



An Investigation into Automating Cardiac MRI Planning Using Deep Learning

Dissertation submitted in partial fulfilment of the requirements of the
Degree of Applied Medical Sciences
University College London
2019/20
UCL Division of Medicine
Faculty of Medical Sciences

Abstract

Cardiac Magnetic Resonance (CMR) imaging is invaluable for evaluating a wide range of heart diseases. The CMR protocol involves placing the patient in an MRI scanner and acquiring cardiac images in several 2D views. To plan these views, a radiographer draws lines through standard cardiac landmarks, which slice the image accordingly. The process is time consuming, requires experts who are not available at every centre and if performed poorly, can impact clinical decisions. This project aims to automate the planning process and overcome these issues.

We explored the use of deep learning (DL) to automate CMR planning. Two approaches were investigated for solving this task— a direct parameter estimation approach and a mask-based segmentation approach. For parameter estimation, a deep convolutional neural network (CNN) was trained to estimate the parameter values that define the planning line equation. For the segmentation approach, a second CNN was trained to predict a binary mask of the line. As proof of concept, we focused on predicting a two-chamber (2Ch) planning line given a four-chamber (4Ch) image. Models were trained on a dataset of 485 patient images and validated on an independent holdout set of 122 images. Model performance was evaluated using the root mean squared distance (RMSD) between the predicted and ground truth line.

Qualitative assessment showed that the parameter estimation approach did not produce plausible lines, but the segmentation approach did. Direct parameter estimation achieved an RMSD of 30.21 ± 27.03 mm while segmentation achieved an RMSD of 5.70 ± 3.77 mm, representing a five-fold increase in performance($p<0.05$).

Declaration of Contribution

I hereby declare that this dissertation is an original report of my research, has been written by me and has not been submitted for any previous degree. This project was designed and supervised by Dr. Rhodri Davies. The datasets used for conducting experimental work were collected and preprocessed by Dr. Davies. All experimental work including implementing the models and computational frameworks and analyzing the data was performed by myself. Due references have been provided on all supporting literatures and resources. Dr. Davies provided critical feedback and helped shape the research, analysis and this final report.

Table of Contents

Abstract	3
Declaration of Contribution	4
Abbreviations	7
1 INTRODUCTION	8
1.1 <i>How is a CMR study performed?</i>	8
1.1.1 Cardiac Imaging Planes.....	9
1.1.2 CMR Protocol	10
1.2 <i>Disadvantages of CMR protocol - Motivations for study</i>	12
2 AIMS AND OBJECTIVES	13
2.1 <i>Previous Studies</i>	13
3 METHODOLOGY	16
3.1 <i>Patients and Dataset Description</i>	16
3.2 <i>Two approaches to planning line prediction</i>	17
3.2.1 Parameter Estimation	17
3.2.2 Segmentation.....	18
3.3 <i>Neural Network Architectures</i>	19
3.3.1 UNet.....	19
3.3.2 AlexNet	20
3.4 <i>Model Training:</i>	21
3.4.1 Dataset Split	21
3.4.2 Loss functions	22

3.4.3	Accuracy metric for UNet training	22
3.4.4	Other Hyperparameters	23
3.5	<i>Software Used</i>	23
3.6	<i>Reconstruction of UNet predicted masks</i>	23
3.7	<i>Accuracy metric for final model evaluation</i>	25
4	RESULTS	27
4.1	<i>Visualized AlexNet Predictions</i>	27
4.2	<i>Visualized UNet Predictions</i>	29
4.3	<i>RMSD values for model predictions</i>	31
5	DISCUSSION	33
5.1	<i>Limitations of study</i>	33
5.2	<i>Next steps in research</i>	34
6	BIBLIOGRAPHY	36
7	APPENDICES	40
7.1	<i>APPENDIX A: Project Code</i>	40
7.2	<i>APPENDIX B: Relevant CNNs operations</i>	40
7.3	<i>APPENDIX C: Model hyperparameter tuning and regularization strategies</i>	43
7.4	<i>APPENDIX D: UNet Raw Data</i>	44
7.5	<i>APPENDIX E: AlexNet Results</i>	47
7.6	<i>APPENDIX F: COVID-19 Impact Statement</i>	50

Abbreviations

CMR Cardiac Magnetic Resonance

SAX Short Axis

LAX Long Axis

HLA Horizontal Long Axis

VLA Vertical Long Axis

LV Left Ventricle

RV Right Ventricle

MV Mitral Valve

4Ch Four-Chamber

2Ch Two-Chamber

DL Deep Learning

ML Machine Learning

NN Neural Network

CNN Convolutional Neural Network

RMSD Root Mean Squared Distance

1 INTRODUCTION

Cardiac Magnetic Resonance (CMR) imaging is an invaluable tool for evaluating various forms of congenital and acquired heart diseases, including cardiomyopathies, coronary artery disease, valvular disease, myocardial ischemia/infarctions, pericardial disease and others. This technique is preferred over other modalities such as CT or PET primarily on account of the high soft tissue contrast without the need for ionizing radiation, which is well tolerated by patients who require regular check-ups. Secondly, CMR offers a large field of view and multiplanar acquisition ability, which gives unrestricted access to the heart and great vessels through multiple, freely chosen slices (Ginat et al., 2011). With cine imaging, the entire cardiac cycle can be visualized, allowing for assessment of temporal changes in morphology. Observing the movement of the myocardium, valves and flowing blood can give insights into cardiac function including biventricular function, flow measurement and myocardial viability (Gatzoulis, Webb and Daubeney, 2011). From images of the heart in end-diastole and end-systole, the ventricular myocardial mass and ventricular chamber volume can also be calculated, which gives quantitative information on cardiac function. Finally, various different types of pulse sequences can also be applied to create different image appearances which, in other modalities, would require a contrast agent. Overall, these features make CMR a highly versatile technique for studying the heart.

1.1 How is a CMR study performed?

CMR studies involve selecting specific imaging planes in order to appropriately visualize anatomical features of the heart and evaluate their relationship with the great vessels. Before describing the protocol of a CMR study, a discussion into the cardiac imaging planes is essential.

1.1.1 Cardiac Imaging Planes

Two main systems of anatomical planes are studied in CMR – the body planes and cardiac planes (Figure 1). The body planes are the traditional anatomical views – axial, sagittal and coronal, through which scout images are acquired for providing a basic overview of cardiac morphology. However, due to the ‘obliquity’ of the heart (Ginat et al., 2011), evaluating the anatomy with respect to the body planes may give a distorted representation of the shape of the cardiac chambers or the thickness of the ventricular myocardium. Furthermore, in the supine position, the traditional long axis of the body is parallel to the B-field vector produced by the MRI scanner, while the cardiac long axis is at an angle to the vector, which can give a false impression of the heart’s functional state (Boxt, 1999; Taylor and Bogaert, 2012). For this reason, a special set of axes for the heart are separately defined, which are the cardiac planes.

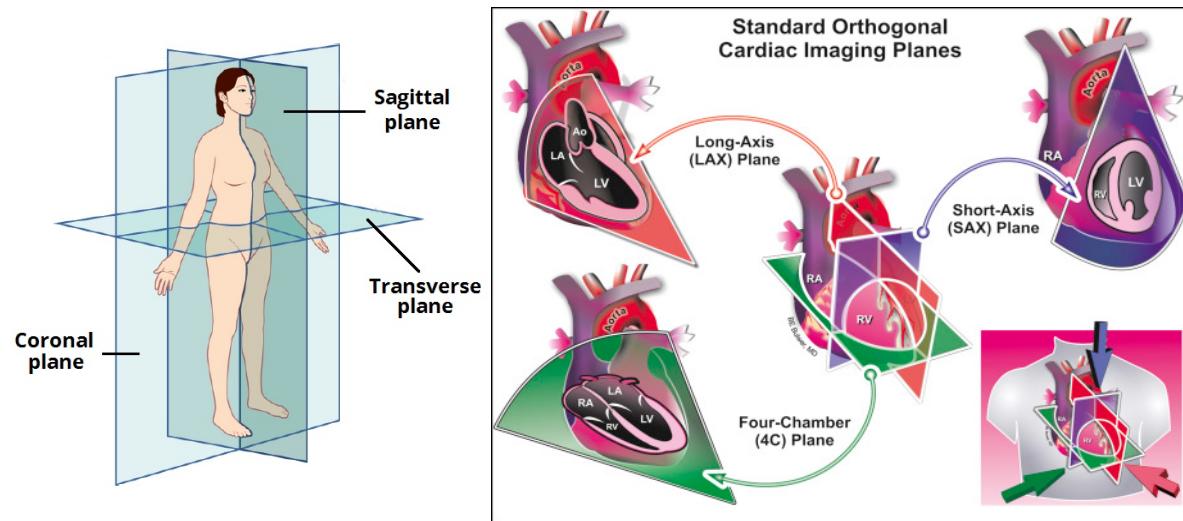


Figure 1: Left - Traditional anatomical planes (Connexions, 2013). Right - Cross-sectional representation of standard cardiac imaging planes (Bulwer, 2019: 76).

The three main cardiac planes are the short axis (SAX), the horizontal long axis (HLA) and vertical long axis (VLA) (Figure 1, Right). The true long axis (LAX) of the heart is an imaginary plane passing through the apex of the heart and the mitral valve (MV). The SAX represents the plane perpendicular to the LAX. The HLA and VLA represent planes running horizontally

and vertically respectively, that are orthogonal to the SAX. With the HLA slice, all four chambers of the heart can be seen, hence it is referred to as the four-chamber (4Ch) view (Figure 2a), while the VLA slice gives a two-chamber (2Ch) view (Figure 2b) (Ginat et al., 2011).

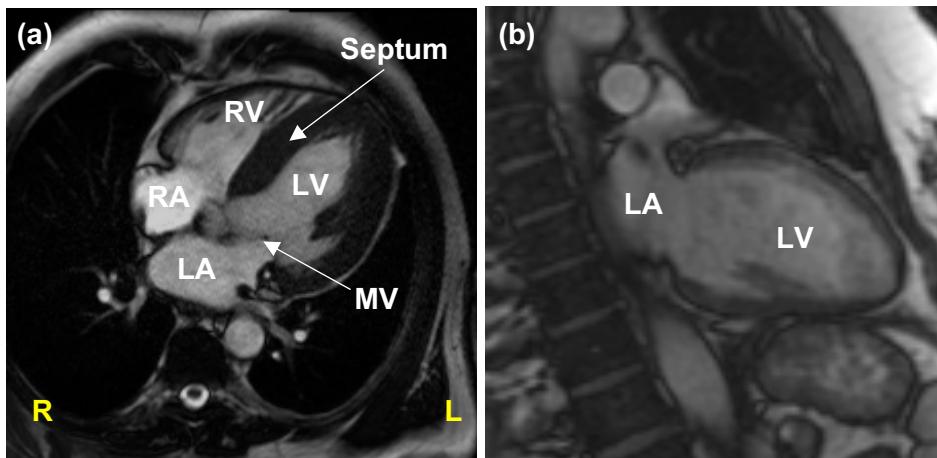


Figure 2: CMR images of 4Ch and 2Ch views. (a) 4Ch view, image taken from the dataset. (b) 2Ch view of heart (Deng and Knipe, n.d.)

RV= right ventricle, RA= right atrium, LV=left ventricle, LA=left atrium, MV=mitral valve.

Other imaging views can also be defined such as the three-chamber (3Ch) view, five-chamber (5Ch) view, left-ventricular and right-ventricular outflow tract (LVOT, RVOT). These offer more specialized views that allow for delineating specific anatomical structures such as the great vessels or coronary arteries, which the main cardiac axes do not offer. Since they are only prescribed in more complex CMR studies, they will not be considered in this report. This report will focus specifically on the 4Ch and 2Ch views, which are mandatory in the CMR workflow.

1.1.2 CMR Protocol

The CMR protocol begins with acquisition of scout views, where three images are obtained in each of the three orthogonal body planes. The purpose of this is to isocenter the heart in the scanner and view the mediastinal structures. An axial stack is constructed with contiguous slices covering the entire thoracic cavity, from the liver to the neck, which includes the heart,

major arteries and veins (Sridharan et al., 2010). This axial stack will form the basis for planning the VLA, HLA and SAX views. From an inferior slice of the axial stack (Figure 3a), at the level of the left ventricle (LV), a line is drawn, running through the cardiac apex and the middle of the MV. This creates the VLA view (Figure 3b) (Taylor and Bogaert, 2012; Joseph, 2013). On the VLA view, a second line is prescribed, transecting the LV apex and the mitral ring, giving the HLA (Figure 3c). From the HLA view, a subjective assessment of the atrial size, biventricular size and ventricular wall motion can be conducted (Sridharan et al., 2010). The SAX (Figure 3d) can now be aligned perpendicular to the VLA and HLA views. On the SAX view, at the level of the mitral valve, the true 4Ch view (Figure 3e) is prescribed by drawing a line passing through the “anterolateral” papillary muscle, the midpoint of the septum and the apex of the right ventricle. A true 2Ch for the left side of the heart can then be prescribed from the true 4Ch view by drawing a line passing through the LA, MV and LV apex (Figure 3e) (Taylor and Bogaert, 2012; Joseph, 2013).

