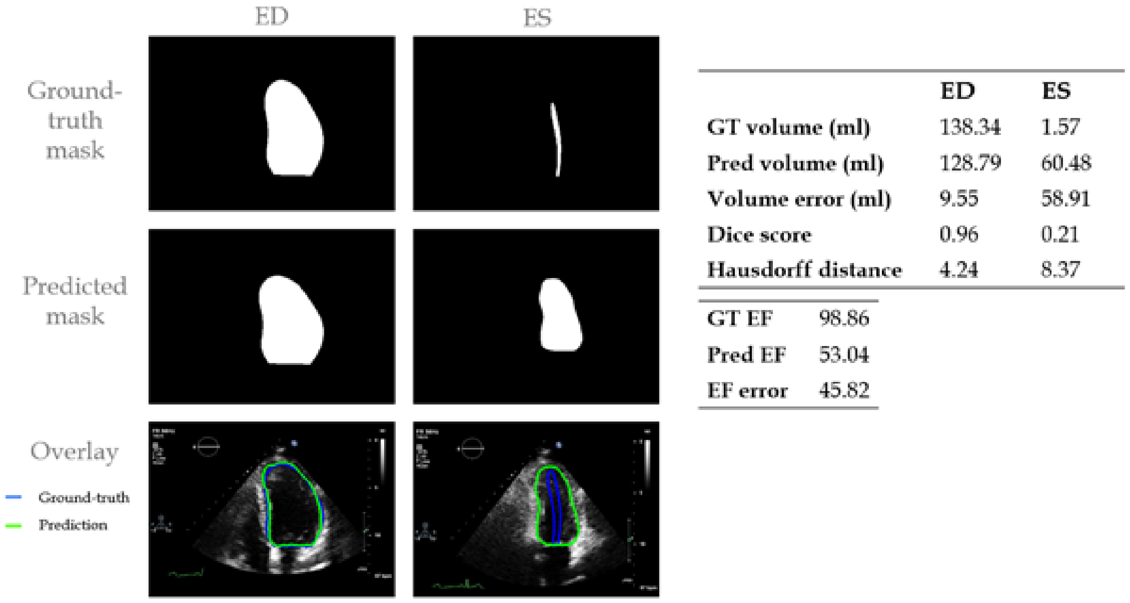


Figure 7.8: Bland-Altman plot for the Unity dataset expert ground-truth annotations and trained network predictions for EF

The results demonstrate pleasing Dice Coefficient scores and average Hausdorff distances, however it is clear these metrics alone are not reliable when considering the average EF and volume calculation metrics, especially for the EchoNet-Dynamic dataset. A possible explanation could be diverse range of image qualities and reliability of expert annotations.

For example, Figure 7.9 highlights the disparity between expert annotations (blue trace lines) and predictions from the automated model (green trace lines) for an example taken from the Unity dataset. Whilst the ground-truth and predicted masks for ED appear reliable and consistent, the annotation for ES appears inconsistent. In this instance, the automated model has predicted, in comparison, a more accurate mask for the segmentation of the LV endocardium, relative to ED.



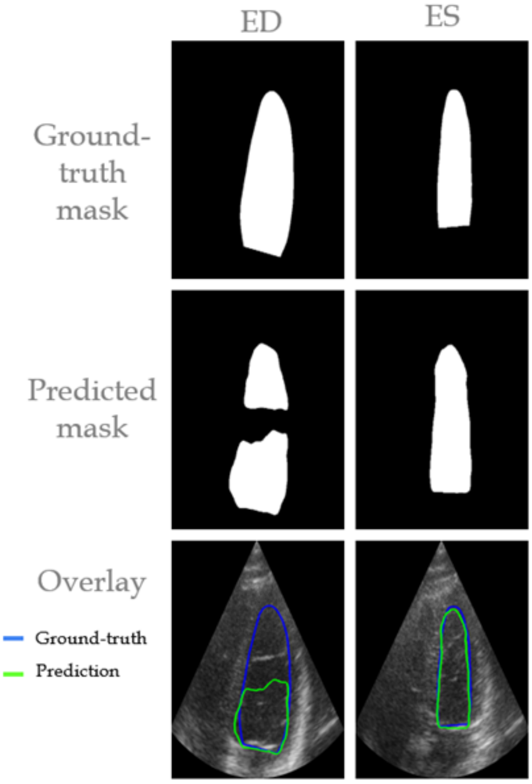
	ED	ES
GT volume (ml)	138.34	1.57
Pred volume (ml)	128.79	60.48
Volume error (ml)	9.55	58.91
Dice score	0.96	0.21
Hausdorff distance	4.24	8.37

GT EF	98.86
Pred EF	53.04
EF error	45.82

Figure 7.9: An Example of inconsistent ground-truth annotations from the Unity dataset

To the right of Figure 7.9, the evaluation metrics are displayed and corroborate the disparity, illustrating the impact the ground-truth annotation accuracy has upon these metrics.

Similarly, figure 7.10 indicates the image quality has significantly impacted upon the networks ability to reliably predict the endocardium trace in alignment with the ground-truth annotation. Here, the predicted mask for ED is separated into two and the volume error is indicative of this. However, the Dice Coefficient score and Hausforff difference are not representative of this error and, in this case, highlight the importance of comparing the estimated volume alongside such established metrics.



	ED	ES
GT volume (ml)	100.40	39.23
Pred volume (ml)	63.31	34.71
Volume error (ml)	37.09	4.52
Dice score	0.83	0.93
Hausdorff distance	5.69	5.78

GT EF	60.92
Pred EF	45.17
EF error	15.75

Figure 7.10: An example from the CAMUS dataset demonstrating that the volume calculation is far more meaningful for failed predictions than standard metrics such as Dice Coefficient and Hausdorff distance

7.5.2 Echocardiographic Phase Detection

As previously stated, the "sliding window" method of pre- and post-processing multiple consecutive beats per echocardiographic video was not possible due to limitations in ground-truth annotations provided for the EchoNet-Dynamic dataset. This can be explained due to the dataset being intended for segmentation tasks, as opposed to phase detection. Therefore, in addition to the exclusion criteria explained in Chapter 6, any videos where ED and ES annotations were $>$ 30 frames apart were also discarded.

The model detailed in Chapter 6, from hereon out referred to as the "PACS model", was used to get ED and ES frame predictions from the EchoNet-Dynamic test set. The network was then trained using EchoNet-Dynamic and tested using the same dataset, and will be referred to as the "EchoNet model". The results are displayed in Table 7.3.

Table 7.3: Errors in ED and ES frame detection between the PACS model and EchoNet-Dynamic phase detection model

Model	ED		ES	
	aaFD	$\mu \pm \sigma$	aaFD	$\mu \pm \sigma$
EchoNet model	1.76	-0.31±2.33	1.99	0.44±2.50
PACS model	2.37	-0.21±3.64	3.60	-2.9±3.58

As expected, the aaFD, mean and standard deviation are more pleasing for the EchoNet model. However, in terms of generalisability in inferring predictions on an unseen dataset from a different centre and ultrasound manufacturer, the range of disagreement is within the range of inter-observer variability between trained human experts previously discussed in Chapter 6. The trained EchoNet model failed to identify the ED in 0.61% of beats and ES in 0.95% of beats. This can be explained because the model was trained on one set of ED/ES annotations within a 30 frame video and in almost all cases, ED comes before ES in the EchoNet-Dynamic dataset. This lack of diversity means the network was unable to learn the temporal relationships between video frames when ES appeared before ED.

As explained, the ED and ES frames in the EchoNet-Dynamic dataset were initially identified using the original PACS model. Table 7.4 compares the results from the ED and ES frames predicted for the EchoNet-Dynamic dataset using the PACS model and EchoNet model. However, because there are no expert annotations for these ED and ES frames, the average results are for average EF and average volume predictions only.

Table 7.4: Average EF results from the ED and ES frames predicted for the EchoNet-Dynamic dataset using the PACS model and EchoNet model

Model	**Avg EF pred mask**
PACS model	49.92
EchoNet model	39.21
EchoNet model GT	56.51

The results are interesting, because comparing the EchoNet GT (ground-truth) measurements, the average volume for the EchoNet model predictions are closer, but in terms of EF the PACS model is more pleasing. The disagreement between the volume and EF averages is possible due to the fact that EF is a ratio between the stroke volume and ED volume, and ratios could obscure subtle details. This means that when the predicted ES frame has a larger endocardium, or when the predicted ED frame has a smaller endocardium, it results in a smaller EF value. The total combined volume from the predicted ED and ES frames could still be relatively close to the volume of the ground truth ED and ES frames.

Figure 7.11 provides a visual illustration of the impact different frames from the same video when calculating EF. In this example, the phase detection model trained on EchoNet-Dynamic (EchoNet model) along the bottom row provides a far less accurate prediction and calculation than that predicted by the PACS model.

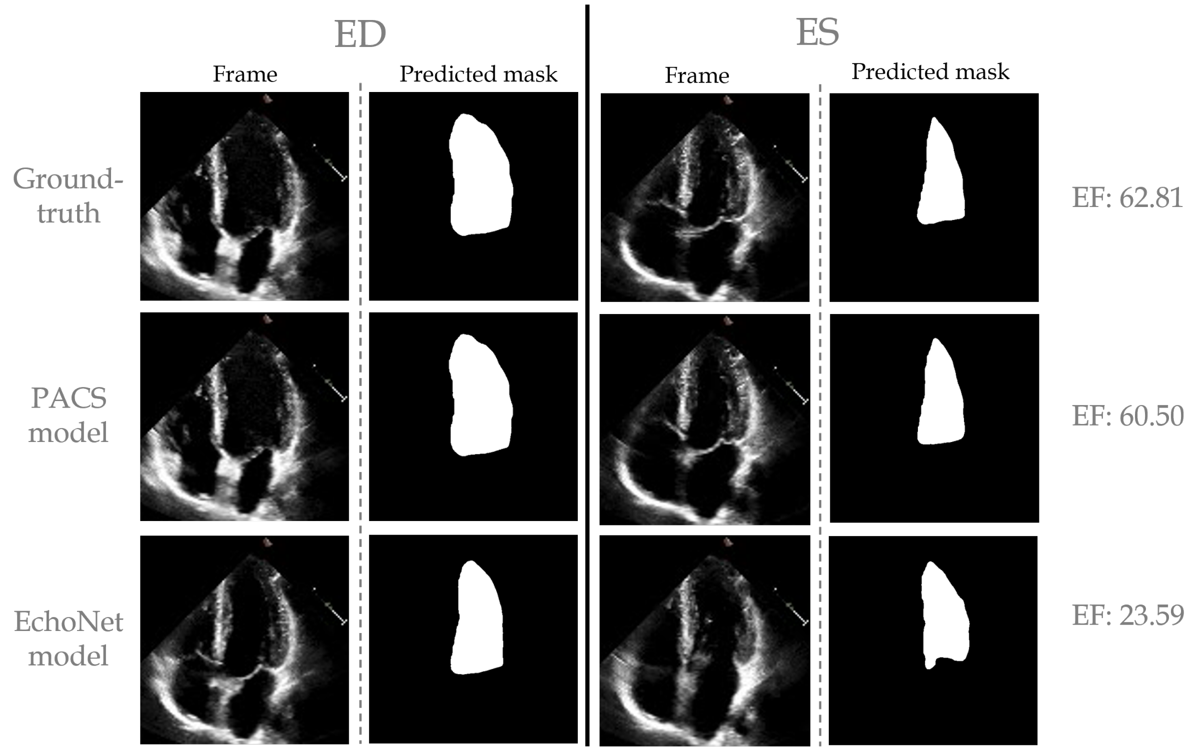


Figure 7.11: An example of the effect different frames have upon the EF calculation taken from the same EchoNet-Dynamic video. The top row represents the original ground-truth annotations, the middle row is the frame predicted by the PACS model and the bottom row is the frame predicted by the EchoNet model

Figure 7.12 provides another example from the EchoNet-Dynamic dataset, in this case the disparity between the size of the ED and ES predicted mask is clear; impacting significantly upon the EF calculation.

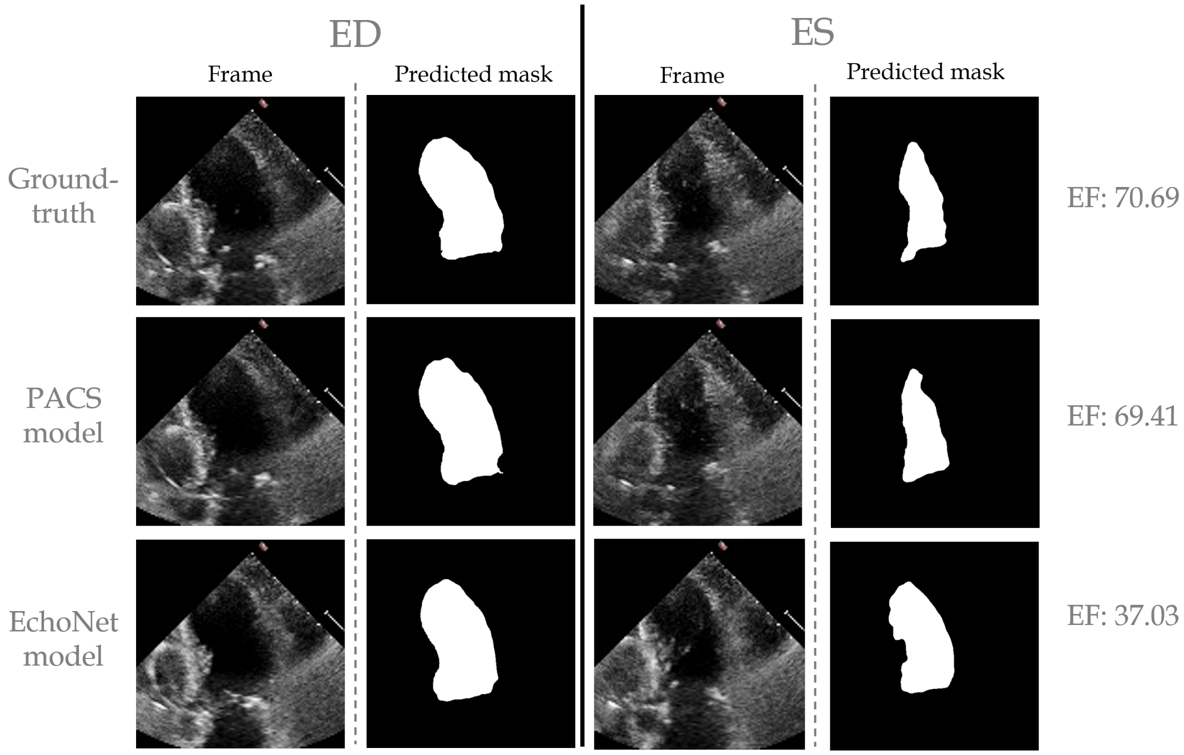


Figure 7.12: As in Figure 7.11 though for a different image from the EchoNet-Dynamic dataset

7.6 Conclusion, Publications and Future Work

7.6.1 Conclusion

Recent literature published in the deep learning for echocardiographic image segmentation domain demonstrate the ability for automated algorithms to accurately perform manual, human annotation tasks. The advancements have undeniably benefited from the publication of large, annotated datasets; allowing for comparison of results and assessing generalisability.

