International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 8s, 2025
https://www.theaspd.com/ijes.php

A Deep Learning Framework For Myocardial Infarction Detection From Apical-4-Chamberechocardiograms: Combining Segmentation And Machine Learning Classification

Noorhan H. I. Mohamed[1], Walaa H. Elashmawi[2,3], Mohamed H. Mohamed[4], Eman M. Gaber Hassan[5,6]

[1]Faculty of Engineering, Ain Shams University, El Sarayat Street, Abassiya, El Weili, Cairo, 11517, Egypt.
[2]Faculty of Computers and Informatics, Suez Canal University, Ring Road, El Salam District, El Sheikh Zayed City, 8366004, Egypt.
[3]Faculty of Computer Science, Misr International University, KM 28 Cairo-Ismailia Road, Ahmed Orabi District, Cairo, 6363001 Egypt.
[4]Faculty of Engineering, University of Freiburg, Fahnenbergplatz, Freiburg im Breisgau, 79085, Germany.
[5]Department of Nursing, Mohammed Al-Mana College for Medical Sciences, Dammam, Saudi Arabia.
[6] Medical Surgical Nursing Department, Faculty of Nursing, Cairo University, Gamaa Street, Giza, 12613, Egypt.
Correspondence: noorhanhatem@gmail.com , https://orcid.org/0009-0006-3521-0042

Abstract

Purpose: *Myocardial infarction is a highly fatal cardiac disease caused by reduced blood flow to parts of or the whole cardiac muscle. Early detection and immediate intervention can greatly reduce the severity of damage to the heart muscle. This work presents a framework to detect MI in echocardiogram videos to increase the swiftness of MI diagnosis.*

Methods: *The framework functions in three key stages. In the first stage, a LadderNet CNN segments the left ventricle wall out of the echocardiogram's frames. In the next Stage, the segmented left ventricle in each frame is divided into six sections whose displacements are tracked across the consecutive frames. In the final stage, a machine learning model classifies the presence of MI by analyzing the displacements of the wall's sections. Five classic algorithms were assessed and compared for the classification task. The HMC-QU benchmark dataset was used for the training and testing of the framework.*

Results: *The segmentation model demonstrated excellent performance with an Intersection over Union of 97.32%, an F1 score of 97.40%, an accuracy of 99.78%, a precision of 97.50%, a sensitivity of 97.31%, and a specificity of 99.89%. For MI classification, the optimal model (Random Forest) achieved 85.71% accuracy, 86.67% precision, 92.86% sensitivity, 71.43% specificity, and 89.66% F1 score.*

Conclusion: *The promising results of this work suggest that the proposed framework has the potential to contribute to the detection of myocardial infarction.*

Keywords: *Echocardiogram, Myocardial infarction, Deep learning, Convolution neural network, Machine learning.*

1. INTRODUCTION

Myocardial infarction (MI) is a common critical health challenge with a relatively high mortality rate. It is defined as the irreversible injury to cardiac muscle due to prolonged inadequate blood flow [1]. The most prevalent cause of MI is coronary artery disease (CAD). The build-up of atherosclerotic plaques within the coronary arteries results in occlusions that reduce blood flow to the heart muscle, thus depriving it partially

or fully of oxygen which can lead to death of the cardiac muscle's cells [2]. Early identification and quick action are essential in improving MI patient outcomes; timely intervention can significantly reduce myocardial damage as well as mortality rate [3, 4].

Echocardiograms, also known as Heart Sonograms, are popular, non-invasive cardio-vascular disease diagnostic tools used by medical practitioners as a reliable and accurate means of detecting MI [5]. There are many types of echocardiograms, the most common of which is a two-dimensional sonogram video of the movement of the heart's four chambers. A cardiologist aiming to identify possible myocardial dysfunction would analyze and evaluate the motion of the heart's walls in the echocardiogram, particularly the left ventricle (LV) wall, as all MIs affect the LV. MI is used interchangeably with "left ventricular infarction" since almost all MIs occur in the LV, and even in the rare cases of right ventricular infraction, the LV is also affected. If a section of the LV wall moves abnormally or displays attenuated (weakened) motion, that indicates it is suffering from reduced blood flow and is diagnosed with MI [6].

This paper was crafted with the aim of creating a framework that uses computer vision and AI techniques to analyze echocardiogram videos and diagnose the presence of MI by tracking the LV wall movement throughout the 2D video and detecting wall motion abnormalities. The proposed framework is composed of 3 main phases. It starts off by extracting and preprocessing each frame of the input video (echocardiogram to be diagnosed) and then passing the frames to the first phase of the framework, Segmentation, in which the LV wall of each frame is segmented using an encoder-decoder CNN. After that, the framework moves on to the next phase, Feature Engineering, where each segmented LV wall is divided into sections, the motion of each of which is tracked by observing its displacement throughout the consecutive frames to create a motion feature vector. In the next and final phase, MI Classification, a machine learning (ML) classifier is used to diagnose MI or lack thereof based on the echocardiogram's feature vector.

The organization of this paper is as follows: Section 2 is a review of published literature concerning MI, echocardiograms, and the use of AI and other technologies in echocardiography. Section 3 goes over the dataset used, Section 4 introduces and details the 3-phase proposed model including the architecture of the encoder-decoder segmentation CNN, how feature engineering was carried out, and a comparison of multiple traditional ML classifiers for MI detection. Section 5 presents the performance results of the segmentation CNN and the tested ML classifiers. Finally, Section 6 concludes the main findings of this research.

2. RELATED WORK

Echocardiography is a popular non-invasive imaging tool for cardiac assessment where sound waves are used to generate detailed images of the heart, allowing for the evaluation of its structure and function. Wall motion abnormalities, cardiac chamber sizes, and overall systolic and diastolic function can all be clearly assessed from the images of an echocardiogram [7]. Not only has it proven itself as an invaluable tool for healthcare providers in the diagnostic assessment of MI [8], it has also demonstrated a prognostic value in risk assessment of patients that have previously suffered from MI [9]. An expert review by [10] found that echocardiograms had an essential prognostic role in early risk stratification of patients after MI by predicting long-term adverse outcomes.

The development of AI-based frameworks for echocardiography to detect MI and for other purposes has garnered significant interest in recent years [11, 12]. Traditionally, MI detection using echocardiography relies on visual assessment by clinicians, which can be subjective and operator-dependent [7, 13]. Therefore, computer-assisted techniques, specifically artificial intelligence models, have been looked into as a way to overcome these limitations by providing objective and reproducible analysis [14, 15]. AI applications in this

domain are not limited to diagnostic tasks but also extend to automating the measurement of various cardiac parameters, potentially increasing efficiency and consistency in patient evaluations [16].

Accurately identifying regions of the heart from echocardiogram videos is the most critical baseline task in any echocardiogram-based diagnostic model [16], especially since publicly-available echocardiography datasets with cardiologist-approved ground truth masks are scarce. Multiple papers have explored and proposed different methods of segmenting the heart's walls and chambers from apical-4-chamber-view echocardiogram videos. Initially, earlier studies relied on contour-based models for LV segmentation [17-21]. As artificial intelligence gained more prominence in biomedical image segmentation, the most popular approach to echocardiogram segmentation became supervised learning using convolution neural networks (CNNs) trained on echocardiogram videos' frames along with their respective masks that highlight the region of interest the researchers want to segment [22-27]. A comparative analysis by [28] examined the performance of both CNNs and active contours in rapid LV segmentation and found no significant difference between the performance metrics of the two techniques.

Almost all CNN-based echocardiographic image segmentation approaches employ an encoder-decoder architecture, which is expected as it is the architecture used in most classic image semantic segmentation algorithms. DeepLab, UNet, and FCN are all examples of encoder-decoder CNNs used for image segmentation [29]. This is because the encoder path (subsampling network) is used to learn multilevel semantic features that the decoder path (oversampling network) then maps to the pixel space of the original resolution for pixel-level classification [30].

Another common approach to echocardiogram segmentation is using unsupervised techniques to separate the heart's regions. A common unsupervised learning technique for echocardiography and medical imaging is Unsupervised Domain Adaptation (UDA). GraphEcho, an unsupervised Graph-Driven Domain Adaptation approach proposed by [31] was able to achieve better echocardiography segmentation results than most traditional UDA and supervised methods. [32] proposed neural network collaborative filtering as an unsupervised, automated method for accurately segmenting the mitral valve from 2D echocardiogram frames.

Straying from traditional techniques, [33] designed a novel semi-supervised proxy- and kernel-based echocardiography video segmentation approach that managed to achieve a better trade-off of performance and efficiency than other state-of-the-art approaches. Furthermore, [34] presented a new extension of active appearance models for the use of echocardiogram segmentation that was able to outperform 2D traditional active appearance models.

