IDENTIFYING TAKOTSUBO SYNDROME IN CT SCANS

A thesis submitted to The University of Manchester
for the degree of Bachelor of Science
in the Faculty of Science and Engineering

2025

Awab Akram

Department of Computer Science

Contents

Word Count: 12,683

List of Tables

List of Figures

Abstract

Identifying Takotsubo Syndrome in CT Scans
Awab Akram
A thesis submitted to The University of Manchester
for the degree of Bachelor of Science, 2025

Takotsubo Syndrome (TTS), often called "broken-heart syndrome," is an acute cardiac condition that mimics a heart attack but without blocked arteries. Its subtle presentation leads to frequent misdiagnosis as an acute coronary event, underscoring the need for better diagnostic tools. This project applies machine learning to routine chest computed tomography (CT) scans to improve early identification of TTS. We developed a pipeline that segments the heart region from CT images and employs a 3D Convolutional Neural Network (CNN) to distinguish TTS cases from normal controls. The model was trained and validated on a dataset of confirmed TTS and non-TTS CT scans, incorporating patient metadata (age and sex) to enhance predictive accuracy. Despite a relatively small training set, the results are promising: the model achieved high diagnostic performance, with an accuracy around 96% and an area under the ROC curve (AUC) of approximately 0.96 on the test data. Notably, the classifier caught all TTS cases in the test set (100% sensitivity), suggesting it can serve as a reliable screening tool. A simple graphical user interface (GUI) was also developed to demonstrate the system in practice, allowing users to upload a CT scan and receive a prediction. In summary, this project successfully prototyped an AI-assisted approach for detecting Takotsubo Syndrome via CT imaging, which could help clinicians avoid misdiagnosis and improve patient outcomes.

Acknowledgements

Identifying Takotsubo Syndrome in CT Scans
Awab Akram
A thesis submitted to The University of Manchester
for the degree of Bachelor of Science, 2025

I would like to express my sincere gratitude to my supervisor, Dr. Fumie Costen, for her continuous support, guidance, and encouragement throughout this project. I am also deeply thankful to Dr. Andrew Crean for providing invaluable clinical insights and access to the CT scan data, without which this research would not have been possible.

My appreciation also goes to the University of Manchester and the School of Computer Science for providing the infrastructure and resources necessary to carry out this work. Finally, I would like to thank my family and friends for their patience, motivation, and unwavering support during this journey.

Chapter 1

Introduction

1.1 Project Motivation

Takotsubo syndrome, also known as "broken heart syndrome," is a temporary heart condition that often mimics the symptoms of a heart attack. Due to this similarity, it is frequently misdiagnosed, leading to inappropriate treatments and potentially adverse outcomes for patients. The condition is characterized by a temporary weakening of the left ventricle, the heart's main pumping chamber, which changes shape to resemble a Japanese octopus trap (takotsubo) – hence the name [2].

The accurate diagnosis of Takotsubo syndrome typically requires a combination of clinical evaluation, electrocardiography, biomarker testing, and advanced imaging techniques. However, the subtle presentation of this syndrome in imaging makes it challenging to identify, particularly in the early stages when it most closely resembles a myocardial infarction.

This project is motivated by the need to improve the diagnostic accuracy of Takotsubo syndrome through the application of machine learning techniques to analyze CT scans. By developing a model capable of identifying the characteristic features of Takotsubo syndrome in imaging data, we aim to facilitate earlier and more accurate diagnosis, ultimately improving patient outcomes.

Recent studies have applied machine learning to echocardiography and cardiac MRI for TTS detection, but CT-based approaches remain largely unexplored. As chest CT scans are already performed in many emergency settings, leveraging them for TTS detection using machine learning could offer a low-cost, widely accessible screening tool [3, 4].

This project aims to bridge this gap by applying a deep learning approach — specifically a 3D Convolutional Neural Network (CNN) — to heart-segmented CT scans to detect Takotsubo Syndrome.

Ultimately, the system could serve as a triage tool in emergency settings, helping

clinicians identify TTS earlier and avoid misdiagnosis.

1.2 Project Objectives

The primary objective of this project is to develop a machine learning model that can accurately identify Takotsubo syndrome in CT scans, with the goal of catching as many true TTS cases as possible. Specifically, the project aims to:

1. Process and prepare DICOM files from both normal and Takotsubo cases for analysis.
2. Implement effective segmentation techniques to isolate the heart region in CT scans.
3. Develop and train a 3D Convolutional Neural Network (CNN) to classify CT scans.
4. Evaluate the model's performance using appropriate metrics.
5. Create an interpretable model that can assist clinicians in diagnosis.

By achieving these objectives, the project seeks to create a tool that can serve as a decision support system for clinicians, flagging potential cases of Takotsubo syndrome for further investigation and appropriate management.

The model's performance is evaluated on a held-out test set using standard classification metrics, including accuracy, sensitivity, and area under the ROC curve (AUC), with particular focus on identifying all true TTS cases.

1.3 Report Structure

The remainder of this report is structured to provide a clear account of the work. First, the **Background** section reviews relevant information on Takotsubo Syndrome and the use of machine learning in medical imaging, setting the context for the project. Next, the **Design and Methodology** section details the approach taken – from data collection and preprocessing to model architecture, training procedures, and evaluation results. Finally, the **Conclusion** summarizes the project's achievements and findings, reflects on limitations, and suggests directions for future work in deploying AI for TTS diagnosis.

Chapter 2

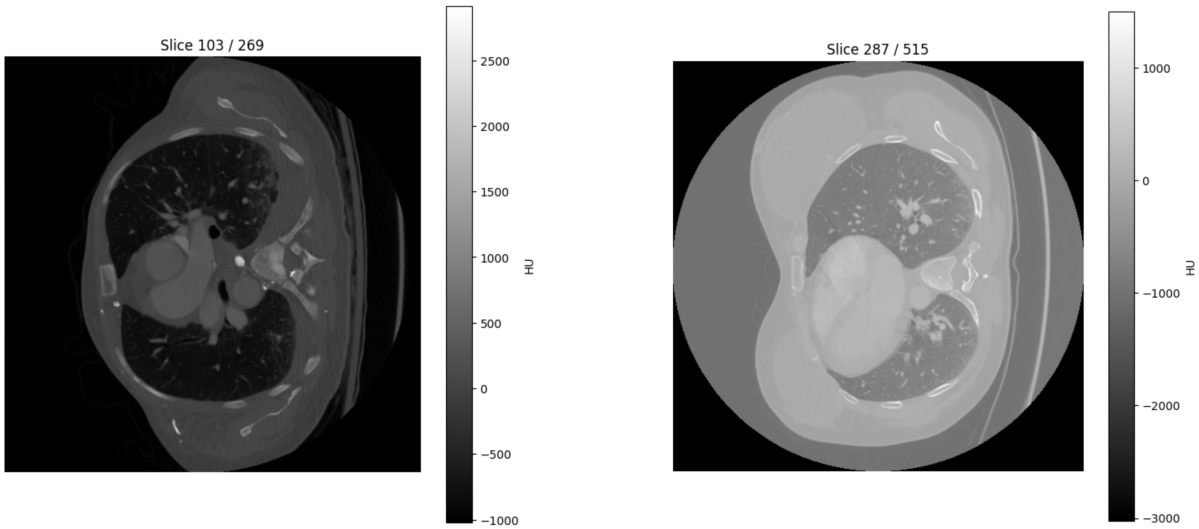
Background

2.1 Takotsubo Syndrome (TTS) Overview

2.1.1 Definition

Takotsubo syndrome (TTS), also known as Takotsubo Cardiomyopathy or "Broken Heart Syndrome," is an acute but reversible form of heart failure that is often triggered by emotional or physical stress [2]. The condition is characterized by transient left ventricular dysfunction in the absence of obstructive coronary artery disease. Internally, TTS presents with a distinct pattern of left ventricular wall motion abnormality, most notably apical ballooning, where the apex of the heart dilates while the base remains hypercontractile. This results in the ventricle adopting a shape reminiscent of a traditional Japanese octopus trap, inspiring the name "Takotsubo" [5].

Externally, patients may present with symptoms such as chest pain, shortness of breath, nausea, and fatigue [6]. These symptoms closely resemble those of acute myocardial infarction (AMI), making initial diagnosis challenging [7, 8]. Unlike AMI, however, TTS is non-ischemic—meaning it is not caused by blocked coronary arteries—though some patients may have coexisting coronary artery disease [7, 9]. The underlying pathophysiology is not yet fully understood, but it is widely believed to involve a surge in catecholamines (stress hormones), resulting in myocardial stunning through mechanisms such as microvascular spasm, energy imbalance, or direct toxicity [10].



(a) Takotsubo Syndrome CT (b) Non-Takotsubo CT

Figure 2.1: Comparison of CT scans: (a) Takotsubo Syndrome showing apical ballooning, (b) Normal cardiac function.

2.1.2 Epidemiology

TTS is relatively rare but likely underdiagnosed. It is estimated to account for approximately 1–3% of all cases of suspected ST-Elevation Myocardial Infarction (STEMI) [11]. The condition disproportionately affects postmenopausal women, with up to 90% of cases occurring in this demographic and a mean age of around 67 years [11]. Women over the age of 55 are particularly at risk, with studies suggesting they may be up to five times more likely to develop TTS than younger women or men [11]. This gender imbalance is thought to be influenced by hormonal changes, particularly reduced estrogen levels following menopause.

Although rare, TTS can also occur in younger individuals and even in children, though such cases are uncommon [12]. Increased clinical awareness in recent years has led to more frequent recognition of the syndrome, including among male patients, particularly when physical stressors are involved. Recurrence is possible, with estimates suggesting a rate of approximately 1.8% per patient-year [13], highlighting the importance of long-term follow-up and patient monitoring.

2.1.3 Why is Takotsubo Syndrome Frequently Misdiagnosed?

Takotsubo syndrome (TS) is often misdiagnosed as acute myocardial infarction (AMI) due to several factors:

1. **Similar clinical presentation:** TTS and AMI present with nearly identical symptoms—chest pain, dyspnea, and syncope — making initial differentiation difficult [11].

2. **ECG and Biomarker Overlap:** TTS commonly shows ST-segment elevations and T-wave inversions, similar to AMI [13]. Cardiac troponins may be modestly elevated, further contributing to confusion [5].

3. **Limited access to immediate advanced imaging:** Definitive diagnosis of TTS often requires advanced imaging techniques like echocardiography or cardiac MRI to identify the characteristic wall motion abnormalities. In emergency settings, these imaging modalities may not be immediately available, leading to initial misdiagnosis based on more readily accessible tests [14].

4. **Coexisting Coronary Artery Disease (CAD):** Some TTS patients may have underlying coronary artery disease, which can confound the diagnosis and lead clinicians to attribute the symptoms to AMI [13].

5. **Lack of awareness:** Despite increasing recognition, many clinicians may still be unfamiliar with TTS, especially in regions where it is less commonly encountered [15].

6. **Evolving diagnostic criteria:** The diagnostic criteria for TTS have evolved over time, and there is still no universal consensus, which can contribute to misdiagnosis [11].

7. **Atypical variants:** While the apical variant is most common, TTS can present with other patterns of left ventricular dysfunction, such as midventricular or basal variants. These atypical presentations may be more challenging to recognize and can be mistaken for other cardiac conditions [11, 1, 13].