In this chapter, three A4C echocardiographic datasets (two public and one private) were used to train an test a U-Net implementation for the task of segmenting the LV endocardium and estimating EF. Additionally, a bespoke algorithm was introduced, based upon the Simpson's mono-plane method, for estimating the LV volume in millilitres (ml). However, it was not possible to directly compare volume measurements with the EchoNet-Dynamic published report as our attempts to contact the centre and ask for the required information were unsuccessful due to data sensitivity issues. The results demonstrate the importance of estimating the LV volume as established segmentation evaluation metrics can be unsuitable for assessing the accuracy of a predicted mask.

Furthermore, the PACS model detailed in Chapter 6 was used to predict ED and ES frames from the EchoNet-Dynamic dataset, allowing for comparison and evaluation of the error between the ground-truth and network predictions trained on different datasets. The "sliding window" method for pre-processing the data of the PACS model was not possible for the EchoNet-Dynamic dataset due to limitations with the provided expert ground-truth. The results indicate an important observation between the data pre-processing methodology. Training for multiple, consecutive heartbeats from various phases in the cardiac cycle allows crucial temporal relationships to be learnt, proving the robustness of the PACS model in predicting ED and ES frames from previously unseen datasets. For example, the EchoNet model was unable to make predictions when, in very few cases, the ES frame appeared before the ED frame. Unlike the original PACS model, the EchoNet model didn't learn to differentiate. Furthermore, the EchoNet-Dynamic images were resized to 512 x 512 to fit the U-Net implementation, however the original size was 112 x 112. This is a considerable limitation as the dataset contains images of varying degrees of quality, meaning they become highly distorted when the resolution is upscaled and impact the accuracy of the network predictions.

7.6.2 Publication

Conference

Naidoo, P., Alajrami, E., Lane, E., Jevsikov, J., Shun-shin, M., Francis, D. and Zolgharni, M., 2022. Influence of Loss Function on Left Ventricular Volume and Ejection Fraction Estimation in Deep Neural Networks. In: *Medical Imaging with Deep Learning.*

7.6.3 Future Work

The volume computation in this study is the Simpson's monoplane method and only considers images of the apical 4-chamber view. Since only a single view is considered, a major geometric assumption is that the disks are circular. However, it may be possible to obtain more accurate volume estimations using Simpson's biplane method, using both apical 4-chamber and 2-chamber views, which provides a major-axis and a minoraxis that form elliptical-shaped disks. However, the only publically available dataset to publish A2C and A4C images is CAMUS.

The adoption of fully automated LVEF assessment in clinical practice is still limited by the fact commercially available software is semi-automatic, often requiring myocardial border tracing by the manual localisation of anatomical landmarks. This means most automated DL models are vendor specific and do not generalisable well. Similarly, the accuracy of automated predictions can be significantly impacted when image quality is sub-optimal [61]. To be able to create a clinically applicable, generalisable network, more multi-centre datasets will be required.

Segmenting cardiac structures in 3D ultrasound is challenging, a recent demonstrated that 3D U-Net performed worse than it's 2D equivalent when evaluated on the same dataset [211]. Whilst there is potential to derive more accurate volume-related clinical indices, 3D echocardiograms suffer from lower temporal resolution and image quality when compared to 2D echocardiograms. Furthermore, 3D DL networks are more computationally demanding are require increased resources. Should these issues be resolved, future work could centre around the application of LV volume and EF calculations for 3D echocardiographic images.

Chapter 8

Automated Tissue Doppler Echocardiography Analysis

8.1 Introduction

Tissue Doppler Imaging (TDI) is an essential echocardiographic technique for the non-invasive assessment of myocardial blood velocity. Recent studies report TDI measurements as a useful prognostic and diagnostic tool, commonly used for measuring global and regional myocardial diastolic and systolic function, left ventricular (LV), right ventricular and atrial function [64, 65, 212–218].

TDI acquisition and interpretation is performed by trained operators who visually localise pixels representing S', E' and A' peak velocities [73, 74]. An example of such manually selected points are shown in Figure 8.1. This is a subjective process which suffers from inter- and intra-observer variability. It has been demonstrated that human factors are the source of the error in peak Doppler velocity measurements [63]. Current clinical guidelines recommend averaging peak velocity measurements over a minimum of three consecutive beats for improved accuracy [64–66]. However, this manual process is time-consuming and highly disruptive to workflow. Therefore, echocardiographers tend to select one heartbeat they consider an average representative sample which may contribute significantly to test-retest variability [75].

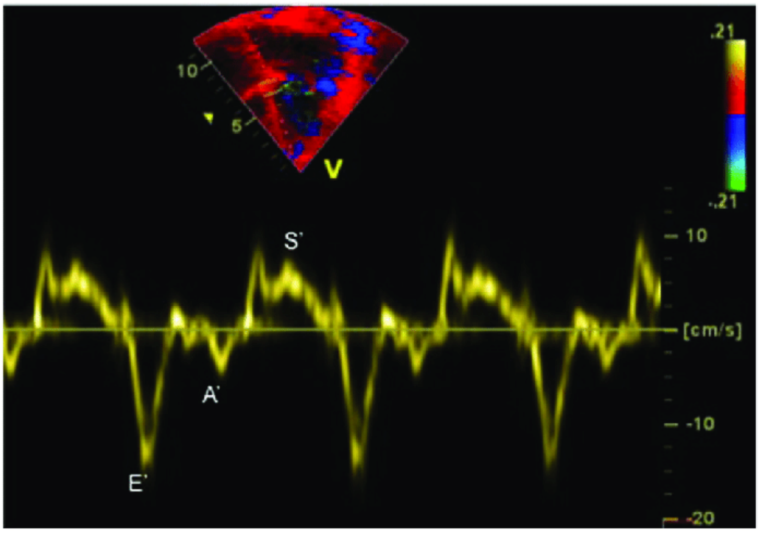


Figure 8.1: Pulsed Tissue Doppler of the septal annulus, showing the S'(systolic velocity), E' (early diastolic velocity), and A' (late diastolic velocity) points selected manually by a human operator

8.1.1 The Value of Automated, Multi-beat Analysis

There is an ongoing challenge of weighing up the cost to clinicians, in terms of time taken to acquire and analyse multiple heartbeats, versus the benefit to patients in terms of improved accuracy. An automated system for accurate analysis of multiple beats from long, uninterrupted tissue Doppler recordings, would allow new protocols to be developed in clinical practice, thus reducing undesirable operator variability which can lead to diagnostic errors. Not only would this system save valuable resources for health services, but it has potential to lead to improved patient outcomes by averaging peak tissue Doppler measurements over a greater number of heartbeats.

Therefore, the potential for an automated model to rapidly quantify crucial measurements from tissue doppler traces is great. Without a reliance upon manual visual detection, specialists' time can be better spent acquiring more high-quality beats, reducing subjectivity and cost.

8.1.2 The Value of Independence From ECG

An electrocardiogram (ECG) signal can be collected from the patient in parallel to an echocardiogram examination. Despite providing some useful data for the computation of clinically relevant parameters (such as temporal intervals from the R-wave peaks), ECG examinations require the connection of multiple cables which is not only time consuming but often inconvenient. Therefore, the ECG trace in the Doppler images could be missing (as in the example shown in Figure 8.1) or, if present, noisy.

Furthermore, due to the increasing popularity of portable, lightweight echocardiographic scanners, focused studies can be performed in a variety of settings lasting just a few minutes [50]. The need for automated peak velocity detection, independent from the ECG signal, is prevalent when implementing such automated technology in handheld devices [23, 51].

8.2 Related Work

Several studies have targeted automated methods for spectral envelope segmentation and interpretation of Doppler images using signal processing and machine learning approaches. Such as low-level image-processing based methods [35, 42], texture filter analysis [219] and thresholding and edge detection [43–48], contour-based and model-based methods for Doppler segmentation [36–39, 49] and traditional machine learning [40, 41].

All previously published approaches encounter limitations which need to be addressed should robust, reproducible applications be applicable for use in routine clinical practice. These include sensitivity to image noise (applied to a small number of good-quality images only), a reliance upon manual operator annotations (being semi-automated), relying on ECG signals for cardiac timing information, and an inability to handle a variety of cardiovascular conditions. To the best of our knowledge, no approach utilises current state-of-the-art deep learning methods [220] for fully automated and ECG-free estimation of peak velocities from multiple consecutive tissue Doppler imaging strips.

8.3 Main Contributions

This study presents a deep learning pipeline for the automated detection of peak velocity measurements from tissue Doppler imaging strips of arbitrary length, containing varying numbers of beats. The main contributions are:

- acquired, prepared, and made publicly available a dataset of tissue Doppler images, each annotated by three accredited and experienced cardiology experts, to be used for deep learning developments

- investigated the feasibility of using convolutional neural networks (CNN) to isolate complete heartbeats from TDI strips of arbitrary length, independently from the ECG information

- achieved accurate landmark localisation, to the pixel, for S', E' and A' peak velocities for each isolated heartbeat

8.4 Methodology

8.4.1 Patient Dataset and Ethics

Pulsed-wave tissue Doppler traces were acquired from 48 patients (30 male), with a mean age of 64 ± 11 years old, who underwent a standard tissue Doppler examination at Imperial College Healthcare NHS Trust, London. Only patients in sinus rhythm were included. No other exclusion criteria were applied. The study was approved by the local ethics committee and written informed consent was obtained.

During acquisition, the echocardiographer was asked to optimise the Doppler images as he or she would in routine clinical practice. The sample volume size was 5 mm with a sweep speed of 75 mm/s and traces from both the septal and lateral annuli were acquired for 30 seconds each. The acquisition process was repeated three times, with the ultrasound probe removed from the patient's chest and re-positioned optimally each time.

Images were acquired using a standard video capture device, live from the echocardiography machine's external display output. In this study, we used a Philips iE33 ultrasound machine (Guildford, UK) with a VGA output, and the VGA2USB Pro (Epiphan Systems, Canada). In total, six 30 second recordings were acquired for each patient (an equal mix of septal and lateral) and reconstructed (using a bespoke MatLab script) into a continuous Doppler strip with a resolution of 900×1300 pixels.

The original images were automatically cropped to include only the Doppler trace. The horizontal zero-velocity axis was detected, and optical character recognition was used to convert velocities from pixels into cm/s. More information about the dataset used in this study, the clinical characteristics of the patients recruited for image acquisition, and the reconstruction of long Doppler strips can be found in the following previously published reports [44, 221].

8.4.2 Expert Annotations

Three accredited and experienced cardiology experts manually annotated (selecting the S', E', and A' peak coordinates), each blinded to the judgement of the other. We developed a custom-made program closely replicating the interface of clinical echocardiography hardware, and the experts were asked to select the peak velocities as they would in clinical practice. Selections were made in one or more sessions at their convenience and the time taken was recorded. Where an operator judged an image to be of low quality, they declared it invalid and did not make a measurement.

Complete beats, for which all three operators annotated systolic and diastolic peak velocities (S', E', and A') were selected. In total, the dataset comprised 280 (out of 288) Doppler strips (5,327 beats). For the definition of ground truth labels, the consensus of the three human experts, as well as the annotations from each expert, were used.

8.4.3 Neural Network Architecture

The neural network consists of a two-step beat detection and keypoint localisation model:

ECG-free Heartbeat Isolation

Detection of complete beats from the reconstructed tissue Doppler strips is performed by a Mask R-CNN implementation, originally developed by Facebook AI Research (FAIR) [222]. A third-party implementation of Mask R-CNN code [223] was adopted, updated to Tensorflow 2.0, and extended for use in this study.

The Mask R-CNN approach builds upon the Faster R-CNN [224] network by efficiently detecting objects in an image with the benefit of the ROIAlign module for improved performance and can produce an object segmentation mask in parallel with the existing bounding box branch. A mask is not required for this study; however, it has future potential for extending the proposed network to other Doppler modalities (e.g., left ventricular outflow tract) which would require tracing of the Doppler envelope. Each heartbeat is recognised, localised and classified as a complete (containing all three Doppler peak velocities) or incomplete (containing only one or two peak points for heartbeats appearing at the beginning or the end of the strip). Only those classified as complete heartbeats were subsequently processed.

The CNN backbone used for the extraction of a spatial feature vector was ResNet101, initialised with pre-trained weights from the large-scale object detection and segmentation COCO dataset [225]. Images are resized and zero padded to get a square image of 1024×1024 pixels for compatibility with the network, and to maintain important spatial information. A high-level illustration of the Mask R-CNN implementation can be found in Figure 8.2.

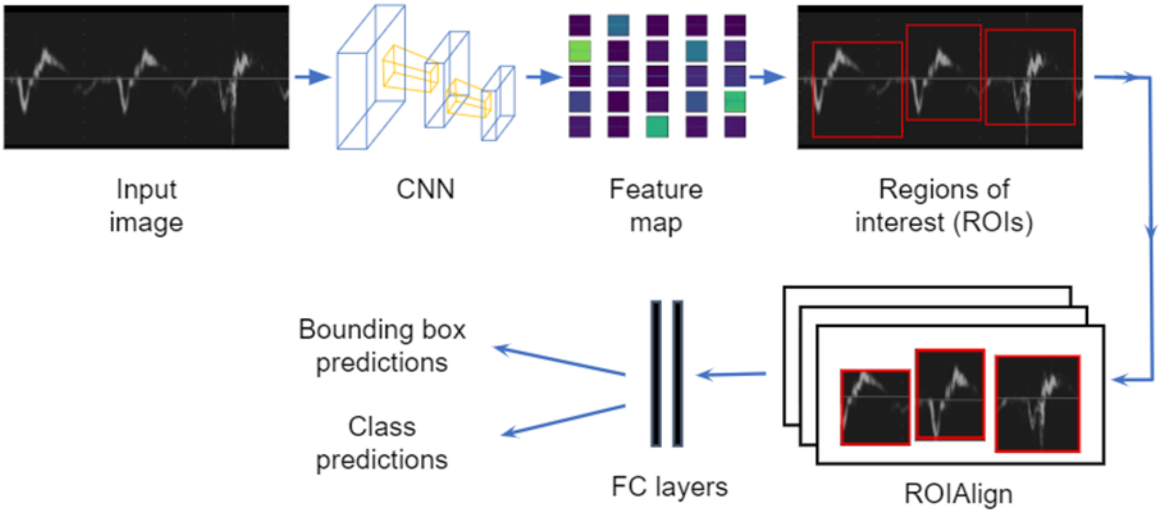


Figure 8.2: An high-level overview of the Mask R-CNN network with ResNet101 used as backbone CNN feature extractor to detect and isolate complete heartbeats (including all three Doppler peak velocities) present in arbitrary length tissue Doppler strips

Doppler Peak Velocity Detection

After successfully detecting each beat, the region of interest is cropped from the input, capturing the entire y-axis and taking x-axis coordinates from the Mask R-CNN bounding box prediction. The cropped image is resized to 192 x 192 pixels, and peak velocity keypoints are detected using a multi-Stage heatmap regression network.