Choosing the best technique to detect wall motion irregularities after segmenting the LV wall is the next critical step in echocardiogram-based diagnostic models. Some proposed frameworks opted for deep learning (DL) technique for MI classification like for echocardiogram image segmentation and achieved great results [24, 25], other proposed frameworks either opted for a singular ML technique [23] or an ensemble of ML techniques [22] with results on par with the DL models. Some papers explicitly did not favor the DL techniques due to the inconsistency of their outcomes and their dependency on the quality of the training data as well as the difficulty in determining the optimal hyperparameters for them [22]. ML models were also seen as more appropriate for such small, imbalanced datasets [22, 23].

3. UTILIZED HMC-QU DATASET

The proposed framework was trained and tested using the HMC-QU benchmark dataset [23, 35] as it is the first and only publicly available dataset serving MI detection in echocardiograms. It consists of 162 echocardiography apical-4-chamber-view (A4C) recordings and 160 echocardiography apical-2-chamber-view (A2C) recordings that were obtained by a collaboration between Hamad Medical Corporation, Tampere

University, and Qatar University. The recordings are of temporal resolution 25 fps (frames per second) and varied spatial resolution of 422x636 to 768x1024 pixels. The echocardiograms were captured using Phillips and GE Vivid ultrasound machines. The dataset is accompanied by 109 224x224 LV ground-truth masks for 109 of the 162 aforementioned A4C view videos, but no masks for the A2C view. Furthermore, the dataset comes with two Excel files, one for A4C and one for A2C, that detail whether or not each of the six segments of the LV wall displays MI for each video in the dataset (the LV wall is typically divided into seven sections, but the Apical Cap was ignored in the dataset as it does not exhibit inward motion). Additionally, the Excel files list the start and end reference frames for 1 full cardiac cycle for each video. These reference frames also indicate the subset of frames for each A4C video that have ground-truth segmentation masks. Finally, the A4C Excel file has a column indicating whether each video has ground-truth segmentation masks or not. Figure 1 shows a snippet of the A4C Excel file.

ECHO	LABEL						One cardiac-cycle frames		LV Wall Ground-truth Segmentation Masks	
	SEG3	SEG9	SEG14	SEG16	SEG12	SEG6	Reference Frame	End of Cycle	Available: ✓	Not Available: ✗
ES0001 _4CH_1	MI	MI	MI	MI	MI	non-MI	1	20		✓
ES00010 _4CH_1	non-MI	non-MI	MI	non-MI	non-MI	non-MI	1	21		✓
ES000102 _4CH_1	non-MI	non-MI	MI	MI	non-MI	non-MI	1	22		✓
ES000103 _4CH_2	MI	MI	MI	MI	non-MI	non-MI	1	16		✓
ES000105 _4CH_1	non-MI	MI	MI	non-MI	non-MI	non-MI	1	20		✓
ES000106 _4CH_1	MI	MI	MI	MI	non-MI	non-MI	1	18		✓
ES000107 _4CH_1	MI	MI	MI	MI	MI	MI	45	61		✓
ES000108 _4CH_1	non-MI	non-MI	MI	non-MI	non-MI	non-MI	1	16		✓
ES000109 _4CH_2	non-MI	non-MI	MI	MI	non-MI	non-MI	1	16		✓
ES00011 _4CH_1	non-MI	non-MI	non-MI	non-MI	non-MI	non-MI	1	17		✓
ES000112 _4CH_3	non-MI	non-MI	MI	MI	non-MI	non-MI	1	21		✓
ES000113 _4CH_2	MI	MI	MI	MI	MI	MI	1	20		✓
ES000114 _4CH_1	MI	MI	MI	MI	MI	MI	1	13		✓

Figure 1: HMC-QU A4C File Snippet

For the purposes of this project, the focus will be placed only on the A4C echocardiograms, as medical-specialist-approved ground-truth segmentation masks for A2C echocardiograms are not available. Furthermore, only the subset of 109 videos with the ground truth masks show the entire LV wall in a4C view and since a medical professional would typically need to access the entire wall to make a diagnosis, the remaining 53 echocardiograms that do not display the full LV wall will also be excluded.

4. PROPOSED FRAMEWORK

The echocardiography framework should be able to accurately segment the region of interest (LV wall) out of each frame of an input 2D A4C echocardiogram and track the motion of different segments of the LV to detect the presence of MI, or lack thereof. The full proposed framework, inspired by the paper published by Nguyen et al. [22], consists of three main stages. First, the framework takes a 2D A4C echocardiography video as input extracts its frames, and processes (re-shapes) them. Then starts the first phase, LV Wall Segmentation, where the processed frames are passed through an encoder-decoder CNN to segment the LV wall out of each frame. In the second phase, Feature Engineering, each of the segmented LV walls is divided into six sections which are analyzed to obtain an overall motion feature vector for the input echocardiogram. Each motion feature vector contains six elements, each element represents the greatest displacement achieved by each of the six segments throughout the frames of the input video. Finally, in the last phase, MI Classification, an ML model is utilized to classify the motion feature vector as “MI” or “non-MI”. Figure 2 illustrates the overall architecture of the proposed framework.

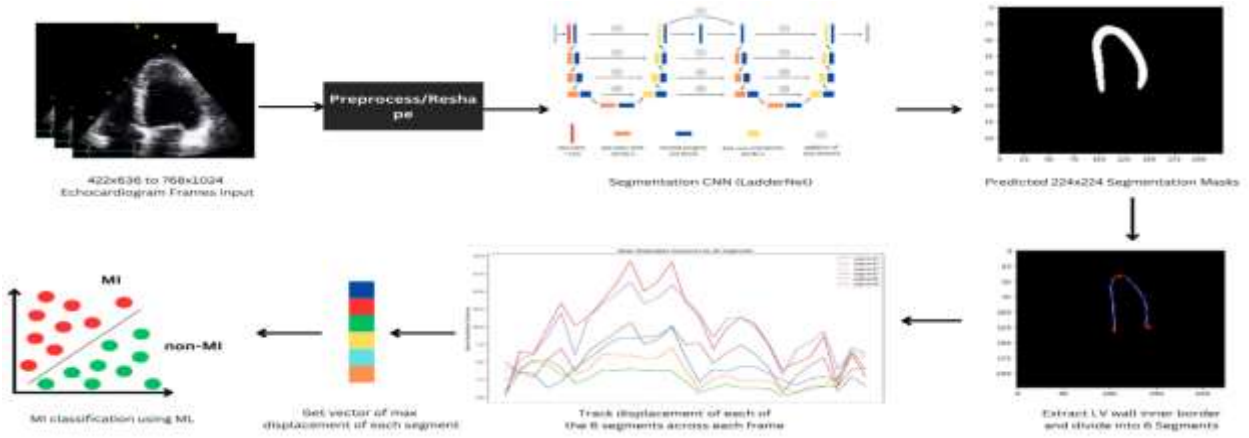

Figure 2: The Framework Pipeline

4.1. Segmentation model

The most critical task of the proposed framework is LV wall segmentation as it forms the basis upon which the other phases of the framework are built. The efficacy or lack thereof of the LV segmentation will affect the next phase's ability to track the LV wall sections' displacements and ultimately, the final phase's ability to reliably identify MI. Figure 3 shows an example of the desired segmentation task.