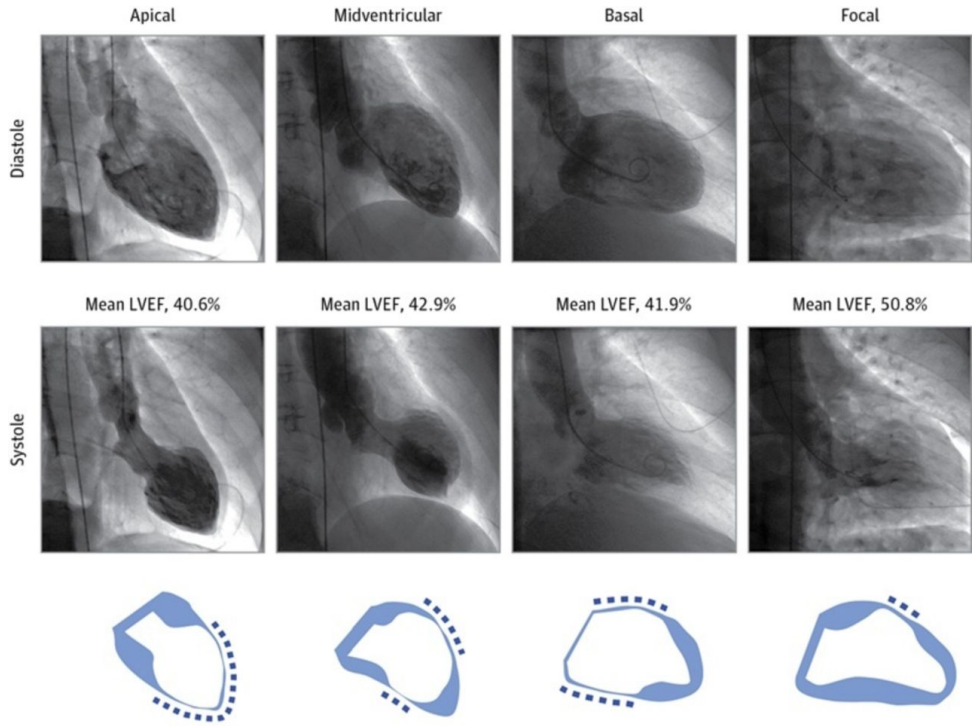


Figure 2.2: Variants of Takotsubo Syndrome: Apical, Midventricular, and Basal Ballooning. Adapted from [1].

2.1.4 Diagnostic Criteria and Guidelines

Diagnosing Takotsubo Syndrome (TTS) typically follows one of two established frameworks: the **Mayo Clinic Criteria** or the **InterTAK Diagnostic Criteria**, supported by the European Society of Cardiology (ESC).

Mayo Clinic Criteria

The Mayo Clinic Criteria, revised in 2008, require all four of the following to be met:

1. Transient left ventricular wall motion abnormalities [7]
2. Absence of obstructive coronary artery disease on angiography [7]
3. New ECG changes (such as ST elevation or T-wave inversion) or elevated cardiac troponin [7]
4. No evidence of pheochromocytoma or myocarditis [7]

Although these criteria originally excluded patients with any coronary artery disease (CAD), it is now accepted that **TTS can occur alongside mild or unrelated CAD**, as long as it doesn't explain the wall motion abnormality[13].

ESC and InterTAK Criteria

The ESC and InterTAK guidelines build on the Mayo framework but allow for more flexibility. They accept that TTS can occur with incidental CAD and include neurological triggers like seizures and stroke as possible causes [11]. **The diagnosis remains one of exclusion**, confirmed after ruling out heart attacks and myocarditis, usually using a combination of imaging and lab results.

2.1.5 Consequences of Misdiagnosis

While TTS is often reversible, misdiagnosis can lead to inappropriate treatments that may cause further complications. Since its symptoms resemble those of a heart attack, patients with TTS are frequently subjected to therapies designed for ischemic heart disease, such as anti-thrombotic medications or invasive procedures[2, 5].

Additionally, TTS can lead to serious complications such as heart failure, arrhythmias, left ventricular outflow tract obstruction, or even cardiogenic shock in severe cases [2]. Early and accurate diagnosis is essential to ensuring proper management and reducing the risk of adverse outcomes.

2.2 Diagnostic Imaging for Takotsubo Syndrome

2.2.1 Heart Tracing (ECG)

An electrocardiogram (ECG) records the electrical activity of the heart. In Takotsubo syndrome (TTS), ECG often exhibits abnormalities that mimic acute myocardial infarction, such as ST-segment elevation and T-wave inversion. Additionally, QT interval prolongation is commonly observed [12]. However, these changes are typically less pronounced than those seen in true myocardial infarctions and lack reciprocal changes, which are indicative of transmural ischemia [7].

2.2.2 Blood Tests

Cardiac biomarkers can help differentiate TTS from myocardial infarction. Troponin levels, which indicate myocardial injury, are only mildly elevated in TTS, whereas they are significantly higher in myocardial infarction [12]. Conversely, B-type natriuretic peptide (BNP) or its precursor NT-proBNP is usually substantially elevated in TTS due to myocardial stretching without permanent damage [2].

2.2.3 Heart Imaging

Echocardiography

This ultrasound-based technique reveals distinctive patterns of left ventricular wall motion abnormalities in TTS, most commonly apical ballooning, where the apex is akinetic while the base contracts normally [16]. These abnormalities usually extend beyond the distribution of a single coronary artery.

Coronary Angiography and CT

In TTS, coronary angiography or CT angiography typically shows unobstructed or only mildly narrowed coronary arteries, helping to distinguish TTS from acute coronary syndromes caused by occlusive coronary artery disease [10].

Cardiac MRI

Cardiac magnetic resonance imaging can demonstrate myocardial edema and hyperemia in TTS but usually lacks the late gadolinium enhancement indicative of necrosis, thus helping to exclude myocardial infarction [17].

2.2.4 Comparison of Diagnostic Modalities

Table 2.1: Comparison of Diagnostic Modalities in TTS vs. Acute Myocardial Infarction (AMI)

Modality	**Typical Findings in TTS**	**Distinction from AMI**
ECG	ST-segment elevation, T-wave inversion, prolonged QT interval [12]	Changes are less pronounced; lacks reciprocal changes [7]
Blood Tests	Mildly elevated troponin [12]; significantly elevated BNP/NT-proBNP [2]	Troponin is much more elevated [12]; BNP is usually lower [2]
Echocardiography	Apical ballooning; apex akinetic, base contracts normally [16]	Wall motion abnormalities extend beyond one artery's territory [7]
Coronary Angiography / CT	Normal or mildly narrowed coronary arteries [10]	AMI typically involves occlusion of major arteries [7]
Cardiac MRI	Myocardial edema and hyperemia without late gadolinium enhancement [17]	AMI shows necrosis with late gadolinium enhancement [17]

2.3 Machine Learning in Medical Imaging

Recent years have seen an explosion in the application of **machine learning (ML)** techniques to medical data, particularly imaging. Machine learning refers to a set of computational methods that enable computers to **learn patterns from data** and improve their performance on tasks without being explicitly programmed for each rule. A subset of ML, **deep learning (DL)**, uses multi-layered neural networks to automatically learn hierarchical representations of data. This section provides an introduction to machine learning and deep learning in the context of medical imaging, focusing on convolutional neural networks (CNNs) which are the workhorse for image analysis tasks. We also explain the difference between 2D and 3D CNN approaches and their relevance to analyzing volumetric scans such as CT images.

2.3.1 What is Machine Learning?

In traditional programming, a human defines explicit rules to process data and produce an output. In machine learning, by contrast, the algorithm **learns the rules or patterns from examples**. In supervised ML (common in medical imaging), we provide the model with input data (e.g., an image) and a desired output (e.g., diagnosis), and the model iteratively adjusts internal parameters to map the input to the correct output. If we have enough examples (training data), the model uncovers statistical patterns that allow it to make predictions on new, unseen data. For instance, an ML model could be trained on many labeled cardiac images (some with TTS, some with AMI) and learn to distinguish them. A classic ML approach might involve manually designing features (like measuring wall motion or shape descriptors) and then using a classifier (like logistic regression or a decision tree) to combine those features. However, manual feature design is challenging for complex images. This is where deep learning revolutionized the field – **deep neural networks can automatically learn features** directly from raw images, often surpassing human-crafted features in efficacy [18].

In medical imaging, ML has been applied to tasks such as detecting tumors in scans, classifying lesions, segmenting organs, etc., with impressive results [19]. Notably, around 2012, deep learning began outperforming previous methods in image recognition, and since then it has become the dominant approach for image analysis [20].

2.3.2 Neural Networks and Deep Learning

Neural networks are computational models inspired by the structure of the brain, consisting of layers of interconnected artificial neurons. Each neuron computes a weighted sum of its inputs and passes it through a nonlinear activation function. In a **deep neural network**, multiple hidden layers allow the model to learn complex hierarchical

representations. Deep learning has enabled automatic feature extraction from raw data, eliminating the need for handcrafted features [21].

In medical imaging, deep learning models—particularly convolutional neural networks—have demonstrated state-of-the-art performance across tasks like detection, classification, segmentation, and registration [22]. The success of deep learning is largely attributed to the availability of large annotated datasets, powerful GPUs, and open-source frameworks such as **PyTorch**, which provide efficient tools for model development and training [23]. PyTorch offers dynamic computation graphs, GPU acceleration, and modular APIs, making it a widely used library in the medical imaging research community.

2.3.3 Convolutional Neural Networks (CNNs)

A convolutional neural network is a type of deep learning model specifically designed for grid-like data such as images. In a regular neural network, each input (pixel) connects to neurons in the next layer (fully-connected), which doesn't scale well to large images. CNNs introduce the concept of **convolutions** – sliding filters (kernels) across the image to detect local patterns. A CNN typically consists of multiple layers: convolutional layers that apply filters to extract features (like edges, textures, shapes), pooling layers that downsample or summarize regions (for spatial invariance and to reduce computation), and eventually fully-connected layers that make a classification decision based on the extracted features.

The convolution operation is key: a small filter (for example 3×3 pixels) is scanned over the image and computes a weighted sum at each position, producing a feature map that highlights where certain features are present. Early layers might detect simple features (edges or blobs), while deeper layers detect complex concepts (like a ventricular shape). The CNN learns the filter weights during training.

2D CNNs operate on 2D images (height × width), which is ideal for single slice images like chest X-rays or pathology slides. **3D CNNs** operate on 3D data (height × width × depth), which is suitable for volumetric scans like CT or MRI where an additional dimension (slices or time) provides context [24].

CNNs have become the dominant architecture for medical image analysis because of their ability to automatically learn relevant visual features. For example, if we wanted a CNN to identify Takotsubo Syndrome on an imaging exam, we would feed in a set of scans labeled as TTS or not TTS. The CNN's convolutional filters in early layers might learn to detect shapes like the ballooned apex or ventricular contours, mid-level layers might combine those to detect a pattern of wall motion abnormality, and the final layers would learn to associate those patterns with the diagnosis of TTS. All this is learned from data, rather than explicitly programmed. This ability to learn subtle and complex patterns is crucial in medicine, where differences can be nuanced. Indeed, deep learning algorithms,

in particular CNNs, have rapidly become a methodology of choice for analyzing medical images [19].

2.3.4 2D vs. 3D CNNs

The distinction between 2D and 3D CNNs is important for medical imaging, because many medical images are inherently three-dimensional (like a stack of CT slices). A **2D CNN** treats input data as flat images. If applied to volumetric data, a 2D CNN would typically either analyze single slices independently or combine results from multiple 2D views. In contrast, a **3D CNN** can take a volume (a stack of slices) as input and has filters that extend in all three dimensions (x, y, and z). This means a 3D CNN can learn features that depend on the relationship between slices (the depth dimension). For example, in a cardiac CT, a 3D convolution could capture an apical ballooning pattern that spans across several adjacent slices of the heart. A 2D CNN might have to infer such a 3D pattern indirectly or require multiple 2D views.

There are trade-offs: 3D CNNs use more memory and computational power because of the extra dimension in filters and data. They also need more training data to learn the additional parameters. However, they can potentially capture the full spatial context of the anatomy. In tasks like detecting tumors in a CT or analyzing heart function from MRI, 3D CNNs have shown superior accuracy because they evaluate the volumetric context [24]. On the other hand, 2D CNNs can leverage pretrained models from broad image databases (like ImageNet) and may be easier to train with limited data. Sometimes a hybrid approach called "2.5D" is used: for instance, feeding multiple adjacent slices into a 2D CNN (as separate channels) to give it some 3D context without full 3D convolution.

For the task of detecting Takotsubo Syndrome in CT scans (which are 3D by nature), a 3D CNN approach could analyze the entire heart volume in one go, potentially learning the three-dimensional shape deformation of the ventricle in TTS. A 2D CNN could be used by analyzing multiple 2D views (like apical 4-chamber, short-axis slices, etc.), but may not capture global shape as directly. Researchers often experiment with both: a 2D CNN might be simpler and need fewer data, while a 3D CNN might capture the telltale features of TTS more holistically if sufficient training data is available [22].

2.4 Image Segmentation in Cardiac Imaging

2.4.1 Definition

Image segmentation is the process of partitioning an image into meaningful regions. In medical imaging, segmentation typically means delineating anatomical structures or abnormalities – for example, outlining the heart chambers on a CT scan, or segmenting a

tumor from healthy tissue. The goal is to assign a label to each pixel or voxel (e.g., "this voxel is part of the left ventricle, this voxel is background"). Segmentation is fundamental because it transforms raw images into quantitative data (such as volumes, shapes, and locations of structures) that can be analyzed or used by diagnostic algorithms. In cardiac imaging, segmentation can isolate the myocardium, ventricles, atria, or even the coronary vessels, enabling measurements of heart function (ejection fraction, wall thickness) and detection of regional motion abnormalities [25, 26, 27].