Instead of predicting Cartesian coordinates, the model predicts a different Gaussian response heatmap, or belief map, for every keypoint of interest. A heatmap is simply an image indicating the likelihood of a specific keypoint residing at that pixel. Subsequently, keypoints are obtained by finding the local maxima in the heatmaps. Indirect inference through a predicted heatmap offers several advantages over direct prediction [174].

Each pixel value in the heatmap encodes the confidence that a nearest keypoint of a particular type occurs. Proxy keypoint heatmaps are generated by putting a symmetric Gaussian distribution with a standard deviation σ at the ground truth location of each annotated keypoint on the input image $I \epsilon R^{W \times H}$:

$$Y_{ijk} = exp(-\frac{(x - G_{ik})^2 + (y - G_{jk})^2}{2\sigma^2})$$

Here, σ is a size-adaptive standard deviation, Y_{ijk} represents the heatmap of the $k-th$ channel, and G_{ik} and G_{ik} represent the ground-truth coordinate of the $k-th$ landmark (i.e. peak Doppler velocity).

One example of the generated Gaussian peak is depicted in Figure 8.3. The standard deviation σ, which controls the spread of the Gaussian peaks, was set to 5. Therefore, balancing the foreground pixels and background pixels and avoiding overlaps between Gaussian labels in each heartbeat.

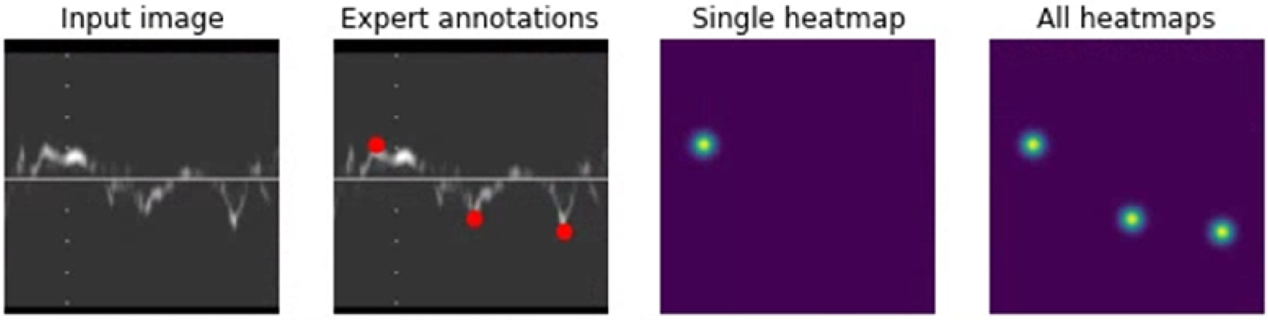


Figure 8.3: A typical isolated heartbeat with the corresponding expert annotations and the generated proxy heatmap as the ground-truth. Each input image has 3 associated keypoints. Therefore, each image has 3 heatmaps; one for each keypoint. As observed above, the areas of the heatmap that are more yellow (brighter) represent pixels that are more likely for a given keypoint.

The model consists of a two-stage sequence of CNNs that produces 2D heatmaps for the location of the keypoint. At the second stage, image features and the heatmaps produced by the first stage, are used as the input. The heatmaps provide the second stage an expressive, non-parametric encoding of the spatial uncertainty for each keypoint location, allowing the network to learn rich image-dependent spatial models of the relationships between keypoints.

Large receptive fields in CNNs are critical for learning long range spatial relationships and can bring about accuracy improvement. Conversely, detailed information (in smaller receptive fields) is needed for fine-grained localisation. Therefore, to consolidate the global and local features, an hourglass module (a special type of fully convolutional network) was adopted as the CNN feature extractor at each stage [174].

The proposed hourglass network is an encoder-decoder architecture, as shown in Figure 8.4. First, two to three convolutional blocks were utilised, followed by the max pooling layer to extract the features from images. After five such convolutional blocks and pooling layers, we obtain a feature map in a small shape with rich information:

$$Y_{i+1} = Pool(Conv(Conv(f_{i-1}))), \text{in modules 1, 2}$$

$$Y_{i+1} = Pool(Conv(Conv(Conv(f_{i-2}))), \text{in modules 3, 4, 5}$$

Here, f_i denotes the features of the $i - th$ stage in the encoder, $Conv$ and $Pool$ represent the convolutional block and max pooling layers. The stacked convolutional layers enlarge the receptive field of the model and enable detection from a large perspective. This is followed by two 1 × 1 convolutional layers, resulting in a fully convolutional architecture. The features are then upsampled and added to the output of former layers by a skip connection:

$$\hat{f}_{i+1} = Up(Conv(\hat{f}_i)) + Conv(f_{i+1})$$

Here, $\hat{f}_i$ denotes the features of the $i - th$ stage in the decoder, Up denotes the upsampling layer. In this way, the model can combine the rich context information in the shallow layers with the adequate semantics in the deep layers for landmark detection. In the final stage, a 1 × 1 convolutional layer is used to generate the output (a heatmap).

An illustration of the architecture can be found in Figure 8.4. A two-stage sequence of CNNs was deemed to be a good compromise between the accuracy and the training/inference time. Increased stages did not generate an improvement in the accuracy in a statistically significant manner.

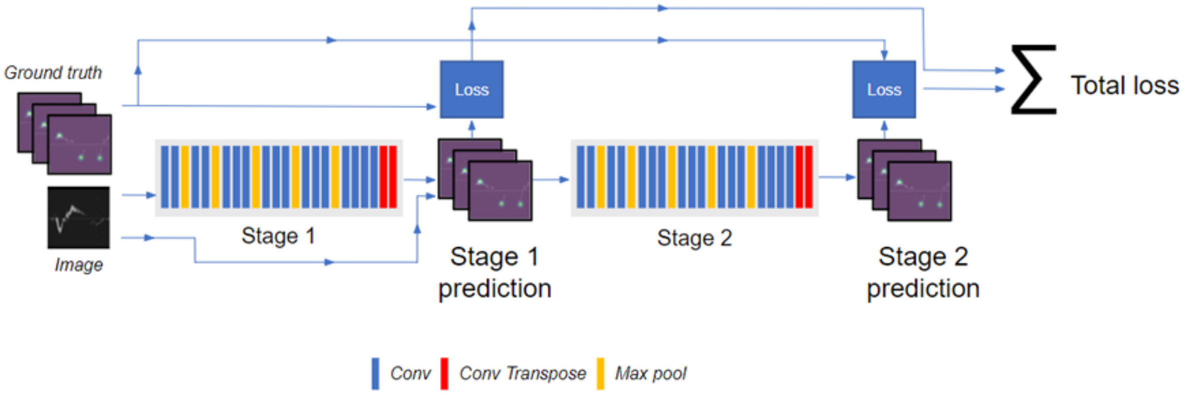


Figure 8.4: Representation of the two-stage convolutional heatmap regression model used to determine Cartesian coordinates for systolic and diastolic peak velocities (S', E' and A').

8.4.4 Deep Learning Framework

For the model to be capable of processing a Doppler strip of arbitrary length (i.e. containing any number of complete heartbeats), a sliding window approach was adopted. The input is chunked into 1,000-pixel wide strips with a stride of 800-pixels (allowing for some overlap to ensure no heartbeats are missed during inference). If required, the final segment is zero-padded to ensure uniform length as the model input. Figure 8.5 illustrates the complete network pipeline:

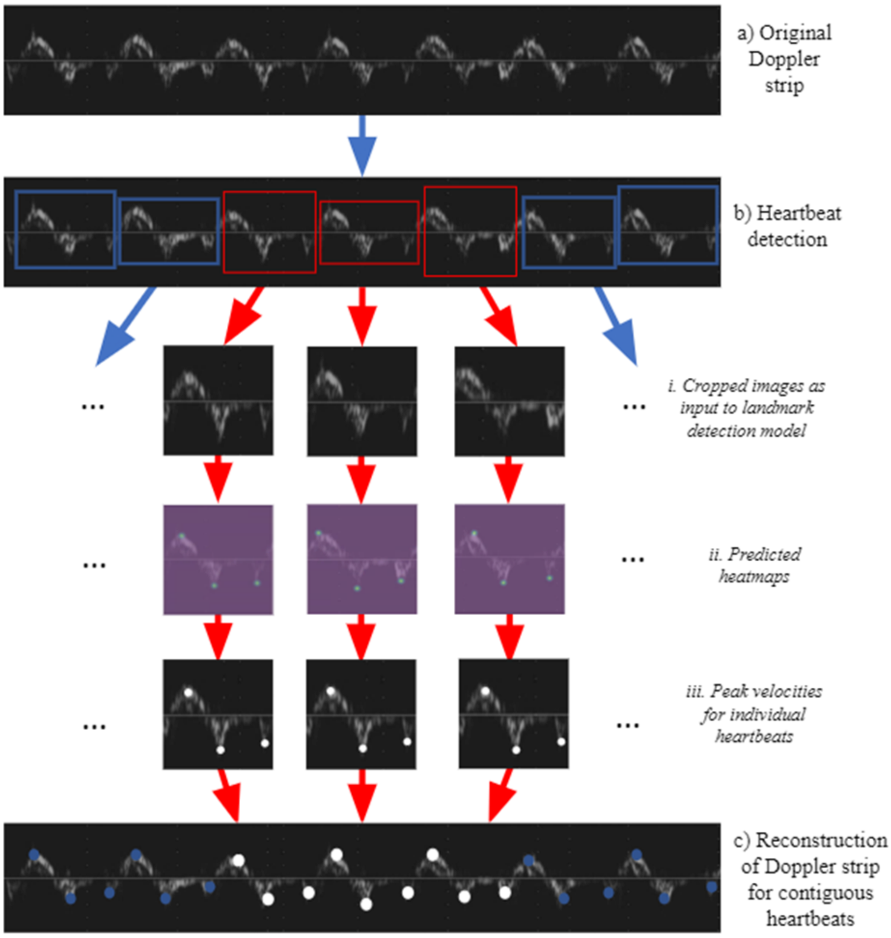

a) Original Doppler strip
b) Heartbeat detection
i. Cropped images as input to landmark detection model
ii. Predicted heatmaps
iii. Peak velocities for individual heartbeats
c) Reconstruction of Doppler strip for contiguous heartbeats

Figure 8.5: Illustration of the complete processing pipeline: (a) 1,000-pixel wide representative Doppler strip; (b) isolated beats (based upon coordinates of bounding boxes from object detection predictions), i. Cropped ROIs from the original Doppler strip, resized to 192×192 pixels, ii. two-stage heatmap regression for the localisation of Cartesian coordinates for systolic and diastolic peak velocities (S', E' and A'), iii. example plots of predicted velocities; (c) predictions are rescaled relative to the original Doppler strip.

8.4.5 Implementation Details

The models were implemented using the TensorFlow 2.0 deep learning framework [143] and trained using an Nvidia GeForce RTX 3090. During Mask R-CNN training a multi-task loss function is defined on each sampled region of interest (ROI), further information can be found in [222]. The classification loss and bounding box loss are the same as specified in [226]. The Mask R-CNN implementation optimiser was Stochastic Gradient Descent (SGD), initialised with a learning rate of $10 - 4$. Training was conducted over 50 epochs with a batch size of 1.

The dataset was used to train the object detector model by trimming long strips into 1,000-pixel wide images with a sliding window of 800-pixels allowing for some overlap to prevent missed beats.

The two-stage heatmap regression model was trained using the ADAM optimiser [163] with a learning rate of $10 - 5$ and mean squared error (MSE) loss. Training was over 335 epochs with a batch size of 64. The original tissue Doppler strips were cropped to each full beat with a border along the x-axis of 50-pixels each side and spanning the entire length of the y-axis. Images were resized to 192x192.

The dataset was randomly split into training (60%), validation (20%), and testing (20%) sub-sets. Doppler images from each patient were ensured to appear in only one sub-dataset to maintain sample independence.

8.4.6 Statistical Analysis

To evaluate the trained deep learning model, the predicted Cartesian coordinates for S', E' and A' peak velocities were converted into cm/s and compared to the consensus of manual annotations by three human experts. The level of disagreement between these experts (inter-observer variability) was also calculated. Additionally, detection time for the automated model and the humans was compared.

Statistical analysis of the levels of agreement was performed using Bland-Altman plots; bias (mean of the signed differences) and standard deviation (SD) were calculated where the confidence interval was defined as ± 1.96SD.

8.5 Results & Discussion

To calculate the time taken by experts to manually annotate the three peak velocities for one heartbeat, compared by automated model inference, average times across a sample of 25 beats were taken. Experts were 4.76 seconds whereas the automated model was 0.18 seconds.

Examples of successful and failed predictions are shown in Figure 8.6. The middle illustrates a failed prediction for the S' peak, where the model has wrongly identified pre-systolic velocity during the isovolumetric contraction as the peak systolic velocity. The right panel demonstrates another example of an inaccurate prediction in which the model has underestimated the peak velocity by presumably being drawn to the brightest point on the Doppler envelope which represents the dominant volume of blood flowing at velocity. For this case, the experts too seem to have differences of opinion as to what the peak velocity is.

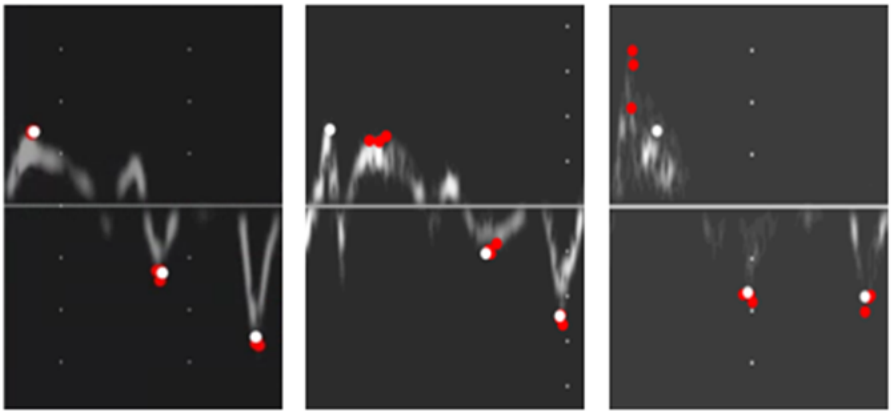


Figure 8.6: Examples of successful and failed predictions, where expert annotations and model predictions are shown as red and white circles, respectively. Left: a typical good prediction in which there is a good agreement between all three experts and the model. Middle and Right show examples of inaccurate predictions for S' systolic velocity.