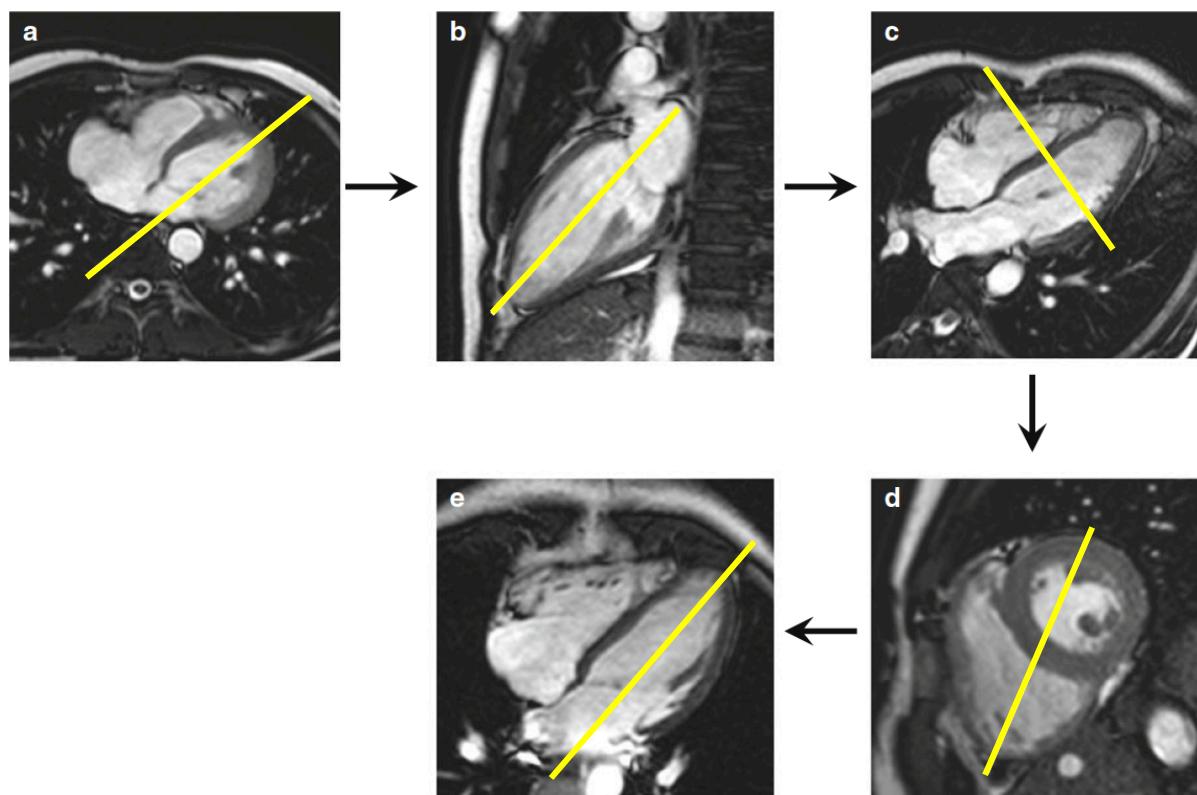


Figure 3: CMR study workflow (Taylor and Bogaert, 2012:94). From axial image (a) line is prescribed through LV apex and MV to give VLA plane (b). HLA (c) is planned from VLA by

drawing line through LV apex and mitral ring. The SAX plane (d) is aligned using the VLA and HLA. The true 4Ch view (e) is aligned using the SAX plane. True 2Ch view (not shown), is created by drawing line through LV apex, MV and LA.

This planning protocol is crucial in ventriculography studies to properly assess cardiac function. From a ventricular stack, important measurements are taken including the end-systolic and diastolic volume, stroke volume, ventricular mass and thickness and ejection fraction. The planning views, therefore, must be carefully planned so that these values are as accurate as possible.

1.2 Disadvantages of CMR protocol - Motivations for study

There are drawbacks to the current CMR protocol. Firstly, it is time-consuming, taking on average 45-60 minutes per study, and even longer for more complex heart conditions (R Davies 2020, personal communication). For this reason, most hospitals use alternative techniques that provide similar information. For example, for myocardial ischemia, stress echocardiography and SPECT are the dominant modalities for diagnosis (von Knobelsdorff-Brenkenhoff and Schulz-Menger, 2015). Second, to acquire and plan CMR images, a detailed knowledge of the cardiac anatomy as well as the competency to operate a CMR machine is essential. This requires specialist radiographers with ideally 15+ years of experience, however they are not available at every imaging centre. Regardless, even with manual annotation of images, because there is an expectation on radiographers to complete the procedure as fast as possible so that no discomfort is caused to the patients, there is a great scope for error. Consequently, improper image acquisition and planning can potentially affect clinical decisions, leading to incorrect diagnoses. Furthermore, the planning process is operator-dependent, hence if annotations are performed by multiple radiographers, there can be inconsistencies in the data due to differences in training. Finally, it must be highlighted that while CMR has been highly recommended for the diagnosis and assessment of a wide array of heart conditions, it is a highly specialized test which is expensive and not accessible to all (von Knobelsdorff-Brenkenhoff and Schulz-Menger, 2015).

2 AIMS AND OBJECTIVES

The aim of this project is to automate the planning process. This would be highly beneficial as it reduces the turnaround time for a study, reduces the need for radiographer expertise, making it accessible and cheaper for patients. More importantly, it ensures that the outputs are highly consistent and accurate. The automation process will be performed using deep learning (DL) techniques. The past decade has seen a revolution in DL, especially in the field of medical computer vision, where neural networks (NN) have been developed for various medical imaging tasks such as disease classification, lesion identification and anatomical segmentation. Due to the increased computational power and availability of advanced GPUs, DL models can be trained on large datasets of medical images. Furthermore, deep NNs learn highly complex functions and feature distributions which traditional machine learning (ML) approaches cannot and therefore are capable of achieving a performance close to human level.

In the field of DL and computer vision, deep convolutional NNs (CNNs) are popularly used when dealing with image data. These NNs essentially learn a set of filters or ‘kernels’ which detect features in the images. In the shallow layers of a CNN, lines and edges are detected, while in the deeper layers, more complex shapes and landmarks are learnt. In this study, we trained CNNs on a dataset of 4Ch images to predict the planning line for the subsequent 2Ch view.

2.1 Previous Studies

In the cardiac imaging literature, more attention has been devoted towards automating anatomical segmentation, while fewer studies have been published regarding automation of cardiac image view planning. Notable works include studies by Lu et al (2011), Blansit et al (2019) and Sundararajan et al (2014) which implement ML techniques to tackle image planning.

Lu et al (2011) planned the SAX, 3Ch, 2Ch and 4Ch views from a 3D volume using a multi-step approach – the LV was first localized and delineated using probabilistic boosting trees, after which the centroids of the LV base and apex were computed using an additionally generated mesh model. Sundararajan et al (2014) used a Support Vector Machines approach, where the model was inputted with anatomical features surrounding the heart, such as the torso height and width, left lung and right lung centroid position (x, y, z), fat estimation and lung size ratio and using these, the planning views were estimated.

Blansit et al (2019) used DL – particularly UNet-based heatmap regression to localize the LAX and SAX anatomical landmarks. For LAX prescription, the MV and apex were localized. Simultaneously, using a SAX stack, the aortic valve, MV, pulmonary valve and tricuspid valve were localized. Using these landmarks, the corresponding SAX, 3Ch, 2Ch and 4Ch views were prescribed.

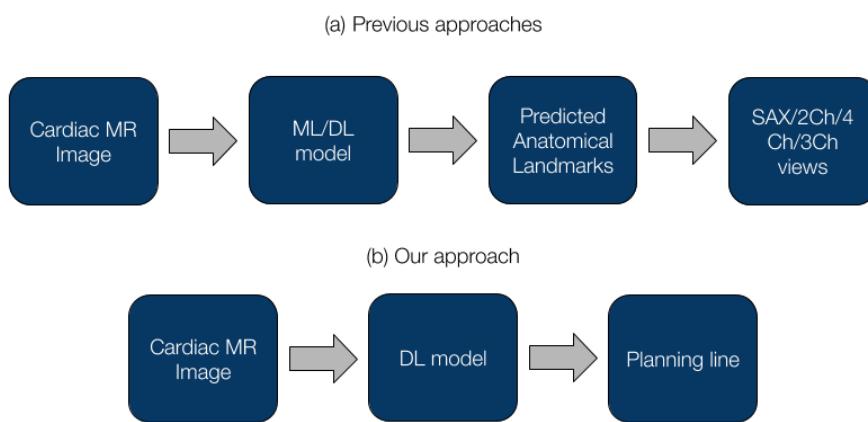


Figure 4: Automation strategies for CMR planning. (a) Previous studies input image into a trained ML/DL model to predict anatomical landmarks, which are used to construct final planning views. (b) Our approach - implement a DL model that inputs image and directly outputs planning line. Diagram created in Google Drawings.

A common theme between these studies is the use of ML to predict the location of anatomical landmarks on the heart in order to prescribe the planning lines (Figure 4a). While this approach has shown promising results, a major challenge is that additional image annotations and

volumetric data is required, which are not typically used in the CMR protocol. Acquiring this data is time consuming, expensive and can be a burden for radiographers. Our method is much more straightforward, the goal being to train a deep CNN that can directly predict the planning line from the image (Figure 4b). With this approach, we expect that the model will automatically learn the anatomical landmarks during training, without having to be explicitly provided the true coordinates of these landmarks.

3 METHODOLOGY

3.1 Patients and Dataset Description

The image dataset used in this study consists of 4Ch images which were retrospectively collected from 607 patient CMR studies performed on a mixture of 1.5T and 3T MRI scanners (Phillips, Siemens and GE).

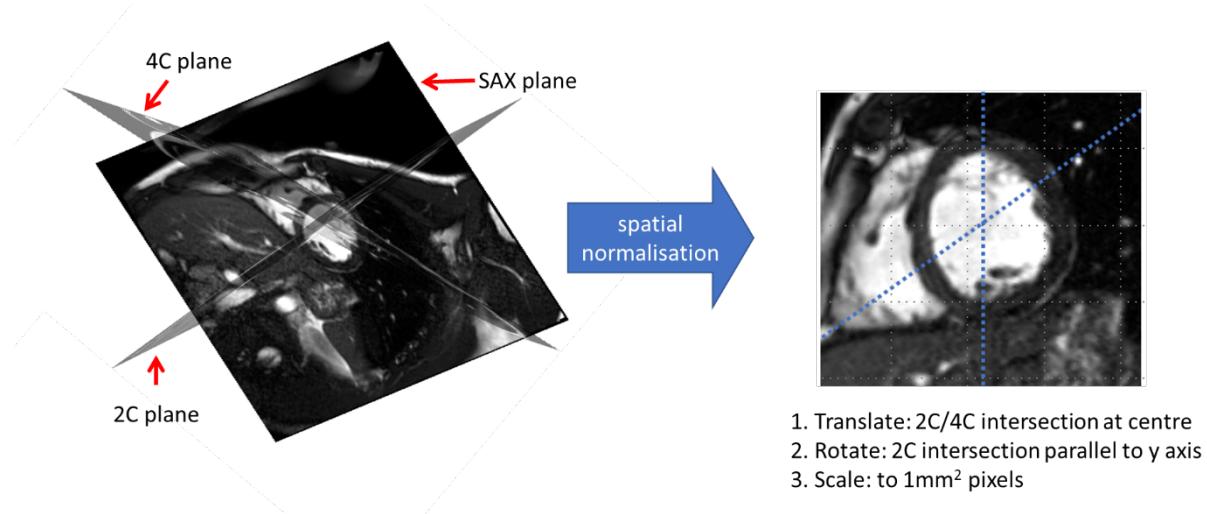


Figure 5: Image Pre-processing conducted on the dataset. Images taken from dataset.

All scans were **SSFP** sequences of the heart in end diastole. To ensure robustness of our CNN models to normal as well as pathological images, the dataset included healthy volunteers, athletes as well as patients of various disease profiles including aortic stenosis, myocardial infarcts, hypertension, Fabry's disease, cardiac amyloid and hypertrophic cardiomyopathy. Prior to model development, various image pre-processing steps were performed, which included transforming the intersection with the 2Ch to be parallel with the y-axis, rescaling the image to an in-plane resolution of 1mm^2 and horizontally translating images so that the intersection of the SAX and the 2Ch was in the middle of the image (Figure 5). All images were of different dimensions; hence resizing was also performed to create 160x160 matrices.

The target labels for model training were the 2Ch line coordinates corresponding to the 4Ch images, which were manually planned by radiographers from Barts Cardiac Imaging Centre. These values were collected in a dataset of form $| x | y | dy | dx |$. Here (x, y) represent a point on the true planning line and (dy/dx) represents the gradient of the line. Using these values, the line can be constructed on the image (Figure 6).

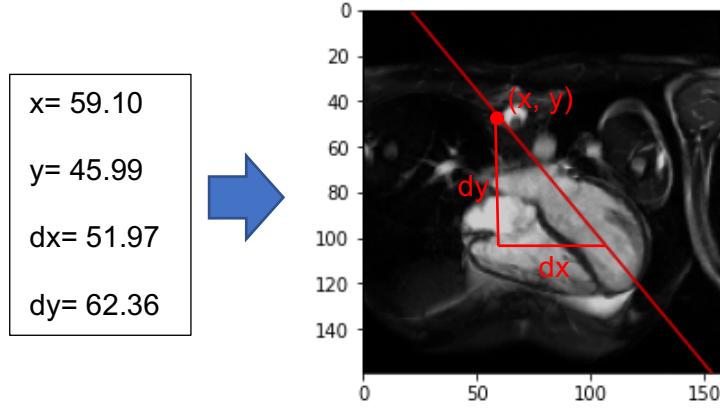


Figure 6: The true 2Ch planning line can be drawn on an image, given coordinates (x, y) and (dy/dx) values. Image taken from dataset.