2.4.2 Importance in Cardiac CT

In the context of cardiac CT scans, segmentation is often a crucial preprocessing step. For instance, to detect TTS features, one might first segment the left ventricle from the CT to focus the analysis on the myocardium. Segmentation allows extraction of metrics like left ventricular volume, which could show an akinetic apex (if segmental volumes or motion are calculated). It also enables visualization tools (like overlaying strain or heatmaps on the heart). Furthermore, if one wants to apply machine learning, having segmented structures can reduce the complexity of the task – e.g., the algorithm can analyze just the heart region instead of the whole chest CT. Segmentation is also used for planning treatments (like identifying myocardial scar vs healthy tissue, though that's more in MRI) and for research (such as measuring the curvature of the LV apex in TTS) [25, 28].

Manual segmentation by expert radiologists or cardiologists is considered the gold standard for accuracy, but it is **time-consuming and labor-intensive** [29, 30]. Given the complex shape of the heart and possibly low contrast between blood and myocardium on non-contrast scans, manual outlining is tedious. Thus, automated segmentation methods are highly desirable.

2.4.3 Segmentation Methods

Over the years, various approaches have been developed:

- **Threshold and Region-Based Methods:** Early techniques segment images by intensity thresholds or region growing. For example, in CT, setting a high threshold can segment bones easily (since bone is very bright). For the heart, thresholding might separate blood pool from myocardium if contrast is used. However, classical segmentation methods (thresholding, edge detection, region growing) often struggle with cardiac CT because of **noise and low contrast** differences between myocardium and surrounding tissues. The heart borders may not be sharply defined and can be confounded by other structures (lungs, liver). These methods lack adaptability – a global threshold might not work across an entire image if lighting varies or if pathologies alter intensities [31, 32].

text

- **Contour-Based (Active Contours and Level Sets):** These techniques introduce a deformable model or "snake" that moves under algorithmic guidance to lock onto edges of structures. Active contour models have been widely used in heart segmentation due to their ability to produce smooth, continuous boundaries. For example, one can initialize a contour roughly around the left ventricle and an energy minimization process pulls the contour towards strong edges (the endocardial border). While active contours can be accurate and **provide continuous boundaries**, they are **sensitive to initialization and noise**. If the initial guess is far off or the image edges are weak, the contour might converge to an incorrect boundary or require heavy tuning of parameters [33, 34].

- **Statistical Shape Models:** Another approach uses predefined shape models of the heart (perhaps built from annotated datasets) and fits them to the new image. The model knows the typical geometry of a left ventricle, for example, and an algorithm deforms this model to match the image data. This incorporates prior anatomical knowledge and can deal with missing or noisy edges. Shape models have shown some success in whole-heart CT segmentation by ensuring the result is anatomically plausible. But they require a robust training of the shape space and can struggle with unusual patient anatomies (e.g., distorted by cardiomyopathy) [35, 36].

- **Classical Machine Learning:** Before deep learning dominance, some methods extracted features (texture, intensity profiles, gradient magnitude) around each pixel and trained a classifier (like random forest) to decide if a pixel is boundary or interior of a structure. These had some success but often needed extensive feature engineering and didn't generalize as well to new scanners or populations [37, 38].

- **Deep Learning (CNN-based Segmentation):** In the past 5-7 years, **deep learning segmentation** models have transformed the field. Chief among these is the **U-Net architecture**, introduced in 2015, which became the de facto standard for biomedical image segmentation. The U-Net is an *encoder-decoder CNN*: the encoder path progressively downsamples the image extracting features (like any CNN), and the decoder path upsamples to produce a segmentation map, with skip connections bridging corresponding levels to retain fine detail. This design allows precise localization (important for segmentation) while utilizing context. U-Net and its variants can automatically segment organs with high accuracy given enough training data. In fact, **U-Net outperformed prior methods** on many biomedical segmentation challenges and is widely used [39, 25, 40]. For cardiac CT, one could train a U-Net to segment the left ventricle, myocardium, atria, etc., by providing

enough CT images with ground truth masks. Deep learning models can implicitly learn the shapes and appearance of the heart, including handling of different contrast phases.

2.4.4 TotalSegmentator and Whole-Body Segmentation

A remarkable recent development is the release of **TotalSegmentator**, a deep learning model capable of segmenting a wide array of anatomical structures from CT scans automatically. TotalSegmentator was trained on over 1000 CT scans and can segment **104 anatomical structures** (organs, bones, muscles, vessels, etc.) with high accuracy. It leverages the powerful **nnU-Net** framework (which automatically optimizes U-Net for a given task) and achieves near human-level performance on many structures [41, 42]. For heart imaging, TotalSegmentator can provide segmentation of the heart, including substructures like ventricles and atria, without any user intervention. This tool is publicly available and can be run quickly (on the order of a couple of minutes for a whole CT).

For our project, using a tool like TotalSegmentator could allow us to automatically extract the heart from chest CT scans, thereby focusing the machine learning analysis on the relevant area for TTS detection. It could also provide precise masks of the left ventricle to compute shape features or apply explainability maps specifically on the myocardium. According to its publication, TotalSegmentator achieved an average Dice similarity coefficient of **0.94** across its 104 segmented structures on test data, which is extremely high agreement with expert segmentation [41].

2.4.5 Segmentation Approaches Comparison

To tie together these approaches, Table 2.2 compares different segmentation methods for cardiac CT imaging, highlighting their characteristics.

Table 2.2: Approaches to Cardiac CT Segmentation – Comparison

Approach	**Description and Example**	**Pros**	**Cons / Challenges**
Manual Segmentation	Expert draws contours of heart or regions on CT slices (ground truth standard).	Very accurate (expert knowledge applied); can account for context and anomalies.	**Labor-intensive**, time-consuming; not feasible for large datasets. Inter-observer variability can occur [29, 30].
Threshold / Region Growing	Use intensity threshold to separate structures (e.g., blood vs myocardium), or grow regions from a seed based on similarity.	Simple, fast, no training needed.	Struggles with **noise and low contrast** differences; often *oversimplistic* [31, 32].
Active Contours / Level Sets	Initialize a contour around LV; it moves to edge boundaries by minimizing energy.	Produces **smooth, continuous boundaries**; can incorporate prior shape knowledge for accuracy.	**Sensitive to initialization & noise**; needs parameter tuning [33, 34].
Statistical Shape Model	A pre-learned 3D heart shape is fitted to the image data.	Brings **anatomical plausibility**; good for noisy images.	Requires a representative shape library; may fail on abnormal geometries [35, 36].
Classical ML (Feature-based)	Compute features per pixel (intensity, gradient, texture) and use a classifier to label pixels.	More adaptive than global threshold; can incorporate contextual features.	**Feature engineering** is needed; limited by chosen features [37, 38].
Deep Learning – U-Net (2D)	Train a 2D U-Net CNN on labeled slices to predict segmentation masks.	**High accuracy**, learns both low-level and global features automatically.	Needs a **large annotated dataset** for training; 2D approach may ignore 3D context [39, 25].
Deep Learning – 3D CNN	Train a 3D network (like 3D U-Net or nnU-Net) on full or sub-volumes of CT for segmentation.	Uses full **3D context**; achieves state-of-the-art Dice scores. Fully automatic.	**Computationally heavy**; requires powerful GPU and more memory [42, 41].

2.4.6 Which segmentation approach to use?

As seen above, traditional methods had various limitations in the complex scenario of cardiac CT. **Deep learning approaches, especially 3D CNNs like U-Net/nnU-Net, have largely overcome these issues**, delivering fast and reliable heart segmentation [39, 42, 41]. For example, TotalSegmentator's developers reported their model "enables robust and accurate segmentation of 104 structures" and made both the trained model and dataset publicly available.

In practical terms, for our project of using ML to detect TTS, we could use an automated segmentation (like TotalSegmentator or a custom-trained U-Net) to first **extract the left ventricular myocardium from the CT**. Once we have that, we can compute shape indices or feed the segmented heart into a classification network that decides if it's TTS or not. This two-step process (segment then classify) can often improve accuracy because the classifier is not distracted by other structures (lungs, ribs, etc.) in the image. Moreover, if we want to measure something like apical-to-basal diameter ratio or ventricular wall thickening, having the segmentation makes it straightforward.

Overall, segmentation turns the qualitative assessment ("apex looks funny") into quantitative data (we can measure the apex size, motion, etc.). With modern ML, segmentation that once took an expert an hour can be done in seconds by a well-trained network with very high fidelity to manual outlines [41, 25]. As a result, segmentation is increasingly a solved component in many imaging pipelines, allowing researchers to focus on interpretation and classification tasks on the segmented outputs.

2.5 Existing Work in ML for Takotsubo Syndrome

While machine learning has been increasingly applied to cardiovascular imaging, studies specifically focused on Takotsubo syndrome (TTS) remain limited. Most of the existing work has leveraged echocardiography or cardiac MRI, with minimal exploration of CT imaging for this purpose.

2.5.1 Echocardiography-Based Approaches

Laumer et al. [43] developed a deep learning pipeline using convolutional and temporal neural networks to differentiate TTS from acute myocardial infarction (AMI) based on echocardiography videos. Trained on 228 patients and tested on an independent cohort of 220, their model achieved an AUC of 0.79 and outperformed expert cardiologists.

Zaman et al. [44] applied a spatiotemporal CNN to apical 4-chamber echo videos, distinguishing TTS from anterior STEMI with high accuracy. Their interpretability analysis revealed that the model relied on basal segment motion, an underappreciated feature by human readers.

2.5.2 Cardiac MRI-Based Approaches

Mannil et al. [4] extracted texture features from T2-weighted and LGE cardiac MRI of 58 TTS patients to predict 5-year major adverse events. A Naïve Bayes classifier yielded an AUC of 0.88, with edema-based texture features providing the strongest prognostic power.

Cau et al. [45] used non-contrast CMR including strain and tissue mapping data to build a classifier distinguishing TTS from myocarditis and controls. Their ensemble model achieved 92% sensitivity and an AUC of 0.94, identifying left atrial strain as the most predictive feature.

2.6 Evaluation Metrics for Classification

When developing machine learning models, especially for a task like binary classification (TTS vs non-TTS), it is crucial to assess their performance using appropriate **evaluation metrics**. Different metrics capture different aspects of performance, and in medical applications – particularly for a relatively rare condition like TTS – choosing the right metric is important. Here we explain common classification metrics (Accuracy, Precision, Recall, F1-Score, AUC-ROC) and discuss why, for Takotsubo detection, an emphasis on sensitivity (recall) might be warranted [46, 3, 47].

2.6.1 Accuracy

Accuracy is the simplest metric – it is the proportion of all instances that the model correctly classified. If we have TP true positives, TN true negatives, FP false positives, and FN false negatives, then:

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

For example, if a model evaluated 100 cases (10 were TTS, 90 were not) and it correctly identified 8 of the TTS and 85 of the non-TTS, the accuracy would be:

$$\frac{8 + 85}{100} = 93\%.$$

While accuracy seems intuitive, it can be misleading in imbalanced datasets [46, 48]. In our example, 93% accuracy sounds high, but note that if the model simply guessed "non-TTS" for all 100 cases, it would be 90% accurate by default (since 90 were not TTS). So accuracy alone does not tell us how well the model is doing on the positive (TTS) cases. In contexts like TTS detection, where TTS cases might be much fewer than non-cases in screening, a high accuracy can be achieved by trivially predicting the majority class,

which is not useful clinically [49].

2.6.2 Precision (Positive Predictive Value)

Precision is the fraction of positive predictions that are actually correct:

$$\text{Precision} = \frac{TP}{TP + FP}.$$

That is, out of all cases the model labeled as "TTS," how many were truly TTS. Precision answers: "When the model says TTS, how often is it right?" A high precision means false positives are low – the model does not cry wolf often. In our example, if the model labeled 10 cases as TTS and 8 were correct (2 were falsely labeled TTS), precision would be:

$$\frac{8}{10} = 0.8 \ (80\%).$$

Precision is important clinically because a low precision would mean many patients are being falsely told they might have TTS, which could lead to unnecessary stress or tests. However, precision must be balanced with recall [50].