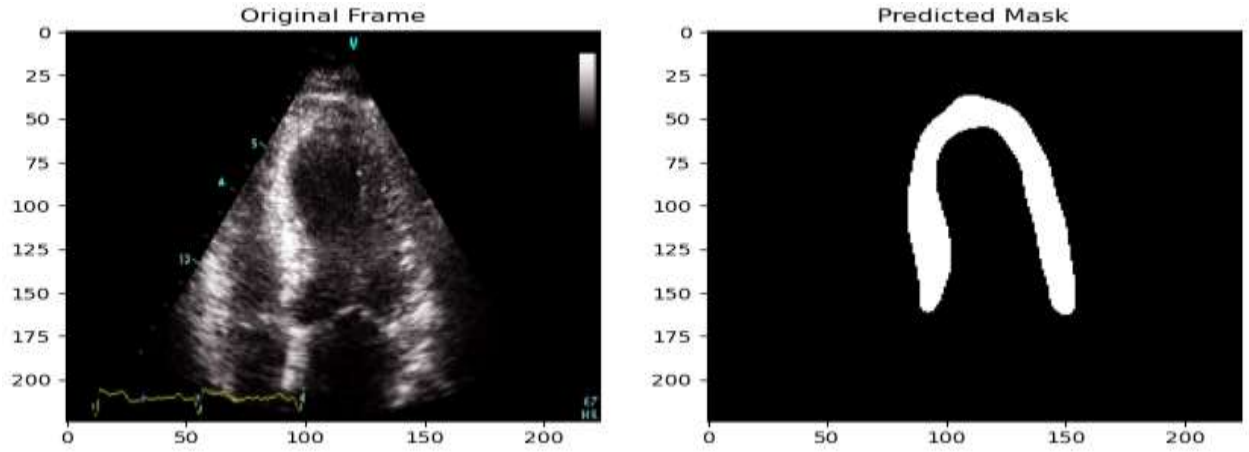

Figure 3: Example of the desired segmentation model output

Unlike the segmentation model proposed by Nguyen et al. [22] which used an ensemble of multiple different segmentation CNNs working together, the proposed framework implemented LV wall segmentation using a single LadderNet [36] which is a CNN architecture based on the design of U-Net [37]. U-Net is an encoder-decoder CNN that was created specifically for biomedical image segmentation and has been revolutionary in its performance compared to previous networks used for biomedical image segmentation. LadderNet, as shown in figure 4, is a modified version of U-Net that consists of multiple pairs of encoder-decoder branches instead of just one pair. Furthermore, there are skip connections in between all adjacent decoder-pairs, which provides more paths for information flow. Laddernet functions as a series of multiple U-Nets combined together.

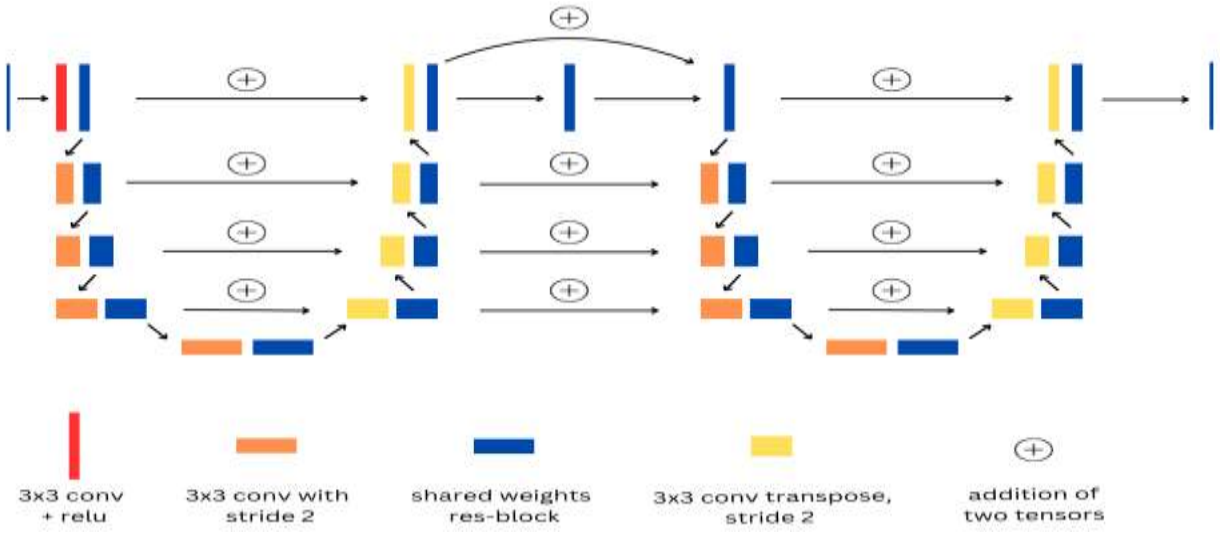

Figure 4: Laddernet Architecture

To train the LadderNet, the relevant frames of the single cardiac cycles that have segmentation ground-truth segmentation masks were extracted using the reference frame numbers provided in the A4C excel file from the 109 videos that have ground-truth segmentation masks for some of their frames. Each video's frames was stored in a distinct directory. These frames were then processed into the shape (224, 224, 3) and used to train the LadderNet along with their respective ground-truth segmentation masks. 20% of the data was put aside for testing and a further 20% were used for validation. The optimizer used was the Adaptive Moment Estimation (Adam) algorithm, the loss function chosen was Sparse Categorical Cross-Entropy. The model was trained with batch size 8 for 50 epochs. Figure 5 displays the training and validation accuracy-across-epoch chart and figure 6 shows the training and validation loss-across-epoch chart.

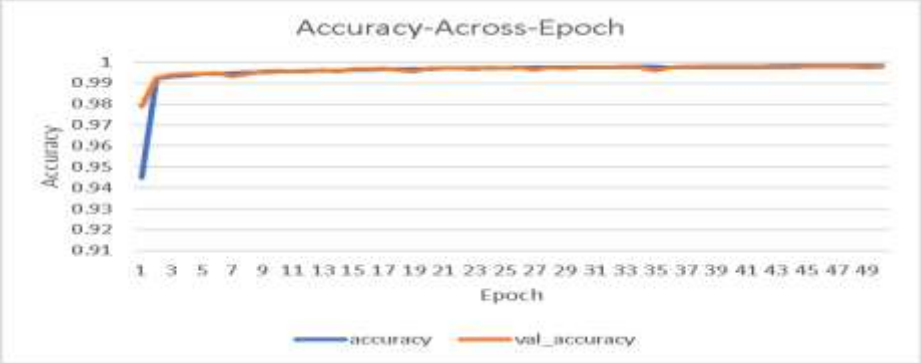

Figure 5: Training-Validation Accuracy Chart

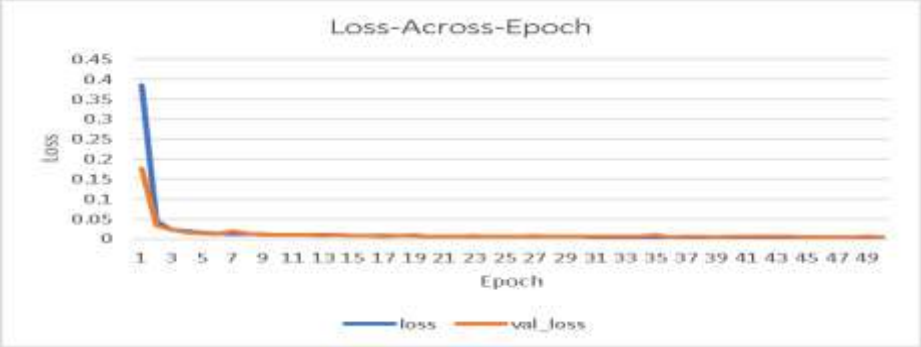

Figure 6: Training-Validation Loss Chart

4.2. Feature engineering

The LV wall is typically divided into seven sections: the Basal Inferoseptum, the Mid Inferoseptum, the Apical Septum, the Apical Cap, the Apical Lateral, the Mid Anterolateral, and the Basal Anterolateral. However, for the proposed framework, only six of those sections will be assessed. The Apical Cap is disregarded as it does not exhibit inward motion according to analysis by done by multiple researchers designing A4C echocardiogram MI detection models [22, 23]. The excel files accompanying the HMC-QU video dataset also disregard the Apical Cap and list MI classification data for the other six sections only.

After the LV wall is segmented in phase I, the inner border (bottom border) is extracted from the set of output predicted LV wall masks for the frames of each particular video. The extracted border is then divided into 6 sections, 3 on the left-side of the wall and 3 on the right-side. To do this, the apex point of the entire LV wall, the bottom-most point on the right-side of the border, and the bottom-most point on the left-side of then border need to be first located. Figure 7 shows an example of inner border extraction along with the apex point, left bottom-most point, and right bottom-most point, highlighted in red, blue, and green respectively. After locating the three reference points, R is measured as the height (y-value) difference between the bottom-most point on the right-side of the border and the apex; L is measured as the height (y-value) difference between the bottom-most point on the left-side of the border and the apex.