3.2 Two approaches to planning line prediction

We treated the task of 2Ch line prediction two ways: using a parameter estimation approach and using a segmentation approach. Separate CNNs were implemented for each approach.

3.2.1 Parameter Estimation

The goal of this task was train a CNN to predict numerical parameters defining the equation of the 2Ch line. We represented the line as an equation in the polar form:

$$r = x \cos \theta + y \sin \theta$$

where r is the length of a line (in px) segment drawn from the origin perpendicular to the 2Ch line and θ represents the angle (in radians) between r and the x-axis (Figure 7) (Duda and Hart, 1972). The reason for using the polar form versus the conventional slope-intercept form is that the values of r and θ are bounded in the ranges $[0, 160\sqrt{2}]$ ($160\sqrt{2}$ = length of image diagonal) and $[0, 2\pi]$ respectively, which makes the learning process effective for the CNN.

The CNN model chosen to perform parameter estimation was a modified AlexNet, which is discussed in section 3.3.2. For model training, the target labels dataset was transformed into an array of form $[[r_1, \theta_1], \dots, [r_m, \theta_m]]$ where each $[r_m, \theta_m]$ corresponds to the m^{th} image in the dataset.

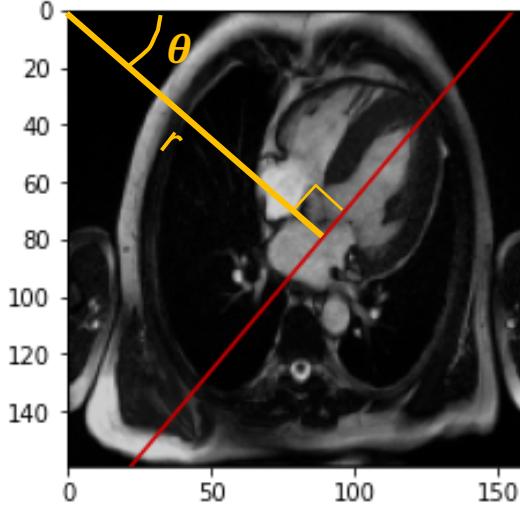


Figure 7: Parameter estimation approach of 2Ch line planning. r is the length of the perpendicular line (px) segment between the origin and the 2Ch line (red). θ is the angle between r and the x-axis (radians). Image taken from dataset.

3.2.2 Segmentation

With the segmentation approach, each pixel in a 4Ch image can belong to one of two classes – the 2Ch line or the background. This means that the line can be represented as a one-hot-encoded mask (binary mask) of the same dimensions as the image, where line pixels store a value of 1.0 and background pixels store a value of 0.0 (Figure 8). The goal of this approach, therefore, is to predict a ‘line mask’ for the 4Ch image. For this, we implemented a fully-convolutional network called the UNet, which is discussed in section 3.3.1. To create the target labels for model training, we developed a mask generator function which inputs the (x, y) and dy/dx values and creates the target line masks (Figure 8).

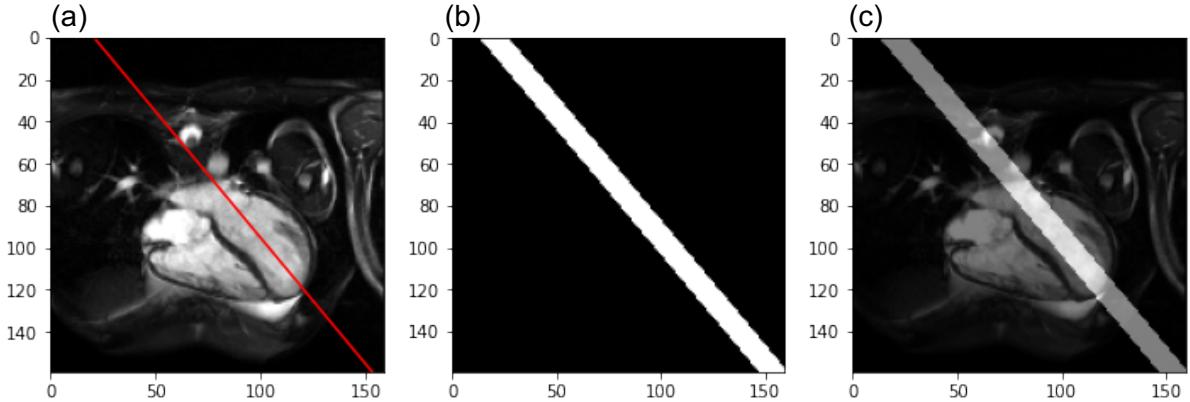


Figure 8: Creation of a binary mask (b) given the 2Ch line (a, red). Pixels in (a) which belong to the background store 0.0 (black) at the same index in (b). Pixels belonging to the 2Ch line store 1.0 (white). For convenience of model training, the line in (b) was drawn thicker (width=10) than the actual line in (a). The line mask can be overlapped on the original image for evaluation (c). Images taken from dataset.

3.3 Neural Network Architectures

The NN architectures implemented in this study are discussed below. A detailed description of the relevant CNN operations is provided in Appendix B.

3.3.1 UNet

The UNet is a fully convolutional network developed for biomedical image segmentation (Ronneberger, Fischer and Brox, 2015). The network architecture we have used is shown in Figure 9; it is a slightly modified version the original architecture, as our images were matrices of dimensions 160x160x1. The architecture consists of three paths: a contracting path, a bottleneck layer and a final expanding path. The contracting path inputs the image and performs a series of convolutions and max-pooling operations 4 times (refer to Appendix B). It starts off with 32 kernels which doubles on every iteration. Essentially, this path compresses the image and learns a set of kernels to detect the anatomical features from the cardiac image. The subsequent bottleneck layer performs 2 convolutions on the image (Lamba, 2020).

The expanding path consists of a series of de-convolution and convolution operations repeated 4 times. Here, the number of kernels halves on each iteration. After every de-

convolution operation, the output feature maps are concatenated with the feature maps produced in the contracting path. By doing this, the model also learns the relative position of each of the anatomical features learnt in the contracting path. Hence, the expanding path plays the role of a localizer, which is essential for image segmentation (Lamba, 2020). The final layer of the UNet is a 1x1 convolution with a sigmoid activation function, which outputs a segmentation map. This is a 160x160x1 matrix, where each entry represents the probability of the pixel at the corresponding index of the image belonging to the 2Ch line or the background. The segmentation map is converted into a binary mask by thresholding at 0.5, such that probabilities greater than 0.5 are set to 1.0 and below 0.5 are set to 0.0.

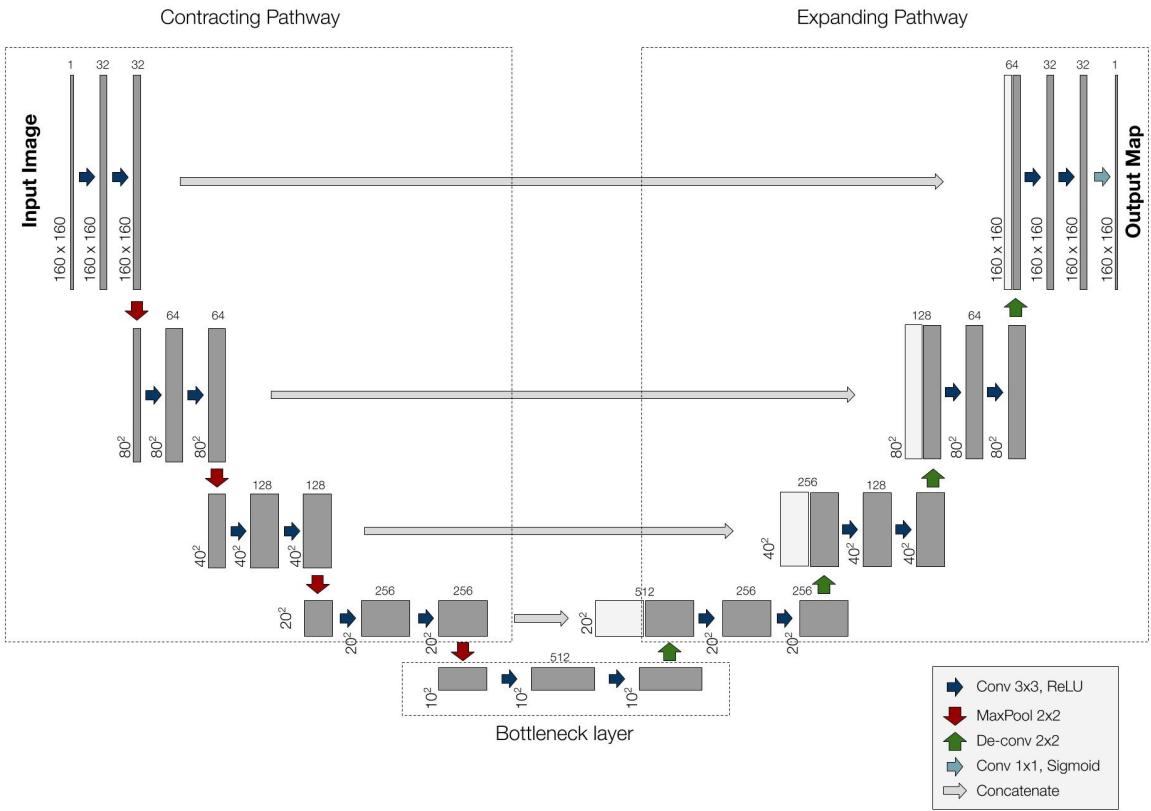


Figure 9: UNet Architecture. Contracting, bottleneck and expanding pathway are highlighted in dotted boxes. Arrows represent different CNN operations, provided in legend. Diagram created in Google Drawings.

3.3.2 AlexNet

The AlexNet architecture was developed for object detection (Krizhevsky, Sutskever and Hinton, 2017), however we have modified it for our parameter estimation task of r and θ

(Figure 10). It consists of 7 convolutional and max pooling layers, which compress the image and learn a set of kernels. Following this, the compressed feature map is flattened into a vector of pixel values and passed through a set of fully connected layers. These layers essentially perform a series of non-linear operations on the pixel values. The final output layer of the network is made up of 2 nodes, each corresponding to the r and θ predictions.

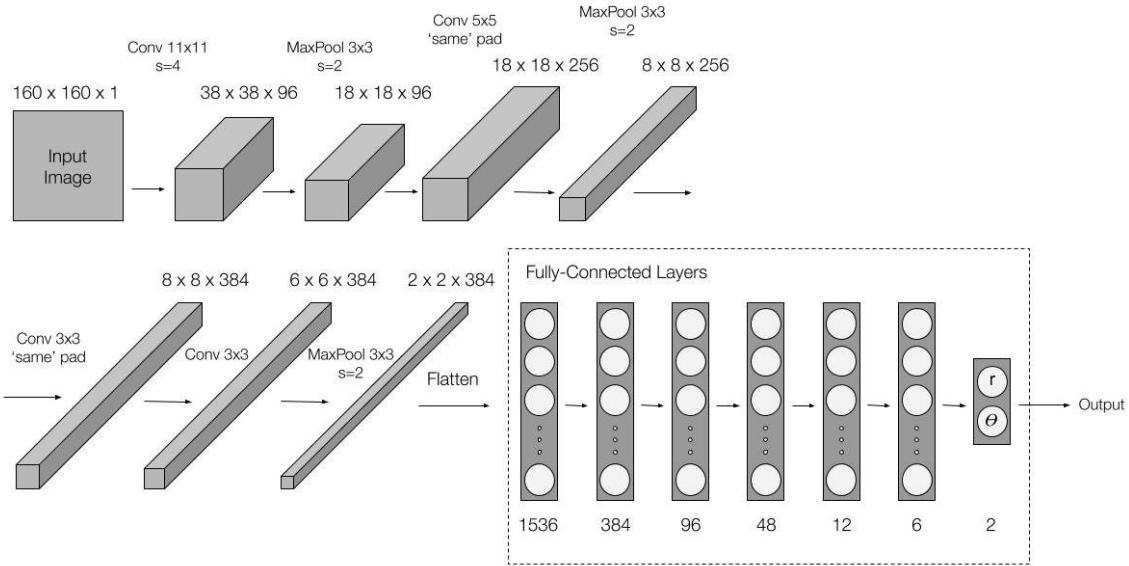


Figure 10: AlexNet architecture. First 8 layers are convolutional layers, followed by 7 fully connected layers of dense nodes. Output layers gives prediction for r and theta. Created in Google Drawings.

3.4 Model Training:

3.4.1 Dataset Split

A training and test dataset was created using an 80:20 split. 80% of the 607 examples (485 images + targets) were used to train the CNNs while the rest 20% (122 images + targets) were used as a holdout set to evaluate the model's ability to generalize to unseen images. To make the training process effective such that the models see different types of examples on every training step, two shuffling operations were performed: first, the images and targets were shuffled before splitting the dataset. Second, the training set images and targets were passed to the CNNs using a Keras generator function, which shuffles the data on every epoch and

passes random images in batches of 128. The model was trained for 30 iterations in steps of 100 per iteration.

