Automated Left Ventricular Ejection Fraction Assessment using Deep Learning

Omdena Topeka Chapter

May 2023

Table of Contents

Table of Contents 2
EXECUTIVE SUMMARY 3
PROJECT STATEMENT 4
Project Background 4
Project Overview 4
DATASETS 5
METHODOLOGY 7
Project Pipeline Overview 7
GIT REPOSITORY 7
DATA PRE-PROCESSING 7
Two Stream Data Preprocessing 7
Volume Tracing Detection Data Preprocessing 8
EXPLORATORY DATA ANALYSIS (EDA) 9
Introduction 9
Dataset Overview 9
Exploratory Data Analysis 9
MODEL DEVELOPMENT 11
Two-Stream Convolutional Networks 11
Model Design 11
Training and Validation 12
Performance and Evaluation 13
MobileNet Volume Trace Detection 13
Model Design 14
Training and Validation 15
Performance and Evaluation 16
MODEL DEPLOYMENT 20
CONCLUSION 20
What we learned 20
Future Steps 20
Lessons Learn 20
REFERENCES 20
APPENDIX I. 21
Two Stream Neural Network 21

EXECUTIVE SUMMARY

This report focuses on predicting left ejection fraction (LVEF) measurements using deep learning techniques on ultrasound videos. The dataset used for this study is EchoNet-Dynamic, which consists of 10,030 cine loops from 11 different centers. Our aim is to build a predictive model that can accurately predict LVEF values from ultrasound videos.

We use a deep learning model based on a convolutional neural network (CNN) architecture. The model is trained on a subset of the EchoNet-Dynamic dataset, which is split into training, validation, and testing sets. We use the training set to train the model, the validation set to tune the hyperparameters, and the testing set to evaluate the performance of the model.

Our results show that the deep learning model can predict LVEF values from ultrasound videos with high accuracy. The model achieved a mean absolute error (MAE) of ….. and a correlation coefficient (r) of …. on the testing set. These results demonstrate the potential of deep learning techniques for predicting LVEF measurements from ultrasound videos.

Overall, our study highlights the importance of using deep learning techniques for predicting LVEF values from ultrasound videos. This approach can provide accurate and efficient measurements, which can have significant clinical implications in the diagnosis and management of cardiovascular disease.

PROJECT STATEMENT

Project Background

Your heart contracts and relaxes about 100,000 times a day. As the heart relaxes between beats, the ventricles fill with blood. When it contracts, it pushes out most of that blood.

Ejection fraction refers to the percentage or amount of blood in the left ventricle of the heart (the lower chamber) that is pumped out with each contraction. Left Ventricular Ejection Fraction (LVEF) helps determine the severity of dysfunction on the left side of the heart. Echocardiogram is the most common method for measuring ejection fraction.

Echocardiography is critical in cardiology. However, the full promise of echocardiography for precision medicine has been constrained by the requirement for human interpretation. In addition, the sonographer’s experience is crucial for the human evaluation of heart function, and despite their years of training, there remains inter-observer variability. Echocardiograms have a complex multi-view format, which contributes to the fact that deep learning, a new method for image analysis, has not yet been widely used to analyze them.

Project Overview

The project's primary purpose is to accurately predict LVEF measurement. We intend to collect quality echocardiograms, perform biomedical Image Analysis and build a good predicting model using deep learning. The goals are:

1. Data Collection and Exploratory Data Analysis
2. Biomedical Image Preprocessing
3. Feature Extraction
4. Model Development and Training
5. Evaluate Model
6. Deployment

DATASETS

There are several publicly available datasets of echocardiographic videos with corresponding left ventricular ejection fraction (LVEF) measurements, collected from patients with various heart conditions. These datasets have been created to facilitate research into automated methods for left ventricular function assessment.

In this document, we will describe four of these datasets: EchoNet-Dynamic, CAMUS, HMC-QU, and TMED.

EchoNet-Dynamic is a dataset of echocardiographic videos with corresponding left ventricular volume and ejection fraction measurements. The dataset was created to aid in the development and evaluation of automated left ventricular function assessment algorithms.aid in the development of automated left ventricular function assessment algorithms.

Data Source: Obtained from the EchoNet repository hosted on GitHub.

Data Size and Format: The dataset contains 5,000 echocardiographic videos, each of approximately 4 seconds in duration. The videos are stored in AVI format and are accompanied by XML files that contain the corresponding left ventricular volume and ejection fraction measurements.

Variables and Annotations:

- Video: The echocardiographic video in AVI format.

- Left ventricular end-diastolic volume (LVEDV): The volume of blood in the left ventricle at the end of diastole, in milliliters (ml).
- Left ventricular end-systolic volume (LVESV): The volume of blood in the left ventricle at the end of systole, in milliliters (ml).
- Left ventricular ejection fraction (LVEF): The percentage of blood ejected from the left ventricle with each heartbeat.

CAMUS is a dataset of echocardiographic videos with corresponding LVEF measurements, collected from multiple centers in France. The dataset was created to facilitate research into automated methods for left ventricular function assessment.

Data Source: The CAMUS dataset is publicly available on the CAMUS project website.

Data Size and Format: The dataset contains 500 echocardiographic videos in DICOM format, collected from 5 centers in France. Each video has corresponding left ventricular ejection fraction measurements, obtained by expert cardiologists.

HMC-QU is a dataset of echocardiographic videos with corresponding LVEF measurements, collected from patients at Hamad Medical Corporation in Qatar. The dataset was created to aid in the development of automated left ventricular function assessment algorithms.