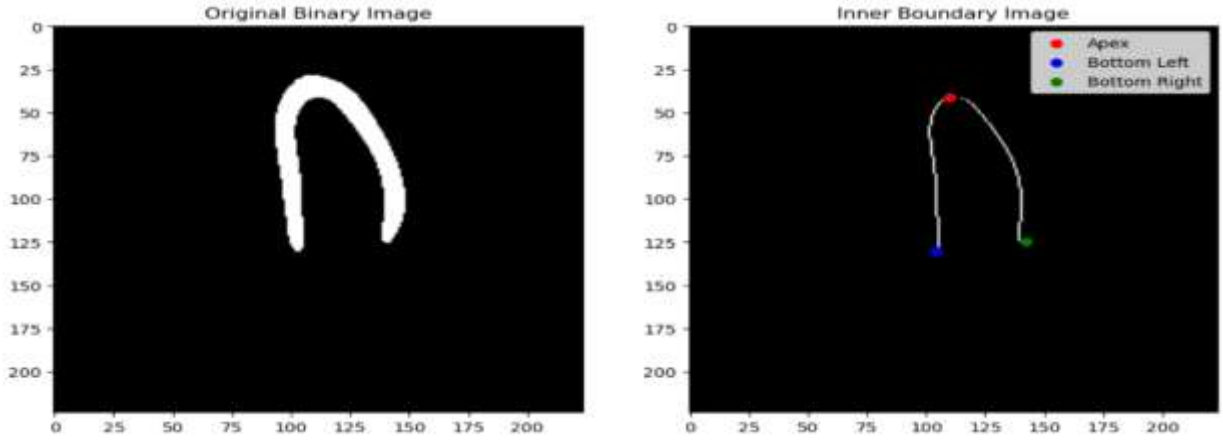

Figure 7: Example of extracting the inner boundary

After calculating L and R, the left side of border is divided into 3 sections, each of height $\frac{2L}{7}$ and the right side of the border is similarly divided into 3 sections, each of height $\frac{2R}{7}$. Figure 8 shows an example of a sectioned border. The apex point as well as the left and right bottom-most points are shown in red; the division points between the sections are shown in blue.

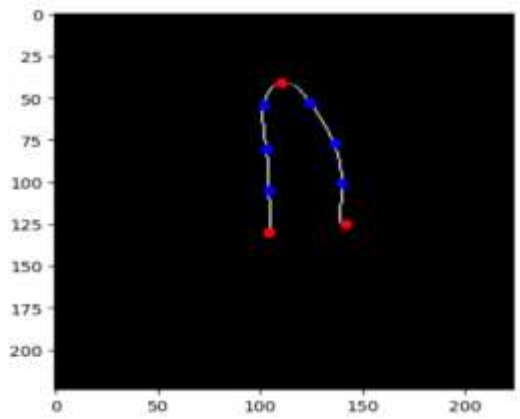

Figure 8: Example of sectioning the extracted border

Now that the inner boundary of the LV wall is sectioned, the model can track the displacement (compared to the first frame) for each section in each subsequent frame of the video. To accomplish that, for each section in each frame, N equally-spaced reference pixels are obtained by dividing each section into $N - 1$ equally-spaced intervals. For the proposed framework, several values for N were tested and it was found that the different values for N did not have a significant effect on MI classification in the next phase as long as there weren't too few of them $(N > 5)$. In the final framework, $N = 10$ was chosen for the feature engineering phase as it appeared to give an adequate number of reference pixels without being too taxing. In figure 10, an $N = 5$ pixel selection is displayed to make it easier to see. The three orange points in figure 9 represent the apex and the two bottom-most points; each section's $N = 5$ points are shown in a distinct color.

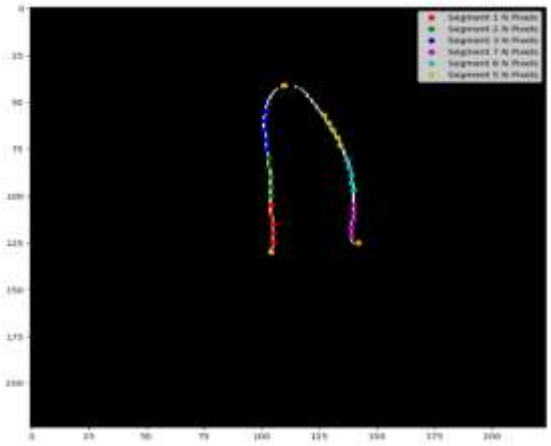

Figure 9: Selecting N equally spaced pixels from each section

Now that the N pixels for each section in each frame are selected, the displacement between each selected pixel in the first frame and its corresponding selected pixel in every other frame of the video is calculated to

track how much each section has moved from its position in the first frame. The displacement is calculated using the Manhattan distance as shown in equation 1.

$$\text{displacement} = |x_1 - x_2| + |y_1 - y_2| \quad (1)$$

Then, for each segment in each frame, we take an average displacement of all the displacements of its N pixels as shown in equation 2.

$$\text{displacement}_{\text{average}} = \frac{1}{N}\sum_{n=1}^{N} \text{displacement}_n \quad (2)$$

If the average/mean displacement for each section across all the frames of a video were plotted, a line graph that looks something like figure 10 would be obtained.

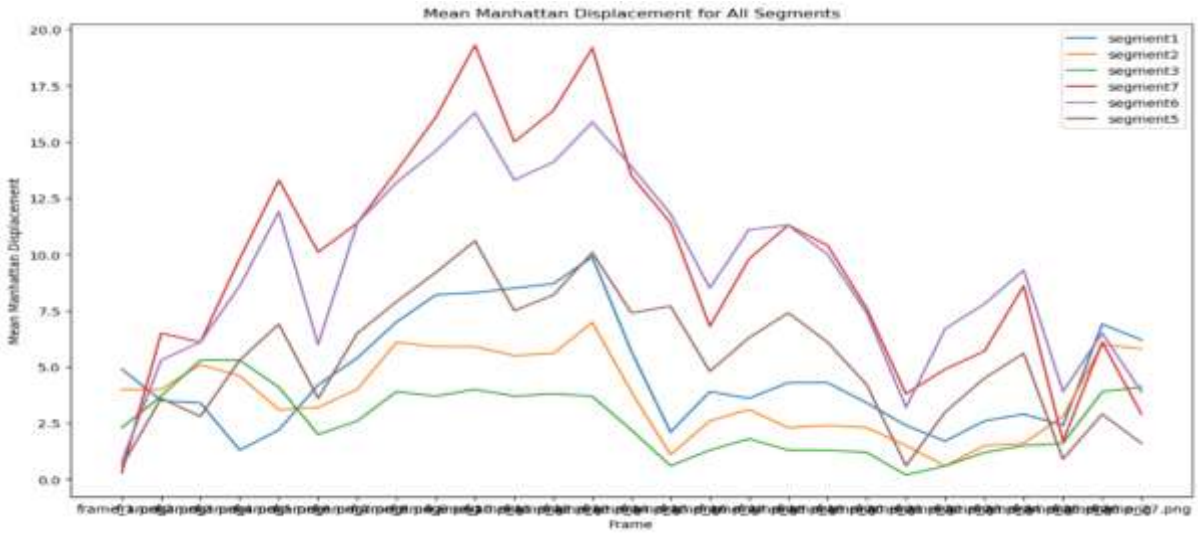

Figure 10: Average displacement of each section across the frames of a video

The final step of feature engineering is to get the motion feature vector for each video. The motion feature vector consists of six elements, one for each section's maximum average displacement (the peak of each section's line plot on the graph). To train the classification ML models in the next phase, 104 motion feature vectors were obtained by applying all the previous feature engineering steps to each of the HMC-QU dataset's 109 videos after extracting each video's frames and passing them through the segmentation LadderNet to get their predicted LV wall masks. For 5 of the 109 echocardiograms, one or more of their frames' predicted LV wall masks had missing sections that made the framework unable to divide the LV wall into the full 6 segments. We chose to disregard those 5 echocardiograms instead of just excluding the defect frame or frames because the diagnosis of MI requires analysis of a full cardiac cycle and removing some frames might lead to incorrect predictions. The defect frames were not a result of the LadderNet's lack of segmentation ability as the ground truth masks of those frames also had missing sections and were not able to be divided them into the required full segments. The cause behind the flaws appears to be the poor quality of the frames they were extracted from. Figure 11 shows examples of the defect ground truth masks from 3 different echocardiograms.

Figure 11: Examples of defect ground truth masks

4.3. MI classification model

Five classic ML classification algorithms: support vector machines (SVM), decision trees (DT), random forest (RF), k-nearest neighbors (KNN), and logistic regression (LR) were trained and tested on the motion featured vectors obtained from tracking the displacements of the LadderNet's predicted masks. Their performances were compared to each other in terms of accuracy, precision, sensitivity, specificity, and F1 score. These algorithms were favored over a DL model during the design of the framework because the HMC-QU dataset is small and imbalanced with much higher non-MI class instances than MI. This is inline with other studies that also favored an ML approach for the imbalanced data [22, 23]. Like in the segmentation model, a 20% random test data split was employed. A grid search was performed on each algorithm to determine its optimal hyperparameter settings for the task at hand. The following subsections will cover each algorithm individually as well as list its grid search results in a table.