2.6.3 Recall (Sensitivity or True Positive Rate)

Recall (also called **sensitivity** in medical terms) is the fraction of actual positives that the model correctly identified:

$$\text{Recall} = \frac{TP}{TP + FN}.$$

It answers: "Out of all real TTS patients, how many did the model catch?" Using the example, recall would be:

$$\frac{8}{8 + 2} = 0.8 \ (80\%),$$

if the model caught 8 of 10 TTS cases. High recall means the model misses very few TTS cases (few false negatives). In rare disease detection, **recall is often considered paramount** – missing a case (false negative) could mean a patient does not get needed treatment [51, 47]. For TTS, a false negative might mean we mistakenly think a patient's chest pain was something else and not monitor them for TTS complications. Generally, in critical diagnoses, we want recall to be as high as possible, even if it means tolerating some false positives [3].

2.6.4 Specificity (True Negative Rate)

Specificity is the counterpart to sensitivity – the fraction of actual negatives correctly identified:

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

It answers: "Out of all non-TTS cases, how many did the model correctly say are not TTS?" High specificity means few false alarms [52]. Precision and specificity are related but not identical (specificity considers all negatives whereas precision considers just the predicted positives). Specificity might be important to avoid overloading clinicians with false alarms. In practice, one balances sensitivity and specificity based on clinical needs [53].

2.6.5 F1-Score

The F1-score is the harmonic mean of precision and recall:

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

It provides a single measure that balances both precision and recall. In our example with precision = 0.8 and recall = 0.8, the F1-score would be:

$$F1 = 0.8.$$

The F1-score is useful when we seek a balance or when comparing models – it ensures a model is only rated highly if both precision and recall are reasonably good. A model with extremely high precision but low recall (or vice versa) will have an F1 closer to the lower number. In rare event detection, often one can trade off precision vs recall by adjusting decision thresholds, and F1 can help find a sweet spot [49]. However, F1 also assumes equal importance of precision and recall, which might not be true in all contexts (sometimes recall is more valued).

2.6.6 ROC Curve and AUC

The Receiver Operating Characteristic (ROC) curve is a plot of the model's **True Positive Rate (Recall)** against the **False Positive Rate** for various threshold settings [54]. As we tune a model to be more sensitive (catch more TTS), typically it will also raise more false alarms, so we get a trade-off. The ROC curve shows this trade-off across all sensitivity settings. Each point on the ROC curve corresponds to a particular threshold of the model's output at which we classify a case as positive. The **Area Under the ROC Curve (AUC-ROC)** is a summary metric that essentially gives the probability

that the model will rank a randomly chosen positive case higher than a randomly chosen negative case [55]. An AUC of 0.5 means no better than chance, 1.0 means perfect separation. AUC is useful to compare models irrespective of a specific threshold, and is often used in machine learning research [56].

2.6.7 Trade-offs in TTS Detection

Consider a scenario with 200 patients, where 10 have Takotsubo syndrome (TTS). A high-recall model might identify all 10 cases (recall 100%) but flag 5 false positives, resulting in Precision $= 10/(10+5) = 67\%$ and Accuracy $= 195/200 = 97.5\%$. Another model might be more conservative, identifying 8 true positives (recall 80%) with no false positives, giving Precision $= 100\%$ and Accuracy $= 198/200 = 99\%$.

Which is better? The first model misses no TTS cases but has some false alarms, while the second misses 2 cases but avoids false positives. Clinically, the first model is often preferred, as missing no cases is critical, even at the cost of a few false alarms. This highlights the importance of prioritizing recall and precision over accuracy in TTS detection.

AUC-ROC for the first model may also be higher, as it demonstrates the ability to identify all positives at some threshold. For TTS detection, high recall with acceptable precision is typically the goal.

2.7 Explainability in Medical AI

The use of artificial intelligence (AI) in clinical settings requires not only high performance but also transparency. Deep learning models, particularly convolutional neural networks (CNNs), are often considered "black boxes" due to their lack of interpretability. However, in medicine, clinicians must be able to understand and trust the AI's decisions, especially when those decisions influence diagnosis or treatment [57, 58].

Explainability techniques help address this issue by providing insight into how and why a model arrives at a specific prediction. This is crucial in high-stakes domains like cardiology, where incorrect predictions can have severe consequences [59]. Moreover, explainability supports regulatory compliance, clinical adoption, and improved patient communication [60].

2.7.1 Grad-CAM for Visual Explanations

Gradient-weighted Class Activation Mapping (Grad-CAM) is a widely used method for visualizing which parts of an input image influenced a CNN's decision [61]. Grad-CAM generates heatmaps by backpropagating the gradients of a specific class prediction through

the model's final convolutional layers. The result is a coarse localization map that highlights the most relevant regions of the image.

In the context of Takotsubo Syndrome (TTS), a model using Grad-CAM can produce a heatmap over the CT scan indicating which areas of the heart contributed to the prediction. For example, if the apical region of the left ventricle is highlighted, this can align with clinical expectations, as this region is commonly affected in TTS. Such visual explanations make it easier for clinicians to validate AI outputs and integrate them into their diagnostic process [57].

Beyond building trust, Grad-CAM can reveal model errors. If the heatmap consistently highlights irrelevant regions—such as image artifacts—it may indicate spurious correlations learned during training. Identifying and correcting such issues helps ensure the model is making decisions for medically valid reasons [62].

Chapter 3

Design, Methodology and Results

3.1 Dataset Overview

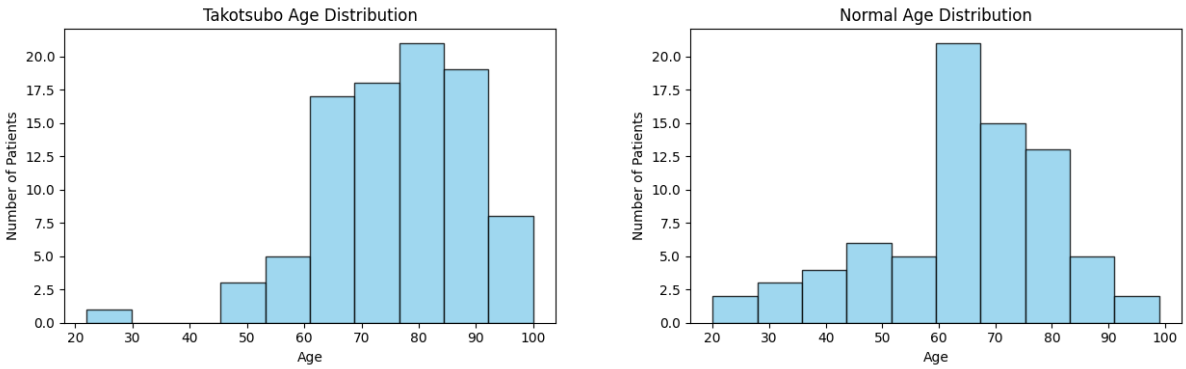
This project utilizes a retrospective dataset of cardiac CT scans provided by a collaborating clinician. The dataset comprises 91 Takotsubo Syndrome cases and 76 normal cases, each stored as a series of DICOM images in a patient-specific folder. Along with the scans, basic patient demographics (age and gender) are included as metadata. The dataset is not publicly available and was collected in a clinical setting. Notably, Takotsubo Syndrome predominantly affects older females (about 90% of cases are post-menopausal women [11]), so incorporating age and gender as input features is biologically relevant. All Takotsubo cases in the dataset were confirmed by the clinician, ensuring reliable ground truth labels. Normal cases were patients with unremarkable cardiac findings on CT. Table 3.1 summarizes the dataset composition. Because this is a private clinical dataset, no additional external data could be used for training or validation.

Table 3.1: Dataset composition by class

Class	**Number of Cases**	**Percentage**
Takotsubo Syndrome	91	54.5%
Normal	76	45.5%
Total	167	100%

Table 3.2: Gender distribution by class

Class	**Female**	**Male**
Normal	53	23
Takotsubo	84	8



(a) Takotsubo age distribution (b) Normal age distribution

Figure 3.1: Age distribution histograms by class

3.2 Data Preprocessing Pipeline

Raw CT volumes (DICOM format) require significant preprocessing before they can be fed into a neural network.

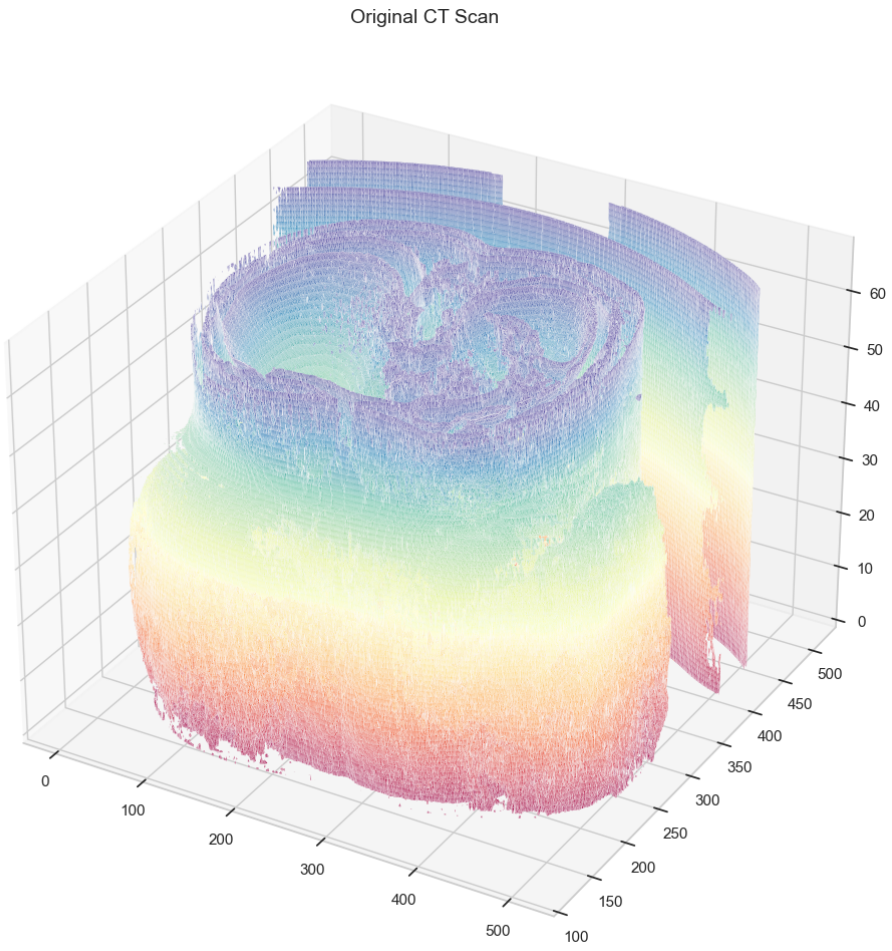


Figure 3.2: Original 3D CT scan before segmentation or preprocessing.

We implemented an end-to-end preprocessing pipeline (illustrated in Figure 3.3) to prepare the data efficiently for model training.

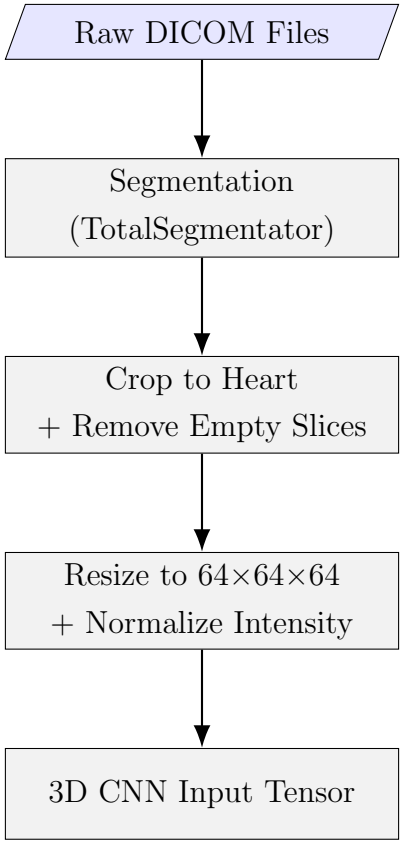


Figure 3.3: Pipeline from raw DICOM to processed input. The CT scan is segmented using TotalSegmentator, cropped to the heart region, resized, and normalized before being passed to the 3D CNN.