3.4.2 Loss functions

In order to effectively train NNs, an objective or ‘loss’ function is required, with the goal being to minimize this function so that the kernels are optimized. For the UNet, we used the binary cross entropy loss function provided by Keras. With the AlexNet, the loss metric chosen was the mean squared error (MSE).

3.4.3 Accuracy metric for UNet training

The DICE coefficient was used to monitor UNet performance during training. This metric measures the overlap between the true mask and the predicted mask. The formula is provided below, along with a visual representation (Figure 11).

$$\text{DICE} = \frac{2 * \text{A} \cap \text{B}}{\text{A} + \text{B}}$$

DICE = 1.0 → Excellent

DICE = 0.80 → Good

DICE = 0.10 → Poor

Figure 11: DICE coefficient. Created using Google Drawings.

$|A \cap B|$ represents the number of elements in the intersection of the true mask A and predicted mask B, and $|A|$ and $|B|$ represents the total number of elements in each mask.

3.4.4 Other Hyperparameters

For effective model training, other hyperparameters and regularization methods were set, which are provided in [Appendix C](#).

3.5 Software Used

The entire project was run on Google Colaboratory, a free cloud service provided by Google for running IPython Notebooks. All data analysis and image preprocessing were performed using NumPy, OpenCV and Matplotlib while all DL experiments were performed using Keras and TensorFlow. The data was trained on an NVIDIA Tesla K80 GPU service provided by Google Colaboratory. The code and results for all models trained during this project can be accessed in Appendix A.

3.6 Reconstruction of UNet predicted masks

After model training, the UNet was used to generate predictions on sample images. Figure 12 shows the true and predicted masks for 5 images from the holdout set.

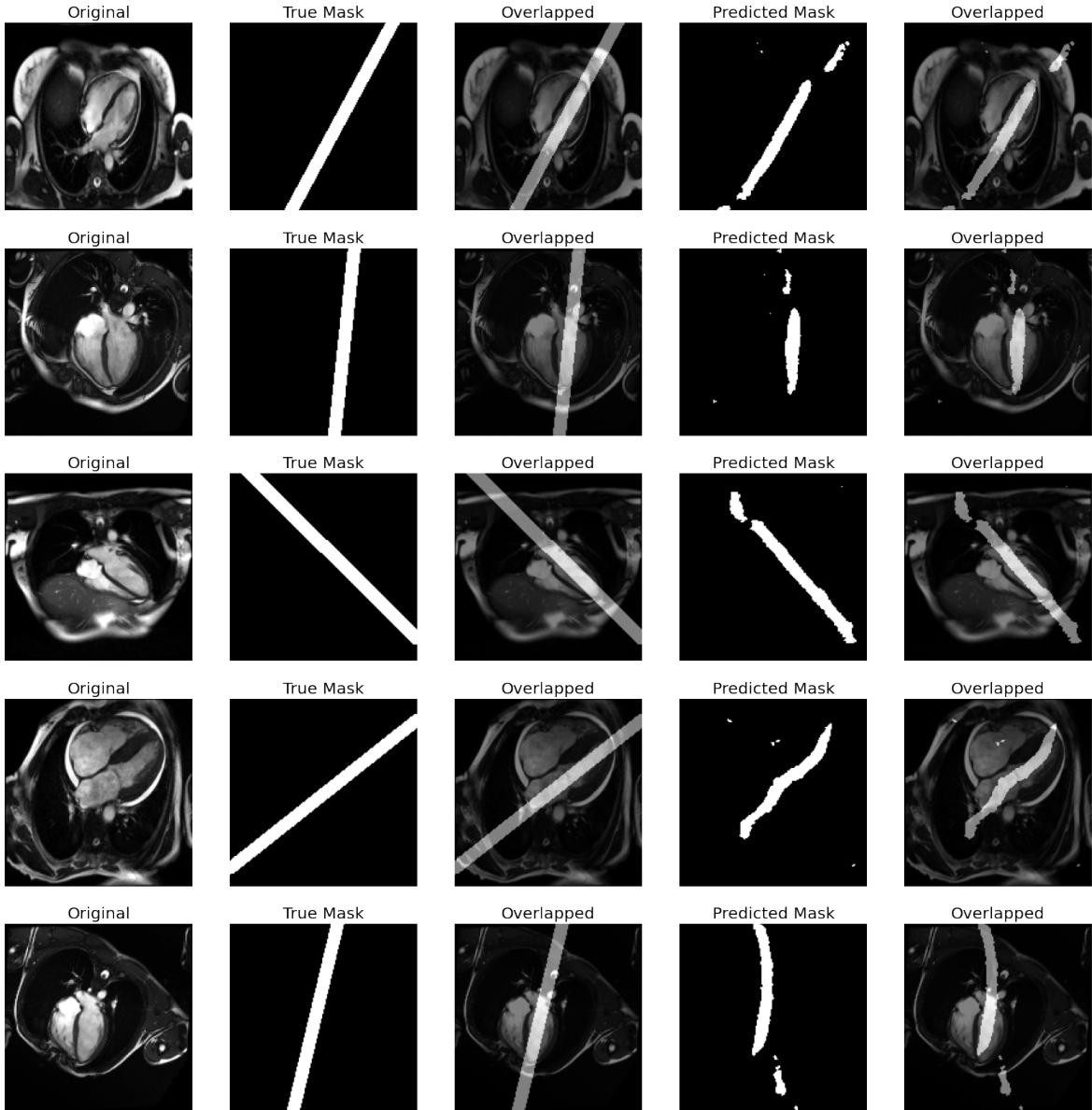


Figure 12: Initial UNet results for 5 sample images. Each row corresponds to one image. Original images in Column 1. True and predicted masks in columns 2 and 4 respectively, overlapped onto the original image in columns 3 and 5 respectively.

Visually inspecting the predicted masks, it is observable that the UNet was unable to produce solid lines, as in the true mask. The segmentation outputs appear broken and patchy, making them unsuitable for comparison with the target masks. In order to rectify this, image processing steps were performed to reconstruct the line equation from the predicted mask. The reconstruction process is shown in Figure 13. From scanning the predictions, it was observable that there were patches of varying areas, however the patch with the largest area resembled the line. Hence, the first step was to erase smaller patches and preserve the

largest. To do this, contours were drawn around smaller patches and bitwise operations were performed to erase them. Finally, using a weighted least squares algorithm, the largest patch was fitted with a line, parametrized by a slope and y-intercept. These parameters were used to construct the final line.

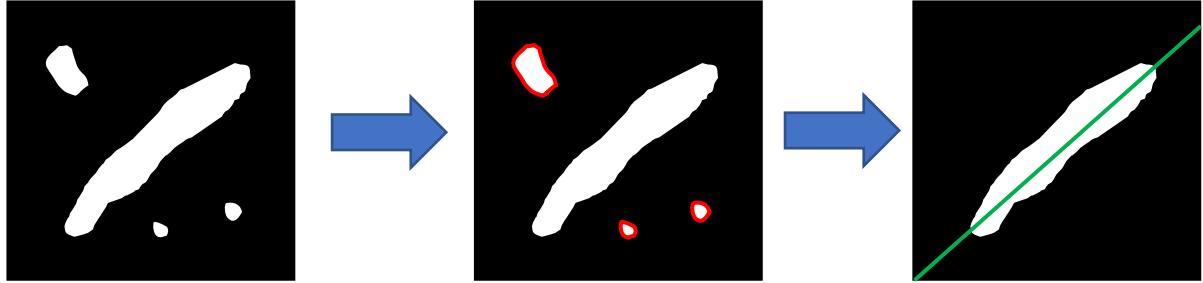


Figure 13: Reconstruction of line from the predicted mask. (a) Predicted mask. (b) Smaller patches are erased, only preserving the largest segment. (c) Line(green) is fitted to largest segment which is the reconstructed line. Created in Google Drawings.

3.7 Accuracy metric for final model evaluation

Since the UNet and AlexNet produce different outputs, it was required to have an accuracy metric to compare both models. Previous studies have used the angle between the true and predicted line as a metric (Blansit et al 2019; Lu et al 2011). The problem with this metric is that it does not take into account the position of the line. For example, in the case of parallel lines at different positions, the angle between the lines is zero, even though they do not overlap. To overcome this, we defined a different metric – the root mean squared distance (RMSD) between the true line and the predicted line. To compute this value for a single image, the true line was first selected as a reference. 200 evenly spaced points were created on the line (within the bounds of the image). The RMSD was computed between the 200 points and the predicted line. The calculation is shown in Figure 14 for 10 points on the true line. This RMSD metric can be averaged over all images to obtain the Mean RMSD.

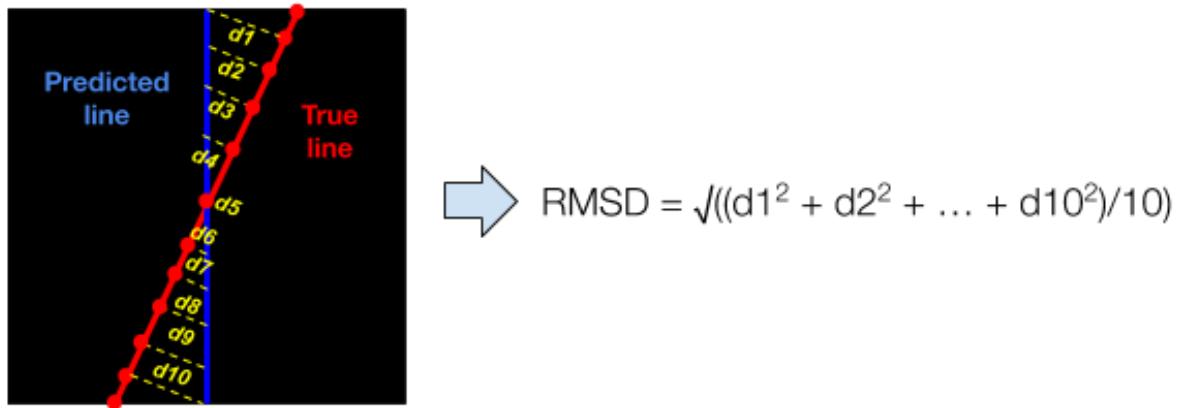
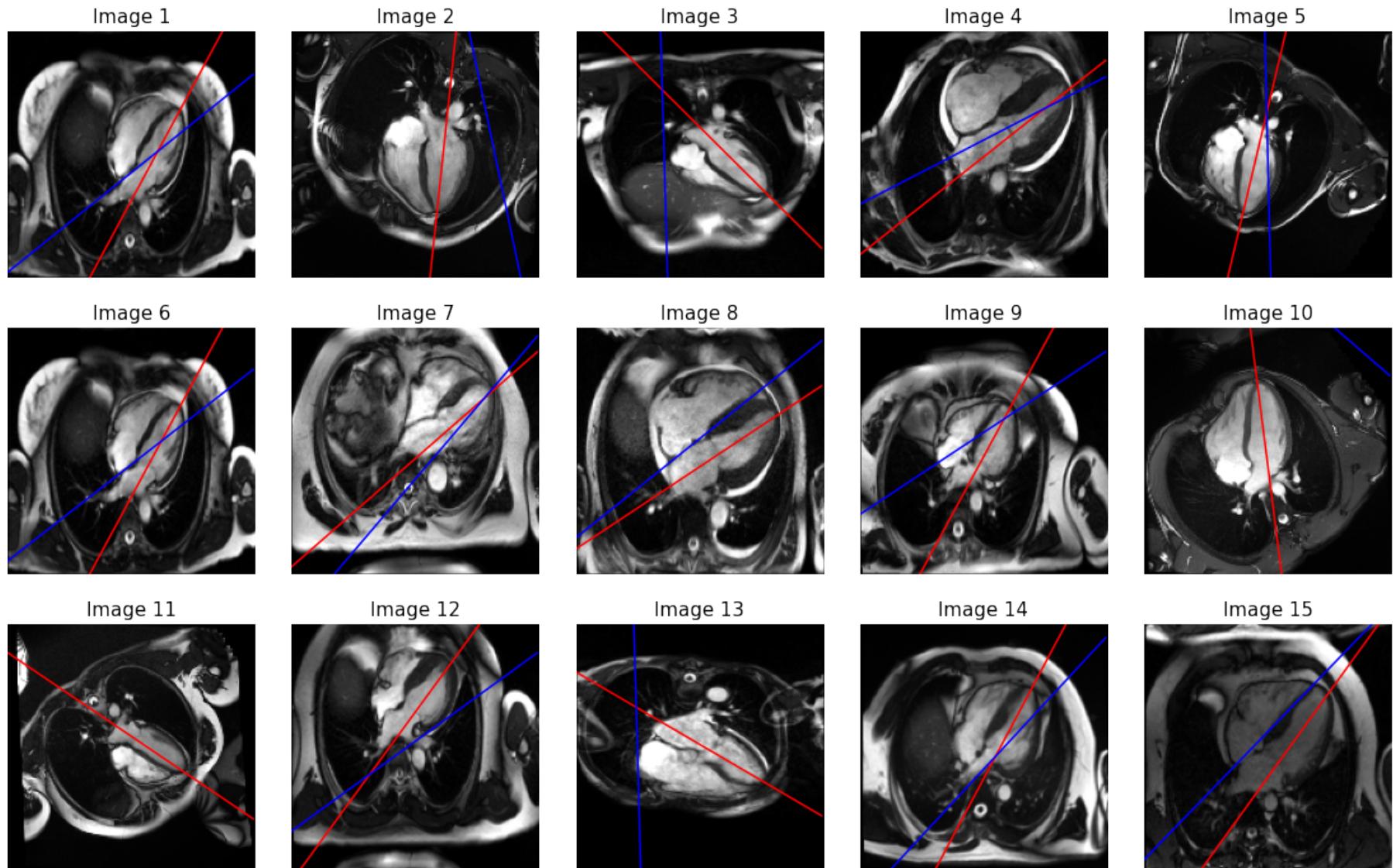


Figure 14: RMSD calculation between true and predicted line. Calculation performed for 10 points on true line. Created in Google Drawings.