4.3.1. Decision tree

DT is a supervised ML classification algorithm that operates by systematically dividing the dataset into subsets according to the characteristics of the data, ultimately forming a tree-like structure of decisions. DT utilizes several important hyperparameters, including:

- **criterion:** Defines the function that evaluates the quality of each split, guiding how effectively the data is partitioned.
- **splitter:** Specifies the strategy utilized to determine how the split is conducted at each node within the tree.
- **max_depth:** Restricts the greatest allowable depth of the DT, which can prevent overfitting.
- **min_samples_split:** Specifies the smallest sample count of an internal node at which a split can occur. at an internal node.
- **min_samples_leaf:** Specifies the smallest sample count allowed at an end (leaf) node, ensuring that each leaf node has sufficient data.
- **max_features:** Restricts the greatest allowable amount of features that can be taken into account when searching for the best split, which can enhance the model's performance and efficiency.
- **max_leaf_nodes:** Restricts the greatest number of leaf nodes that the decision tree can have, allowing for better control over the model's complexity.

Table 1 lists the options we tested for each hyperparameter in DT's grid search. The best hyperparameter option is emphasized in bold and italics.

Table 1: Optimal hyper-parameters for DT

DT Hyper-parameter	Tested Options
criterion	**gini**, entropy
splitter	**best**, random
max_depth	**None**, 10, 20, 30, 40, 50
min_samples_split	2, **5**, 10
min_samples_leaf	**1**, 2, 4
max_features	**None**, sqrt, log2
max_leaf_nodes	**None**, 10, 20, 30, 40, 50

4.3.2. K-nearest neighbors

KNN is a supervised ML classification algorithm whose underlying principle is that similar instances tend to be situated in close proximity to one another. This algorithm identifies the 'k' nearest data points to a specific input data point and classifies any input data according to the classes of the existing data points closest in proximity to the new input nearest neighbors). KNN utilizes several important hyperparameters, including:

- **n_neighbors:** Specifies the 'k' value, how many of the closest data points will be used for making the classification decision.
- **weights:** Specifies the function that determines how much influence each identified neighbor has in the prediction process, allowing for weighted voting among neighbors.
- **metric:** Specifies the function used to measure the distance between two points.

Table 2 lists the options we tested for each hyperparameter in KNN's grid search. The best hyperparameter option is emphasized in bold and italics.

Table 2: Optimal hyper-parameters for KNN

KNN Hyper-parameter	Tested Options
n_neighbors	3, 5, 7, 9, 11, 13, 15
weights	**uniform**, distance
metric	**euclidean**, manhattan, minkowski

4.3.3. Logistic regression

LR is both a statistical and ML algorithm primarily employed for binary classification tasks. It estimates the probability that a given input belongs to a specific class. Although it carries the name "regression," LR functions as a linear model similar to linear regression, but it is specifically designed for classification purposes rather than regression. LR utilizes several important hyperparameters, including:

- **C:** The regularization hyperparameter which helps to prevent overfitting by controlling the trade-off between fitting the training data and maintaining model simplicity.
- **penalty:** Defines the norm utilized for penalization, which influences how the model handles complexity and regularization.
- **solver:** Specifies the optimization algorithm used to find the best-fitting model parameters.

Table 3 lists the options we tested for each hyperparameter in LR's grid search. The best hyperparameter option is emphasized in bold and italics.

Table 3: Optimal hyper-parameters for LR

LR Hyper-parameter	Tested Options
C	0.1, **1**, 10, 100, 1000
penalty	**l1**, l2, elasticnet, none
solver	newton-cg, lbfgs, **liblinear**, sag, saga

4.3.4. Support vector machine

SVM is a supervised classification algorithm whose fundamental concept is to identify a hyperplane that optimally separates different classes of data while maximizing the margin between them. This approach helps in achieving better generalization on unseen data. SVM utilizes several important hyperparameters, including:

- **C:** This regularization hyperparameter which controls the trade-off between maximizing the margin between the classes and minimizing classification errors.
- **gamma:** The kernel coefficient, which specifies the amount of influence each training example extends and can significantly impact the decision boundary.
- **kernel:** Specifies the type of kernel function to be employed, which determines the feature space transformation and influences the model's performance and flexibility.

Table 4 lists the options we tested for each hyperparameter in SVM's grid search. The best hyper-parameter option is emphasized in bold and italics.

Table 4: Optimal hyper-parameters for SVM

SVM Hyper-parameter	Tested Options
C	**0.1**, 1, 10, 100, 1000
gamma	**scale**, auto, 0.001, 0.01, 0.1, 1
kernel	**linear**, poly, rbf, sigmoid

4.3.5. Random forest

RF is a supervised ensemble learning classification algorithm which constructs multiple decision trees during the training process and determines the final output by either taking the mode of the classes (in the case of classification) or calculating the mean prediction (for regression) from the individual trees. RF utilizes several important hyperparameters, including:

- **n_estimators:** Specifies the total number of DTs to be generated, which can influence the model's accuracy and robustness.
- **max_depth:** Restricts the greatest allowable depth of each DT.
- **min_samples_split:** Specifies the smallest sample count of an internal node at which a split can occur.
- **min_samples_leaf:** Specifies the smallest sample count allowed at an end node.

- **max_features:** Restricts the greatest allowable number of features that can be taken into account when searching for the best split.
- **bootstrap:** Specifies whether bootstrap samples are utilized during the generation of trees, influencing how the training data is sampled and impacting model variability.

Table 5 lists the options we tested for each hyperparameter in RF's grid search. The best hyperparameter option is emphasized in bold and italics.

Table 5: Optimal hyper-parameters for RF

RF Hyper-parameter	Tested Options
n_estimators	**50**, 100, 200
max_depth	**None**, 10, 20, 30, 40, 50
min_samples_split	2, **5**, 10
min_samples_leaf	**1**, 2, 4
max_features	**auto**, sqrt, log2
bootstrap	**True**, False

5. RESULTS

The efficacy of the framework was assessed for both phase one (segmentation) and phase three (classification) using the standard performance metrics of accuracy, precision, sensitivity (recall), specificity, and F1 score. These metrics were computed using confusion matrix elements: true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

For segmentation, the definitions of these terms are as follows: TP represents the count of LV wall pixels that were correctly identified as LV wall pixels; TN refers to the count of background (non-LV wall) pixels accurately identified as not belonging to the LV wall; FP denotes the background pixels mistakenly classified as LV wall pixels; and FN represents the LV wall pixels incorrectly categorized as background pixels.

For classification, MI is treated as the positive class, while non-MI is designated as the negative class. Accordingly, TP reflects the count MI-positive echocardiograms correctly diagnosed as having MI; TN signifies the count of MI-negative echocardiograms correctly recognized as non-MI; FP represents the non-MI echocardiograms that were incorrectly classified as MI; and FN indicates the MI echocardiograms that were misclassified as non-MI.

Accuracy was computed as the ratio of correct predictions to total predictions as shown in equation 3. Precision was computed as the ratio of correct positive predictions to total positive prediction as shown in equation 4. Sensitivity was computed as the ratio of correct positive predictions to actual positive instances as shown in equation 5. Specificity was computed as the ratio of correct negative predictions to actual negative instances as shown in equation 6. F1-Score was computed as is the weighted average of precision and sensitivity as shown in equation 7.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{3}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4}$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{6}$$

$$\text{F1 Score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{7}$$

Furthermore, an additional performance metric was evaluated for the segmentation task: Intersection over Union (IoU). IoU was computed as the ratio of the intersection of the predicted LV wall and ground truth LV wall's bounding boxes' areas to their union as shown in equation 8. This metric is particularly valuable in the evaluation of object detection, as it provides a measure of localization accuracy.

$$\text{IoU} = \frac{\text{target} \cap \text{prediction}}{\text{target} \cup \text{prediction}} \tag{8}$$

5.1. Segmentation model results

To evaluate the segmentation LadderNet's performance on the test data, the mean for each of aforementioned performance metrics on the predicted masks in comparison with their ground-truth masks was computed.

Metric	Performance result
IoU	97.31815%
Accuracy	99.78382%
Precision	97.49897%
Sensitivity	97.30910%
Specificity	99.89144%
F1 Score	97.40394%

The performance metric results for the LadderNet segmentation model demonstrate impressive results across all tested evaluation criteria as shown in table 6. The IoU score of 97.32% indicates a substantial overlap between the average predicted mask and its ground truth. This high score suggests that the model is effectively and accurately identifying the target LV wall region, demonstrating strong performance in the segmentation task. This in addition to the near-perfect accuracy of 99.78% show that almost all pixel predictions are correct. The precision of 97.50% reflects that the model makes few false positive pixel errors. The sensitivity of 97.31% indicates that the model is highly effective at capturing the majority of LV wall pixels within the frames. The specificity of 99.89% demonstrates the model's ability to accurately identify most background (non-LV wall) pixels. Moreover, the high F1 score of 97.40% reflects a strong balance between precision and sensitivity. Collectively, these results showcase the robustness of the proposed LadderNet in the task of LV wall segmentation.