Figures 8.7, 8.8 and 8.9 (upper panels) show Bland Altman plots for the degree of disagreement between the experts. Each expert's velocity measurements were compared with the consensus of the other two experts. This was done on a patient-by-patient basis, where we calculated, for each patient, the average of each expert's measurements made from all the beats at the septal and lateral annulus.

Across the three peak velocity measurements, experts showed a tendency to consistently make measurements which over-read (positive bias) or under-read (negative bias) the average of the other two experts. This corroborated our previous findings on other Doppler modalities [227].

The standard deviation of differences between each expert's estimates and the consensus of experts, excluding that particular expert, was 0.60, 0.40 and 0.27 cm/s for S velocity measurements. The performance was similar for E (0.39, 0.50 and 0.22 cm/s) and A (0.53, 0.50 and 0.25 cm/s) peaks.

To investigate whether the models can perform as well as the human experts, three models were trained using the ground-truth obtained from each expert. For example, Model-1 learnt from Expert-1 annotations only. Subsequently, the level of agreement between either candidate (Expert-1 and Model-1) was measured against the consensus of other two experts, to avoid bias.

The lower panels in figures 8.7, 8.8 and 8.9 show Bland Altman plots for the degree of disagreement between the model and experts for the septal and lateral annuli, combined. All results are reported on the testing sub-dataset only.

Expert-1 had a bias of -0.51 to -0.20 cm/s, and standard deviation of the differences of 0.60 cm/s, when compared with the consensus of other two experts. Having learnt from Expert-1, Model-1 had a very similar performance to this expert in terms of bias (-0.12 to -0.15 cm/s) and standard deviation of the differences ($\leq$ 0.46 cm/s) with the consensus of other two experts.

Comparably, Model-2 and Model-3 had indistinguishable performance from their corresponding expert teachers and can therefore be considered as their potential replacements.

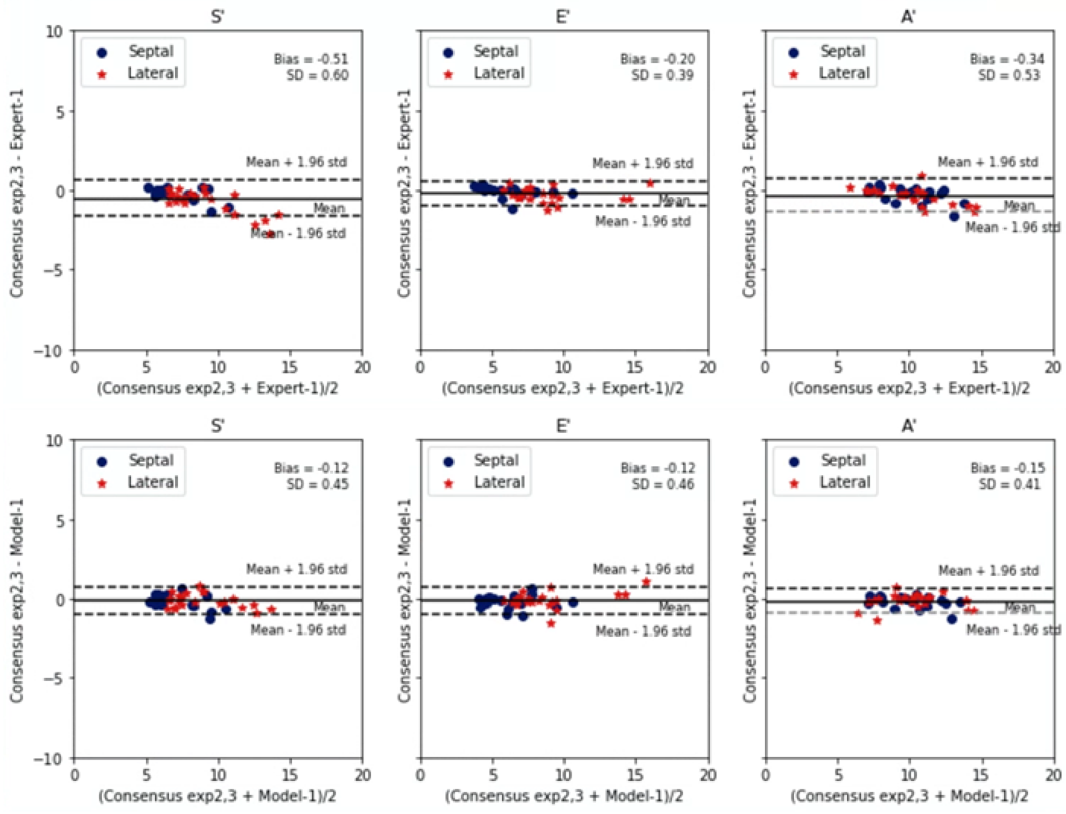


Figure 8.7: Bland-Altman plots for patient-by-patient analysis for peak S', E', and A' velocities where the agreement between Expert-1 and the consensus of the other two experts is shown in the upper panel. The lower panel replaces Expert-1 by Model-1 which has learnt (trained using ground-truth obtained) from Expert-1 only.

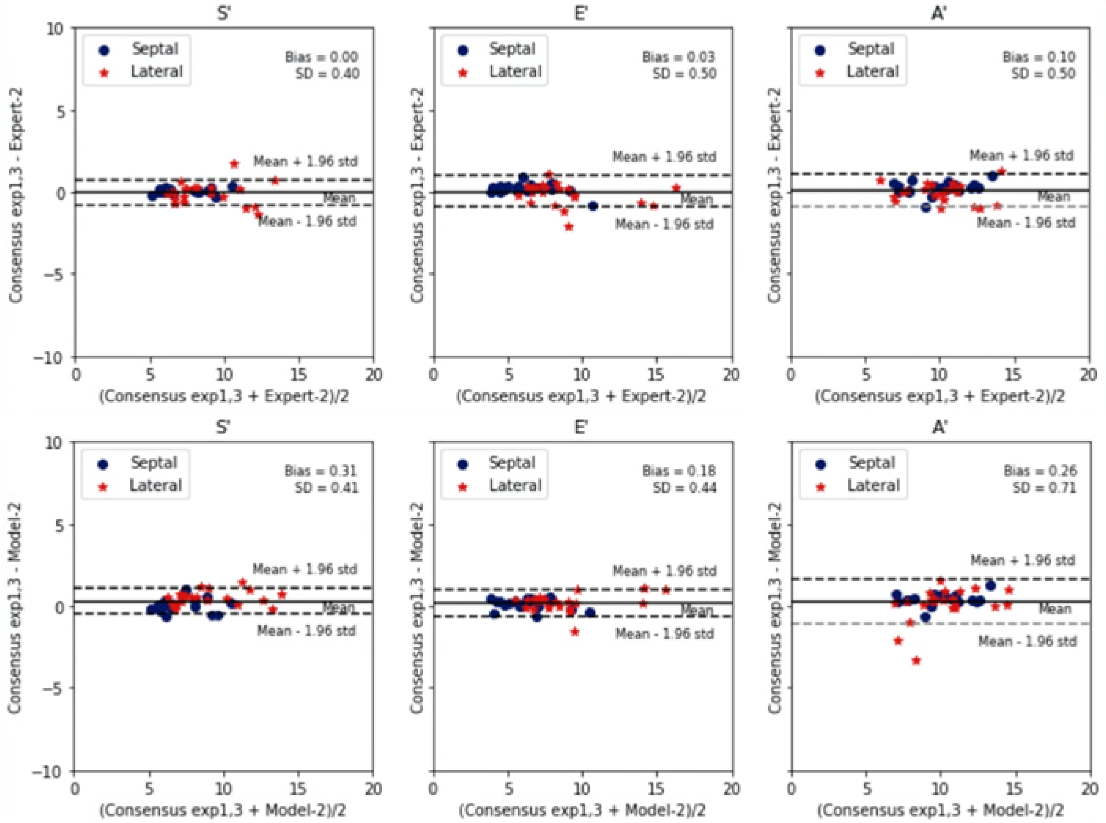


Figure 8.8: As in Figure 7, but for Expert-2.

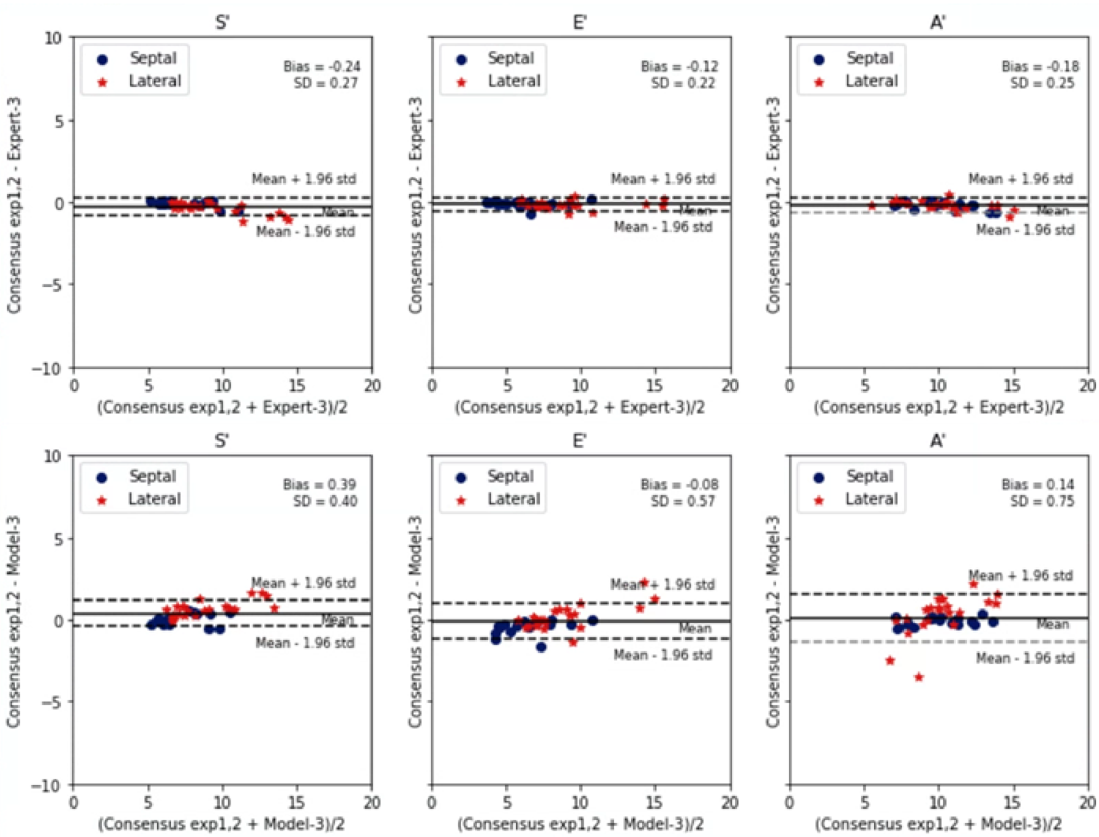


Figure 8.9: As in Figure 7, but for Expert-3.

Additionally, a model was trained on the consensus of annotations from all three experts, and its performance was examined against the testing sub-dataset of the same consensus, as shown in the upper panel of Figure 8.10.

Velocities ranged from 4.59 to 12.69 cm/s, 3.02 to 12.3 cm/s, and 5.37 to 15.03 cm/s for septal S, E and A peaks respectively. At the lateral annulus, the velocities ranged from 5.27 to 15.52 cm/s, 3.9 to 18.74 cm/s and 4.49 to 17.86 cm/s for systolic, early diastolic, and late diastolic phases.

The lower row illustrates a similar assessment where the individual heartbeats, originating from all Doppler strips (i.e. patients) in the testing dataset, are placed in a pool of beats and are used for a beat-by-beat analysis.

Whilst the bias is comparable for all three peak velocities in both analyses, the standard deviation of the differences is greater for beat-by-beat analysis (0.8-1.13 cm/s), compared to that for the patient-by-patient analysis (0.35-0.45 cm/s). This is due to the presence of occasional noisy predictions, mainly in E' and A' peak velocities in both septal and lateral annuli, evident in the Figure. Since the patients were approached without regard to whether they were likely to be good echo subjects, in some cases, image quality was poor which may have contributed to the wide dispersion in measurements between the model and human experts.

This dispersion was present even amongst the humans, as shown in Figure 8.11. This clearly indicates the advantage of taking an average across several heartbeats for each patient, as opposed to relying on a single heartbeat measurement.

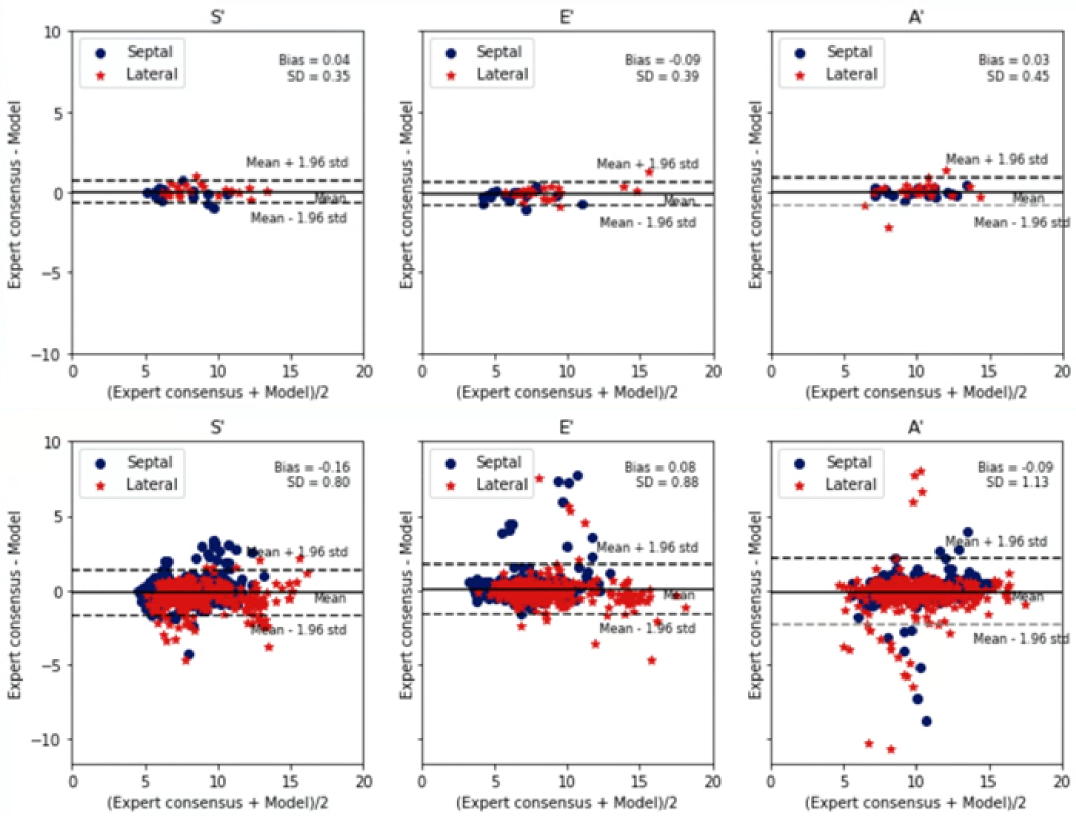


Figure 8.10: Bland-Altman plots for patient-by-patient analysis (upper panel) and beat-by-beat analysis (lower panel) for peak S', E', and A' peak Doppler velocities in both septal and lateral annuli, where the agreement between the expert consensus and the model trained using the consensus ground-truth is shown.