The RMSD scores for the UNet and AlexNet models were compared using a two-sample t-test. A p-value <0.05 was considered to be a significant difference.

4 RESULTS

4.1 Visualized AlexNet Predictions



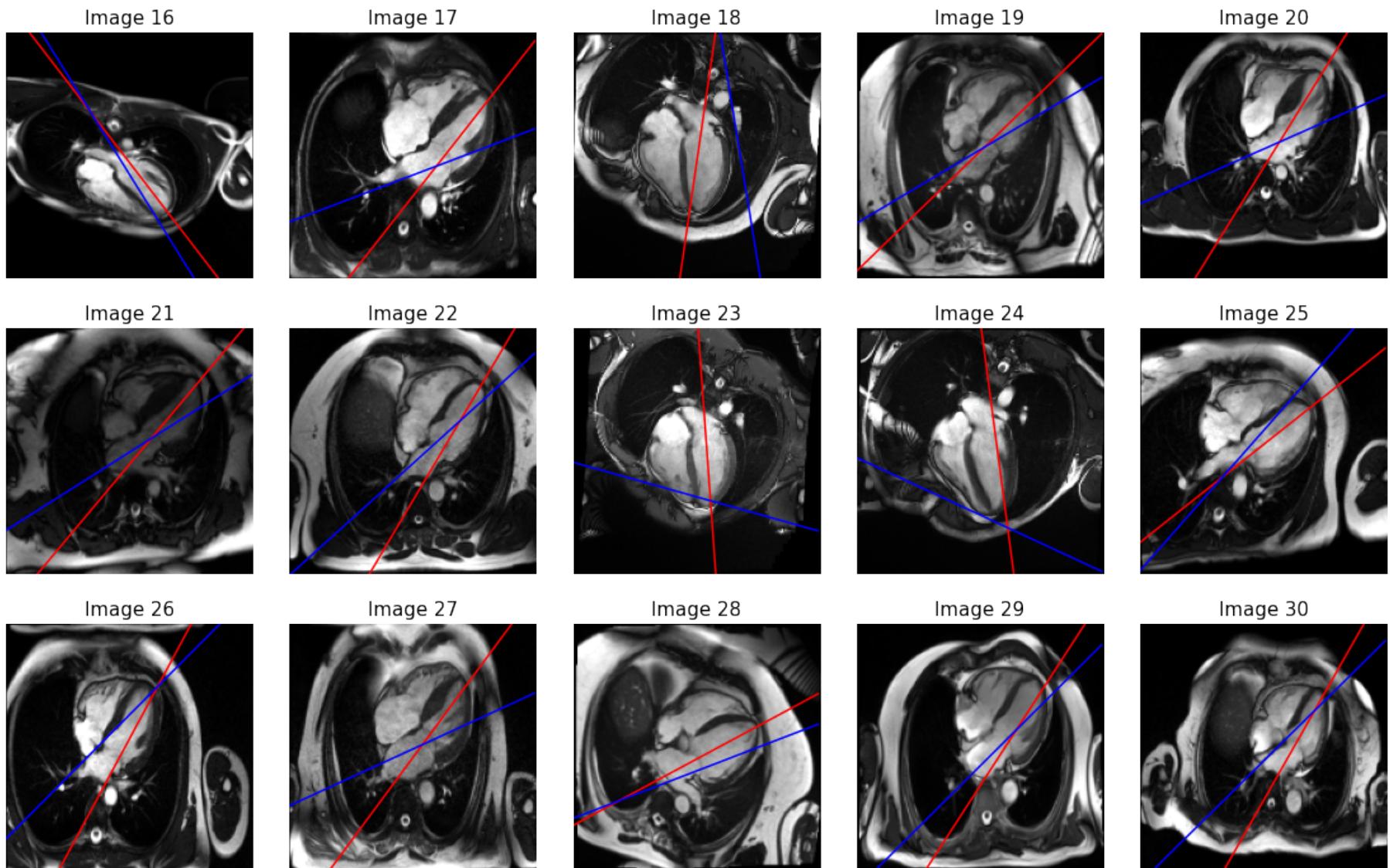
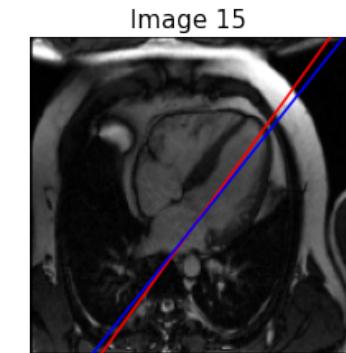
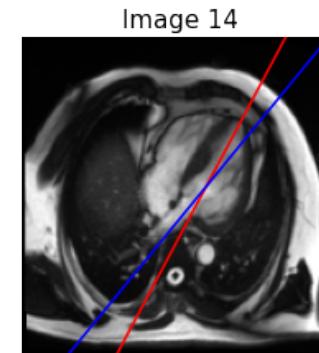
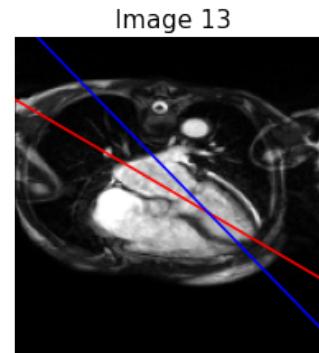
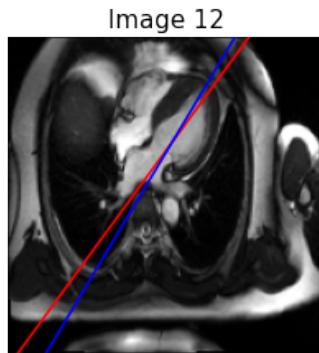
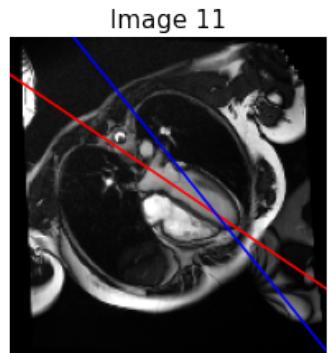
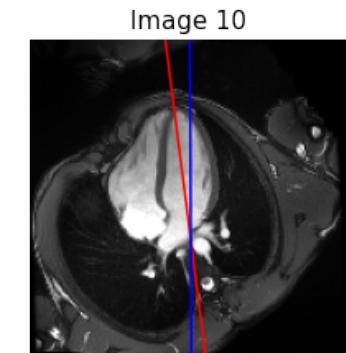
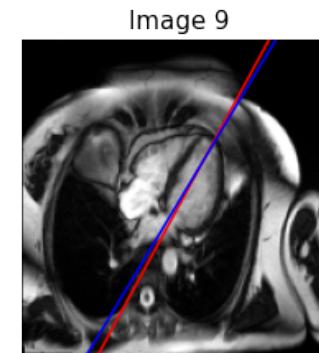
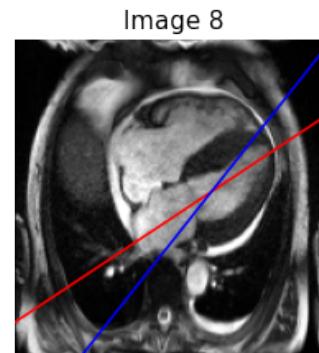
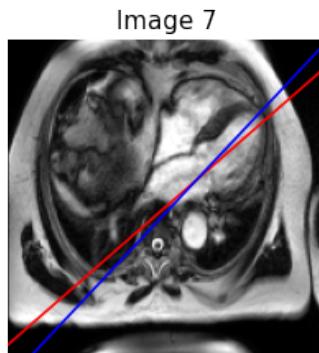
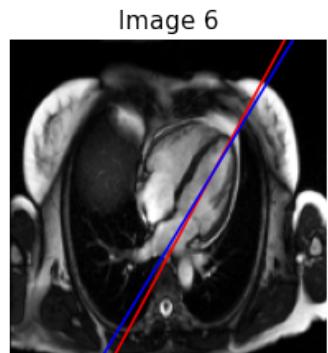
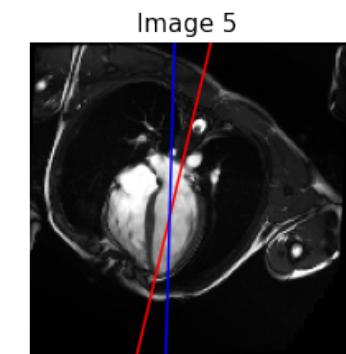
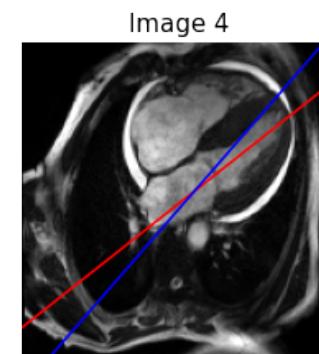
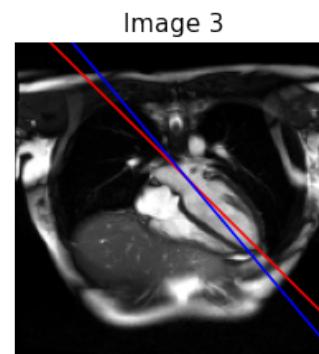
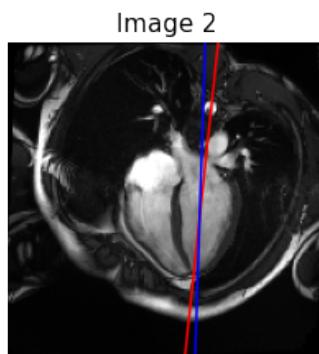
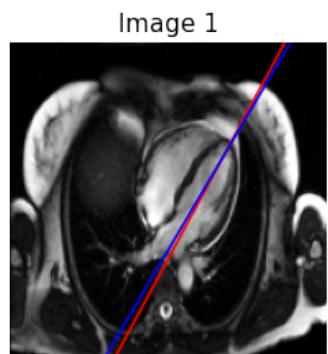


Figure 15: Visualized AlexNet predictions for 30 images from the holdout set. True line=red, Predicted line=blue.

Results for all 122 images can be accessed in Appendix A.

4.2 Visualized UNet Predictions



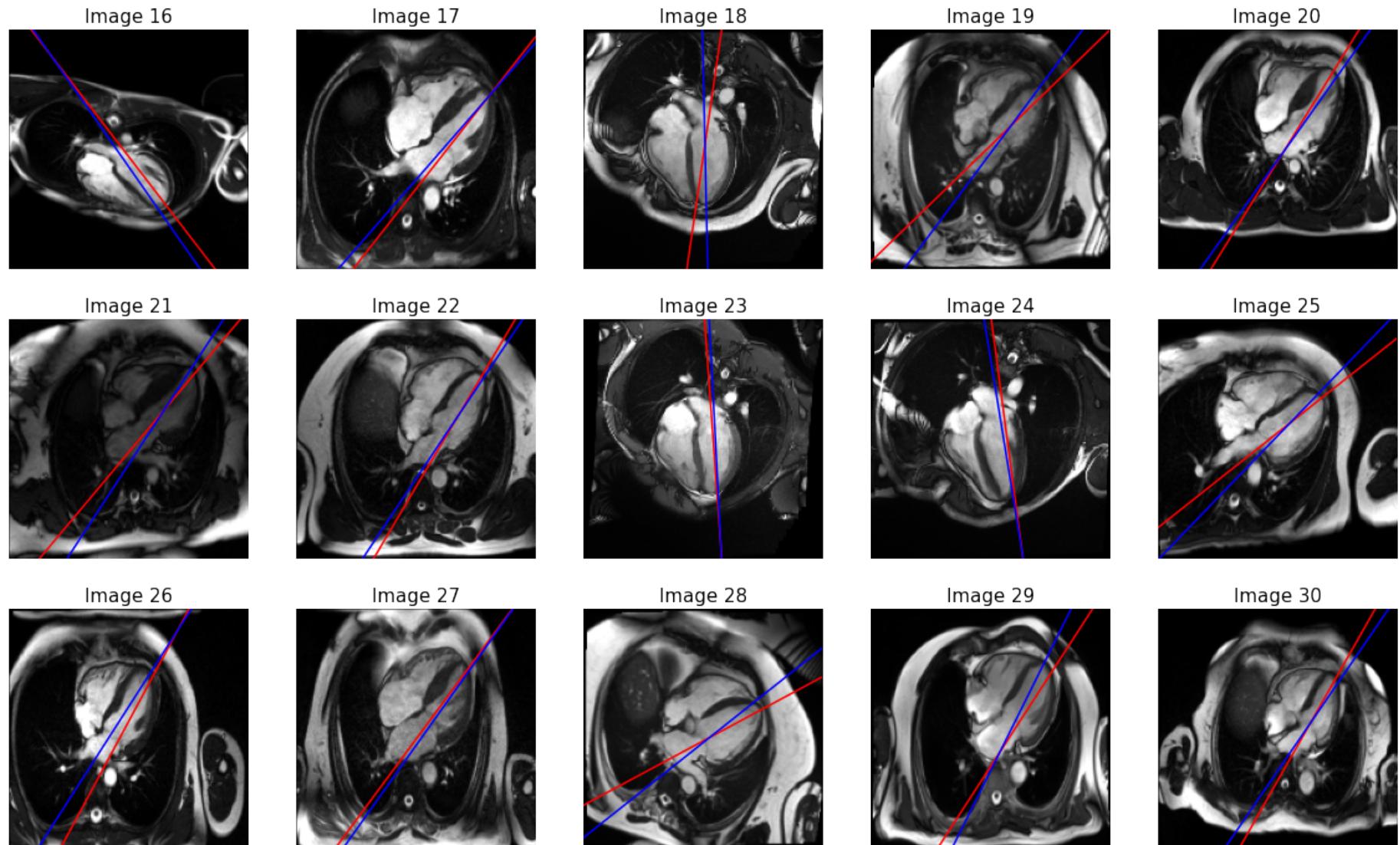


Figure 16: Visualized UNet predictions for 30 images from holdout set. True line = red, Predicted line = blue.