The pipeline includes the following steps:

3.2.1 Heart Segmentation

Raw chest CT volumes often include the entire thorax. We employed an automated segmentation tool (**TotalSegmentator**) to extract the heart region from each CT scan. TotalSegmentator is a state-of-the-art whole-body medical image segmentation model capable of identifying and delineating the heart and other organs. Using this tool, we generated a binary mask of the heart for each scan and then cropped the CT volume tightly around the heart [41]. This focuses the model on cardiac structures by removing unrelated anatomy (lungs, spine, etc.), effectively zooming in on the heart where TTS-related shape changes occur. The result of this step is a roughly heart-sized sub-volume (often on the order of $\sim 250 \times 200 \times 100$ voxels, depending on patient anatomy) for each case.

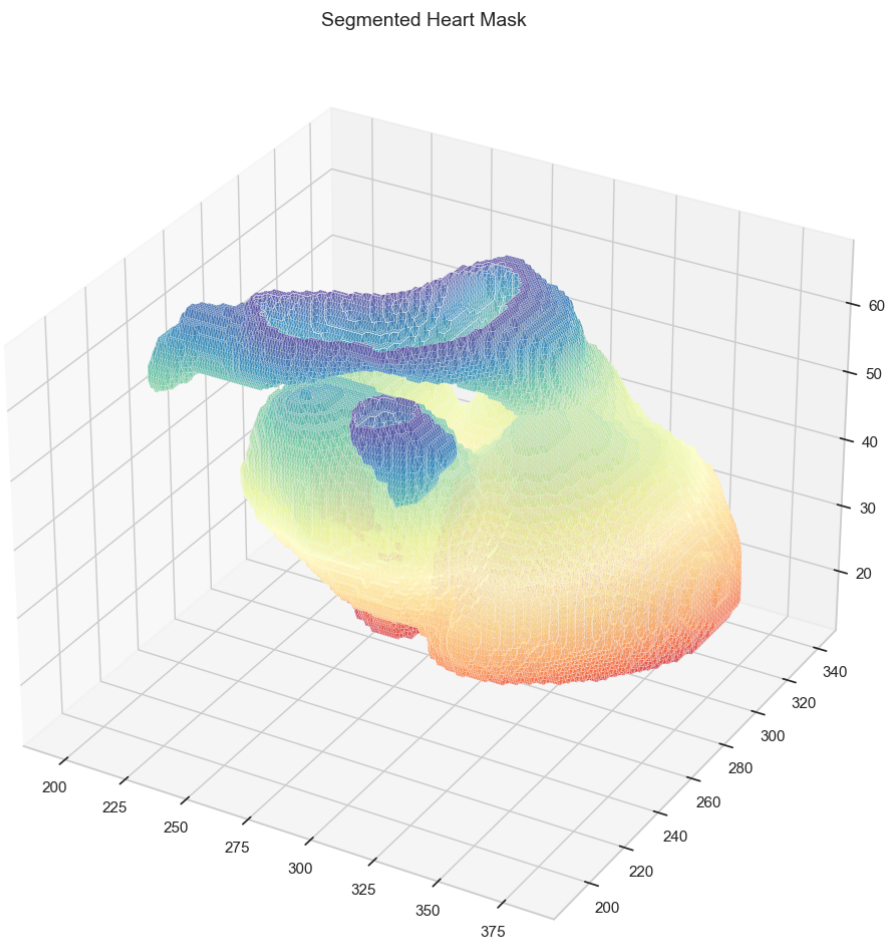


Figure 3.4: Heart segmentation mask generated by TotalSegmentator.

As you can see in Figure 3.5, TotalSegmentator also supports high-resolution segmentation of individual anatomical structures within the heart, such as the left and right ventricles, atria, myocardium, and others. However, we found that enabling this fine-grained mode increased processing time by a factor of approximately 5, making it impractical for our pipeline. As a result, we chose to segment the entire heart as a whole.

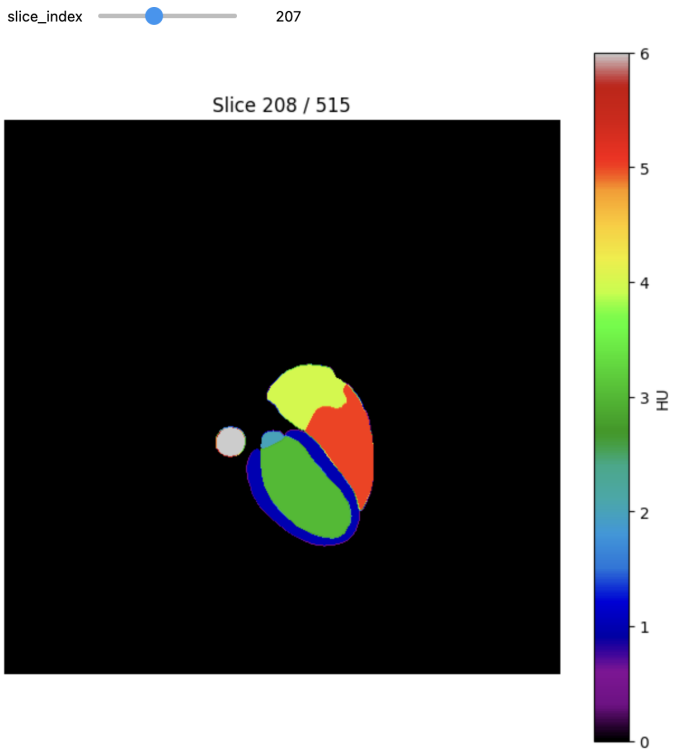


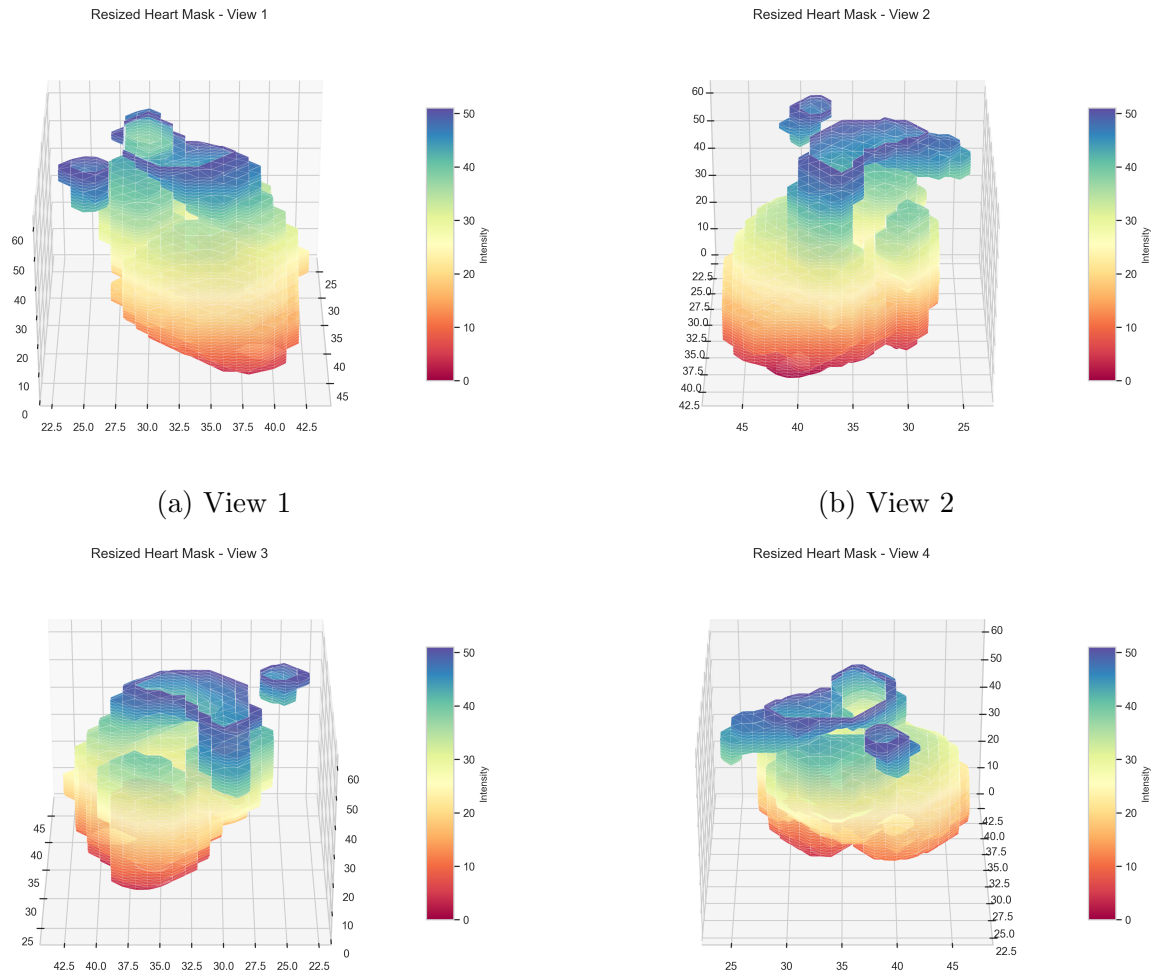
Figure 3.5: Example of individual anatomical heart segmentation using high-resolution mode.

3.2.2 Slice Selection and Alignment

Many CT DICOM series include slices that extend beyond the heart (e.g., base of neck or upper abdomen). We remove slices at the extremities that do not contain heart tissue (based on the mask) to tighten the focus on the heart. Additionally, all volumes are reoriented to a consistent axis (e.g., axial view) and any inconsistencies in patient positioning are addressed, ensuring that the heart is similarly oriented across all samples. This was done by using orientation transforms (via MONAI library) to standardize the axes of the volume so that the model does not need to learn orientation variations. Consistent alignment means, for instance, that the apex of the heart is always in a similar position across scans, making patterns easier to learn.

3.2.3 Resizing and Intensity Normalization

The segmented heart volumes are then resampled and resized to a standard voxel dimension of $64 \times 64 \times 64$. This fixed input size ensures a uniform tensor shape for the 3D CNN. Resizing also substantially reduces memory requirements while retaining enough resolution to capture anatomical details. We chose 64^3 after experimentation as a balance between detail and computational load. Essentially, each 64^3 voxel input to the model represents a standardized, focused 3D image of the patient's heart.



(a) View 1

(b) View 2

(c) View 3

(d) View 4

Figure 3.6: Resized heart mask from different perspectives, prepared for model input.

3.2.4 Split into Train/Validation/Test Sets

The dataset is divided into training, validation, and test subsets (approximately 70% train, 15% validation, 15% test). We ensure each split contains a balanced mix of Takotsubo and normal cases to prevent class imbalance in any subset. Patient scans are kept entirely in one subset (no patient's data is split) to avoid data leakage. The training set ($\approx$ 117 cases) is used to optimize the model's weights, the validation set ($\approx$ 25 cases) is used for tuning hyperparameters and early stopping (to avoid overfitting), and the held-out test set ($\approx$ 25 cases) is reserved for final performance evaluation. This separation guarantees that performance metrics reflect the model's ability to generalize to new, unseen patients.

3.2.5 Pipeline summary

By the end of this pipeline, each scan is converted into a $64 \times 64 \times 64$ voxel 3D image of the heart plus two patient metadata values (age and sex), ready for input into the neural network. This preprocessing approach standardizes diverse raw data into a uniform format, isolating the heart where Takotsubo's effects manifest and reducing extraneous variation. Figure **??** illustrates an example of this pipeline: starting from a raw CT (with the heart region outlined), to a cropped heart volume, to the final 64^3 input (post-alignment and normalization). This ensures that the subsequent model training focuses on relevant anatomy with minimal noise from other structures.

3.3 3D CNN Model Design

We designed a custom 3D Convolutional Neural Network (3D CNN), implemented as `class CNN3D` in our code, to perform binary classification (Takotsubo vs. Normal). A 3D CNN was chosen over a 2D CNN to exploit the volumetric nature of CT scans – whereas 2D models see only slices, a 3D model learns from the entire volume, capturing spatial context in the superior-inferior axis that is invisible to 2D slice-based methods. In medical imaging tasks, 3D CNNs have demonstrated superior performance by leveraging 3D spatial information [24]. This is particularly important for Takotsubo detection, as subtle shape changes in the left ventricle (e.g., apical ballooning) may only be apparent when considering the whole heart geometry across slices.