5.2. Classification models results

After training each of the classifiers with its optimal hyper-parameters and running it on the test data, the training and testing confusion matrices for each of the models was plotted then the training and testing performance metrics of each model were computed. In the following section each classifier will be analyzed

by looking at its confusion matrices to compare its performance during training with its performance during testing. In the section after that, all classifiers' training and testing evaluation metrics will be compared to identify the optimal algorithm to pick for the echocardiogram framework.

5.2.1. Analysis of models' performance confusion matrices

To get a comprehensive view of the classification models' capabilities during training and during testing, the respective confusion matrices were evaluated. They provide a detailed review of each model's classifications, showing how many instances were correctly and incorrectly classified. Training confusion matrices provide an insight into how well each algorithm performed on the data it was trained on and testing confusion matrices illustrates each model's ability to generalize to new data. Comparing both matrices is essential for detecting overfitting (if the training confusion matrix has much better results than the testing confusion matrix) and underfitting (if both matrices show subpar results). Overfitting occurs when a model learns its training data too thoroughly, including its noise and outliers. This hinders the model's ability to generalize to new data. Conversely, underfitting indicates that the model was unable to learn from its training data well and failed to identify the underlying patterns in it.

Figure 12 shows the confusion matrices for the DT model on the motion features obtained from the predicted masks. DT demonstrates strong performance during training as evidenced by its very few false positives (1) and negatives (2). As expected, it's performance during testing drops. While the number of false positives (2) and negatives (2) may still be low, they form a higher ratio of overall classifications since the size of the testing data is much small than that of the training data. However, the drop is not severe and the overall performance is still good.

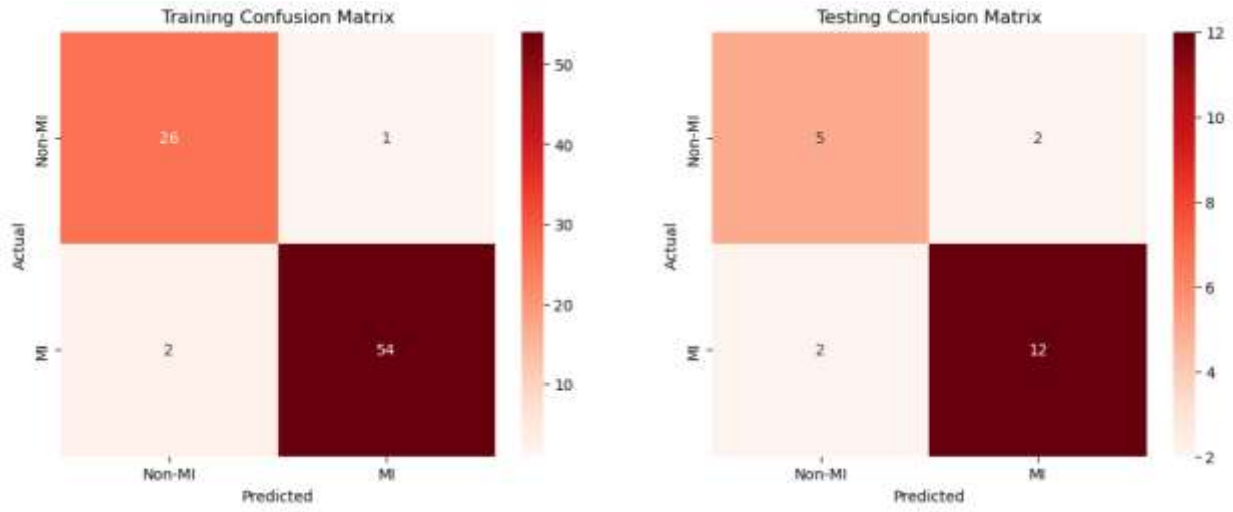

Figure 12: DT Training and Testing Confusion Matrices

Figure 13 shows the confusion matrices for the KNN model on the motion features obtained from the predicted masks. During training, KNN did not classify any false negatives meaning it has perfect sensitivity; however, it classified several false positives (9) meaning it has relatively low specificity. During testing, there was a significant rise in the ratio of false positives and negatives especially when considering the smaller test data size. Overall, KNN seemed to be much more adept at identifying true negatives (non-MI) than true positives (MI) and therefore, had more false classifications than DT.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 8s, 2025
https://www.theaspd.com/ijes.php

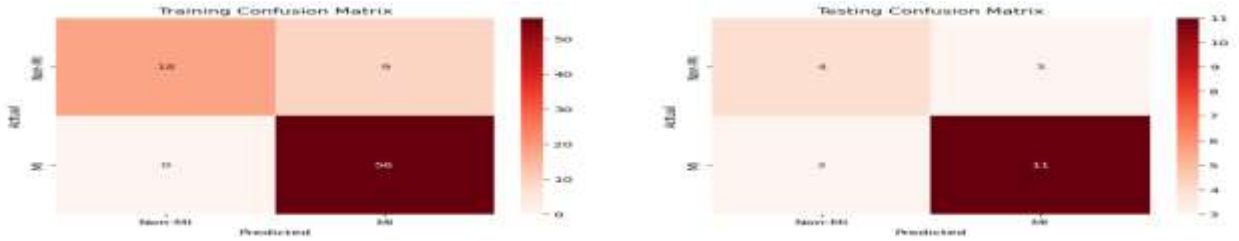

Figure 13: KNN Training and Testing Confusion Matrices

Figure 14 shows the confusion matrices for the LR model on the motion features obtained from the predicted masks. LR performed slightly poorer than KNN during training with an equal number of false positives but with marginally higher false negatives (2 instead of KNN's 0). Surprisingly, LR performed much better than KNN and DT on the test data with a lower ratio of false negatives and a much lower ratio of false positives. The reduction in LR's ratio of false negatives on testing in comparison to training is unusual and probably not reflective of LR's actual capability due to test data's small size.

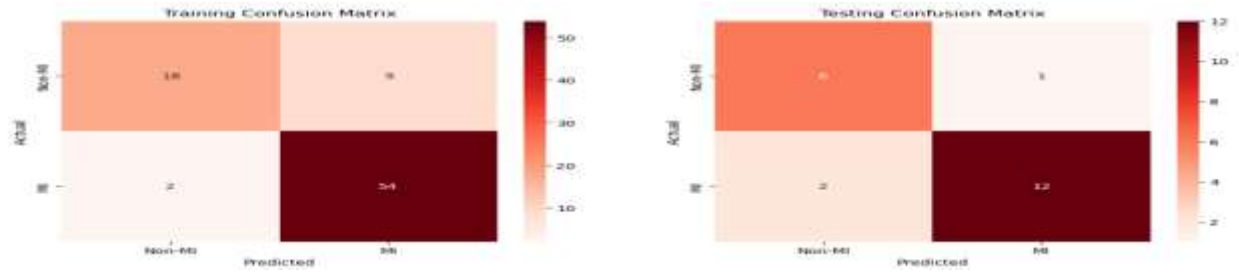

Figure 14: LR Training and Testing Confusion Matrices

Figure 15 shows the confusion matrices for the SVM model on the motion features obtained from the predicted masks. During training, SVM showed a middle-of-the-road performance with 2 less false negatives than KNN and LR but 5 more and 3 more false negatives than KNN and LR respectively. SVM had the greatest number of overall false classifications during training out of all the models. On the other hand, SVM performed very well during testing with an identical testing confusion matrix to that of LR. As aforementioned in the LR subsection, this is unusual and probably not reflective of SVM's actual capability.