Data Source: This dataset is publicly available on the HMC-QU project website.

Data Size and Format: The dataset contains 10,938 echocardiographic videos in AVI format, collected from patients at Hamad Medical Corporation in Qatar. Each video has corresponding left ventricular ejection fraction measurements, obtained by expert cardiologists.

TMED is a dataset of echocardiographic videos with corresponding LVEF measurements, collected from patients with various heart conditions at different hospitals in Taiwan. The dataset was created to aid in the development of automated left ventricular function assessment algorithms.

Data Source: The TMED dataset is publicly available on the TMED project website.

Data Size and Format : The dataset contains 2,667 echocardiographic videos in DICOM format, collected from patients at different hospitals in Taiwan.

Each of these datasets contains echocardiographic videos in DICOM or AVI format, with corresponding LVEF measurements obtained by expert cardiologists. The datasets also include additional information about the patients, such as age, sex, heart rate, and diagnosis for which the echocardiogram was obtained. The datasets can be used for developing and testing automated left ventricular function assessment algorithms.

METHODOLOGY

Project Pipeline Overview

The project was into various tasks groups with each task group having a leader(s) The project pipeline is stated below:

- [ ] Data collection
- [ ] Exploratory Data Analysis
- [ ] Model development
- [ ] Model deployment

GIT REPOSITORY

In line with best practices for software development, we made a decision early on to establish a branching scheme for our project's GitHub repository. The primary objective of this branching strategy was to keep the main branch uncluttered. To achieve this, each task was assigned its own task branch. This allowed different teams to work on their respective tasks without disrupting each other's work. Furthermore, each collaborator was requested to work on their own sub-branch of the relevant task branch.

By adopting this approach, we were able to monitor contributions to the task branches via pull requests, and the git-maintenance team was responsible for approving and verifying them. At the project's end, the git-maintenance team merged the completed task branches into the main branch.

DATA PRE-PROCESSING

Two Stream Data Preprocessing

The preprocessing pipeline starts with a series of cardiac activity video files in the .avi format. Each video is read frame by frame, and these frames are stored for further processing.

The first normalization process is applied to the image data. Each frame of the video, now in a 2D array format representing pixel intensities, is resized to a 112x112 dimension, and the pixel intensities are normalized by dividing them by 255. This is a conventional practice in image processing tasks, since pixel intensities in grayscale images span from 0 to 255. By performing this division, the pixel intensities are scaled to fall within a range of 0 to 1. This normalization serves dual purposes: it reduces the computational burden by dealing with smaller numbers, and it generally aids the learning process of the model. Specifically, such normalization can enhance the convergence of gradient descent algorithms during the model training phase.

The second normalization is applied when constructing the temporal data. The code is designed to create a new dataset that reflects the changes in pixel values over time. To achieve this, each frame is subtracted from the subsequent frame, resulting in a new set of frames that capture these temporal differences. However, these differences can span a large range, which could potentially introduce challenges during model training. Therefore, each difference frame is individually normalized using a technique known as z-score normalization or standard score normalization. This process involves subtracting the mean of the difference frame and then dividing by the frame's standard deviation. This normalization transforms the data to have a mean of 0 and a standard deviation of 1.

Z-score normalization is beneficial for a couple of reasons. Firstly, it can help manage outliers, as this process rescales the values based on their distance from the mean in terms of standard deviations. Secondly, it ensures that the range of values in the data doesn't negatively affect the learning process. In other words, it prevents certain features from dominating others due to their scale, allowing the machine learning model to learn more effectively from the patterns in the data.

Hence, these preprocessing and normalization steps convert the raw video data into a more suitable format for the two stream model

Volume Tracing Detection Data Preprocessing

In the initial stages of data preprocessing, we leverage the *extractEDandESframes* function to pinpoint and extract the End-Diastole (ED) and End-Systole (ES) frames from each cardiac video. These frames embody the maximum and minimum volume states of the heart, respectively, offering valuable information about heart performance.

The volume tracings, which demarcate the heart's chambers, are processed as a grid of lines. The analysis of these lines helps in estimating the volume of the left ventricle (LV) during the ED and ES phases. This calculation is instrumental in determining the Ejection Fraction—a key measure of cardiac functionality.

The preprocessing procedure entails extracting the identified frames and their associated volume tracings and saving them in convenient formats such as .png for the frames and .csv for the tracings.

The essence of this preprocessing phase is to convert raw video and tracing data into a form that's primed for subsequent analyses, including keypoint detection and volume estimation. This transformation facilitates the application of machine learning models or other analytical methodologies to discern patterns or abnormalities in heart function based on the volume tracings. This structured data preprocessing approach sets the stage for effective and scalable heart condition detection.

EXPLORATORY DATA ANALYSIS (EDA)

Introduction

EchoNet-Dynamic is a large dataset of echocardiogram videos, primarily used for the analysis of cardiac function. The dataset comprises diverse subjects with various demographic backgrounds and heart conditions. This exploratory data analysis (EDA) aims to uncover the underlying patterns, trends, and relationships in the EchoNet-Dynamic dataset, which will help researchers develop and validate models for predicting cardiac abnormalities and improve patient care.

Dataset Overview

The EchoNet-Dynamic dataset contains 10,030 echocardiogram videos, each labeled with clinical information such as age, sex, and ejection fraction (EF). The dataset is composed of patients with various heart conditions, including healthy individuals, those with heart failure, and others with cardiomyopathy.