3.3.1 Architecture Overview

The network architecture (illustrated in Figure 3.7) consists of several components arranged sequentially:

Convolutional Blocks

The model has three 3D convolutional layers, each followed by a ReLU activation and a 3D max-pooling layer. The first convolutional layer uses a small kernel ($3 \times 3 \times 3$) to extract low-level features (edges, textures) from the input volume. Deeper convolutional layers use a similar kernel but operate on progressively lower-resolution feature maps. Pooling layers ($2 \times 2 \times 2$) after each convolution reduce the spatial dimensions by half, achieving downsampling and translation invariance. The number of filters increases with depth (we used 32, 64, and 128 filters in `conv1`, `conv2`, `conv3` respectively for the final model), so that the network can learn a richer set of features at each stage. These convolutional blocks enable the model to detect patterns like ventricular shape or wall motion abnormalities in 3D.

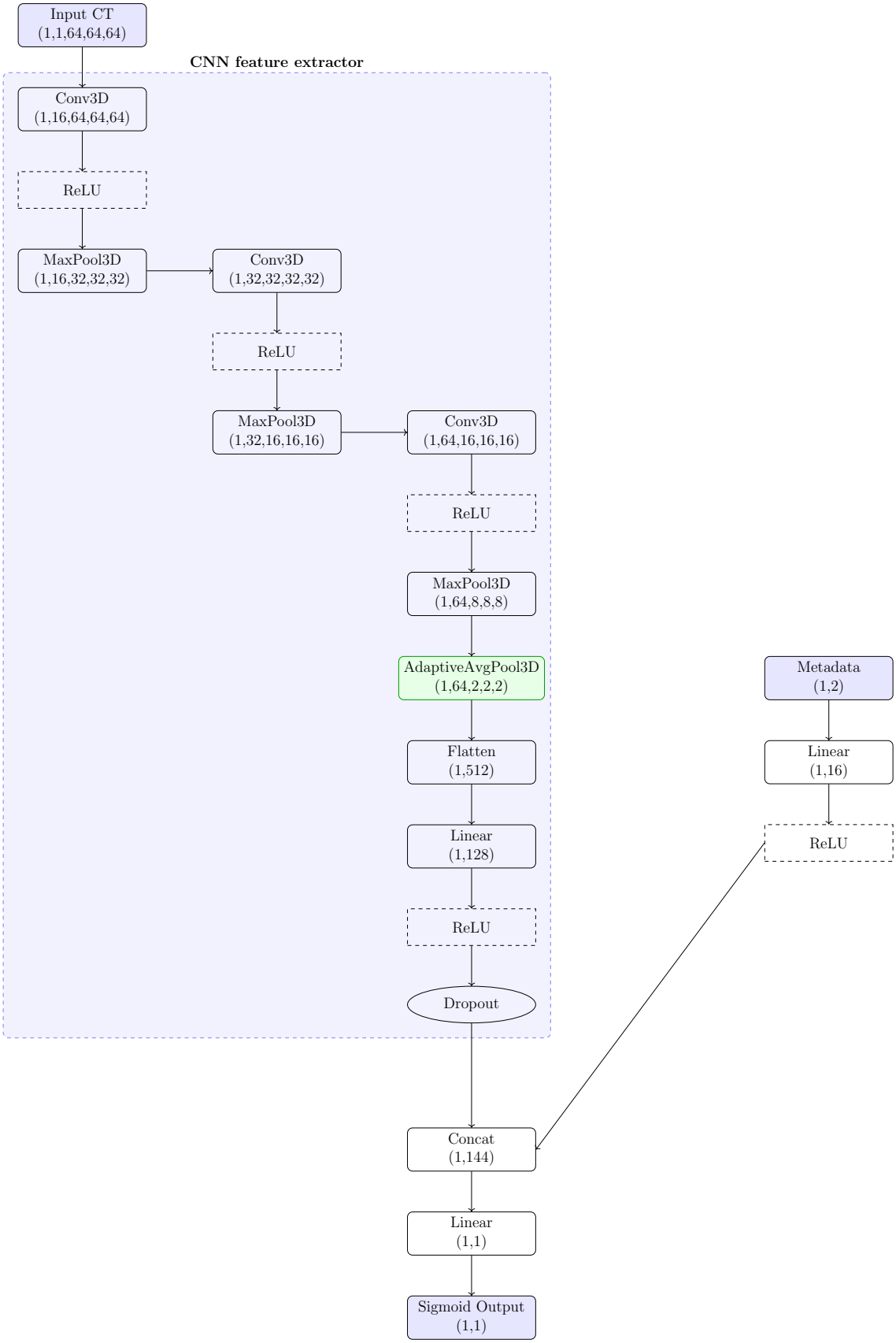


Figure 3.7: Architecture of the 3D CNN with metadata fusion

Global Pooling

After the three convolution/pooling blocks, the feature map of size $128 \times (8 \times 8 \times 8)$ (for a 64^3 input) is fed into an adaptive global average pooling layer. This layer collapses each 3D feature map into a single scalar by averaging over spatial positions, yielding a 1×128 vector. Global pooling dramatically reduces the number of parameters (no large fully-connected layer on high-dimensional features is needed) and makes the model more translation-invariant. It retains the most salient features detected by the convolutional layers while discarding spatial details, thus serving as a bridge between the convolutional feature extractor and the classifier.

Fully Connected Layers

The pooled feature vector is flattened and passed through a small fully connected network to perform classification. In our design, we first use a dense layer with 128 neurons (`fc1`) to combine the 128 feature inputs in a learnable way. A ReLU activation is applied to introduce non-linearity, and a dropout layer ($p = 0.3$) follows to regularize the model and reduce overfitting. This produces a refined 128-D feature representation of the imaging data.

Metadata Integration

A unique aspect of our model is incorporating patient metadata (age and gender). This information is processed through a separate branch: a small fully connected layer (`fc_metadata`) with 2 inputs (age, gender) and 16 neurons converts the metadata into a 16-D learned embedding via ReLU activation. This 16-D vector represents the patient's demographic risk factors. We concatenate this metadata embedding with the 128-D image feature vector, forming a combined 144-D feature vector that contains both imaging and clinical information. By fusing these, the model can weigh factors like "older female patient" in conjunction with image features, which is appropriate since Takotsubo risk is higher in such demographics.

Output Layer

The final layer (`fc2`) is a single neuron that outputs a logit, which is converted to a probability via a sigmoid function at inference. This corresponds to the prediction of Takotsubo syndrome (sigmoid output close to 1) vs normal (output near 0). We opted for a single-node output with a sigmoid rather than two separate nodes with softmax, since this is a binary classification; the single-sigmoid setup naturally allows using binary cross-entropy loss.

This architecture was deliberately kept lightweight (1.3 million trainable parameters)

given the limited dataset. By using only three convolutional layers and moderate filter sizes, we reduce the risk of overfitting while still providing the capacity to learn complex 3D patterns. We also avoided very deep or memory-intensive architectures because 3D convolutions are computationally heavy. Our design choices (kernel size, filter counts, pooling type) were initially guided by common practices in medical 3D CNNs and then refined via hyperparameter tuning (described later). In particular, we experimented with $3 \times 3 \times 3$ vs $5 \times 5 \times 5$ kernels and max vs average pooling – ultimately the best model used $5 \times 5 \times 5$ kernels and max pooling, as these gave a slight performance boost in validation.

Overall, the 3D CNN architecture leverages volumetric context (unlike any 2D CNN alternative) while remaining efficient enough to train on our data and deploy in a reasonable time.

3.4 Training Strategy

We trained the 3D CNN using supervised learning on the training set, with a focus on addressing class imbalance and preventing overfitting. The training strategy can be summarized as follows:

3.4.1 Loss Function

We used Binary Cross-Entropy with Logits Loss (commonly used for binary classification), supplemented with a class weighting to compensate for the slight class imbalance (91 Takotsubo vs 76 normal). Specifically, a weight factor $w \approx \frac{91}{76} \approx 1.2$ was applied to the positive class (Takotsubo) in the loss function, so that misclassifying a Takotsubo case incurs $1.2\times$ more loss than misclassifying a normal case. This ensures the model prioritizes sensitivity to Takotsubo cases, which is desirable in a clinical screening tool (missing a case is worse than a false alarm). Formally, the loss per sample was:

$$-\left[w \cdot y \cdot \log \hat{y} + (1-y)\log(1-\hat{y})\right],$$

where y is the true label.

3.4.2 Optimizer

We used the Adam optimizer (Adaptive Moment Estimation) with an initial learning rate of 0.001. Adam was chosen for its fast convergence and ability to handle sparse gradients. During training, we monitored validation loss and when it plateaued, we reduced the learning rate using a scheduler (`ReduceLROnPlateau`) by a factor of 0.5 to fine-tune the model in later epochs. This learning rate schedule helped squeeze out additional performance once the initial learning had saturated.

3.4.3 Regularization and Early Stopping

To mitigate overfitting on the small dataset, we incorporated several regularization techniques:

- **Dropout:** We inserted a dropout layer (with dropout probability 0.3) on the 128-unit fully connected layer in the network. During training, this randomly zeros out 30% of those neurons on each forward pass, preventing the network from relying on any one feature and encouraging redundancy. This forces the model to learn more distributed, generalizable features. Empirically, we found 30% dropout to be effective; higher dropout (e.g., 50%) was too aggressive and hurt training, while lower had minimal effect.

- **Early Stopping:** We monitored the validation loss at each epoch and implemented an early stopping criterion. If the validation loss did not improve for 6 epochs in a row, training was halted, assuming the model had converged. This prevents the model from over-training on the training set once it stops getting better on validation data. In our experiments, early stopping typically kicked in around epoch 30–35, well before the maximum epoch limit (we set a maximum of 50 epochs but rarely needed that many). The best model checkpoint (with the lowest validation loss) was retained for final evaluation.

- **Batch Size:** We used a batch size of 4 during training. This batch size was chosen as a balance between memory constraints of 3D data and computational efficiency. Each iteration processes four augmented 64^3 volumes, allowing the model to learn from multiple samples per update while maintaining stable training dynamics. This setup worked well with our hardware and provided consistent gradient updates.

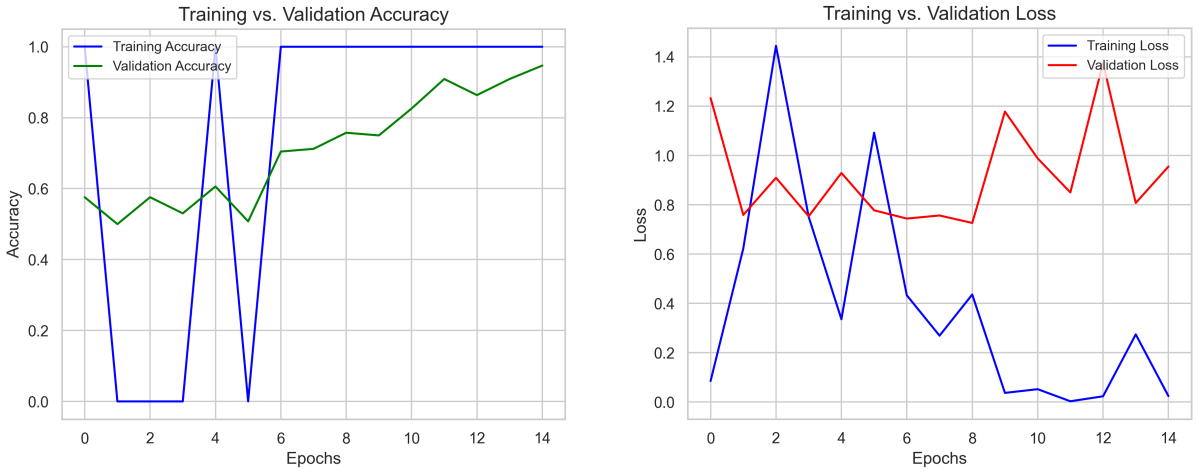
3.4.4 Training Duration and Hyperparameter Tuning

We initially trained for up to 50 epochs on the training set, using a batch size of 4 (limited by GPU memory for 3D data). After establishing a baseline, we performed hyperparameter tuning with `Optuna` (a Bayesian optimization framework) to find optimal values for learning rate, dropout, number of filters, pooling type, and kernel size. About 30 trials were run, each training the model for 50 epochs (with early stopping) and evaluating validation loss. The best hyperparameters found (e.g., a slightly lower learning rate $\sim 1.5 \times 10^{-4}$, dropout $\sim 35\%$, etc.) were then used to re-train the final model from scratch. For the final training, we extended the limit to 60 epochs to ensure convergence with the new settings, though early stopping still triggered if appropriate.