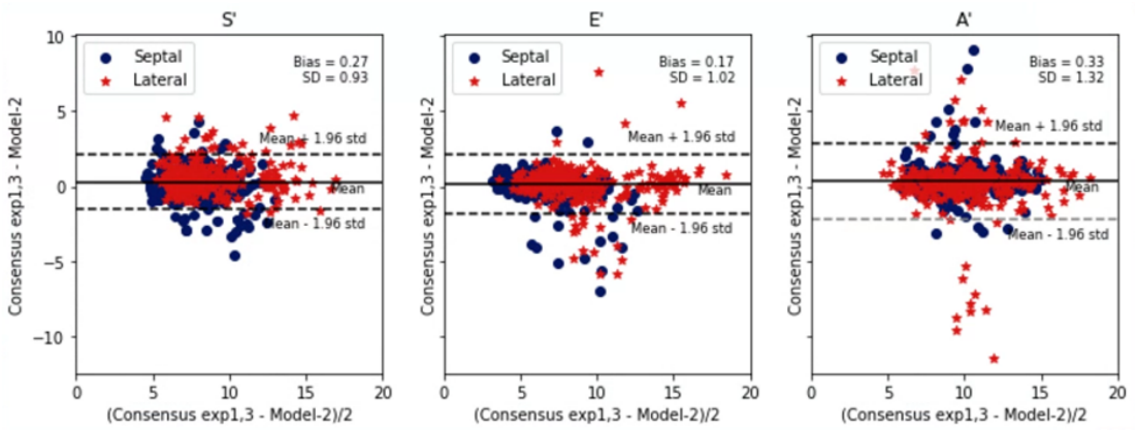


Figure 8.11: As in Figure 8.8 (upper panel), but on a beat-by-beat analysis.

Bland-Altman biases and 95% limits of agreement on a patient-by-patient basis are summarised in Table 8.1, but for the measurements in the septal and lateral walls separately. Here, the ‘expert combinations’ refer to the pool of all differences between each pair of experts, hence indicating the inter-observer variability with a standard deviation averaged 0.49 and 0.80 cm/s for the septal and lateral walls, respectively.

In all septal measurements, the performance of the models are indistinguishable from their teacher expert, when compared with the other two human experts.

The lateral annulus measurements showed the same pattern, albeit the biases and the standard deviation of the differences were greater than those for septal measurements, the same as for inter-observer variability for both human and machine performances. This is somewhat to be expected, since the lateral wall is less visible than the septal, due to the lower spatial resolution in the ultrasound images, thus making the measurements in the lateral annulus less reliable.

Table 8.1: Bland-Altman bias and 95% limits of agreement comparing manual and automated peak tissue Doppler velocities (S', E', and A') measurements at the septal and lateral annuli on a patient-by-patient basis. All values are provided in cm/s.

	Septal annulus			Lateral annulus		
	S'	E'	A'	S'	E'	A'
Human Performance						
Expert-1 vs other experts	-0.20 ± 0.42	-0.05 ± 0.32	-0.27 ± 0.49	-0.76 ± 0.73	-0.32 ± 0.45	-0.38 ± 0.57
Expert-2 vs other experts	0.06 ± 0.16	0.21 ± 0.32	0.26 ± 0.41	0.33 ± 0.55	0.05 ± 0.55	0.16 ± 0.43
Expert-3 vs other experts	0.14 ± 0.34	-0.16 ± 0.33	0.01 ± 0.58	0.43 ± 0.73	0.26 ± 0.62	0.22 ± 0.61
Expert combinations	±0.41	±0.42	±0.63	±1.00	±0.68	±0.70
Machine Performance						
Model-1 vs other experts	-0.13 ± 0.46	-0.19 ± 0.43	-0.18 ± 0.37	-0.12 ± 0.44	-0.06 ± 0.50	-0.13 ± 0.45
Model-2 vs other experts	0.07 ± 0.40	0.13 ± 0.34	0.43 ± 0.37	0.52 ± 0.42	0.21 ± 0.54	0.12 ± 1.05
Model-3 vs other experts	0.01 ± 0.33	-0.43 ± 0.41	-0.08 ± 0.26	0.70 ± 0.46	0.20 ± 0.73	0.32 ± 1.25
Model vs expert consensus	-0.08 ± 0.37	-0.21 ± 0.35	0.01 ± 0.24	0.14 ± 0.32	0.01 ± 0.42	0.05 ± 0.66

Model-1 is a model trained on ground-truth obtained from Expert-1 only.

Model is the model trained on ground-truth obtained from experts' consensus.

Human performance is looking at the pool of all expert comparisons (i.e., Expert-1 vs Expert-2; Expert-1 vs Expert-3, and Expert-2 vs Expert-3).

Table 2 shows the corresponding data for the beat-by-beat analysis. The inter-observer variability was greater than that in the patient-by-patient analysis, with a standard deviation averaging 0.74 and 0.98 cm/s for the septal and lateral walls, respectively. Again, the automated measurements of the peak Doppler velocities showed good agreement with the reference consensus expert values; average standard deviations of 0.86 and 1.01 cm/s in septal and lateral walls, respectively; comparable with inter-observer variability.

Table 8.2: As in table 8.1, but for a beat-by-beat analysis.

	Septal annulus			Lateral annulus		
	S'	E'	A'	S'	E'	A'
Human performance						
Expert-1 vs other experts	0.19 ± 0.63	0.04 ± 0.56	0.24 ± 0.69	0.66 ± 0.94	0.27 ± 0.77	0.27 ± 0.78
Expert-2 vs other experts	-0.06 ± 0.50	-0.22 ± 0.60	-0.29 ± 0.70	-0.33 ± 0.87	-0.12 ± 0.80	-0.17 ± 0.75
Expert-3 vs other experts	0.19 ± 0.63	0.04 ± 0.56	0.24 ± 0.69	0.66 ± 0.94	0.27 ± 0.77	0.27 ± 0.78
Expert combinations	±0.69	±0.68	±0.86	±1.06	±0.96	±0.91
Machine performance						
Model-1 vs other experts	0.12 ± 0.97	0.17 ± 0.94	0.15 ± 0.99	0.11 ± 1.02	0.04 ± 0.93	0.04 ± 1.35
Model-2 vs other experts	-0.04 ± 0.93	-0.15 ± 0.94	-0.44 ± 0.98	-0.50 ± 0.93	-0.19 ± 1.10	-0.21 ± 1.66
Model-3 vs other experts	0.01 ± 0.82	0.42 ± 1.00	0.11 ± 0.85	-0.59 ± 0.93	-0.08 ± 1.32	-0.38 ± 1.87
Model vs expert consensus	0.07 ± 0.78	-0.22 ± 0.92	0.02 ± 0.88	-0.38 ± 0.81	0.06 ± 0.84	-0.19 ± 1.38

8.6 Conclusion, Publications and Future Work

8.6.1 Conclusion

In this Chapter an open-source automated method to make measurements on tissue Doppler traces is presented, independent from an accompanying ECG signal, and using deep neural networks.

A dataset of tissue Doppler images was acquired, where the patients were a convenience sample drawn from those attending a cardiology outpatient clinic. DL models were trained using the ground-truth obtained from the annotations by three accredited and experienced cardiology experts.

The performance of the proposed method was examined by comparisons to the gold-standard reference data. The inter-observer variability in our study was similar to that reported in the literature [228], and the trained models' performance was indistinguishable from this.

There are multiple sources of variability in Doppler measurements. The proposed approach can help in two ways. Firstly, it eliminates the variability that arises when different operators select different positions to make velocity measurements from the same images. Secondly, by allowing the analysis of multiple beats in the same time a human would take to measure a smaller number of beats, it can reduce the contribution of beat-to-beat variability.

At present, no echocardiography dataset, and corresponding annotations specifically prepared for tissue Doppler measurements, is publicly available. The patient dataset used in this research has therefore been made available online, thereby providing a benchmark for future studies.

8.6.2 Publications

Journal Publication

Lane, E., Jevsikov, J., Dhutia, N., Shun-shin, M., Francis, D. and Zolgharni, M., 2022. Automated Multibeat Tissue Doppler Echocardiography Analysis Using Deep Neural Networks. *Medical & Biological Engineering & Computing.*

Conference Proceeding

Lane, E., Jevsikov, J., Dhutia, N., Shun-shin, M., Francis, D. and Zolgharni, M., 2022. Automated Multibeat Tissue Doppler Echocardiography Analysis Using Deep Neural Networks. In: *Medical Imaging with Deep Learning.*

8.6.3 Future work

The Mask R-CNN approach builds upon the Faster R-CNN [224] network by efficiently detecting objects in an image with the benefit of the ROIAlign module for improved performance and can produce an object segmentation mask in parallel with the existing bounding box branch. A mask is not required for this study; however, it has future potential for extending the proposed network to other Doppler modalities (e.g. left ventricular outflow tract) which would require tracing of the Doppler envelope.

Chapter 9

Conclusion and Future Work

9.1 Conclusion

The overall aim of this thesis was to develop several deep learning (DL) computer vision (CV) models capable of reliably automating important processes in the image and video analysis stage of the modern echocardiographic laboratory. More specifically, view classification, image quality assessment, detection of cardiac phases from cine loops, segmenting the LV and calculating crucial diagnostic metrics and automatic detection of velocity measurements from Tissue Doppler images.

Such automated systems are capable of revolutionising clinical practice in assisting clinicians in the decision making process, analysing a greater amount of data in a shorter time than humans and with less variability, hence improving accuracy, saving valuable resources for healthcare services and improving patient outcomes.

Chapter 2 presents the clinical context including an overview of clinical cardiography and a brief discussion of several conditions affecting the heart. Imaging modalities were discussed, with a focus upon echocardiography. **Chapter 3** discusses the recent advancements in AI and DL and how they have paved the way for several cutting-edge computer vision tasks, each lending themselves well to automated medical image interpretation. Throughout this chapter, state-of-the-art architectures are discussed, including the history of CNNs, segmentation and object detection networks and RNNs for sequence analysis. A thorough description of network parameters, hyper-parameters and regularisation techniques for optimisation are also included. Furthermore, an analysis of current trends in programming languages and DL frameworks, libraries and packages is provided.

In **Chapter 4** a two-stage DL network for accurate classification of 21 echocardiographic views (both video and images) is presented alongside a novel approach to pre-processing videos for unbalanced classes. Four state-of-the-art CNNs were compared for image classification, with ResNet50 achieving the highest accuracy of 92.6% across five view classes with an average inference time of 49 milliseconds. The most challenging views to classify were tricuspid CW and aortic CW predominantly due of their similarity. A time-distributed DenseNet201 CNN with the addition of 1x LSTM layer produced 98.5% accuracy when classifying 15 complex video views. The most challenging being PSAX AV and PSAX LV, again due to similarities between views. This chapter addresses research questions 1 and 2, and research objectives 1 and 6.

Chapter 5 presents a novel approach for echocardiographic image quality assessment using the performance of each image when inference is performed on a trained left ventricle (LV) segmentation and Mitral Valve (MV) hinge point localisation models. The proposed network is a multitask learning network with a ResNet50 backbone for feature extraction and subsequent classification and regression branches. Early experimental results demonstrate the accuracy of the regression branch significantly outperformed a 10-point classification approach. However, the Unity dataset used in this Chapter is imbalanced, due to the inclusion of a significant proportion of high quality images, some low quality and very few in between and further work is needed to improve the accuracy and inference time. Chapter 5 addresses research questions 3, 4 and 6, and research objectives 2 and 6.

The feasibility of fully automated identification of ED and ES frames derived from 2D echocardiographic images and independent from an accompanying ECG signal using deep neural networks is explored in **Chapter 6**. The performance of the proposed method was examined by comparisons to gold standard reference data, obtained from multiple cardiologist experts. It has been demonstrated that the performance of the proposed model is like that of human experts, with its detection error falling within the range of inter-observer variability and can therefore be used to reliably identify multiple ED and ES frames from videos of arbitrary length. Additionally, the performance of the automated model, measured as the processing time, is superior to that of human operators, where an improvement of ¿20 times was observed. One of the private datasets used in this Chapter has since been published for the benefit of the research community. It addresses research questions 5 and 6, and research objectives 3, 6 and 7.

In **Chapter 7**, three A4C echocardiographic datasets (two public and one private) were used to train an test a U-Net implementation for the task of segmenting the LV endocardium and estimating EF. Additionally, a bespoke algorithm was introduced, based upon the Simpson's method, for estimating the LV volume in millilitres (ml). However, it was not possible to directly compare volume measurements with the EchoNet-Dynamic published report as our attempts to contact the centre and ask for the required information were unsuccessful due to data sensitivity issues. The results demonstrate the importance of estimating the LV volume as established segmentation evaluation metrics can be unsuitable for assessing the accuracy of a predicted mask.

Additionally, in Chapter 7 the PACS model detailed in Chapter 6 was used to predict ED and ES frames from the public EchoNet-Dynamic dataset, allowing for comparison and evaluation of the error between the ground-truth and network predictions trained on different datasets. This chapter addresses research questions 6 and 7, and research objectives 4 and 6.

Finally, **Chapter 8** presents an open-source automated method to make measurements on tissue Doppler traces, independent from an accompanying ECG signal. The performance of the proposed method was examined by comparisons to the gold-standard reference data provided by human experts. The inter-observer variability in this Chapter was similar to that reported in the recent literature [228], and the trained models' performance was indistinguishable.

Presently, no echocardiography dataset, and corresponding annotations specifically prepared for tissue Doppler measurements, is publicly available. The patient dataset used in Chapter 8 has therefore been made available to the research community, providing a benchmark for future studies. This Chapter addresses research question 8 and research objectives 5-7.