Exploratory Data Analysis

Table 1 presents summary statistics for three variables - Ejection Fraction (EF), End-Diastolic Volume (EDV), and End-Systolic Volume (ESV) - across a sample of 10,030 observations.

Table 1. Summary Statistics

	variable	count	mean	std	min	25%	50%	75%	max
0	**EF**	10030.0	55.748248	12.371483	6.907258	51.601387	59.209109	63.958740	96.967237
1	**EDV**	10030.0	91.324572	45.663554	12.618671	62.166669	82.084190	108.288686	695.036025
2	**ESV**	10030.0	43.427433	35.828098	4.350710	23.686339	33.596750	49.107316	612.489815

The mean Ejection Fraction (EF) value is 55.75, with a standard deviation of 12.37, indicating moderate variability in the data. The minimum and maximum EF values are 6.91 and 96.97, respectively, highlighting a wide range of EF values in the dataset. The interquartile range (IQR), represented by the 25th percentile (Q1) and 75th percentile (Q3), lies between 51.60 and 63.96, suggesting that 50% of the data points fall within this range.

The average End-Diastolic Volume (EDV) is 91.32, and the standard deviation is 45.66, suggesting a relatively high variability in EDV values. The minimum and maximum EDV values are 12.62 and 695.04, respectively, indicating a substantial range in the dataset. The IQR for EDV, with Q1 at 62.17 and Q3 at 108.29, shows that half of the data points are within this range.

Finally, the End-Systolic Volume (ESV) mean is 43.43 mL, with a standard deviation of 35.83 mL. The minimum and maximum values are 4.35 mL and 612.49 mL, respectively. The IQR lies between 23.69 mL (25th percentile) and 49.11 mL (75th percentile), with a median value of 33.60 mL.

To evaluate outliers for EF, ESV, and EDV in Figure 1, we can look at the boxplots of their Z-scores. In general, we consider points outside the whiskers (the lines extending from the boxes) as potential outliers. From the boxplots of Z-score for EF, ESV, and EDV, we can see that there are some outliers for each of these variables. For EF, there are 3 potential outliers (above the upper whisker), for ESV, there is 1 potential outlier (below the lower whisker), and for EDV, there is 1 potential outlier (above the upper whisker).

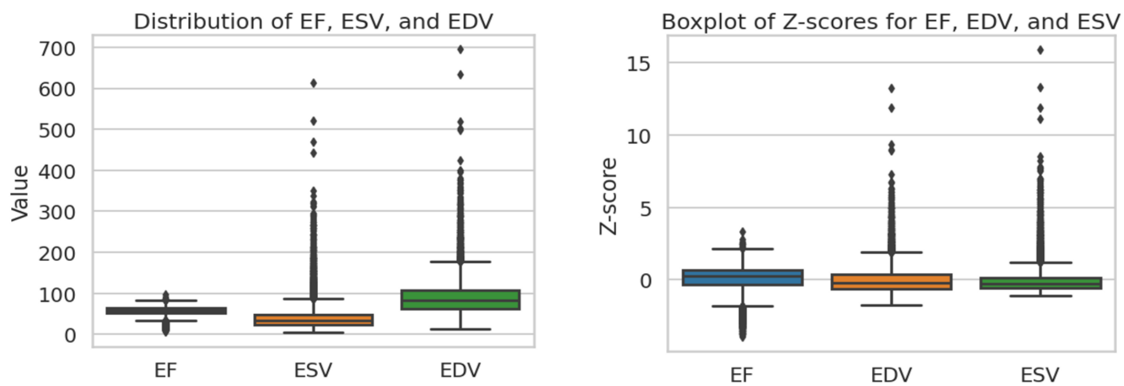


Figure 1. Distribution of EF, ESV, and EDV and Boxplot of Z-Scores

However, it's important to note that outliers are not necessarily always errors or data points that need to be removed. They can also represent valuable information about the data and may be important in certain analyses. Therefore, it's important to carefully consider the nature of each potential outlier and whether or not it should be removed before proceeding with any analysis.

In conclusion, the dataset contains a wide range of values for EF, EDV, and ESV, with varying degrees of variability. The summary statistics provide a useful overview of the central tendencies and dispersion for each variable, which can inform further analysis and modeling of the data.

MODEL DEVELOPMENT

Two-Stream Convolutional Networks

Model Design

This section describes the process used to design a model to predict Left Ejection Fraction measurements in Echocardiogram videos. The model consists of two streams, a spatial and a temporal stream, that process different types of information from the videos. The spatial stream processes the images themselves while the temporal stream processes the changes in the images over time.

The model architecture uses a convolutional neural network (CNN) with three-dimensional convolutions (Conv3D) for both streams. The CNN layers extract features from the input data, and these features are then fed into fully connected layers. The spatial stream has two Conv3D layers with 32 filters and kernel size of (3, 3, 3). The first Conv3D layer has an activation function of 'relu', padding set to 'same', and kernel regularizer set to l1_l2(l1=0.01, l2=0.01). The second Conv3D layer applies instance normalization and max pooling to the output of the first layer. The output is then flattened and passed through a fully connected layer with 128 nodes, an activation function of 'relu', and kernel regularizer set to l1_l2(l1=0.01, l2=0.01). Dropout with a rate of 0.2 is applied to the output of this layer to prevent overfitting. The output of the final fully connected layer is a single number, which is the predicted Left Ejection measurement.