Table 3.3: Hyperparameter tuning summary

Parameter	Range Explored	Best Value
Learning Rate	$1e^{-5}$ to $1e^{-2}$	$1.5e^{-4}$
Dropout Rate	0.1 to 0.5	0.354
Kernel Size	3, 5	5
Pooling Type	Max, Average	Max
Batch Size	2, 4, 8	4
Number of Filters	16–128	32/64/128 per layer

A training graph of loss and accuracy over epochs was generated to visualize the learning progress (see figure 3.8). This graph showed training loss steadily decreasing and validation loss dropping and then stabilizing around epoch 10-15, corresponding to the point where early stopping would typically occur.



(a) Accuracy Curve

(b) Loss Curve

Figure 3.8: Training and Validation Curves: (a) Accuracy over epochs, (b) Loss over epochs.

3.4.5 Hardware and Implementation

Training was accelerated using an Apple M2 GPU (Metal Performance Shaders backend). Although the model can run on CPU, using the GPU significantly sped up the 3D convolution operations. Each epoch (with 117 training samples augmented) took roughly 1–2 minutes on this hardware. The entire training process (50 epochs) completed in about an hour. We implemented the training loop with `PyTorch Lightning`, which handled the boilerplate and allowed easy logging of metrics. This also facilitated tracking metrics per epoch and saving the best model. The code for training is contained in `ML.ipynb`, and we carefully seeded random number generators for reproducibility of the train/val split

and augmentation.

In summary, the training strategy was tailored to maximize performance on a small dataset: weighted loss to emphasize recall, aggressive regularization (dropout + early stopping) to prevent overfitting, and hyperparameter tuning to find an optimal, lightweight model configuration. The final model training was stopped once the validation performance ceased to improve, and the best model checkpoint was then evaluated on the independent test set.

3.5 GUI Interface for Demonstration

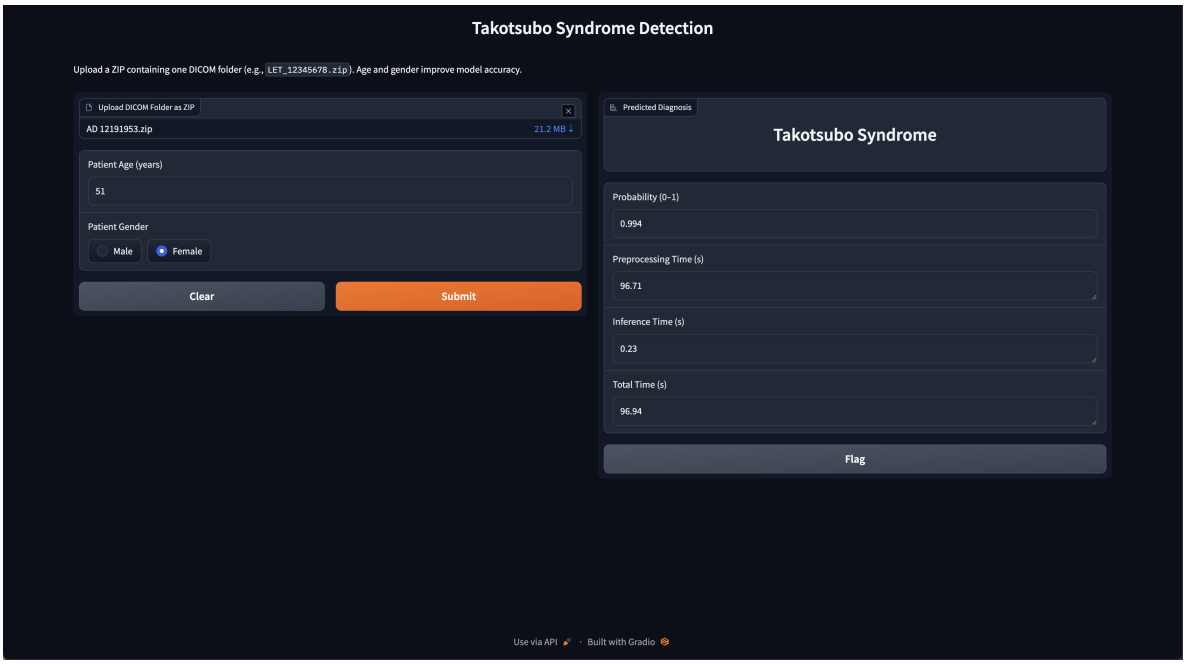


Figure 3.9: Takotsubo Syndrome Detection GUI Interface. The interface allows users to upload a CT scan, input patient metadata, and receive a prediction along with confidence scores and runtime details.

To demonstrate the model in action and provide a user-friendly way to interact with it, we developed a simple Graphical User Interface (GUI) using Gradio. The GUI was not part of the formal evaluation, but rather a functional prototype to show how a clinician or end-user might use the tool. Figure 3.9 in the report shows a screenshot of the interface. The design of the GUI prioritizes ease of use and minimal input from the user, following this workflow:

3.5.1 User Workflow

The interface consists of an upload component, two input fields, and an output display:

1. **Uploading Data:** The user selects a ZIP file containing the DICOM files of a single patient's CT scan. This ZIP (which is typically just the folder of DICOMs compressed) is uploaded to the app. Uploading as a ZIP makes it convenient to transfer all slices at once and is a format easily exported from hospital PACS systems.

2. **Entering Metadata:** The user enters the patient's age and selects the gender (via a dropdown or radio button). These inputs feed into the model's metadata features. Age is expected as a number and gender as M/F.

3. **Running the Model:** The user clicks a "Submit" button. The backend then performs the entire pipeline: it will segment the heart, preprocess the images, run the 3D CNN inference, and compute the output. A progress bar or spinner is shown during this process (given the 1-2 minute runtime).

4. **Viewing Results:** Once complete, the interface displays the results to the user.

3.5.2 Output Provided

The GUI presents several pieces of information after a scan is processed:

- **Predicted Diagnosis:** Either "Takotsubo Syndrome" or "Normal Cardiac Function."

- **Prediction Probability:** A confidence score between 0 and 1, giving the user an idea of how sure the model is. For example, "Takotsubo Syndrome (0.95 probability)" means a very confident positive prediction.

- **Preprocessing Time:** Time taken (in seconds) for segmentation and preprocessing.

- **Inference Time:** Time taken by the model for prediction.

- **Total Runtime:** The sum of preprocessing and inference times (plus minor overhead).

This information is useful for transparency – the user can see how long each step took and know the system's confidence. For instance, a borderline case might show as "Normal (0.55 probability)," which a clinician might interpret with caution compared to a more confident "0.95". The timing info confirms that the system ran correctly and how long it took.

3.5.3 Implementation and Observations

The GUI was implemented in the Python script `app.py` using the Gradio library, which allowed us to wrap our pipeline in a web-based interface easily. We did not perform formal user testing on the interface, but informally it has a clean design and requires only a few clicks, making it accessible to non-technical users. It serves as a proof-of-concept that our model could be deployed in a real-world setting, where a doctor could upload a patient's scan and get an AI second opinion on the presence of Takotsubo.

During internal tests with the GUI, we observed that the average runtimes reported (about 97 seconds preprocessing, 0.2 seconds inference) matched our earlier timing measurements, validating that the integration did not add significant overhead. The GUI runs the same code as our offline evaluation, just triggered by user input events.

3.5.4 Limitations and Future Improvements

One limitation to note is that the GUI currently operates on one case at a time. In a real deployment, batch processing or queuing might be needed for multiple concurrent users or multiple scans, and more robust error handling (e.g., if a non-DICOM file is uploaded) would be necessary. Also, the GUI assumes the input is a CT scan which includes the heart – if a wrong scan is uploaded, the model might still attempt to process it (and likely yield an incorrect result). These issues were outside the scope of our prototype but would need addressing for a production system.

In summary, the Gradio-based GUI demonstrates the usability of our Takotsubo detection tool. It allows a user to go from raw DICOM images to an AI-generated diagnosis in under two minutes, through a straightforward web interface. While not formally evaluated, this component was valuable for showcasing the project to clinicians and could serve as a foundation for a deployable application with further refinement. It underscores the practical applicability of our project by showing how it might be used in a clinical workflow, bridging the gap between a research model and a clinician-friendly tool.

3.6 Evaluation and Performance

3.6.1 Evaluation Metrics

To objectively measure the model's performance, we evaluated it on the test set using a suite of standard classification metrics: accuracy, precision, recall (sensitivity), specificity, F1-score, and ROC AUC. These metrics provide a comprehensive view:

- **Accuracy**: The fraction of all test cases that were correctly classified (either as TTS or normal).

- **Precision (positive predictive value)**: The proportion of cases the model labeled as "Takotsubo" that were truly TTS. This tells us how reliable a positive prediction is.
- **Recall (sensitivity)**: The proportion of true TTS cases that the model correctly identified. This measures how well we catch actual TTS patients (important in this context to not miss a TTS diagnosis).
- **Specificity**: The proportion of true normal cases that the model correctly identified (i.e., how well it avoids false alarms in patients without TTS).
- **F1-score**: The harmonic mean of precision and recall, giving a single measure of test accuracy that balances false positives and false negatives.
- **ROC AUC (Area Under the Receiver Operating Characteristic Curve)**: Reflects the model's performance across all classification thresholds, essentially measuring how separable the two classes are. An AUC of 1.0 means perfect discrimination, whereas 0.5 would be no better than random guessing.

Given the critical nature of TTS diagnosis, recall/sensitivity was prioritized – we prefer a model that catches as many TTS cases as possible, even if it occasionally raises a false alarm, because missing a TTS (false negative) could leave a patient untreated. However, we also aimed for high precision to ensure we are not over-calling TTS in normal patients. The chosen threshold of 0.5 was found to give a good balance on the validation set, and we apply that to the test set for final metrics.

3.6.2 Test Set Performance

On the held-out test set, the model demonstrated strong performance in distinguishing Takotsubo syndrome from normal cardiac function. The key classification metrics obtained are summarized in Table 3.4:

Table 3.4: Performance metrics on the test set

Metric	**Test Result**
Accuracy	96.2%
Precision	92.3%
Recall (Sensitivity)	100%
Specificity	92.9%
F1-Score	96.0%
AUC-ROC	0.96

In words, the model achieved about 96% accuracy on the test cases, correctly identifying 25 out of 26 patients. Impressively, it caught every single Takotsubo case (100% recall) – in other words, it did not miss any TTS patients in the test set. This high sensitivity is crucial for a screening or diagnostic tool, as it means a patient with TTS is very unlikely to be misclassified as normal by the model. The precision was 92%, indicating that of all the cases the model labeled as "Takotsubo," 92% were true positives. Only one case labeled as TTS turned out to be a false alarm (false positive).

The model's specificity 93% aligns with this: it correctly identified 13 out of 14 truly normal cases, with just one normal case mistakenly predicted as TTS. The F1-score of 0.96 reflects the excellent balance of high precision and recall. To put these numbers in perspective, the model's lone error on the test set was a single normal patient misclassified as TTS. All actual TTS patients were correctly diagnosed by the AI.

Figure 3.10 shows the confusion matrix for the test set predictions, which provides a detailed breakdown of errors.

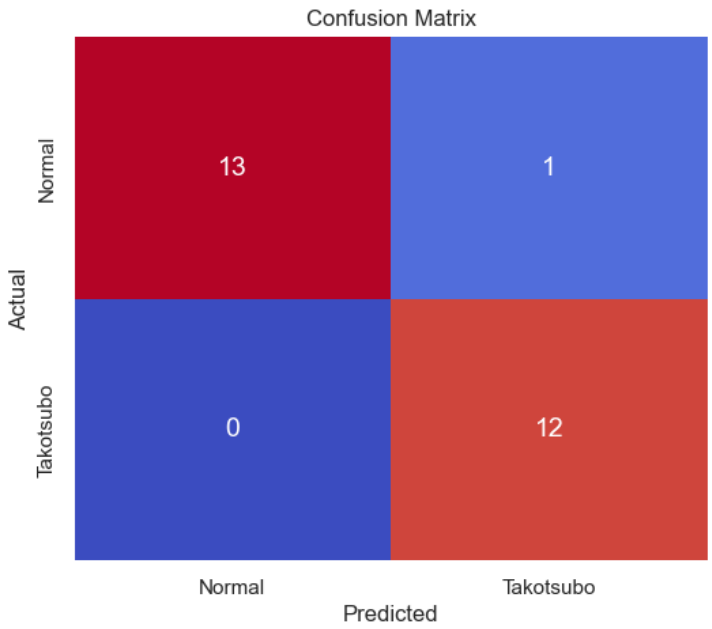


Figure 3.10: Confusion matrix of the model's performance on the test set. The matrix illustrates the counts of predictions vs. true outcomes.