9.2 Future Work

9.2.1 Chapter 4: Echocardiographic View Classification

This chapter investigates view classification using a two-dimensional (2D) echocardiographic (echo) dataset. Three-dimensional (3D) echocardiography currently suffers from considerable reduction in frame rates and image quality, rendering it's adoption in clinical practice limited [93]. When such issues are resolved, classification of echo views could be applied to 3D. Until then, 2D echocardiography remains unrivalled. Additionally, the data used in this Chapter was acquired using ultrasound equipment from GE and Philips manufacturers. To be considered vendor-neutral, data captured using a wide range of manufacturers from multiple medical centres should be included. Thus, future work could focus on the collection and annotation of a large-scale, diverse echocardiographic view dataset from multiple vendors across several sites. For the classification results to be truly comparable, a dataset should be made public for evaluation of reported studies.

9.2.2 Chapter 5: Real-time Quality Assessment of Echocardiographic Images

Chapter 5 proposes an initial prototype for real-time echo image quality analysis via a DL model and a real-time web application. However, the A4C dataset used is not currently diverse enough to offer a thorough range of image qualities (from poor to excellent). Therefore, future work could centre around the curation of a greater range of image qualities, views and ultrasound manufacturers. The novel method of labelling image quality (by running inference on trained DL models) could also be expanded to encompass other analysis tasks, such as phase detection.

To be considered real-time, improvements need to be made to increase the inference time of the trained network. Thus speeding up the web application to render more results per second. This could be done by removing the classification branch of the network as the results detailed in this Chapter have proved it is not an effective method of assessing image quality and convert the model to a Tensorflow Lite instance

9.2.3 Chapter 6: Echocardiographic Phase Detection Using Deep Neural Networks

In clinical practice, many echo views are considered then computing complex diagnostic markers relating to cardiac function. In Chapter 6, only the A4C view was considered. In line with the research objectives of this project, and to improve the accuracy and efficiency of the network, future research could focus on expanding echocardiographic views. Additionally, the DL model was trained upon data from one centre, but tested upon data from multiple centres, from a range of ultrasound machines. An amalgamated dataset could be formed to be more representative of multi-centre data, thus more representative of current clinical practice.

9.2.4 Chapter 7: Left Ventricular Volume and Ejection Fraction Estimation With Deep Neural Networks

The volume computation in this Chapter is the Simpson's monoplane method which only considers images of the apical 4-chamber view. However, it may be possible to obtain more accurate volume estimations using Simpson's biplane method, using both apical 4-chamber and 2-chamber views. The accuracy of automated predictions can be significantly impacted when image quality is sub-optimal [61], as with some of the datasets used in this Chapter. To be able to create a clinically applicable, generalisable network, more multi-centre datasets would be required.

9.2.5 Chapter 8: Automated Tissue Doppler Echocardiography Analysis

The DL algorithm developed and proposed in Chapter 8 utilises the Mask R-CNN transfer learning approach. This network builds upon the Faster R-CNN [224] architecture by efficiently detecting objects in an image with the benefit of an ROIAlign module to produce an object segmentation mask, in parallel with the existing bounding box branch. There is future potential to extend the proposed network to other Doppler modalities (e.g. LV outflow tract) which would require tracing of the Doppler envelope, and thus, utilise the segmentation capabilities of the network.

Bibliography

[1] United Nations. *Population.* URL: https://www.un.org/en/global-issues/population. (accessed: 19.05.2022).

[2] G. Roth et al. “Demographic and Epidemiologic Drivers of Global Cardiovascular Mortality”. In: *New England Journal of Medicine* 372.14 (2015), pp. 1333–1341.

[3] E. Braunwald. “The war against heart failure: the Lancet lecture”. In: *The Lancet* 385.9970 (2015), pp. 812–824.

[4] J. Zhang et al. “Fully Automated Echocardiogram Interpretation in Clinical Practice”. In: *Circulation* 138.16 (2018), pp. 1623–1635.

[5] R. Luengo-Fernandez. “Cost of cardiovascular diseases in the United Kingdom”. In: *Heart* 92.10 (2006), pp. 1623–1635.

[6] Cebr. *The economic cost of cardiovascular disease from 2014-2020 in six European economies.* URL: https://cebr.com/reports/the-rising-cost-of-cvd/. (accessed: 16.06.2022).

[7] K. Siegersma et al. “Artificial intelligence in cardiovascular imaging: state of the art and implications for the imaging cardiologist”. In: *Netherlands Heart Journal* 27.9 (2019), pp. 403–413.

[8] Mayo Clinic. *Echocardiogram.* URL: https://www.mayoclinic.org/tests-procedures/echocardiogram/about/pac-20393856. (accessed: 01.07.2022).

[9] C. Knackstedt et al. “Fully Automated Versus Standard Tracking of Left Ventricular Ejection Fraction and Longitudinal Strain.” In: *Journal of the American College of Cardiology* 66.13 (2015), pp. 1456–1466.

[10] C Raynaud et al. “Handcrafted features vs ConvNets in 2D echocardiographic images”. In: 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017), 2017, pp. 1116–1119.

[11] X. Gao et al. “A fused deep learning architecture for viewpoint classification of echocardiography”. In: *Information Fusion* 36 (2017), pp. 103–113.

[12] A. Abdi et al. “Automatic quality assessment of apical four-chamber echocardiograms using deep convolutional neural networks”. In: 2017.

[13] Ho. Vaseli et al. “Designing lightweight deep learning models for echocardiography view classification”. In: 2019, p. 14.

[14] Østvik A et al. “Real-Time Standard View Classification in Transthoracic Echocardiography Using Convolutional Neural Networks”. In: *Ultrasound Med Biol.* 45(2) (2019), pp. 374–384.

[15] A. Chartsias et al. “Contrastive Learning for View Classification of Echocardiograms”. In: *CoRR* abs/2108.03124 (2021).

[16] N. Azarmehr et al. “Neural architecture search of echocardiography view classifiers”. In: *Journal of Medical Imaging* 8.3 (2021).

[17] J. Dong et al. “A Generic Quality Control Framework for Fetal Ultrasound Cardiac Four-Chamber Planes”. In: *IEEE J Biomed Health Inform.* (2020), pp. 931–942.

[18] DA. Tighe et al. “Influence of image quality on the accuracy of real time three-dimensional echocardiography to measure left ventricular volumes in unselected patients: a comparison with gated-SPECT imaging”. In: *Echocardiography* (2007), pp. 1073–80.

[19] A. Abdi et al. “Quality Assessment of Echocardiographic Cine Using Recurrent Neural Networks: Feasibility on Five Standard View Planes”. In: *Medical Image Computing and Computer Assisted Intervention MICCAI 2017*. Springer International Publishing, 2017, pp. 302–310.

[20] G. Toporek et al. “User Guidance for Point-of-Care Echocardiography Using a Multi-Task Deep Neural Network”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019: 22nd International Conference*. Springer-Verlag, 2019, pp. 309–317.

[21] D. Medvedofsky et al. “Three-Dimensional Echocardiographic Automated Quantification of Left Heart Chamber Volumes Using an Adaptive Analytics Algorithm: Feasibility and Impact of Image Quality in Nonselected Patients”. In: *J Am Soc Echocardiogr.* (2017), pp. 879–885.

[22] R. Labs et al. “Automated assessment of transthoracic echocardiogram image quality using deep neural networks”. In: *Intelligent Medicine* (2022).

[23] E. Lane et al. “Multibeat echocardiographic phase detection using deep neural networks”. In: *Computers in Biology and Medicine* 133 (2021), p. 104373.

[24] Badrinarayanan. V, Kendall. A, and Cipolla. R. “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.12 (2015), pp. 2481–2495.

[25] H. A. Omar et al. “Quantification of cardiac bull’s-eye map based on principal strain analysis for myocardial wall motion assessment in stress echocardiography”. In: *IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)* 3 (2018), pp. 1195–1198.

[26] R. M. Abazid et al. “Visual versus fully automated assessment of left ventricular ejection fraction”. In: *Avicenna journal of medicine* 8(2) (2018), pp. 41–45.

[27] P. Naidoo et al. “Influence of Loss Function on Left Ventricular Volume and Ejection Fraction Estimation in Deep Neural Networks”. In: Medical Imaging with Deep Learning (MIDL), 2022.

[28] C. A. Frederiksen et al. “Clinical utility of semi-automated estimation of ejection fraction at the point-of-care”. In: *Heart, lung and vessels* 7(3) (2015), pp. 208–216.

[29] S. Leclerc et al. “Deep Learning for Segmentation Using an Open Large-Scale Dataset in 2D Echocardiography”. In: *IEEE Trans Med Imaging* 38(9) (2019), pp. 2198–2210.

[30] O. Ronneberger, P. Fischer, and T. Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *CoRR* abs/1505.04597 (2015).

[31] C. Chen et al. “Deep Learning for Cardiac Image Segmentation: A Review”. In: *Frontiers in Cardiovascular Medicine* 7 (2020).

[32] E. Smistad et al. "Fully Automatic Real-Time Ejection Fraction and MAPSE Measurements in 2D Echocardiography Using Deep Neural Networks". In: *2018 IEEE International Ultrasonics Symposium (IUS)*. 2018, pp. 1–4.

[33] S. Leclerc et al. "Deep Learning Applied to Multi-Structure Segmentation in 2D Echocardiography: A Preliminary Investigation of the Required Database Size". In: *2018 IEEE International Ultrasonics Symposium (IUS)*. 2018, pp. 1–4.

[34] MH. Jafari et al. "Automatic biplane left ventricular ejection fraction estimation with mobile point-of-care ultrasound using multi-task learning and adversarial training". In: *Int J Comput Assist Radiol Surg* (2019), pp. 1027–1037.

[35] M. Zolgharni et al. "Feasibility of using a reliable automated Doppler flow velocity measurements for research and clinical practices". In: *Medical Imaging 2014: Ultrasonic Imaging and Tomography* (2014).

[36] G. Zamzmi et al. "Fully automated spectral envelope and peak velocity detection from Doppler echocardiography images". In: *Medical Imaging 2020: Computer-Aided Diagnosis* (2020).

[37] S. Zhu and R. Gao. "A novel generalized gradient vector flow snake model using minimal surface and component-normalized method for medical image segmentation". In: *Biomedical Signal Processing and Control* 26 (2016), pp. 1–10.

[38] H. Kalinić et al. "Image registration and atlas-based segmentation of cardiac outflow velocity profiles". In: *Computer Methods and Programs in Biomedicine* 106.3 (2012), pp. 188–200.

[39] V. Baličević et al. "A computational model-based approach for atlas construction of aortic Doppler velocity profiles for segmentation purposes". In: *Biomedical Signal Processing and Control* 40 (2018), pp. 23–32.

[40] J. Park et al. "Automatic Mitral Valve Inflow Measurements from Doppler Echocardiography". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2008* (2008), pp. 983–990.

[41] S. Zhou et al. "A probabilistic, hierarchical, and discriminant framework for rapid and accurate detection of deformable anatomic structure". In: *2007 IEEE 11th International Conference on Computer Vision* (2007).

[42] N. Kiruthika, B. Prabhakar, and M. Reddy. "Automated Assessment of Aortic Regurgitation using 2D Doppler Echocardiogram". In: *Proceedings of the 2006 IEEE International Workshop on Imagining Systems and Techniques (IST 2006)* (2006).

[43] A. Taebi et al. "Estimating Peak Velocity Profiles from Doppler Echocardiography using Digital Image Processing". In: *2018 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)* (2018).

[44] N. Dhutia et al. "Open-source, vendor-independent, automated multi-beat tissue Doppler echocardiography analysis". In: *The International Journal of Cardiovascular Imaging* 33.8 (2017), pp. 1135–1148.

[45] H. Kalinic et al. "Model-based segmentation of aortic ultrasound images". In: *In: ISPA 2011 - 7th International Symposium on Image and Signal Processing and Analysis* (2011), pp. 739–743.

[46] M. Higa et al. "A computational tool for quantitative assessment of peripheral arteries in ultrasound images". In: *In: 2009 36th Annual Computers in Cardiology Conference (CinC)* (2009), pp.41–44.

[47] V. Magagnin et al. "Semi-Automated Analysis of Coronary Flow Doppler Images: Validation with Manual Tracings". In: *2006 International Conference of the IEEE Engineering in Medicine and Biology Society* (2006).

[48] H. Greenspan et al. "Doppler echocardiography flow-velocity image analysis for patients with atrial fibrillation". In: *Ultrasound in Medicine Biology* 31.8 (2005), pp. 1031–1040.

[49] E. Gaillard et al. "Optimization of Doppler Echocardiographic Velocity Measurements Using an Automatic Contour Detection Method". In: *Ultrasound in Medicine Biology* 36.9 (2010), pp. 1513–1524.

[50] A. Testuz et al. "Diagnostic accuracy of pocket-size handheld echocardiographs used by cardiologists in the acute care setting." In: *European Heart Journal - Cardiovascular Imaging* 14.1 (2012), pp. 38–42.

[51] M. Zolgharni et al. “Automatic detection of end-diastolic and end-systolic frames in 2D echocardiography”. In: *Echocardiography* 34.7 (2017), pp. 957–967.

[52] C. Mitchell et al. “Guidelines for Performing a Comprehensive Transthoracic Echocardiographic Examination in Adults: Recommendations from the American Society of Echocardiography”. In: *J Am Soc Echocardiography* 32(1) (2019), pp. 1–64.

[53] F. Wegner et al. “Accuracy of Deep Learning Echocardiographic View Classification in Patients with Congenital or Structural Heart Disease: Importance of Specific Datasets”. In: *Journal of Clinical Medicine* 11(3) (2022).

[54] M. Kurt et al. “Impact of contrast echocardiography on evaluation of ventricular function and clinical management in a large prospective cohort”. In: *J Am Coll Cardiol.* (2009), pp. 802–810.

[55] JC. Plana et al. “Expert consensus for multimodality imaging evaluation of adult patients during and after cancer therapy: a report from the American Society of Echocardiography and the European Association of Cardiovascular Imaging”. In: *J Am Soc Echocardiogr.* (2014), pp. 911–39.

[56] LD. Jacobs et al. “Rapid online quantification of left ventricular volume from real-time three-dimensional echocardiographic data”. In: *Eur Heart J.* (2006), pp. 460–468.

[57] Y. Nagata et al. “Impact of image quality on reliability of the measurements of left ventricular systolic function and global longitudinal strain in 2D echocardiography”. In: *Echo Res Pract* 5 (2018), pp. 28–39.

[58] S. Darvishi et al. “Measuring Left Ventricular Volumes in Two-Dimensional Echocardiography Image Sequence Using Level-set Method for Automatic Detection of End-Diastole and End-systole Frames”. In: *Research in Cardiovascular Medicine* 1.2 (2012), pp. 39–45.

[59] R. Mada et al. “How to Define End-Diastole and End-Systole?” In: *JACC: Cardiovascular Imaging* 8.2 (2015), pp. 148–157.

[60] B. Amundsen. “It Is All About Timing!” In: *JACC: Cardiovascular Imaging* 8.2 (2015), pp. 158–160.