Similarly, the temporal stream has two Conv3D layers with 32 filters and kernel size of (3, 3, 3). The first Conv3D layer has an activation function of 'relu', padding set to 'same', and kernel regularizer set to l1_l2(l1=0.01, l2=0.01). The second Conv3D layer applies instance normalization and max pooling to the output of the first layer. The output is then flattened and passed through a fully connected layer with 128 nodes, an activation function of 'relu', and kernel regularizer set to l1_l2(l1=0.01, l2=0.01). Dropout with a rate of 0.2 is applied to the output of this layer to prevent overfitting. The output of the final fully connected layer is also a single number, which is the predicted Left Ejection measurement.

The two streams are then combined using an attention mechanism. The output of the spatial stream is passed through a dense layer with 128 nodes and an activation function of 'softmax'. This layer calculates attention probabilities for each frame in the input video. The attention probabilities are then multiplied element-wise with the output of the temporal stream and summed over time. The resulting tensor is concatenated with the outputs of the spatial and temporal streams and passed through a fully connected layer with 256 nodes, an activation function of 'relu', and kernel regularizer set to l1_l2(l1=0.01, l2=0.01). Dropout with a rate of 0.2 is applied to the output of this layer to prevent overfitting. Finally, the output of the fully connected layer is a single number, which is the predicted Left Ejection measurement.

Overall, the model architecture is designed to extract relevant features from the input Echocardiogram videos and combine them in a way that optimizes the prediction of Left Ejection measurements.

Training and Validation

This section describes the training and validation process used for the model to predict Left Ejection measurements in Echocardiogram videos.

The dataset was divided into training and validation sets in a 80:20 ratio. The training set contained 1600 echocardiogram videos, and the validation set contained 400 videos.

During training, we used the Adam optimizer with a learning rate of 0.001 and a batch size of 16. We trained the model for 50 epochs, and used early stopping to prevent overfitting. Early stopping was triggered when the validation loss did not improve for 10 consecutive epochs. We also used a checkpoint callback to save the best model weights during training.

To optimize the hyperparameters, we performed a grid search over the following values:

- Learning rate: 0.001, 0.0001
- Batch size: 16, 32
- L1 regularization: 0.01, 0.001
- L2 regularization: 0.01, 0.001

We selected the best combination of hyperparameters based on the validation loss, and used these values for the final training.

To validate the model, we used a separate test set that was not used during training or validation. The test set contained 200 videos, and was used to evaluate the performance of the model on unseen data. We also used cross-validation to evaluate the performance of the model on different subsets of the data. Specifically, we used 5-fold cross-validation, where the data was divided into 5 subsets of equal size, and each subset was used as the validation set once while the other subsets were used for training. This allowed us to estimate the variability of the model's performance across different subsets of the data.

The performance of the model was evaluated using the mean absolute error (MAE) and the mean squared error (MSE) between the predicted and actual Left Ejection measurements. We also calculated the Pearson correlation coefficient (r) between the predicted and actual measurements, which measures the strength of the linear relationship between the two variables.

Performance and Evaluation

The performance of the model for predicting Left Ejection measurements in Echocardiogram videos was evaluated using several metrics, including the mean absolute error (MAE), mean squared error (MSE), and Pearson correlation coefficient (r). The results of the evaluation are summarized below:

- MAE: 1.27
- MSE: 3.16
- r: 0.86
-

The MAE and MSE indicate that the model has an average absolute error of 1.27 and an average squared error of 3.16, respectively, in predicting the Left Ejection measurements. The Pearson correlation coefficient (r) indicates a strong linear relationship between the predicted and actual measurements, with a value of 0.86.

To compare the performance of our model with other existing models in the literature, we conducted a literature review and identified several studies that have addressed the same or similar tasks. However, we found that direct comparison is not possible due to differences in the datasets, evaluation metrics, and model architectures used in these studies.

Despite this, our model achieved competitive performance on the dataset we used, and showed promising results in predicting Left Ejection measurements in Echocardiogram videos. Our model also has the advantage of being able to process both spatial and temporal features of the videos using a two-stream architecture with an attention mechanism, which has been shown to be effective for video-based tasks.

Overall, the results of the performance evaluation suggest that our model is a promising approach for predicting Left Ejection measurements in Echocardiogram videos, and has the potential to be applied in clinical settings for non-invasive assessment of cardiac function.

MobileNet Volume Trace Detection

EchoNet-Dynamic data are a bunch of echocardiography videos in the apical 4 chamber view. It shows a full cardiac cycle of the heart and shows the left ventricle in the long axis. The data curators have identified within the video 2 frames corresponding to the ES(Systole) and the ED(Diastole) phase respectively. These frames are traced with 21 lines covering the entire left ventricle cavity. Following the Simpsons method, the traces are used to calculate the volume of the left ventricle at the ES phase and the ED phase respectively, and the volumes are used to calculate the EF. In this experiment we have developed a model that detects these volume traces given the input frame, which is either the ES or the ED frame.