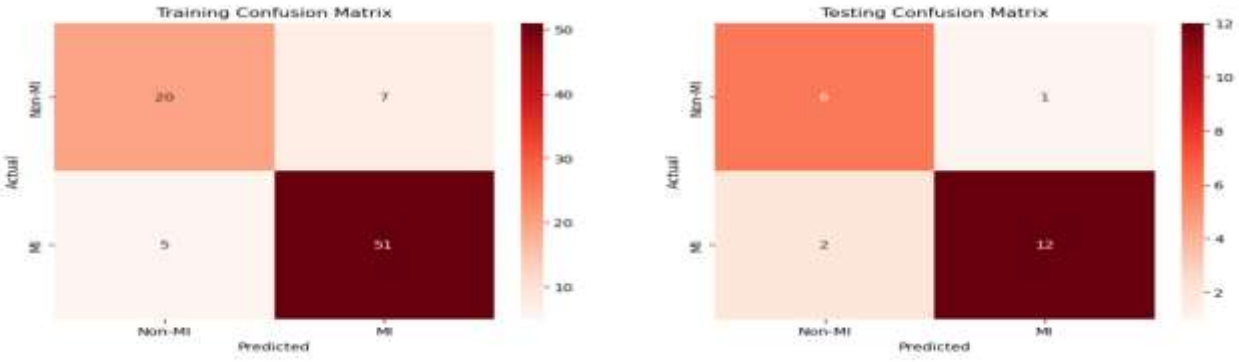

Figure 15: SVM Training and Testing Confusion Matrices

Figure 16 shows the confusion matrices for the RF model on the motion features obtained from the predicted masks. RF had a great performance during training, with only 1 false positive and 1 false negative, outperforming DT. Furthermore, during training, RF had less of a performance drop than DT, with 1 less false negative. RF's testing confusion matrix is has 1 higher false positive and 1 lower false negative than LR and SVM, meaning an overall equal number of false classifications (3) as them; however, unlike LR and SVM, RF's testing results are probably much more reflective of RF's capability considering its unmatched training results.

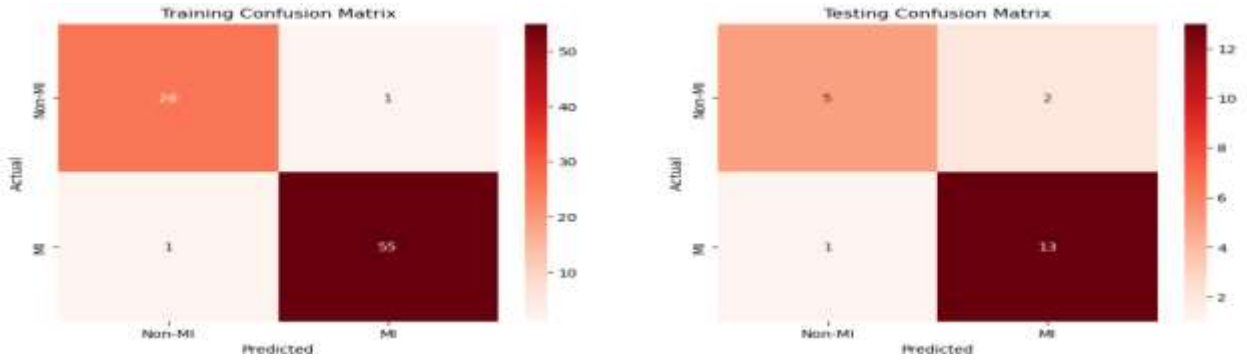

Figure 16: RF Training and Testing Confusion Matrices

5.2.2. Models' performance evaluation metrics

The performance evaluation metric results for each of the classification models are listed in Table 7 for the training results and Table 8 for the testing results. The top performance result for each category (column) is emphasized in bold and italics.

Table 6: MI classification training performance metric results

Classifier	Accuracy	Precision	Sensitivity	Specificity	F1 Score
DT	96.39%	98.18%	96.43%	**96.30%**	97.30%
KNN	89.16%	86.15%	**100.00%**	66.67%	92.56%
LR	86.75%	85.71%	96.43%	66.67%	90.76%
SVM	85.54%	87.93%	91.07%	74.07%	89.47%
RF	**97.59%**	**98.21%**	98.21%	**96.30%**	**98.21%**

Table 7: MI classification testing performance metric results

Classifier	Accuracy	Precision	Sensitivity	Specificity	F1 Score
DT	80.95%	85.71%	85.71%	71.43%	85.71%
KNN	71.43%	78.57%	78.57%	57.14%	78.57%
LR	**85.71%**	**92.31%**	85.71%	**85.71%**	88.89%
SVM	**85.71%**	**92.31%**	85.71%	**85.71%**	88.89%
RF	**85.71%**	86.67%	**92.86%**	71.43%	**89.66%**

As per the training results shown in table 7, RF achieved the highest results in all metrics except sensitivity where it placed second, slightly behind KNN. DT also performed very well, second in overall performance only to RF. DT matched RF in specificity and scored just slightly less than it in other categories. KNN had a good performance that was better than LR and SVM but not as good as RF and DT; however, KNN's perfect sensitivity score outdid all over models' sensitivity results. LR and SVM both performed the worst; however, their metrics were still satisfactory. All in all, each of the five models achieved over 85% accuracy and precision as well as over 91% sensitivity and over 89% F1 score. It appears that specificity is the metric where all models had their lowest performance especially KNN and LR whose specificity of 66.67% is much lower than their other metrics, indicating that they had a high false positive rate which means they were classifying many non-MI cases as MI. RF and DT's worse performing metric was also specificity at 96.30%; however, it is only very slightly lower than their other metrics. As per the testing results shown in table 8, RF had the least overall drop in performance and achieved the best testing accuracy, sensitivity, and F1 score, demonstrating its reliability in generalizing well to new data. On the other hand, KNN appears to have the biggest drop in performance and achieved the lowest metric results in all categories. DT had solid results, matching RF again in its specificity score; and even though it had a higher drop in performance than RF, it still had good metrics overall. LR and SVM had identical results across all categories which was unsurprising given their aforementioned identical testing confusion matrices. They tied with RF in terms of accuracy and outperformed it in precision and specificity. However, the reliability of LR and SVM's results is uncertain and could be skewed by the small size of the test data considering their testing precision and specificity are significantly higher than their training counterparts. We chose the Random Forest model for MI classification as it provided top and consistent results while in training and testing. It demonstrated its effectiveness in avoiding both false positives and especially false negatives. Its testing sensitivity of 92.86% ensures that it effectively identifies MI-positive cases, which is crucial in medical diagnosis scenarios. That combined with its consistent high performance across the other metrics and its robust generalization capabilities make Random Forest the most reliable model for this task.

6. CONCLUSION

To conclude, this work explored the use of a LadderNet CNN for segmenting the LV wall out of echocardiogram frames as well as various ML algorithms for diagnosing MI from the motion of the LV wall's sections. After comprehensive investigation and experimentation, the efficacy and reliability of LadderNet's performance in accurately segmentation the LV wall out of low-quality echocardiograms were established. RF's ability to classify MI to a high degree of accuracy, precision, sensitivity, and specificity from the motion of the LV wall's sections was also demonstrated after comparing its performance with various other ML models. However, the limitations of these investigations and experiments need to be acknowledged. The LadderNet was both trained and tested on only one dataset; its performance results have not been explored on echocardiograms obtained from other sources. Furthermore, while the RF model provided a very high sensitivity score of 92.86%, its specificity score was lower at 71.43%. More work needs to be done to raise the specificity score to ensure that the ratio of correct MI-negative classification is as high as the ratio of correct MI-positive classifications. This could possibly be done by extracting more features in the feature engineering stage in addition to the maximum displacement for each section, e.g., the overlap of section areas across the frames. In summary, this paper has actualized its intended purpose of contributing to the advancement of early MI diagnoses by using the help artificial intelligence. It purposed a full framework that could detect motion abnormalities, and hence MI, in the LV by analyzing A4C echocardiograms. Its findings validate the results of previous, similar studies and hopefully offer a valuable insight for future similar work.

DECLARATIONS

- Funding: No funding was received to assist with the preparation of this manuscript.
- Conflict of interest: The authors declare no competing interests.
- Ethics approval: Not applicable.
- Data availability: The dataset utilized in this paper is available from the corresponding author upon request.