[61] X. Liu et al. “Deep learning-based automated left ventricular ejection fraction assessment using 2-D echocardiography”. In: *Am J Physiol Heart Circ Physiol.* 321(2) (2021), pp. 390–399.

[62] M. Cameli et al. “Echocardiographic assessment of left ventricular systolic function: from ejection fraction to torsion”. In: *Heart Fail Rev.* 21(1) (2016), pp. 77–94.

[63] E. Lui et al. “Human factors as a source of error in peak Doppler velocity measurement”. In: *Journal of Vascular Surgery* 42.5 (2005), pp. 972–972.

[64] S. Nagueh et al. “Recommendations for the Evaluation of Left Ventricular Diastolic Function by Echocardiography: An Update from the American Society of Echocardiography and the European Association of Cardiovascular Imaging”. In: *Journal of the American Society of Echocardiography* 29.4 (2016), pp. 277–314.

[65] S. Nagueh et al. “Recommendations for the Evaluation of Left Ventricular Diastolic Function by Echocardiography”. In: *European Journal of Echocardiography* 10.2 (2008), pp. 165–193.

[66] T. Matthew et al. “A Guideline Protocol for the Echocardiographic assessment of Diastolic Dysfunction”. In: *British Society of Echocardiography* (2013), pp. 1–6.

[67] A. Voulodimos et al. “Deep Learning for Computer Vision: A Brief Review”. In: *Computational Intelligence and Neuroscience,* (2018), pp. 1–13.

[68] J. Howard et al. “Improving ultrasound video classification: an evaluation of novel deep learning methods in echocardiography”. In: *Journal of Medical Artificial Intelligence* 3 (2020), pp. 4–4.

[69] H. Baumgartner et al. “Recommendations on the echocardiographic assessment of aortic valve stenosis: a focused update from the European Association of Cardiovascular Imaging and the American Society of Echocardiography”. In: *European Heart Journal - Cardiovascular Imaging* 18.3 (2016), pp. 254–275.

[70] A. Kosaraju et al. “Left Ventricular Ejection Fraction”. In: *Treasure Island (FL): StatPearls Publishing* (2022).

[71] R. Lang et al. “Recommendations for chamber quantification”. In: *European Journal of Echocardiography* 7.2 (2006), pp. 79–108.

[72] RM. Lang et al. “Recommendations for cardiac chamber quantification by echocardiography in adults: an update from the American Society of Echocardiography and the European Association of Cardiovascular Imaging”. In: *Eur. Heart J.-Cardiovasc. Imaging* 16(3) (2015), pp. 233–271.

[73] F. Flachskampf and M. Martensson. “How should tissue Doppler tracings be measured?” In: *European Heart Journal - Cardiovascular Imaging* 15.7 (2014), pp. 828–829.

[74] N. Dhutia et al. “Guidance for accurate and consistent tissue Doppler velocity measurement: comparison of echocardiographic methods using a simple vendor-independent method for local validation”. In: *European Heart Journal - Cardiovascular Imaging* 15.7 (2014), pp. 817–827.

[75] J. Finegold et al. “Choosing between velocity-time-integral ratio and peak velocity ratio for calculation of the dimensionless index (or aortic valve area) in serial follow-up of aortic stenosis”. In: *International Journal of Cardiology* 167.4 (2013), pp. 1524–1531.

[76] G. Gyurkovics. *Human Biology.* URL: `https://bio.libretexts.org/Courses/Community_College_of_Vermont/Human_Biology_(Gabor_Gyurkovics)`. (accessed: 18.07.2022).

[77] Institute for Quality and Efficiency in Health Care. *How does the blood circulatory system work?* Cologne, Germany: National Library of Medicine (NLM), 2019.

[78] G. Buckley. *Cardiac Cycle.* URL: `https://biologydictionary.net/cardiac-cycle/`. (accessed: 18.07.2022).

[79] S. Jarvis and S. Saman. “Cardiac System 1: anatomy and physiology”. In: *Nursing Times* 114.2 (2018), pp. 34–37.

[80] Texas Heart Institute. *Heart Anatomy.* URL: `https://www.texasheart.org/heart-health/heart-information-center/topics/heart-anatomy/`. (accessed: 18.07.2022).

[81] National Health Service (NHS). *NHS Long Term Plan.* URL: `https://www.longtermplan.nhs.uk/`. (accessed: 18.07.2022).

[82] M. Hoffman. *Human Anatomy.* URL: `https://www.webmd.com/heart/picture-of-the-heart`. (accessed: 18.07.2022).

[83] D. Katritsis, B. Gersh, and A. Camm. *Clinical cardiology: Current Practice Guidelines.* Oxford, England: Oxford: Oxford University Press, 2021.

[84] R. Rehman, Yelamanchili. V, and Makaryus A. *Cardiac Imaging.* Floria, USA: StatPearls, 2022.

[85] Mayo Clinic. *CT Coronary Angiogram.* URL: `https://www.mayoclinic.org/tests-procedures/ct-coronary-angiogram/about/pac-20385117#:~:text=A%5C%20computerized%5C%20tomography%5C%20(CT)%5C%20coronary,a%5C%20variety%5C%20of%5C%20heart%5C%20conditions.`. (accessed: 04.08.2022).

[86] Heart.org. *Magnetic Resonance Imaging (MRI).* URL: `https://www.heart.org/en/health-topics/heart-attack/diagnosing-a-heart-attack/magnetic-resonance-imaging-mri`. (accessed: 04.08.2022).

[87] AA. Mohamed, AA. Arifi, and A. Omran. "The basics of echocardiography". In: *J Saudi Heart Assoc* 22.2 (2010), pp. 71–76.

[88] WP. Mason. "Hysteresis losses in solid materials". In: *Piezoelectric crystals and their application in ultrasonics* (1950).

[89] Adobe. *Echocardiogram.* URL: `https://stock.adobe.com/uk/search?k=echocardiogram&asset_id=387784829`. (accessed: 03.08.2022).

[90] ECG Waves. *Clinical Echocardiography.* URL: `https://ecgwaves.com/course/clinical-echocardiography/`. (accessed: 04.08.2022).

[91] MD. Cerquira. "American Heart Association Writing Group on Myocardial Segmentation and Registration for Cardiac Imaging : Standardized myocardial segmentation and nomenclature for tomographic imaging of the heart : A statement for healthcare professionals from the Cardiac Imaging Committee of the Council on Clinical Cardiology of the American Heart Association". In: *Circulation* 105 (2002), pp. 539–542.

[92] e-Echocardiography. *Transthoratic Views.* URL: `https://e-echocardiography.com/page/page.php?UID=1429484711`. (accessed: 04.08.2022).

[93] K. Cheng et al. "3D echocardiography: benefits and steps to wider implementation". In: *British Journal of Cardiology* (2018).

[94] A. Lange et al. "Three-dimensional echocardiography: Historical development and current applications". In: *Journal of the American Society of Echocardiography* 14.5 (2001), pp. 403–412.

[95] B. Kong et al. "Recognizing End-Diastole and End-Systole Frames via Deep Temporal Regression Network". In: *Lecture Notes in Computer Science* (2016), pp. 264–272.

[96] F. Taheri Dezaki et al. "Cardiac Phase Detection in Echocardiograms With Densely Gated Recurrent Neural Networks and Global Extrema Loss". In: *IEEE Transactions on Medical Imaging* 38.8 (2019), pp. 1821–1832.

[97] Regunath. G. *Advancing Analytics.* URL: `https://www.advancinganalytics.co.uk/blog/2021/12/15/understanding-the-difference-between-ai-ml-and-dl-using-an-incredibly-simple-example`. (accessed: 21.07.2022).

[98] *The Oxford English Dictionary.* New York, USA: Oxford University Press, 2009.

[99] Merriam-Webster. *Artificial Intelligence.* URL: `https://www.merriam-webster.com/dictionary/artificial%5C%20intelligence`. (accessed: 21.07.2022).

[100] F. Chollet. *Deep learning with Python.* New York, USA: Manning Publications, 2018.

[101] A. Samuel. "Some studies in machine learning using the game of checkers". In: *IBM Journal of Research and Development* 44.1.2 (1959), pp. 206–226.

[102] IBM. *Deep Learning.* URL: `https://www.ibm.com/cloud/learn/deep-learning`. (accessed: 21.07.2022).

[103] IBM. *Neural Networks.* URL: `https://www.ibm.com/cloud/learn/neural-networks`. (accessed: 21.07.2022).

[104] Goodfellow. I, Bengio. Y, and Courville. A. *Deep Learning.* http://www.deeplearningbook.org. MIT Press, 2016.

[105] Mishra. M. *Neural Networks Explained.* URL: `https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939`. (accessed: 22.07.2022).

[106] Amini. A. *Deep Computer Vision.* http://introtodeeplearning.com/slides/6S191$_{M}IT_{D}eepLearning_{L}$
2022. (accessed: 22.07.2022).

[107] Saha. S. *A Comprehensive Guide to Convolutional Neural Networks*. URL: `https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53`. (accessed: 22.07.2022).

[108] Muhamad Yani, S Irawan, and Casi Setianingsih. "Application of Transfer Learning Using Convolutional Neural Network Method for Early Detection of Terry's Nail". In: *Journal of Physics: Conference Series* 1201 (2019), p. 012052.

[109] Basavarajaiah. M. *Maxpooling vs minpooling vs average pooling*. URL: `https://medium.com/@bdhuma/which-pooling-method-is-better-maxpooling-vs-minpooling-vs-average-pooling-95fb03f45a9`. (accessed: 23.07.2022).

[110] Shubham. J. *An Overview of Regularization Techniques in Deep Learning (with Python code)*. URL: `https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/`. (accessed: 26.07.2022).

[111] Keras. *tf.keras.preprocessing.image.ImageDataGenerator*. URL: `https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator`. (accessed: 24.08.2022).

[112] Imgaug. *Imgaug*. URL: `https://imgaug.readthedocs.io/en/latest/`. (accessed: 24.08.2022).

[113] R. Holbrook and A. Cook. *Overfitting and Underfitting*. URL: `https://www.kaggle.com/code/ryanholbrook/overfitting-and-underfitting/tutorial`. (accessed: 24.08.2022).

[114] Keras. *Callbacks API*. URL: `https://keras.io/api/callbacks/`. (accessed: 24.08.2022).

[115] L Ping et al. "Towards Understanding Regularization in Batch Normalization". In: *CoRR* abs/1809.00846 (2018).

[116] S. Ioffe and C. Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *CoRR* abs/1502.03167 (2015).

[117] Keras. *Module: tf.keras.applications*. URL: `https://www.tensorflow.org/api_docs/python/tf/keras/applications`. (accessed: 26.07.2022).

[118] Y. Lecun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[119] Y. Lecun. *The MNIST Database of Handwritten Digits.* URL: http://yann.lecun.com/exdb/mnist/. (accessed: 27.07.2022).

[120] Stanford Vision Lab. *ImageNet.* URL: https://image-net.org/index.php. (accessed: 26.07.2022).

[121] A. Krizhevsky, I. Sutskever, and G. Hinton. "ImageNet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (2012), pp. 84–90.

[122] Simonyan. K and Zisserman. A. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *ICLR* (2014).

[123] Zheng. Y, Yang. C, and Merkulov. A. "Breast cancer screening using convolutional neural network and follow-up digital mammography". In: (2018).

[124] Szegedy. C et al. "Going Deeper With Convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015).

[125] He. K et al. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[126] Google. *Classification: ROC Curve and AUC.* URL: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc#:~:text=An%5C%20ROC%5C%20curve%5C%20(receiver%5C%20operating,False%5C%20Positive%5C%20Rate. (accessed: 27.07.2022).

[127] Russakovsky O et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252.

[128] Ronneberger. O, Fischer. P, and Brox. T. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Lecture Notes in Computer Science* (2015), pp. 234–241.

[129] Chen. L et al. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (2017), pp. 834–848.

[130] Girshick. R et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *CoRR* abs/1311.2524 (2013).

[131] Girshick. Ross. "Fast R-CNN". In: *CoRR* abs/1504.08083 (2015).

[132] R. Shaoqing et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *CoRR* abs/1506.01497 (2015).

[133] Redmon. J et al. “You Only Look Once: Unified, Real-Time Object Detection”. In: *CoRR* abs/1506.02640 (2015).

[134] Redmon. J and Farhadi. A. “YOLO9000: Better, Faster, Stronger”. In: *CoRR* abs/1612.08242 (2016).

[135] Redmon. J and Farhadi. A. “YOLOv3: An Incremental Improvement”. In: *CoRR* abs/1804.02767 (2018).

[136] OpenCV. *Intersection over Union (IoU) in Object Detection and Segmentation.* URL: `https://learnopencv.com/intersection-over-union-iou-in-object-detection-and-segmentation/#IoU-in-Image-Segmentattion`. (accessed: 02.12.2022).

[137] W. Zaremba, I. Sutskever, and O. Vinyals. “Recurrent Neural Network Regularization”. In: *CoRR* abs/1409.2329 (2014).

[138] H. Sak, A. Senior, and F. Beaufays. “Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition”. In: *CoRR* abs/1402.1128 (2014).

[139] K. Greff et al. “LSTM: A Search Space Odyssey”. In: *CoRR* abs/1503.04069 (2015).

[140] M. Schuster and K.K. Paliwal. “Bidirectional recurrent neural networks”. In: *IEEE Transactions on Signal Processing* 45.11 (1997), pp. 2673–2681.

[141] K. Korakitis et al. *State of the Developer Nation.* Tech. rep. 22. London, England: SlashData Ltd, 2022.

[142] Python. *About Python.* URL: `https://www.python.org/about/`. (accessed: 20.07.2022).

[143] Tensorflow. *Tensorflow Core — Machine Learning For Beginners And Experts.* URL: `https://www.tensorflow.org/overview`. (accessed: 04.07.2022).

[144] Keras. *Keras.* URL: `https://keras.io/`. (accessed: 21.07.2022).

[145] Pytorch. *Pytorch.* URL: `https://pytorch.org/`. (accessed: 21.07.2022).

[146] M Wasfy and M. Picard. *Transthoracic Echocardiography: Nomenclature and Standard Views.* URL: `https://thoracickey.com/transthoracic-echocardiography-4/`. (accessed: 08.09.2022).

[147] S.K. Zhou et al. "Image-Based Multiclass Boosting and Echocardiographic View Classification". In: 2. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), 2006, pp. 1559–1565.

[148] J. H. Park et al. "Automatic Cardiac View Classification of Echocardiogram". In: 2013 IEEE 10th International Symposium on Biomedical Imaging, 2007, pp. 1–8.

[149] D. Agarwal, K S. Shriram, and N. Subramanian. "Automatic view classification of echocardiograms using Histogram of Oriented Gradients". In: 2013 IEEE 10th International Symposium on Biomedical Imaging, 2013, pp. 11368–1371.