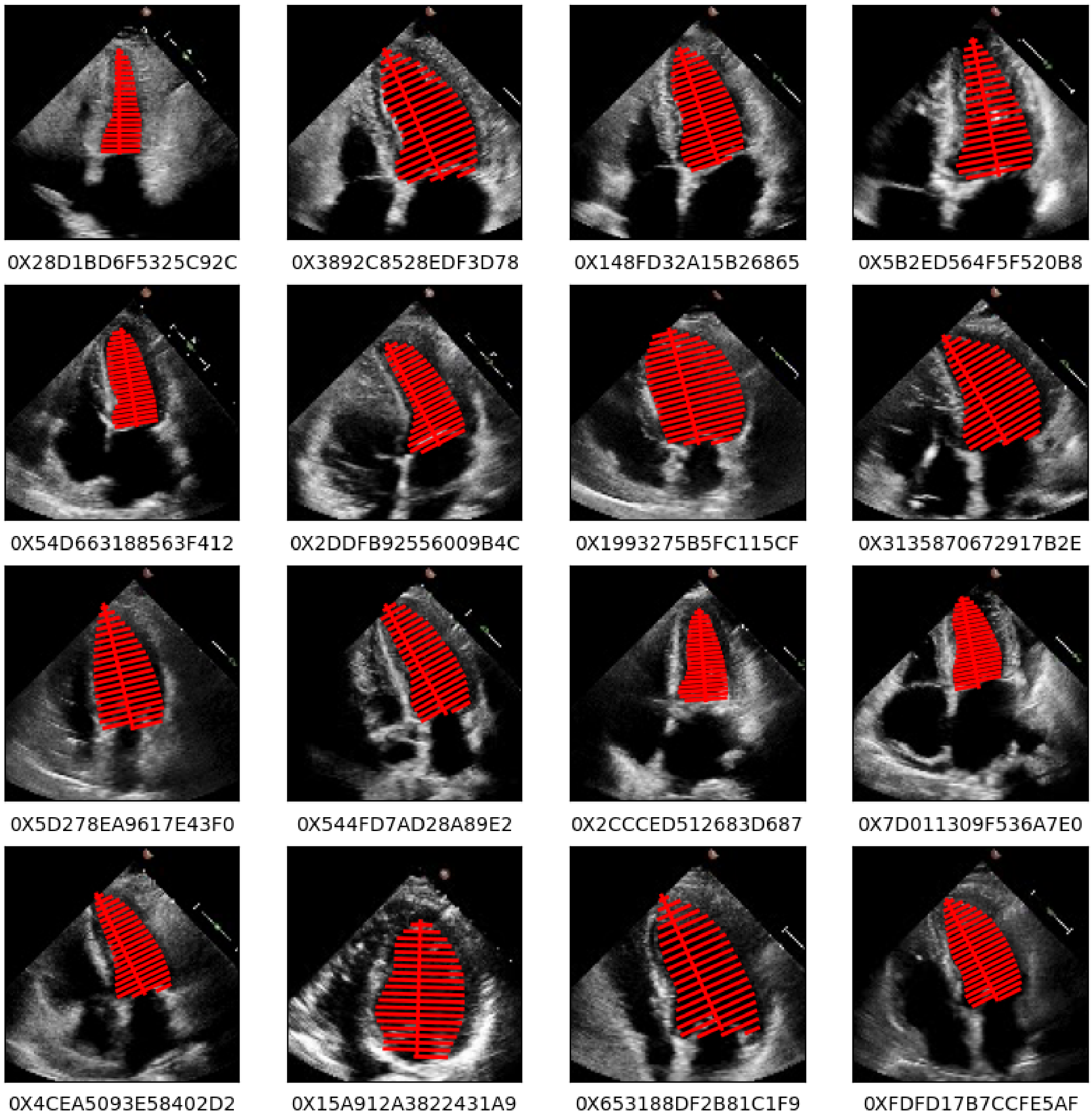


Figure 2: Sample ES and ED frames with the corresponding volume traces.

Model Design

The model is illustrated in Figure 3 and was built using the Tensorflow 2 framework. The model uses the MobileNetV2 architecture as the backbone. The MobileNetV2 model is pre-trained with ImageNet weights, and is frozen for the duration of the training. The original output head is removed and replaced with two Separable Convolutions layers. The first layer has a filter size of 3, and uses a RELU activation function. The second layer uses a filter size of 2, a sigmoid

activation function, and outputs the coordinates for the volume traces in the input image. Given that the model has to detect 21 lines, this translates to 84 values predicted by the model: 2 endpoints per line, and each endpoint has an x-coordinate and a y-coordinate (21*2*2). The input image is the ES or the ED frame, and has an image size of 112x112x3.
For the loss function the mean absolute error(MAE) is used, and the Adam optimizer is used for gradient descent with a constant learning rate of 1e-4. A Dropout layer is added after the MobileNetV2 output for regularization during training.