Results for all 122 images can be accessed in Appendix A.

4.3 RMSD values for model predictions

The performance metrics for the model predictions are provided in Tables 1 and 2 below. Since the UNet was unable to create solid lines and reconstruction techniques were applied (as described in section 3.6), the DICE score values were unreliable for evaluating the model, hence they have been excluded from analysis and can be accessed in Appendix D.

Table 1: RMSD between the true and predicted 2Ch lines, computed for 30 AlexNet and UNet predictions. All raw data can be accessed in Appendix D and E.

Image	RMSD (mm)	
	ALEXNET	UNET
1	20.332	2.203
2	35.605	3.483
3	50.944	5.113
4	14.710	10.952
5	13.925	9.807
6	21.288	2.586
7	12.392	6.303
8	14.971	18.615
9	26.979	2.423
10	94.003	6.257
11	181.863	18.467
12	18.757	6.120
13	60.451	15.253
14	13.467	10.336
15	18.816	2.709
16	6.678	4.460
17	29.551	3.824
18	30.459	7.732
19	14.869	10.689
20	31.083	3.424
21	18.528	7.760
22	17.197	3.125
23	53.084	1.772
24	57.685	2.709
25	12.009	7.934
26	19.519	6.965
27	27.690	2.545
28	9.706	9.314
29	11.898	6.536
30	16.106	4.472

Table 2: Summary statistics for describing the distribution of the RMSD values for AlexNet and UNet model. Values computed over entire holdout set (n=122).

	RMSD (mm)	
	ALEXNET	UNET
Mean	30.209*	5.697*
Std	27.038	3.768
Min	4.035	0.288
25%	14.200	2.999
50%	21.446	5.055
75%	38.142	7.567
Max	181.863	20.368

*p <0.05

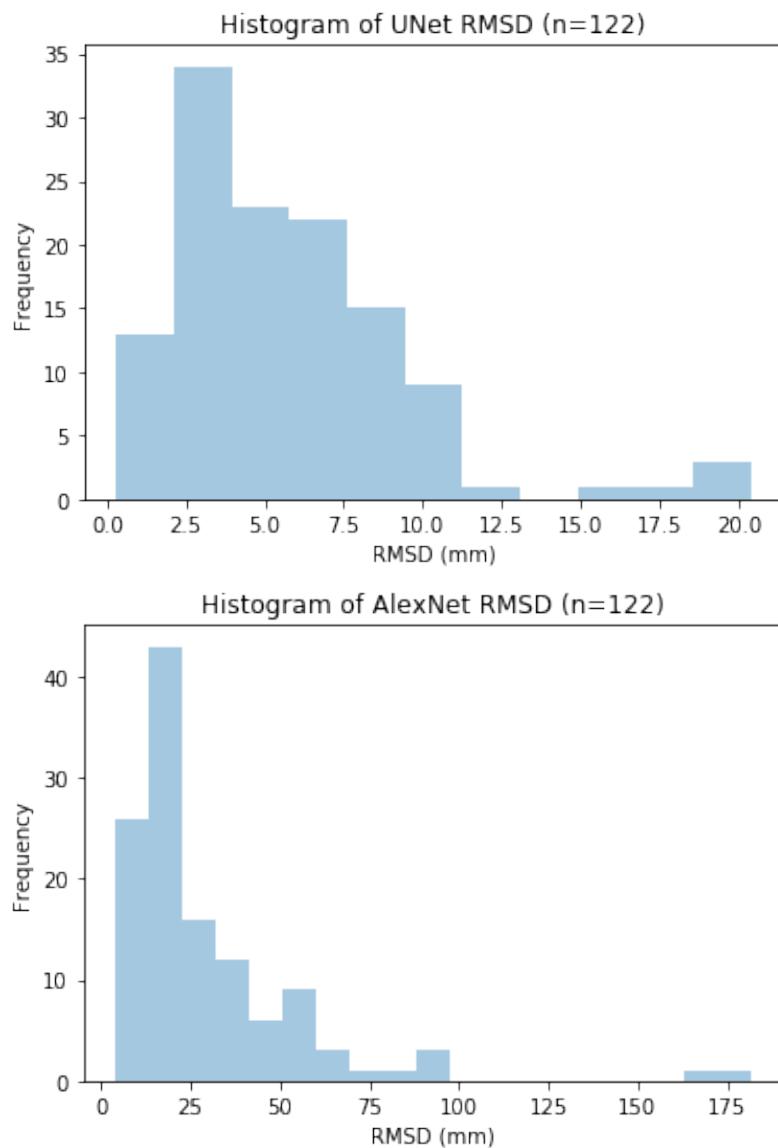


Figure 17: Distribution of the RMSD values for the AlexNet and UNet models.

5 DISCUSSION

This study aimed to demonstrate the feasibility of DL techniques to automate the CMR planning process. As a proof of concept, we focused on the task of predicting a 2Ch line from a 4Ch image – this was conducted by testing two separate CNNs which deal with the problem through different approaches. The first approach was estimating parameters r and θ that describe the 2Ch line equation, which was performed using the AlexNet. The second approach was segmentation, which creates a binary mask of the 2Ch line, for which we used the UNet. Visually comparing the predictions of both models (Figure 15 and 16), it was found that the AlexNet was unable produce plausible 2Ch lines, however the UNet predictions showed greater agreement with the ground truth lines. The mean RMSD between the true and predicted lines, computed over the holdout set, was found to be 30.21 ± 27.04 mm for the AlexNet and 5.697 ± 3.768 mm for the UNet (Table 2), which demonstrates a five-fold superior performance by the UNet ($p < 0.05$). Comparing the distributions of the RMSD values (Figure 17) as well, it was observable that the UNet values demonstrate a smaller spread with 75% of the dataset having an RMSD < 8.0 . From these results, it can be concluded that the segmentation approach is effective for automating the planning process.

The superior performance of the UNet can be attributed to its autoencoder-style architecture. As described in section 3.3.1, the contracting path captures coarse contextual information from images which is passed to the expanding path via skip connections, enabling the localization of these features (Lamba, 2020). Comparatively, the architecture of the AlexNet is less complex, consisting of down sampling layers between the input and output layers. The lack of a localizer network could explain its inability to position the 2Ch line on the image.

5.1 *Limitations of study*

While this study demonstrates the effectiveness of DL techniques for planning line prediction, there were limitations. Firstly, our models were trained and evaluated using a small dataset of

607 images. Typically, DL studies require at least 1000 examples for training NNs to achieve a human-level performance. It must be mentioned that at the beginning of this study, our initial models (results included in appendix A), demonstrated poor generalizability. This was managed by including data augmentation which adds random transformations like rotations, shears and translations on image batches. While this improved performance by a small amount, it merely replaces existing images with transformed copies, rather than expanding the dataset and creating a larger feature space for the model to learn from. Furthermore, while the UNet demonstrated an overall satisfactory performance, 25% of the images demonstrated an RMSD > 7, representing poor prediction accuracy on these images. This can be reduced by obtaining more cardiac images of varied shapes, sizes, and disease profiles, allowing the model to capture new features. Regarding model evaluation, which is crucial for translating this approach into a clinical setting, it must be highlighted that in the cardiac medical imaging literature, there is no robust metric to evaluate quality of a planning line, and assessment is visual. Development of better accuracy metrics will allow for developing better objective functions, thus improving the effectiveness of model training and evaluation.

5.2 Next steps in research

Our study primarily focused on single shot sequences. However, in the conventional CMR workflow, cine studies are regularly performed where 3D cardiac volumes are analyzed. Hence, the next steps in research would be to generalize our models to an entire cine study. The challenge with cine MRI, however, is that the cardiac chambers are in motion. CNNs will be computationally expensive for this case, thus it would be worth exploring more complex architectures such as recurrent NNs instead. Recently with neuroimaging, there has been research into deep reinforcement learning strategies where a learning agent tries to mimic experienced radiographers at the view planning task (Alansary et al., 2018). This approach could be transferable to CMR planning as well. Eventually, the goal is to implement a whole ML system that can be seamlessly integrated into the CMR workflow. By doing so, it is hoped that the limitations of CMR, including the need for specialist radiographers and the lengthy

turnaround time will be eliminated, making this high quality imaging modality cheaper and accessible to patients.

6 BIBLIOGRAPHY

Alansary, A., Folgoc, L., Vaillant, G., Oktay, O., Li, Y., Bai, W., Passerat-Palmbach, J., Guerrero, R., Kamnitsas, K., Hou, B., McDonagh, S., Glocker, B., Kainz, B. and Rueckert, D. 2018. Automatic View Planning with Multi-scale Deep Reinforcement Learning Agents. *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, 11070, pp.277-285.

Blansit, K., Retson, T., Masutani, E., Bahrami, N. and Hsiao, A., 2019. Deep Learning-based Prescription of Cardiac MRI Planes. *Radiology: Artificial Intelligence*, 1(6), p.e180069.

Boxt, L., 1999. From the RSNA Refresher Courses. Cardiac MR Imaging: A Guide for the Beginner. *RadioGraphics*, 19(4), pp.1009-1025.

Duda, R. and Hart, P., 1972. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1), pp.2-3.

Gatzoulis, M., Webb, G. and Daubeney, P., 2011. Diagnosis And Management Of Adult Congenital Heart Disease. 2nd ed. Edinburgh: Churchill Livingstone, p.52.

Ginat, D., Fong, M., Tuttle, D., Hobbs, S. and Vyas, R., 2011. Cardiac Imaging: Part 1, MR Pulse Sequences, Imaging Planes, and Basic Anatomy. *American Journal of Roentgenology*, 197(4), pp.808-815.

Hayes, C., Daniel, D., Lu, X., Jolly, M. and Schmidt, M., 2013. Fully automatic planning of the long-axis views of the heart. *Journal of Cardiovascular Magnetic Resonance*, 15(S1).

Joseph, E., 2013. *How To Perform A Cardiac MRI Study*. Available at:
<https://www.youtube.com/watch?v=wKa-qOBVCxA&t=99s>.

Krizhevsky, A., Sutskever, I. and Hinton, G., 2017. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), pp.84-90.

Lamba, H., 2020. Understanding Semantic Segmentation With UNET. [online] Towards Data Science. Available at: <<https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47>>.

Lu, X., Jolly, M., Georgescu, B., Hayes, C., Speier, P., Schmidt, M., Bi, X., Kroeker, R., Comaniciu, D., Kellman, P., Mueller, E. and Guehring, J., 2011. Automatic View Planning for Cardiac MRI Acquisition. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Berlin, Heidelberg: Springer, Berlin, Heidelberg, pp.479-486.

Ronneberger, O., Fischer, P. and Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Cham, Switzerland: Springer, Cham, pp.234-241.

Sridharan, S., Price, G., Tann, O., Hughes, M., Muthurangu, V. and Taylor, A., 2010. *Cardiovascular MRI In Congenital Heart Disease*. Berlin, Heidelberg: Springer, Berlin, Heidelberg, pp.3-13.

Sundararajan, R., Patel, H., Shanbhag, D. & Vaidya, V. 2014, *An SVM Based Approach for Cardiac View Planning*, Cornell University Library, arXiv.org, Ithaca.

Taylor, A. and Bogaert, J., 2012. Cardiovascular MR Imaging Planes and Segmentation. In: J. Bogaert, S. Dymarkowski, A. Taylor and V. Muthurangu, ed., *Clinical Cardiac MRI*, 2nd ed. Berlin, Heidelberg: Springer, Berlin, Heidelberg, pp.93-107.

von Knobelsdorff-Brenkenhoff, F. and Schulz-Menger, J., 2015. Role of cardiovascular magnetic resonance in the guidelines of the European Society of Cardiology. *Journal of Cardiovascular Magnetic Resonance*, 18(1).

Images:

Bulwer, B., 2019. Principles of Transthoracic Imaging Acquisition: The Standard Adult Transthoracic Echocardiographic Examination. In: S. Solomon, J. Wu and L. Gillam, ed., *Essential Echocardiography: A Companion to Braunwald's Heart Disease*. Elsevier, pp.73-95.

Connexions, 2013. *The Three Planes Most Commonly Used In Anatomical And Medical Imaging Are The Sagittal, Frontal (Or Coronal), And Transverse Plane..* [image] Available at: <<https://cnx.org/contents/FPtK1zmh@8.108:F-TuqKAF@5/Anatomical-Terminology>>.