[150] R. Kumar et al. "Echocardiogram view classification using edge filtered scale-invariant motion features". In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 723–730.

[151] A. Madani et al. "Fast and accurate view classification of echocardiograms using deep learning". In: *NPJ digital medicine* 1.6 (2018).

[152] G. Litjens et al. "A Survey on Deep Learning in Medical Image Analysis". In: *CoRR* abs/1702.05747 (2017).

[153] S. Ebadollahi, S. Chang, and H. Wu. "Automatic view recognition in echocardiogram videos using parts-based representation". In: 2. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004, pp. II–2.

[154] J A. Roy et al. "State-Based Modeling and Object Extraction From Echocardiogram Video". In: *IEEE Transactions on Information Technology in Biomedicine* 12(3) (2008), pp. 366–376.

[155] R. Deo et al. "An End-to-End Computer Vision Pipeline for Automated Cardiac Function Assessment by Echocardiography". In: (2017).

[156] B. Georgescu et al. "Database-guided segmentation of anatomical structures with complex appearance". In: vol. 2. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005, pp. 429–436.

[157] G. Allan et al. "Simultaneous Analysis of 2D Echo Views for Left Atrial Segmentation and Disease Detection". In: *IEEE transactions on medical imaging* 36(1) (2017), pp. 40–50.

[158] H. Khamis et al. "Automatic apical view classification of echocardiograms using a discriminative learning dictionary". In: *Med Image Anal.* 36 (2017), pp. 15–21.

[159] SR. Snare et al. "Real-time scan assistant for echocardiography". In: *IEEE Trans Ultrason Ferroelectr Freq Control* 59(3) (2012), pp. 583–589.

[160] H. Pham et al. "Efficient Neural Architecture Search via Parameters Sharing". In: 2018. URL: http://proceedings.mlr.press/v80/pham18a/pham18a.pdf.

[161] H. Hanxiao Liu, K. Karen Simonyan, and Y. Yang. "DARTS: Differentiable Architecture Search". In: *CoRR* abs/1806.09055 (2018).

[162] K. Huang et al. "Artificial Intelligence Aids Cardiac Image Quality Assessment for Improving Precision in Strain Measurements". In: *JACC Cardiovasc Imaging* (2021), pp. 335–345.

[163] D. Kingma and 2015 Ba J. "Adam: A Method For Stochastic Optimization". In: *ICLR* (2015), pp. 1–15.

[164] Scikit Learn. *3.3. Metrics and scoring: quantifying the quality of predictions.* URL: https://scikit-learn.org/stable/modules/model_evaluation.html. (accessed: 08.09.2022).

[165] J. Zhang et al. "Fully Automated Echocardiogram Interpretation in Clinical Practice". In: *Circulation* 138(16) (2018), pp. 1623–1635.

[166] Z. Liao et al. "On Modelling Label Uncertainty in Deep Neural Networks: Automatic Estimation of Intra- Observer Variability in 2D Echocardiography Quality Assessment". In: *IEEE Transactions on Medical Imaging* 39.6 (2020), pp. 1868–1883.

[167] siemens. *ACUSON Freestyle Series Ultrasound Systems.* URL: https://www.siemens-healthineers.com/en-us/ultrasound/ultrasound-point-of-care/acuson-freestyle-ultrasound-machine. (accessed: 23.09.2022).

[168] GE Healthcare. *Vscan Extend Handheld Ultrasound.* URL: https://www.gehealthcare.co.uk/en/products/ultrasound/vscan-family/vscan-extend. (accessed: 23.09.2022).

[169] Clarius. *Clarius HD3 Ultra-portable Ultrasound*. URL: https://clarius.com/. (accessed: 23.09.2022).

[170] Philips. *Lumify*. URL: https://www.philips.co.uk/healthcare/product/HCNOCTN481/lumify-exceptional-ultrasound-from-your-smart-device. (accessed: 23.09.2022).

[171] M. Chamsi-Pasha, P. Sengupta, and W. Zoghbi. "Handheld Echocardiography". In: *Circulation* 136.22 ().

[172] N. Van Woudenberg et al. "Quantitative Echocardiography: Real-Time Quality Estimation and View Classification Implemented on a Mobile Android Device". In: *POCUS/BIVPCS/CuR* 2018.

[173] Unity Imaging. *Unity Imaging*. URL: https://unityimaging.net. (accessed: 22.09.2022).

[174] V. Belagiannis and A. Zisserman. "Recurrent Human Pose Estimation". In: *12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)* (2017).

[175] Flask. *Welcome to Flask*. URL: https://flask.palletsprojects.com/en/2.2.x/#. (accessed: 08.10.2022).

[176] Jinja. *Jinja*. URL: https://jinja.palletsprojects.com/en/3.1.x/. (accessed: 08.10.2022).

[177] Werkzeug. *Werkzeug*. URL: https://werkzeug.palletsprojects.com/en/2.2.x/. (accessed: 08.10.2022).

[178] Epiphan. *VGA2USB*. URL: https://www.epiphan.com/products/vga2usb/. (accessed: 08.10.2022).

[179] OpenCV. *OpenCV*. URL: https://github.com/opencv/opencv-python. (accessed: 08.10.2022).

[180] Imutils. *Imutils*. URL: https://github.com/PyImageSearch/imutils. (accessed: 08.10.2022).

[181] Turbo-Flask. *Turbo-Flask*. URL: https://turbo-flask.readthedocs.io/en/latest/. (accessed: 08.10.2022).

[182] Chart.js. *Chart.js*. URL: https://www.chartjs.org/. (accessed: 08.10.2022).

[183] Mayo Clinic. *Cardiac cycle*. URL: `https://en.wikipedia.org/wiki/Cardiac_cycle#/media/File:Wiggers_Diagram_2.svg`. (accessed: 01.07.2022).

[184] M. Zolgharni et al. "Automated Aortic Doppler Flow Tracing for Reproducible Research and Clinical Measurements." In: *IEEE Transactions on Medical Imaging* 33.5 (2014), pp. 1071–1082.

[185] T. Jahren et al. "Estimation of End-Diastole in Cardiac Spectral Doppler Using Deep Learning." In: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 67.12 (2020), pp. 2605–2614.

[186] A. Shalbaf et al. "Automatic detection of end systole and end diastole within a sequence of 2-D echocardiographic images using modified Isomap algorithm." In: *2011 1st Middle East Conference on Biomedical Engineering* (2011).

[187] S. Aase et al. "Echocardiography without electrocardiogram". In: *European Journal of Echocardiography* 12.1 (2010), pp. 3–10.

[188] S. Tridandapani, J. Fowlkes, and J. Rubin. "Echocardiography-Based Selection of Quiescent Heart Phases". In: *Journal of Ultrasound in Medicine* 24.11 (2005), pp. 1519–1526.

[189] C. Wick et al. "Detection of Cardiac Quiescence From B-Mode Echocardiography Using a Correlation-Based Frame-to-Frame Deviation Measure". In: *IEEE Journal of Translational Engineering in Health and Medicine* 1 (2013), pp. 1900211–1900211.

[190] L. Ravichandran et al. "Detection of Quiescent Cardiac Phases in Echocardiography Data Using Nonlinear Filtering and Boundary Detection Techniques". In: *Journal of Digital Imaging* 27.5 (2014), pp. 625–632.

[191] F. Dezaki et al. "Deep Residual Recurrent Neural Networks for Characterisation of Cardiac Cycle Phase from Echocardiograms". In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support* (2017), pp. 100–108.

[192] A. Fiorito et al. "Detection of Cardiac Events in Echocardiography Using 3D Convolutional Recurrent Neural Networks". In: *2018 IEEE International Ultrasonics Symposium (IUS)* (2018).

[193] P. Pabari et al. “When is an optimization not an optimization? Evaluation of clinical implications of information content (signal-to-noise ratio) in optimization of cardiac resynchronization therapy, and how to measure and maximize it”. In: *Heart Failure Reviews* 16.3 (2010), pp. 277–290.

[194] M. Moraldo et al. “Evidence-based recommendations for PISA measurements in mitral regurgitation: systematic review, clinical and in-vitro study”. In: *International Journal of Cardiology* 168.2 (2013), pp. 1220–1228.

[195] M. Shun-Shin and D. Francis. “Why Are Some Studies of Cardiovascular Markers Unreliable? The Role of Measurement Variability and What an Aspiring Clinician Scientist Can Do Before It Is Too Late”. In: *Progress in Cardiovascular Diseases* 55.1 (2012), pp. 14–24.

[196] J. Zech et al. “Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study”. In: *PLOS Medicine* 15.11 (2018), e1002683.

[197] Renal Fellow Network. *Focused Cardiac Ultrasound for the Nephrologist: The apical window - Renal Fellow Network.* URL: `https://www.renalfellow.org/2019/09/20/focused-cardiac-ultrasound-for-the-nephrologist-the-apical-window/`. (accessed: 02.07.2022).

[198] Echonet Dynamic. *Echonet Dynamic.* URL: `https://echonet.github.io/dynamic/`. (accessed: 02.07.2022).

[199] D. Ouyang et al. “Video-based AI for beat-to-beat assessment of cardiac function”. In: *Nature* 580(7802) (2020), pp. 252–256.

[200] K. He et al. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[201] C. Szegedy et al. “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence, 31(1)* (2017).

[202] Grepmed. *Apical Windows - POCUS Echocardiogram Anatomy - Apical*. URL: https://www.grepmed.com/images/11120/apical2chamber-echocardiogram-apical4chamber-windows-anatomy-pocus-apex. (accessed: 04.07.2022).

[203] E. Maret et al. "Computer-assisted determination of left ventricular endocardial borders reduces variability in the echocardiographic assessment of ejection fraction". In: *Cardiovascular Ultrasound* (2008), pp. 1476–7120.

[204] MT. Nolan and P. Thavendiranathan. "Automated Quantification in Echocardiography". In: *JACC Cardiovasc Imaging* 12(6) (2019), pp. 1073–1092.

[205] V. Tavakoli and A. Amini. "A survey of shaped-based registration and segmentation techniques for cardiac images". In: *Computer Vision and Image Understanding* 117 (2013).

[206] Y Hu et al. "AIDAN: An Attention-Guided Dual-Path Network for Pediatric Echocardiography Segmentation". In: *IEEE Access* 8 (2020), pp. 29176–29187.

[207] Ö. Çiçek et al. "3D U Net: learning dense volumetric segmentation from sparse annotation". In: *19th International Conference on Medical Image Computing and Computer Assisted Intervention—MICCAI* (2016), pp. 424–32.

[208] M. Jafari et al. "A Unified Framework Integrating Recurrent Fully-Convolutional Networks and Optical Flow for Segmentation of the Left Ventricle in Echocardiography Data". In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer International Publishing, 2018, pp. 29–37.

[209] Stanford AIMI Center. *EchoNet-Dynamic Cardiac Ultrasound*. URL: https://aimi.stanford.edu/echonet-dynamic-cardiac-ultrasound. (accessed: 21.09.2022).

[210] S. Salehi, D. Erdogmus, and A. Gholipour. "Tversky loss function for image segmentation using 3D fully convolutional deep networks". In: *International workshop on machine learning in medical imaging*. Springer. 2017, pp. 379–387.

[211] C. Baumgartner et al. "An Exploration of 2D and 3D Deep Learning Techniques for Cardiac MR Image Segmentation". In: *Statistical Atlases and Computational Models of the Heart. ACDC and MMWHS Challenges*. Springer International Publishing, 2018, pp. 111–119.

[212] M. Quiñones et al. “Recommendations for quantification of Doppler echocardiography: A report from the Doppler quantification task force of the nomenclature and standards committee of the American Society of Echocardiography”. In: *Journal of the American Society of Echocardiography* 15.2 (2002), pp. 167–184.

[213] C. Ho and S. Solomon. “A Clinician’s Guide to Tissue Doppler Imaging”. In: *Circulation* 113.10 (2006).

[214] M. Cikes and S. Solomon. “Beyond ejection fraction: an integrative approach for assessment of cardiac structure and function in heart failure”. In: *European Heart Journal* 37.21 (2015), pp. 1642–1650.

[215] K. Kadappu and L. Thomas. “Tissue Doppler Imaging in Echocardiography: Value and Limitations”. In: *Heart Lung Circulation* 24.3 (2015), pp. 224–233.

[216] N. Nikitin. “Prognostic value of systolic mitral annular velocity measured with Doppler tissue imaging in patients with chronic heart failure caused by left ventricular systolic dysfunction”. In: *Heart* 92.6 (2005), pp. 775–779.

[217] J. Meluzín. “Pulsed Doppler tissue imaging of the velocity of tricuspid annular systolic motion. A new, rapid, and non-invasive method of evaluating right ventricular systolic function”. In: *European Heart Journal* 22.4 (2001), pp. 340–348.

[218] T. Yamamoto et al. “Prognostic value of the atrial systolic mitral annular motion velocity in patients with left ventricular systolic dysfunction”. In: *Journal of the American Society of Echocardiography* 16.4 (2003), pp. 333–339.

[219] N. Biradar, M. Dewal, and M. Kumar Rohit. “Automated delineation of Doppler echocardiographic images using texture filters”. In: *International Conference on Computing for Sustainable Global Development (INDIACom)* (2015), pp. 1903–1907.

[220] G. Zamzmi et al. “Harnessing Machine Intelligence in Automatic Echocardiogram Analysis: Current Status, Limitations, and Future Directions”. In: *IEEE Reviews in Biomedical Engineering* 14 (2021), pp. 181–203.

[221] M. Zolgharni et al. “Automated Aortic Doppler Flow Tracing for Reproducible Research and Clinical Measurements”. In: *IEEE Transactions on Medical Imaging* 33.5 (2014), pp. 1071–1082.

[222] K. He et al. "Mask R-CNN". In: *IEEE International Conference on Computer Vision (ICCV)* (2017).

[223] Matterport. *GitHub - matterport/Mask*$_R CNN : MaskR{-}CNNforobjectdetectionandinstancesegm$
URL: `https://github.com/matterport/Mask_RCNN`. (accessed: 13.07.2022).

[224] S. Ren et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (2017), pp. 1137–1149.

[225] Cocodataset.org. *COCO - Common Objects in Context*. URL: `https://cocodataset.org`. (accessed: 13.07.2022).

[226] R. Girshick. "Fast R-CNN". In: *2015 IEEE International Conference on Computer Vision (ICCV)* (2015).

[227] S. Sacchi et al. "Doppler assessment of aortic stenosis: a 25-operator study demonstrating why reading the peak velocity is superior to velocity time integral". In: *European Heart Journal - Cardiovascular Imaging* 19.12 (2018), pp. 1380–1389.

[228] D. Vinereanu, A. Khokhar, and A. Fraser. "Reproducibility of Pulsed Wave Tissue Doppler Echocardiography". In: *Journal of the American Society of Echocardiography* 12.6 (1999), pp. 492–499.