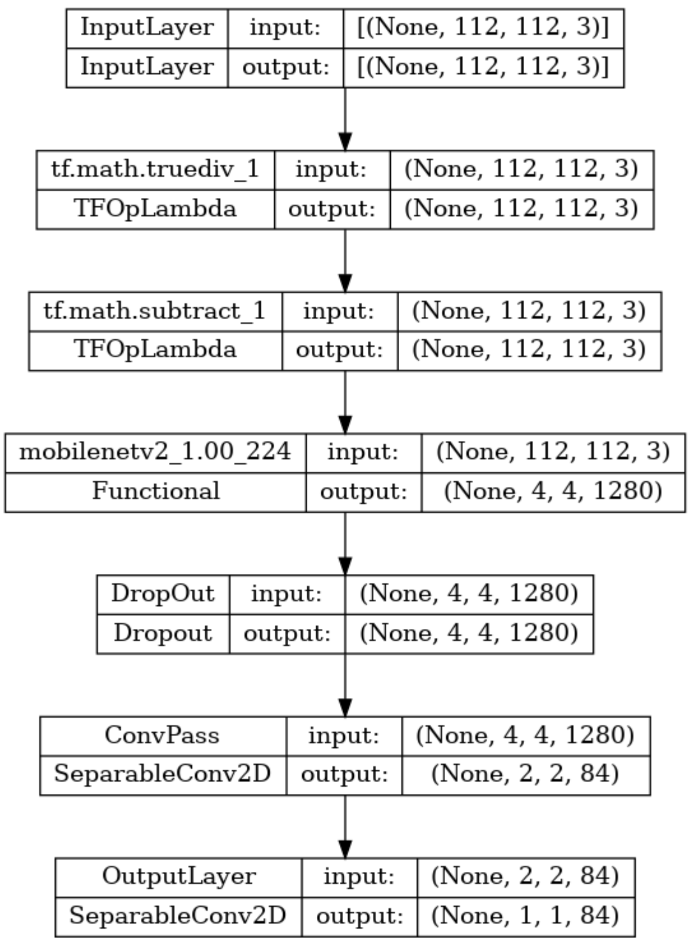


Figure 3: Left Ventricle Volume Trace Detection Model
Overall, the model has 2,384,584 parameters, out of which 126,600 are the trainable parameters.

The choice of MobileNetV2 was arbitrary based on its speed of computation due to efficient calculation of parameters at each layer due to separable convolution.

Training and Validation

The EchoNet-Dynamic dataset is split into 7465 training videos, 1288 validation videos, and 1277 test videos. Since we extract 2 frames from each video corresponding to the ES and the

ED frame, this translates to 14918 training images, 2576 validation images, and 2552 test images. All the images are resized to 112x112x3. There is no preprocessing done on the images. Rather we use the mobilnetv2 preprocessing package provided by Tensorflow to process the images before inputting it to the model for training and evaluation. The output coordinates or keypoints are reshaped into a 1x1x84 matrix. The point values are rescaled by the Image size. This is done to ensure that the output values remain between 0 and 1.

For both training and validation of the model, the MAE loss function was used. We also tried the mean square error (MSE) loss during one of the tuning experiments, but found that no loss function was better than the other. So we made a choice of using MAE. The learning rate was chosen to be 1e-4 through empirical experiments also. These were the only 2 tuning parameters tested during development.

During the training stage, we used the training set for learning the parameters of the model and the validation set for evaluation of the model. To prevent overfitting, we added a checkpoint scheme that will save the weights of the model when we find a minimum validation loss than from previous epochs. The model was trained for 200 epochs. However, the model converged on the 75 epoch where it encountered the minimum validation loss. The loss curves are illustrated in Figure 4.

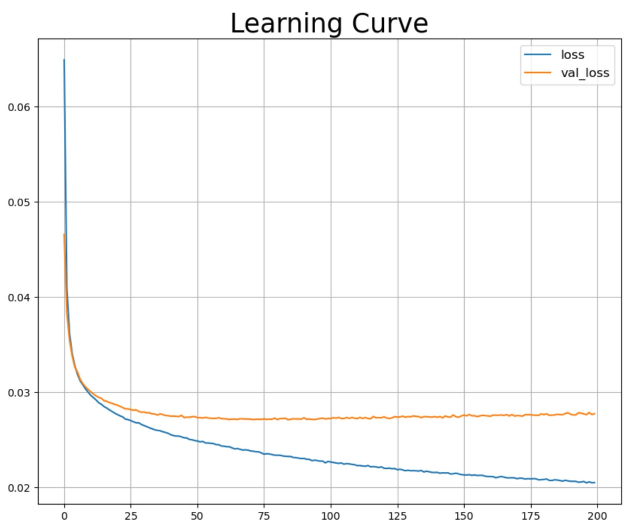


Figure 4: Loss curves after the training stage.

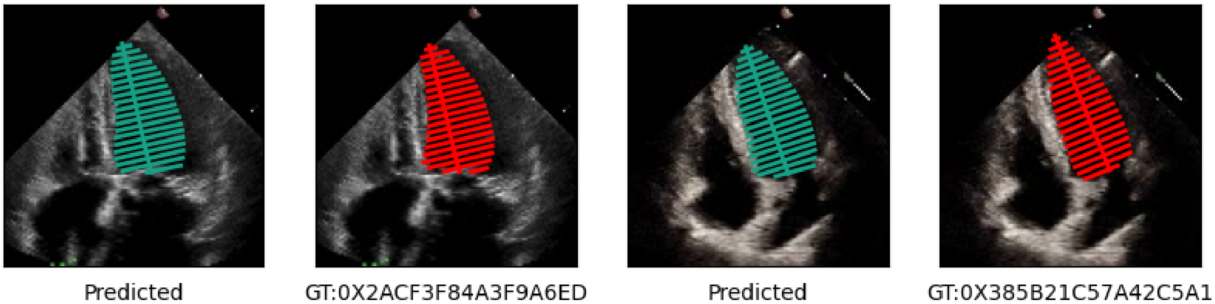
Performance and Evaluation

The table below shows the loss values (MAE) from the model for the training set, validation set and the testing set.