REFERENCES

1. O. J. Mechanic, M. Gavin, and S. A. Grossman. Acute myocardial infarction. In StatPearls. StatPearls Publishing, Treasure Island (FL), 2024.
2. N. Ojha and A. S. Dhamoon. Myocardial infarction. In StatPearls. StatPearls Publishing, Treasure Island (FL), 2024.
3. J. Park, K. H. Choi, J. M. Lee, H. K. Kim, D. Hwang, T. Rhee, J. Kim, T. K. Park, J. H. Yang, Y. B. Song, J. Choi, J. Hahn, S. Choi, B. Koo, S. C. Chae, M. C. Cho, C. J. Kim, J. H. Kim, M. H. Jeong, H. Gwon, H. Kim, and KAMIR-NIH (Korea Acute Myocardial Infarction Registry–National Institutes of Health) Investigators. Prognostic implications of door-to-balloon time and onset-to-door time on mortality in patients with ST-segment-elevation myocardial infarction treated with primary percutaneous coronary intervention. Journal of the American Heart Association, 8(9):e012188, 2019.
4. L. Martin, M. Murphy, A. Scanlon, C. Naismith, D. Clark, and O. Farouque. Timely treatment for acute myocardial infarction and health outcomes: An integrative review of the literature. Australian Critical Care, 27(3):111–118, 2014.
5. H. Feigenbaum. Clinical applications of echocardiography. Progress in Cardiovascular Diseases, 14(6):531–558, 1972.
6. K. Thygesen, J. S. Alpert, A. S. Jaffe, M. L. Simoons, B. R. Chaitman, and H. D. White. Third universal definition of myocardial infarction. Circulation, 126(16):2020–2035, 2012.
7. R. M. Lang, L. P. Badano, V. Mor-Avi, J. Afilalo, A. Armstrong, L. Ernande, F. A. Flachskampf, E. Foster, S. A. Goldstein, T. Kuznetsova, P. Lancellotti, D. Muraru, M. H. Picard, E. R. Rietzschel, L. Rudski, K. T. Spencer, W. Tsang, and J.-U. Voigt. Recommendations for Cardiac Chamber Quantification by Echocardiography in Adults: An Update from the American Society of Echocardiography and the European Association of Cardiovascular Imaging. European Heart Journal – Cardiovascular Imaging, 16(3):233–271, 2015.
8. M. Esmaeilzadeh, M. Parsaee, and M. Maleki. The role of echocardiography in coronary artery disease and acute myocardial infarction. J. Tehran Heart Cent., 8(1):1–13, 2013.
9. S. A. Mollema, G. Nucifora, and J. J. Bax. Prognostic value of echocardiography after acute myocardial infarction. Heart, 95(21):1732–1745, 2009.
10. M. Prastaro, E. Pirozzi, N. Gaibazzi, S. Paolillo, C. Santoro, G. Savarese, M. A. Losi, G. Esposito, P. Perrone Filardi, B. Trimarco, and M. Galderisi. Expert review on the prognostic role of echocardiography after acute myocardial infarction. Journal of the American Society of Echocardiography, 30(5):431–443.e2, 2017.
11. A. Madani, R. Arnaout, M. Mofrad, and R. Arnaout. Fast and accurate classification of echocardiograms using deep learning. npj Digital Medicine, 1, 2017.
12. Z. Akkus, Y. Aly, I. Attia, F. Lopez-Jimenez, A. Arruda-Olson, P. Pellikka, S. Pislaru, G. Kane, P. Friedman, and J. Oh. Artificial intelligence (ai)-empowered echocardiography interpretation: A state-of-the-art review. Journal of Clinical Medicine, 10:1391, 2021
13. C. Frederiksen, P. Juhl-Olsen, J. Hermansen, N. Andersen, and E. Sloth. Clinical utility of semi-automated estimation of ejection fraction at the point-of-care. Heart, Lung and Vessels., 7:208–216, 2015.
14. S. A. Coulter and K. Campos. Artificial intelligence in echocardiography. Tex. Heart Inst. J., 49(2), 2022.
15. T. Barry, J. M. Farina, C.-J. Chao, C. Ayoub, J. Jeong, B. N. Patel, I. Banerjee, and R. Arsanjani. The role of artificial intelligence in echocardiography. Journal of Imaging, 9(2), 2023.
16. J. Zhou, M. Du, S. Chang, and Z. Chen. Artificial intelligence in echocardiography: detection, functional evaluation, and disease diagnosis. Cardiovascular Ultrasound, 19(1):29, 2021.
17. V. Chalana, D. R. Haynor, and Y. Kim. Left-ventricular boundary detection from short-axis echocardiograms: the use of active contour models. In M. H. Loew, editor, Medical Imaging 1994: Image Processing, volume 2167, pages 786 – 798. International Society for Optics and Photonics, SPIE, 1994.
18. V. Chalana, D. Linker, D. Haynor, and Y. Kim. A multiple active contour model for cardiac boundary detection on echocardiographic sequences. IEEE Transactions on Medical Imaging, 15(3):290–298, 1996.
19. A. Mishra, P. Dutta, and M. Ghosh. A ga based approach for boundary detection of left ventricle with echocardiographic image sequences. Image and Vision Computing, 21(11):967–976, 2003.
20. M. Landgren, N. Overgaard, and A. Heyden. Segmentation of the left heart ventricle in ultrasound images using a region based snake. Medical Imaging 2013: Image Processing, volume 8669, page 866945. International Society for Optics and Photonics, SPIE, 2013.

21. N. Lin, W. Yu, and J. S. Duncan. Combinative multi-scale level set framework for echocardiographic image segmentation. Medical Image Analysis, 7(4):529–537, 2003.
22. T. Nguyen, P. Nguyen, D. Tran, H. Pham, Q. Nguyen, T. Le, H. Van, B. Do, P. Tran, V. Le, T. Nguyen, L. Tran, and H. Pham. Ensemble learning of myocardial displacements for myocardial infarction detection in echocardiography. Frontiers in Cardiovascular Medicine, 10, 2023.
23. A. Degerli, M. Zabihi, S. Kiranyaz, T. Hamid, R. Mazhar, R. Hamila, and M. Gabbouj. Early detection of myocardial infarction in low-quality echocardiography. IEEE Access, 9:34442–34453, 2021.
24. Y. Guo, G.-Q. Du, W.-Q. Shen, C. Du, P.-N. He, and S. Siuly. Automatic myocardial infarction detection in contrast echocardiography based on polar residual network. Computer Methods and Programs in Biomedicine, 198:105791, 2021.
25. O. Hamila, S. Ramanna, C. J. Henry, S. Kiranyaz, R. Hamila, R. Mazhar, and T. Hamid. Fully automated 2d and 3d convolutional neural networks pipeline for video segmentation and myocardial infarction detection in echocardiography. Multimedia Tools and Applications, 81(26):37417–37439, 2022.
26. Y. Chen, X. Zhang, C. Haggerty, and J. Stough. Fully automated multi-heartbeat echocardiography video segmentation and motion tracking. In Medical Imaging 2022: Image Processing, page 29, 2022.
27. S. V. Wifstad, H. A. Kildahl, B. Grenne, E. Holte, S. W. Hauge, S. Sæbø, D. Mekonnen, B. Nega, R. Haaverstad, M.-E. Estensen, H. Dalen, and L. Lovstakken. Mitral valve segmentation and tracking from transthoracic echocardiography using deep learning. Ultrasound in Medicine & Biology, 50(5):661–670, 2024.
28. X. Zhu, Y. Wei, Y. Lu, M. Zhao, K. Yang, S. Wu, H. Zhang, and K. K. Wong. Comparative analysis of active contour and convolutional neural network in rapid left-ventricle volume quantification using echocardiographic imaging. Computer Methods and Programs in Biomedicine, 199:105914, 2021.
29. Y. Xing, L. Zhong, and X. Zhong. An encoder-decoder network based fcn architecture for semantic segmentation. Wireless Communications and Mobile Computing, 2020(1):8861886, 2020.
30. A. Norelyaqine, R. Azmi, and A. Saadane. Architecture of deep convolutional encoder-decoder networks for building footprint semantic segmentation. Scientific Programming, 2023(1):8552624, 2023.
31. J. Yang, X. Ding, Z. Zheng, X. Xu, and X. Li. Graphecho: Graph-driven unsupervised domain adaptation for echocardiogram video segmentation, 2023 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 11844-11853, 2023.
32. L. Corinzia, F. Laumer, A. Candreva, M. Taramasso, F. Maisano, and J. M. Buhmann. Neural collaborative filtering for unsupervised mitral valve segmentation in echocardiography. Artificial Intelligence in Medicine, 110:101975, 2020.
33. H. Wu, J. Lin, W. Xie, and J. Qin. Super-efficient echocardiography video segmentation via proxy- and kernel-based semi-supervised learning. Proceedings of the AAAI Conference on Artificial Intelligence, 37(3):2803–2811, 2023.
34. J. Bosch, S. Mitchell, B. Lelieveldt, F. Nijland, O. Kamp, M. Sonka, and J. Reiber. Automatic segmentation of echocardiographic sequences by active appearance motion models. IEEE Transactions on Medical Imaging, 21(11):1374–1383, 2002.
35. A. Degerli, S. Kiranyaz, T. Hamid, R. Mazhar, and M. Gabbouj. Early myocardial infarction detection over multi-view echocardiography. Biomedical Signal Processing and Control, 87:105448, 2024.
36. J. Zhuang. Laddernet: Multi-path networks based on U-Net for medical image segmentation, 2019.
37. O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, pages 234–241, Cham, 2015. Springer International Publishing.