Deng, F. and Knipe, H., n.d. *Cardiac Imaging Planes | Radiology Reference Article | Radiopaedia.Org.* [online] Radiopaedia.org. Available at: <<https://radiopaedia.org/articles/cardiac-imaging-planes>> [Accessed 3 May 2020].

Mohamed, I., 2017. *An Example Of Convolution Operation In 2D 2 ..* [image] Available at: <https://www.researchgate.net/publication/324165524_Detection_and_Tracking_of_Pallets_using_a_Laser_Rangefinder_and_Machine_Learning_Techniques>.

Sharma, S., 2020. *Relu V/S Logistic Sigmoid*. [image] Available at:
<https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>.

7 APPENDICES

7.1 APPENDIX A: Project Code

The entire project was written in Python on Google Colaboratory. All the code and results can be accessed at the following link:

<https://drive.google.com/drive/folders/1LdUITyN803wXyP5b3RfOngLpiesSZBv>

The organization of the directories is described in the file ‘README’.

7.2 APPENDIX B: Relevant CNNs operations

Convolution operation: This operation involves a sliding window or a ‘kernel’ of dimensions $(f \times f)$ that operates on an image/mapping of dimension $(n_{\text{input}} \times n_{\text{input}} \times c)$ to produce a new mapping of dimension $(n_{\text{output}} \times n_{\text{output}} \times c)$. A convolution operation is shown below, where I is the input, K is the kernel and $I * K$ represents the convolution of K with I.

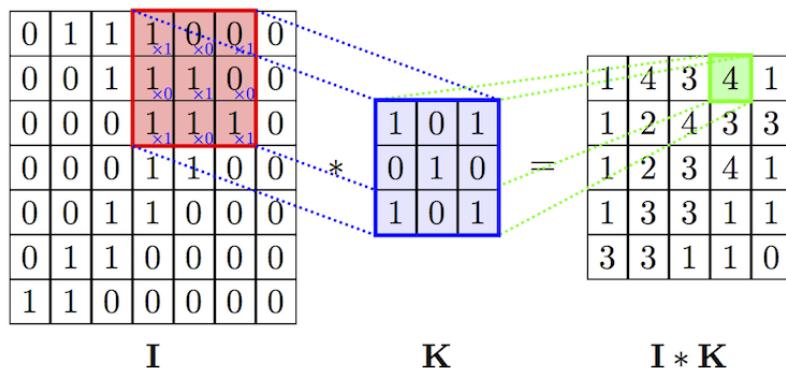


Figure 18: Convolution operation (Mohamed, 2017). I=image, K=kernel

The values stored in the kernel are what help them detect edges, shapes and complex features. For example, to detect a vertical line edge, a 3x3 kernel could be:



The goal of a CNN is to optimize these kernels to learn important features from the image. When performing convolutions, other parameters can be set including the stride s and the amount of padding p that is applied to the image. The stride s represents the size of the steps with which the kernel slides across the image. Padding an image means adding extra p pixels (usually storing value 0) around the image borders. This is usually done if we want the input and output of the convolution to have the same dimensions (also known as ‘same’ padding).

The relationship between the n_{input} and n_{output} is related to the values of f , p and s by the formula:

$$n_{\text{output}} = \left\lceil \frac{n_{\text{input}} - f + 2p}{s} \right\rceil \quad (1)$$

De-convolution: The mathematics behind a deconvolution is complex, however intuitively, it performs the opposite of a convolution. Rather than down-sampling a feature map, the de-convolution up-samples or increases the resolution of the image.

Pooling: This operation is always performed after a convolution, where a window of dimension (k, k) passes slides over a region of the feature map and can either take the maximum (max pooling) or average (average pooling) of the pixel values in the receptive field. This also can be varied by a stride s and the dimension of the output can be determined using equation (1). Pooling layers essentially reduce the dimension of a mapping by retaining important features.

Activation functions: These are essentially non-linear mathematical operations that are performed on the pixel values in feature maps. There are many types of activation functions – the ones that are commonly used in NNs and that are relevant to the models implemented in this study are the sigmoid and ReLU, for which the formulas are provided below in Table 3. Activations can be performed immediately after a convolution operation or after a pooling operation.

Table 3: Common activation functions for neural networks

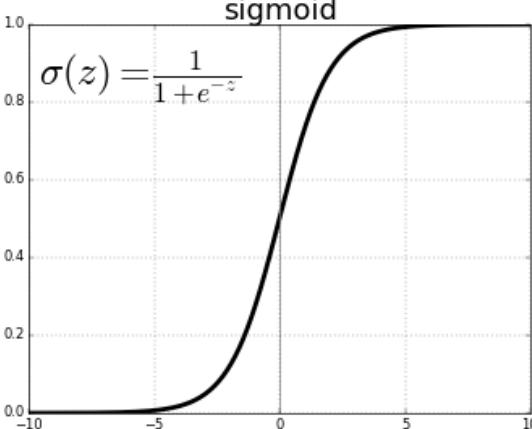
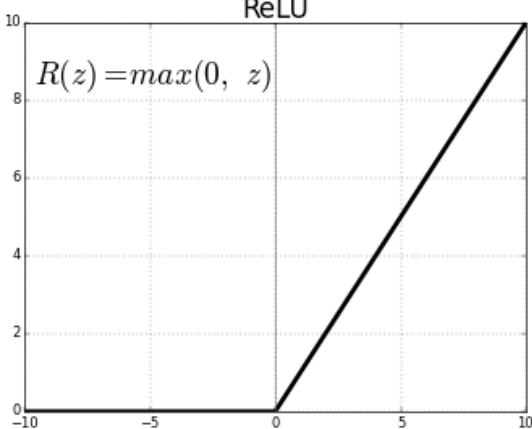
<p>Sigmoid/Logistic function: Operates on input z to create output $\sigma(z)$ in the range of $[0, 1]$. Used when the output of interest is a probability mapping.</p>	<p>ReLU: Rectified linear unit function. It essentially takes the maximum between 0 and the value z. This retains highly relevant information in the neural network while discarding irrelevant values.</p>
$\sigma(z) = \frac{1}{1 + e^{-z}}$ 	$R(z) = \max(0, z)$ 

Figure 19: Activation functions (Sharma, 2020)

Flattening: This operation converts a map of dimension $(n_{\text{input}} \times n_{\text{input}} \times c)$ into a vector of shape $((n_{\text{input}} * n_{\text{input}} * c) \times 1)$. This is performed when passing the feature map values into Dense fully connected layers.

7.3 APPENDIX C: Model hyperparameter tuning and regularization strategies

The additional hyperparameters tuned for our models are provided in Table 4, with specification of which model they were applied to.

Table 4: Hyperparameters tuned for the NNs.

Model	Hyperparameter	Details
UNet	Data augmentation	<ul style="list-style-type: none"> • rotation range = 10° • horizontal and vertical translation range = [-15, 15] px • Zoom range = [0.9, 1.2] • Vertical flips = True
UNet and AlexNet	Batch normalization	<ul style="list-style-type: none"> • UNet - Batchnorm added after every convolution operation, before performing the activation function. • Alex-Net - Batchnorm added after the pooling operations in the convolutional layers, and after each dense layer in the FC layers.
Alex-Net	Dropout=0.5	Dropout only added after each Dense layer.

7.4 APPENDIX D: UNet Raw Data

Table 5: Raw data for UNet predictions. m_{true} , m_{pred} are the slopes of the true and predicted lines. b_{true} and b_{pred} are the y-intercepts of the true and predicted lines. RMSD is computed between true and predicted line. The predicted line is the line created after image reconstruction.

*Note: DICE Scores are between the original UNet mask predictions and the true masks.

Image	DICE Scores*	m_{true}	m_{pred}	b_{true}	b_{pred}	RMSD (mm)
1	0.5916310324	-1.856	-1.699	256.926	240	2.203
2	0.4949534161	-9.431	-31.279	997.901	3104	3.483
3	0.6018833377	0.995	1.188	-17.302	-34	5.113
4	0.4160974161	-0.790	-1.147	144.010	175	10.952
5	0.3842087876	-4.226	-36.173	386.199	2639	9.807
6	0.5987163814	-1.856	-1.661	256.926	237	2.586
7	0.3699329716	-0.874	-1.063	154.032	172	6.303
8	0.3048977395	-0.664	-1.264	142.722	202	18.615
9	0.6646464646	-1.845	-1.670	230.041	214	2.423
10	0.6008518406	7.842	217.373	-535.225	-17536	6.257
11	0.08974358974	0.677	1.242	18.615	-40	18.467
12	0.518973519	-1.369	-1.676	165.702	191	6.120
13	0.2424153166	0.585	1.028	31.041	-12	15.253
14	0.3876181763	-1.879	-1.219	249.313	188	10.336
15	0.6683730247	-1.384	-1.261	209.032	199	2.709
16	0.5257792019	1.302	1.433	-19.347	-23	4.460
17	0.5726907631	-1.274	-1.147	207.271	191	3.824
18	0.4680924457	-6.888	45.366	633.166	-3592	7.732
19	0.3743354277	-0.970	-1.341	154.357	189	10.689
20	0.634253819	-1.616	-1.409	216.244	199	3.424
21	0.4652004454	-1.192	-1.530	183.132	218	7.760
22	0.6515272244	-1.690	-1.495	246.025	225	3.125
23	0.5562068966	13.829	19.349	-1113.784	-1619	1.772
24	0.555774926	7.561	6.129	-607.455	-463	2.709
25	0.5631632137	-0.793	-1.036	138.569	160	7.934
26	0.2813609467	-1.861	-1.560	222.662	189	6.965
27	0.6398619958	-1.353	-1.409	194.284	203	2.545
28	0.4079295154	-0.538	-0.794	130.445	152	9.314
29	0.501880243	-1.545	-2.011	227.963	268	6.536
30	0.6224947652	-1.761	-1.460	254.559	224	4.472
31	0.4958064516	-1.897	-1.803	212.155	213	3.778
32	0.5909531502	-14.852	42.029	1265.047	-3429	5.568
33	0.5884020619	1.099	1.091	3.251	8	2.848
34	0.4240292628	-1.564	-1.137	234.940	192	8.318
35	0.6871249318	-1.661	-1.661	206.603	202	2.395
36	0.4981032973	-1.667	-1.252	210.037	171	8.245
37	0.7669792555	-1.824	-1.790	228.418	222	1.843
38	0.4290598291	12.812	116.105	-964.685	-9486	3.324

Image	DICE Scores*	m_true	m_pred	b_true	b_pred	RMSD(mm)
39	0.4542635659	-1.054	-1.068	184.463	179	4.645
40	0.6302128901	-1.393	-1.606	192.088	210	3.740
41	0.4839679359	-2.483	-1.729	299.724	231	7.073
42	0.5225248155	-1.224	-1.013	180.616	161	5.791
43	0.5326560232	-2.581	-1.696	334.342	248	8.051
44	0.3980711673	-6.154	-16.815	646.488	1518	8.105
45	0.6103238866	-53.937	29.021	4195.008	-2089	2.910
46	0.4524555904	-1.827	-1.322	222.287	180	7.877
47	0.524185336	0.851	1.042	10.174	-3	6.257
48	0.6819251896	-1.252	-1.213	177.115	170	2.754
49	0.5428485214	-1.672	-1.531	212.690	200	2.272
50	0.6869017632	-1.682	-1.472	236.616	215	3.427
51	0.3481416957	-1.001	-1.195	200.129	207	9.102
52	0.5942756716	-0.953	-1.104	150.979	164	4.694
53	0.5821678322	-0.997	-1.187	175.193	193	5.125
54	0.5342514124	-0.848	-1.057	198.799	216	5.501
55	0.4617461746	-2.092	-1.493	289.362	225	7.726
56	0.5034195659	-1.831	-1.332	248.272	203	7.558
57	0.5270969969	-5.053	-12.968	564.631	1332	5.585
58	0.6154458599	-1.764	-1.892	236.836	250	1.762
59	0.6024064871	-1.136	-1.261	175.874	185	3.300
60	0.4957378796	-0.851	-1.064	147.352	165	6.730
61	0.4636504254	32.407	-13.387	-3086.074	1393	4.876
62	0.3824570273	-0.946	-1.329	161.122	190	10.528
63	0.5166973442	-0.774	-1.034	141.147	158	8.762
64	0.451242319	-1.059	-1.251	159.447	177	5.559
65	0.4377256318	-1.472	-1.154	222.866	183	8.761
66	0.6639852907	-1.139	-1.088	177.829	174	1.449
67	0.4480892797	0.695	0.833	17.379	7	4.892
68	0.5932518597	-1.681	-1.503	217.877	201	2.972
69	0.4210106383	-1.048	-1.495	167.800	209	11.064
70	0.5700962696	-0.888	-0.914	167.319	167	2.056
71	0.7252717391	-1.321	-1.307	189.072	188	0.288
72	0.3361691784	-1.001	-1.411	176.472	216	9.929
73	0.6541420118	-2.010	-2.305	252.617	282	3.125
74	0.4735735736	-1.768	-2.400	226.725	289	7.354
75	0.5485178617	-1.591	-1.279	230.014	198	5.782
76	0.4692090395	-1.333	-1.212	201.925	181	6.785
77	0.6866478004	-1.488	-1.557	208.597	210	2.697
78	0.3966796185	1.325	1.459	-18.373	-29	2.667
79	0.5834048641	-1.792	-1.501	241.892	209	5.504
80	0.4907302308	-14.872	-13.614	1433.077	1307	0.896
81	0.2898391813	-2.833	-2.019	328.355	260	6.049
82	0.5460609911	-1.999	-1.797	236.431	213	4.336
83	0.5638034641	-37.266	-15.861	3129.169	1389	1.817