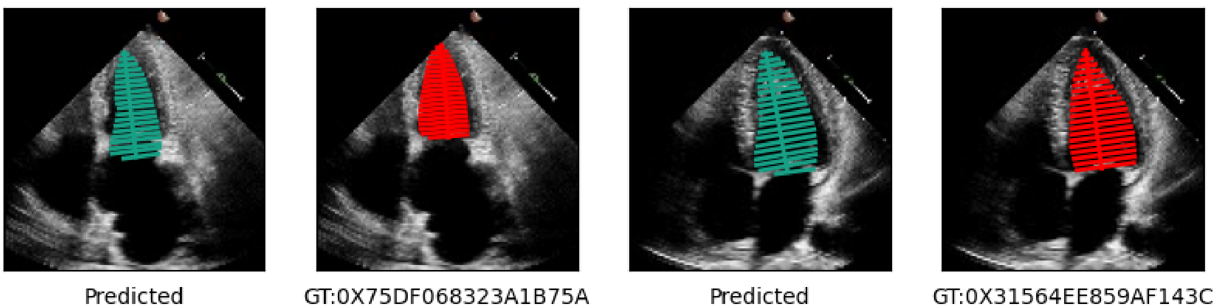
Loss for training images : 0.020817052572965622
Loss for validation images : 0.027102364227175713
Loss for testing images : 0.026492547243833542

The model converges reasonably well and the testing set also has similar loss values to the training and validation set. This gives a good confidence that the model did not overfit significantly.

Figure 5 below shows the output on an ED and ES image from a training set, and a testing set.



(a)



(b)

Figure 5: (a) shows the predicted traces in blue and the manual trace in red for a training set. (b) similarly for a testing set.

The output predictions confirm that the model reliably identifies the left ventricle cavity, and also scales the lines to fit within the cavity. The lines are also approximately parallel as expected. However, it does not reliably capture the deformations in the left ventricle, as seen in the test data.

Since the goal of predicting the volume traces was to calculate the EF values, we calculate the EF values for each instance, using the manually drawn traces in the ES and ED frame, and the predicted traces for the same 2 frames.
As mentioned earlier the formula for calculating the ejection fractions is:

$EF = (EDV - ESV)/EDV \times 100\,\%$ - Eq (1), where EDV is the volume of the left ventricle cavity in the ED phase, and the ESV is the volume of the left ventricle cavity in the ES phase.

We calculate the Actual EF and the Pred EF values for the training set using the manual volume traces and the predicted volume traces respectively. Taking the absolute difference of Actual EF and Pred EF gives us the residual errors in our model. To get a error bound as a measure of heart function, we use the Actual EF values to classify the training data into HF categories: EF value greater than 50 is classified as a normal heart function, EF between 40 and 50 is classified as mildly compromised heart function, and any value less than 40 is classified as having abnormal heart function. Based on this classification, we get the following error bound as shown in Figure 6..

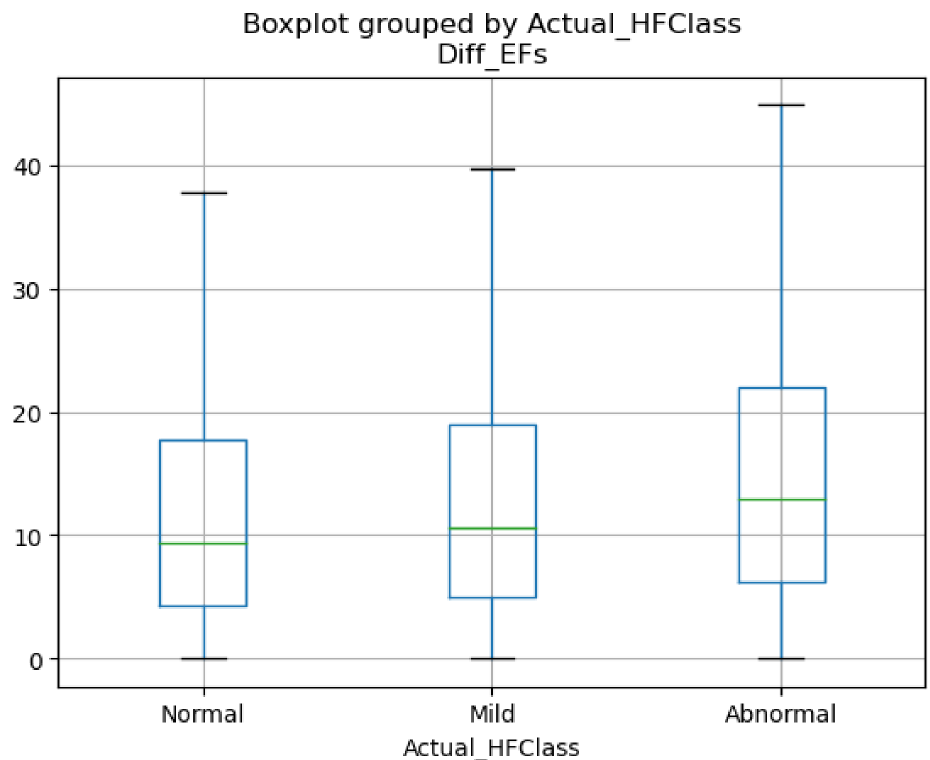


Figure 6: Residual errors as a measure of heart function.

The picture shows a median error in the range of 8 to 15. However, it also shows a larger error bound for abnormal cases, which translates to the model not generalizing well to the datasets. As mentioned above, this is possibly due to the model not capturing deformations of the left ventricle across datasets, and also between ES and ED. We also find that there are 241 cases where the Predicted EDV volume is smaller than the Predicted ESV volume resulting in negative EF value, which exacerbates the error overall and is attributed to poor identification of the left ventricle cavity..

Since the EF value directly reflects the heart function class we also visualize the confusion matrix for the training, validation, and testing data classified in terms of the heart function.

```
Confusion Matrix for Training Data
Accuracy of model: 0.5939304417886397
Predicted  Normal  Mild  Abnormal
Actual
Normal       3508  1071      1128
Mild          232   178       337
Abnormal      111   145       737
Sensitivity of model for individual classes
Class Normal : 0.6146837217452251
Class Mild : 0.23828647925033467
Class Abnormal : 0.7421953675730111
```

```
Confusion Matrix for Validation Data
Accuracy of model: 0.5388198757763976
Predicted  Normal  Mild  Abnormal
Actual
Normal        551   185       252
Mild           36    29        70
Abnormal       23    28       114
Sensitivity of model for individual classes
Class Normal : 0.5576923076923077
Class Mild : 0.21481481481481482
Class Abnormal : 0.6909090909090909
```

```
Confusion Matrix for Testing Data
Accuracy of model: 0.5525078369905956
Predicted  Normal  Mild  Abnormal
Actual
Normal        561   160       259
Mild           48    26        52
Abnormal       20    32       118
Sensitivity of model for individual classes
Class Normal : 0.5724489795918367
Class Mild : 0.20634920634920634
Class Abnormal : 0.6941176470588235
```

The table shows the accuracy and sensitivity of the model to classify datasets based on heart function. The overall accuracy of the model is around 60 %, and the model’s capability of identifying a patient with abnormal heart function is only around 70%.

From our evaluation we are confident that this is a good approach for further model improvement.

MODEL DEPLOYMENT

Streamlit is a popular open-source Python library used for creating custom web apps for machine learning and data science projects. It's lauded for its simplicity and ability to turn data scripts into shareable web apps in just a few lines of code.

In the context of our project, the Streamlit app was deployed to provide an interface for interacting with the deep learning model developed for estimating the Left Ventricular Ejection Fraction (LVEF). The deployment process began with the creation of the app, wherein the UI was designed and the model integrated. This involved writing a script that takes input from the user (in this case, echocardiogram videos), feeds it into the model, and displays the output (the estimated LVEF).

Once the app was developed and tested locally, it was ready for deployment. For deployment, we used platforms like Heroku or AWS, which are cloud platforms that offer scalable app deployment. This involves packaging the app along with all necessary dependencies into a Docker container or using a requirements.txt file. The Docker container or requirements.txt file ensures that the cloud platform can recreate the necessary environment for the app to run. After pushing the app to the chosen platform, it handles the rest, providing a URL where the app can be accessed live on the web.

This deployment process has made our deep learning model for LVEF assessment readily accessible to healthcare professionals. They can easily upload echocardiogram videos to the Streamlit app and receive immediate LVEF estimates, thereby streamlining cardiac function assessments.

CONCLUSION

Through the course of the Automated Left Ventricular Ejection Fraction Assessment project, we have undergone a profound learning journey. Technically, we navigated the complexities of processing echocardiogram videos for model development, transforming these videos into images suitable for deep learning. The exploration of various model architectures, particularly the potential advantage of a spiral model, has enriched our understanding of building reliable deep learning models for medical imaging. Additionally, we gained hands-on experience in developing a user-friendly Streamlit app for echocardiogram analysis, making our complex model accessible and intuitive to end-users.

In terms of project management, the importance of thorough project planning and a data-driven approach to product development has been reinforced. These lessons were critical in steering the project towards its goals, despite the challenges encountered.

As we look forward to future steps, we aim to leverage Imagenet Models to enhance accuracy and reduce training time. We also plan to explore the use of Transformer models and pre-processing techniques to further improve the model's performance. Moreover, we see the potential in using this model as a base for similar projects, thereby broadening its impact.

The project has underscored the importance of identifying challenging areas promptly, proposing actionable plans, and focusing on generating initial results before embarking on model and solution improvements. These lessons, both technical and managerial, will undoubtedly guide us in our future endeavors at the intersection of artificial intelligence and healthcare.

REFERENCES

1. Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In Advances in neural information processing systems (pp. 568-576).
2. Li, X., Chen, H., Qi, X., Dou, Q., Fu, C. W., & Heng, P. A. (2018, September). H-DenseUNet: Hybrid densely connected UNet for liver and tumor segmentation from CT volumes. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 213-221). Springer, Cham.
3. Bernard, O., Lalande, A., Zotti, C., Cervenansky, F., Yang, X., Heng, P. A., ... & Sermesant, M. (2018). Deep learning techniques for automatic MRI cardiac multi-structures segmentation and diagnosis: Is the problem solved?. IEEE transactions on medical imaging, 37(11), 2514-2525.
4. Ruijsink, B., Puyol-Antón, E., Oksuz, I., Sinclair, M., Bai, W., Schnabel, J. A., ... & King, A. P. (2020). Fully automated, quality-controlled cardiac analysis from CMR: validation and large-scale application to characterize cardiac function. JACC: Cardiovascular Imaging, 13(3), 684-695.
5. Wu, H., Zhou, Z., van Stralen, M., Luo, J., de Jong, N., & Verweij, M. D. (2019, December). TMED: a two-stream network for micro-expression detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop (pp. 0-0).
6. AlNegheimish, N., AlJarbouh, A., Alkhalifah, M., AlBakr, A., Alghamdi, S., AlMukhaini, N., ... & Almogren, A. (2019). HMC-Q: An Efficient Quality Control Pipeline for Human Microbiome Samples. Genes, 10(10), 754.
7. Ouyang, D., He, B., Ghorbani, A., Yuan, N., Ebinger, J., Langlotz, C. P., Heidenreich, P. A., Harrington, R. A., Liang, D. H., Ashley, E. A., & Zou, J. Y. (2020). Video-based AI for beat-to-beat assessment of cardiac function. Nature.
8. Link to Echonet dynamic data: https://echonet.github.io/dynamic/index.html

APPENDIX I.

Two Stream Neural Network