Image	DICE Scores*	m_true	m_pred	b_true	b_pred	RMSD(mm)
84	0.6928451649	-2.083	-1.773	256.021	227	3.635
85	0.5336651054	-1.327	-1.660	177.059	204	6.111
86	0.6001655629	1.208	1.347	-9.262	-18	3.274
87	0.6958256599	11.089	13.335	-801.775	-978	0.718
88	0.2507727975	2.036	7.511	-115.842	-733	20.368
89	0.7134486607	-1.344	-1.280	192.280	184	2.237
90	0.4989669421	-1.029	-1.233	190.415	215	5.822
91	0.4465753425	1.292	2.026	-39.195	-111	11.767
92	0.5059994147	-1.414	-1.644	205.997	231	4.528
93	0.5311827957	-16.863	-25.174	1657.534	2432	0.911
94	0.495186863	-1.978	-1.748	244.712	214	6.304
95	0.4314205738	3.154	6.021	-181.237	-420	6.876
96	0.3327561328	-2.451	-1.535	292.385	211	9.465
97	0.6953787257	-1.536	-1.371	213.027	197	3.083
98	0.4209265176	-0.781	-1.087	149.497	176	9.683
99	0.5826306914	-1.337	-1.399	187.037	195	2.137
100	0.5679046563	-1.679	-1.608	220.604	207	4.132
101	0.4861111111	-1.921	-1.391	260.450	208	7.571
102	0.2901376147	6.000	124.569	-446.002	-10750	7.345
103	0.1975623583	-1.712	-1.468	199.774	191	5.984
104	0.6631944444	-1.546	-1.328	202.674	184	3.992
105	0.7814029364	-17.213	-13.574	1457.158	1163	0.747
106	0.5119363395	-1.642	-1.366	218.172	192	4.882
107	0.6690920595	-74.864	18.171	6055.393	-1359	3.218
108	0.3424162837	2.517	1.639	-101.230	-31	9.177
109	0.5935828877	-1.464	-1.452	217.474	217	0.412
110	0.5555555556	-0.840	-0.993	154.805	167	4.999
111	0.4692090395	-1.333	-1.212	201.925	181	6.785
112	0.3614729459	-1.359	-1.156	211.757	196	5.240
113	0.6783159053	-1.561	-1.827	205.376	226	3.794
114	0.170115894	-0.229	-0.729	98.672	136	18.751
115	0.5977045321	-1.447	-1.652	192.407	213	4.145
116	0.3002904866	7.657	5.707	-570.328	-384	4.126
117	0.4226954492	-1.011	-1.072	161.361	170	3.150
118	0.5377735938	-1.298	-1.086	191.428	170	5.531
119	0.6449120938	-1.242	-1.313	174.184	179	1.637
120	0.6557142857	-1.421	-1.479	195.102	196	2.365
121	0.4404322615	-2.043	-1.379	283.759	213	9.167
122	0.4776441566	0.982	1.070	-8.404	-15	2.670

7.5 APPENDIX E: AlexNet Results

Table 6: Raw data for AlexNet predictions. r_{true} , r_{pred} are for the true and predicted lines respectively. $r_{error} = (r_{true} - r_{pred})$. θ_{true} and θ_{pred} are for the true and predicted lines respectively. $\theta_{error} = (\theta_{true} - \theta_{pred})$. RMSD is computed between true and predicted line.

Image	r_{true} (px)	r_{pred} (px)	r_{error} (px)	θ_{true} (rad)	θ_{pred} (rad)	θ_{error} (rad)	RMSD (mm)
1	121.865	121.028	0.837	0.494	0.892	-0.398	20.332
2	105.223	111.625	-6.402	0.106	6.077	-5.971	35.605
3	12.268	54.537	-42.269	5.495	6.255	-0.760	50.944
4	112.975	99.053	13.922	0.902	1.093	-0.191	14.710
5	88.932	77.561	11.371	0.232	6.258	-6.026	13.925
6	121.865	118.620	3.245	0.494	0.909	-0.414	21.288
7	115.950	123.927	-7.976	0.852	0.706	0.146	12.392
8	118.896	105.764	13.131	0.985	0.895	0.090	14.971
9	109.628	100.404	9.223	0.497	0.986	-0.489	26.979
10	67.707	79.902	-12.196	6.156	5.426	0.731	94.003
11	15.417	69.901	-54.484	2.166	4.120	-1.954	181.863
12	97.743	107.964	-10.221	0.631	0.943	-0.312	18.757
13	26.789	37.303	-10.514	2.100	6.255	-4.155	60.451
14	117.108	122.029	-4.921	0.489	0.747	-0.258	13.467
15	122.423	105.561	16.862	0.626	0.768	-0.143	18.816
16	11.784	19.020	-7.236	5.628	5.727	-0.098	6.678
17	127.964	114.577	13.387	0.665	1.208	-0.543	29.551
18	90.972	93.534	-2.562	0.144	6.121	-5.977	30.459
19	110.809	106.082	4.727	0.801	1.034	-0.233	14.869
20	113.813	101.392	12.421	0.554	1.153	-0.599	31.083
21	117.682	110.511	7.171	0.698	1.006	-0.308	18.528
22	125.267	118.404	6.863	0.534	0.835	-0.300	17.197
23	80.327	83.686	-3.359	6.211	1.844	4.367	53.084
24	79.649	76.100	3.548	6.152	2.005	4.146	57.685
25	108.590	103.664	4.927	0.901	0.722	0.178	12.009
26	105.408	98.072	7.336	0.493	0.783	-0.290	19.519
27	115.455	106.938	8.517	0.636	1.140	-0.504	27.690
28	114.895	117.519	-2.624	1.078	1.204	-0.126	9.706
29	123.848	121.420	2.428	0.574	0.787	-0.212	11.898
30	125.674	119.199	6.475	0.516	0.791	-0.275	16.106
31	98.929	99.452	-0.523	0.485	0.902	-0.417	22.026
32	84.985	88.895	-3.910	0.067	5.623	-5.555	80.721
33	2.189	27.897	-25.708	2.403	5.982	-3.579	30.261
34	126.557	124.717	1.840	0.569	0.768	-0.199	11.012
35	106.567	110.153	-3.586	0.542	0.934	-0.392	20.608
36	108.027	105.889	2.138	0.540	0.699	-0.159	9.983
37	109.829	120.860	-11.031	0.502	0.735	-0.234	14.030
38	75.066	98.558	-23.492	6.205	4.882	1.323	169.190
39	126.985	123.216	3.769	0.759	1.155	-0.395	22.632

Image	r_{true} (px)	r_{pred} (px)	r_{error} (px)	θ_{true} (rad)	θ_{pred} (rad)	θ_{error} (rad)	RMSD (mm)
40	111.995	116.986	-4.991	0.622	0.966	-0.343	19.913
41	111.958	114.814	-2.857	0.383	0.831	-0.448	21.851
42	114.289	124.652	-10.363	0.685	0.872	-0.187	15.359
43	120.806	129.981	-9.175	0.370	0.701	-0.331	16.398
44	103.690	118.729	-15.039	0.161	0.840	-0.679	29.427
45	77.763	85.317	-7.553	0.019	6.204	-6.185	16.204
46	106.725	109.168	-2.443	0.501	0.864	-0.363	18.773
47	7.748	70.984	-63.236	2.276	6.114	-3.838	39.265
48	110.555	94.475	16.080	0.674	0.941	-0.267	22.129
49	109.189	117.572	-8.383	0.539	0.923	-0.384	21.034
50	120.942	121.369	-0.427	0.537	0.721	-0.185	9.946
51	141.444	136.526	4.918	0.785	0.705	0.080	5.915
52	109.273	106.054	3.219	0.809	0.969	-0.160	10.144
53	124.052	115.635	8.417	0.787	1.073	-0.286	16.849
54	151.612	95.290	56.322	0.867	0.956	-0.089	55.287
55	124.806	131.839	-7.033	0.446	1.009	-0.563	29.481
56	119.010	110.947	8.063	0.500	0.798	-0.298	18.564
57	109.611	109.571	0.039	0.195	1.003	-0.808	35.174
58	116.820	117.550	-0.730	0.516	1.567	-1.051	59.499
59	116.220	102.650	13.570	0.722	0.866	-0.145	15.625
60	112.207	115.762	-3.555	0.866	0.889	-0.023	4.035
61	95.182	86.853	8.329	6.252	1.657	4.595	48.563
62	117.037	121.368	-4.331	0.813	0.795	0.018	4.488
63	111.610	119.616	-8.006	0.912	1.000	-0.088	10.977
64	109.474	126.089	-16.615	0.757	0.800	-0.043	16.687
65	125.245	93.832	31.413	0.597	1.166	-0.569	34.832
66	117.328	110.534	6.795	0.721	1.353	-0.632	38.988
67	14.269	34.278	-20.009	2.178	5.315	-3.136	49.105
68	111.382	119.367	-7.985	0.537	0.928	-0.391	21.426
69	115.853	117.766	-1.913	0.762	0.991	-0.229	15.187
70	125.111	129.695	-4.585	0.845	0.943	-0.099	7.818
71	114.123	121.898	-7.775	0.648	0.839	-0.191	13.184
72	124.737	80.502	44.235	0.785	1.228	-0.443	40.710
73	112.525	106.646	5.879	0.462	0.931	-0.469	24.819
74	111.602	93.442	18.160	0.515	0.867	-0.352	27.876
75	122.381	115.624	6.757	0.561	0.801	-0.240	14.870
76	121.189	125.336	-4.148	0.644	0.719	-0.075	5.785
77	116.335	95.472	20.862	0.592	0.877	-0.286	26.285
78	11.068	29.647	-18.579	5.637	5.013	0.624	90.248
79	117.877	141.609	-23.733	0.509	0.630	-0.121	22.470
80	96.144	109.890	-13.746	0.067	1.318	-1.251	45.038
81	109.286	94.340	14.946	0.339	1.910	-1.570	68.925
82	105.792	117.330	-11.537	0.464	0.585	-0.121	10.109
83	83.937	77.383	6.554	0.027	1.235	-1.208	49.691

Image	r_{true} (px)	r_{pred} (px)	r_{error} (px)	θ_{true} (rad)	θ_{pred} (rad)	θ_{error} (rad)	RMSD (mm)
84	110.808	99.496	11.311	0.448	0.759	-0.311	22.893
85	106.573	92.793	13.780	0.646	0.885	-0.239	20.537
86	5.906	27.119	-21.213	5.592	5.545	0.047	26.473
87	72.009	71.959	0.050	6.193	1.660	4.533	45.573
88	51.076	37.184	13.892	5.827	2.537	3.289	71.360
89	114.788	108.617	6.170	0.640	0.849	-0.209	13.603
90	132.725	153.746	-21.021	0.771	0.831	-0.060	21.467
91	23.995	43.519	-19.524	5.624	6.230	-0.606	55.192
92	118.964	135.038	-16.074	0.616	0.822	-0.207	19.677
93	98.120	112.433	-14.313	0.059	5.893	-5.834	59.615
94	110.422	94.340	16.082	0.468	0.850	-0.382	28.141
95	54.772	53.560	1.212	5.976	6.055	-0.079	9.778
96	110.464	110.510	-0.046	0.387	1.305	-0.917	41.056
97	116.236	91.505	24.732	0.577	1.092	-0.515	33.160
98	117.836	113.480	4.356	0.908	1.305	-0.398	24.115
99	112.035	120.365	-8.330	0.642	0.781	-0.139	10.784
100	112.863	99.663	13.200	0.537	0.779	-0.241	20.500
101	120.282	126.701	-6.419	0.480	0.835	-0.355	18.648
102	73.317	62.422	10.895	6.118	6.150	-0.032	13.948
103	100.769	109.427	-8.657	0.529	0.703	-0.174	10.271
104	110.080	100.979	9.101	0.574	0.922	-0.348	21.482
105	84.511	66.695	17.816	0.058	2.545	-2.487	92.602
106	113.470	111.351	2.118	0.547	1.318	-0.771	39.917
107	80.878	60.218	20.660	0.013	1.418	-1.405	54.774
108	37.374	23.348	14.026	5.905	5.226	0.679	65.242
109	122.661	134.423	-11.762	0.599	0.590	0.009	11.886
110	118.517	111.670	6.847	0.872	1.048	-0.176	11.390
111	121.189	125.336	-4.148	0.644	0.719	-0.075	5.785
112	125.479	86.022	39.458	0.634	0.956	-0.322	39.439
113	110.779	112.408	-1.628	0.570	0.741	-0.172	9.388
114	96.187	140.426	-44.239	1.346	1.118	0.228	34.944
115	109.364	122.793	-13.429	0.605	1.385	-0.781	49.737
116	73.855	75.345	-1.490	6.153	6.267	-0.114	9.755
117	113.502	102.754	10.747	0.780	0.844	-0.064	11.292
118	116.835	123.460	-6.625	0.656	0.974	-0.317	20.380
119	109.230	118.417	-9.187	0.678	1.046	-0.368	24.202
120	112.281	109.144	3.137	0.613	1.006	-0.393	21.635
121	124.727	102.113	22.614	0.455	0.971	-0.516	32.241
122	5.997	33.538	-27.541	5.489	6.193	-0.705	58.683