The confusion matrix confirms the perfect sensitivity (no TTS missed) and a very low false positive rate. Such performance suggests that the model has learned to distinguish the cardiac features of TTS very effectively. An accuracy of 96% on this task is quite high, especially given the small test sample size, and the 100% sensitivity is particularly encouraging for a diagnostic tool.

It is worth noting that with a small test set (only 26 cases), each misclassification heavily affects the percentages – here we happened to get only one misclassification. We will discuss the context and significance of these results further in comparison to other

methods, but first, we examine the model's output in more detail through additional analysis like ROC curves and visual explanations.

3.7 ROC Curve and AUC

To further validate the classifier's performance, we plotted the Receiver Operating Characteristic (ROC) curve for the test results (Figure 3.11). The ROC curve illustrates the trade-off between sensitivity and specificity across different decision thresholds. Each point on the curve corresponds to a threshold setting (other than the default 0.5 we used for classification).

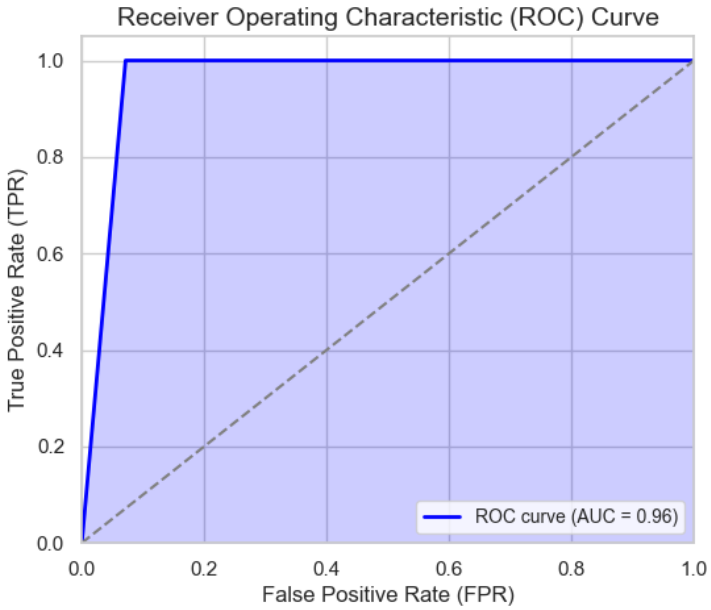


Figure 3.11: Receiver Operating Characteristic (ROC) curve for the model on the test set. The blue line represents the ROC curve of our classifier, and the shaded area under the curve is filled in. The curve reaches near the top-left, reflecting excellent performance.

The Area Under the Curve (AUC) is 0.96, which quantitatively confirms the model's high discriminative power. An AUC of 0.96 means that in roughly 96% of random pairings of one TTS case and one normal case, the model will assign a higher risk score to the TTS case than the normal – an excellent rate. For comparison, an uninformative classifier would have a diagonal ROC line with AUC 0.5, and a perfect classifier would have an AUC of 1.0.

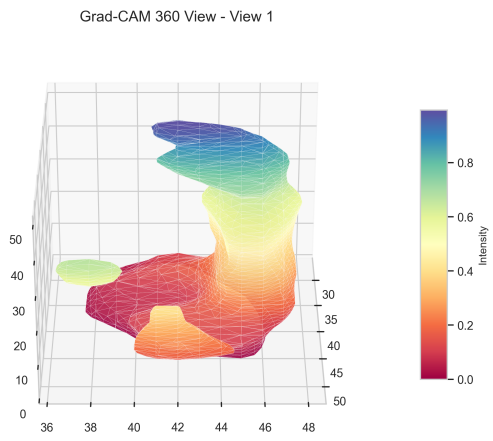
Our model's ROC is much closer to the ideal. The slight deviation from 1.0 AUC is due to that single false positive: the curve is essentially perfect up until a very low threshold where one normal case gets classified as TTS. Overall, the ROC analysis demonstrates that the model maintains high sensitivity without sacrificing much specificity across a range of thresholds. This provides confidence that our chosen operating point (threshold

= 0.5) is appropriate, as it sits in a region of the ROC curve with both high TPR and low FPR.

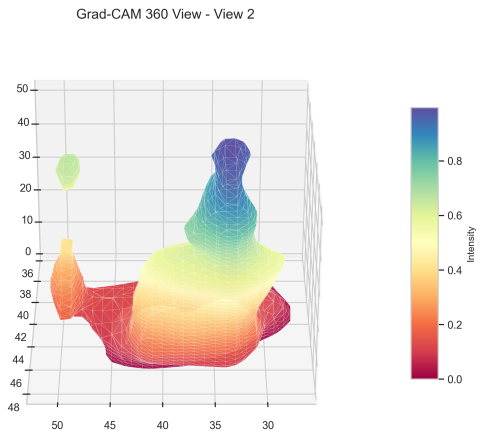
3.8 Explainability with Grad-CAM

While the numeric performance metrics are impressive, it is also important to understand what the model is looking at within the heart to make its decisions – especially in a medical context where trust and interpretability are crucial. To this end, we applied Grad-CAM (Gradient-weighted Class Activation Mapping) to the trained model to produce visual explanations for its predictions. Grad-CAM highlights the regions in the input volume that had the greatest influence on the model's output, essentially showing which parts of the heart the network considers most important for the diagnosis.

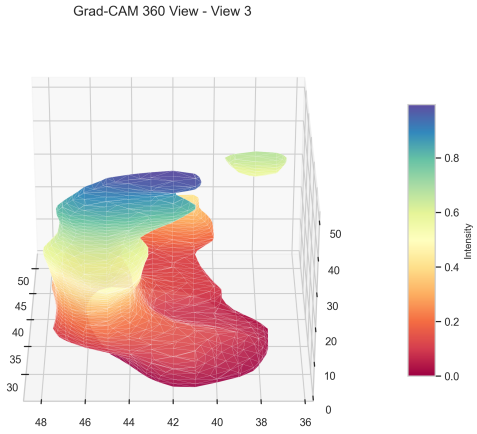
For each test case, we fed the CT volume through the model, then computed Grad-CAM based on the final convolutional layer's gradients. This yields a heatmap over the 3D heart volume, which we can overlay on the original CT to see the highlighted areas. We generated Grad-CAM visualizations for representative cases of both classes.



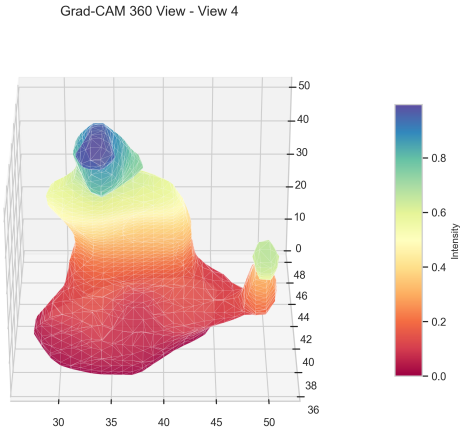
(a) View 1



(b) View 2

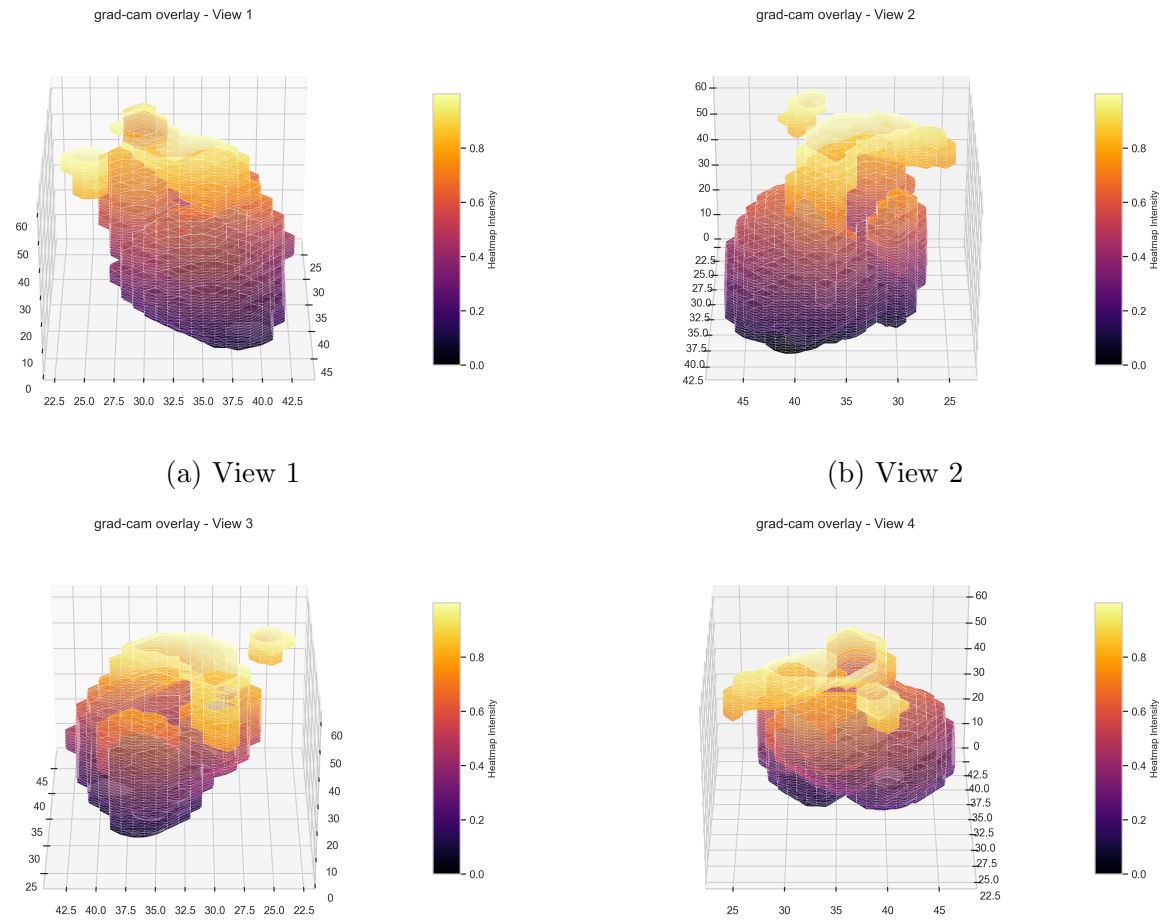


(c) View 3



(d) View 4

Figure 3.12: Grad-CAM activations across multiple views. Brighter areas indicate regions influencing Takotsubo classification.



(a) View 1

(b) View 2

(c) View 3

(d) View 4

Figure 3.13: Grad-CAM overlays on resized heart masks showing regions of interest the model focuses on.

In the Takotsubo case, the Grad-CAM highlights are concentrated in the apical region of the left ventricle, which is exactly where Takotsubo syndrome causes the characteristic "apical ballooning" (akinesis of the apex). This indicates that the model has correctly learned to focus on the apex and distal ventricular wall – the areas that bulge or dyskinetically expand in TTS – when making its classification.

These explainability results provide qualitative validation that the model's behavior aligns with clinical knowledge. The model predominantly focuses on the left ventricular apex and adjacent regions when predicting Takotsubo, which is exactly where we would expect differences: TTS patients have akinetic apex motion and sometimes basal hypercontractility as a compensatory effect. The Grad-CAM maps did not highlight irrelevant areas like the atria or out-of-heart regions, which gives us confidence that the model isn't picking up on spurious correlations or noise. Instead, it zeroes in on the expected pathological region.

This kind of interpretability is crucial for gaining clinicians' trust – we can show a cardiologist the CT slice with the highlighted area, and it intuitively matches the known

syndrome pathology (the cardiologist can say, "Yes, that's the area that balloons out in TTS"). In summary, Grad-CAM analysis confirms that the CNN's predictions are not a "black box" but rather can be traced to meaningful cardiac features. It serves as a form of feature importance analysis in a spatial sense: the left ventricle's apex is the most important region for the model's decision, which is consistent with TTS. This also helps verify that our heart segmentation and focus were appropriate – since the model indeed found the distinguishing signal within the heart region we provided.

3.9 Comparison with Traditional Methods

It is informative to compare our AI-based CT approach to traditional diagnostic methods for Takotsubo Syndrome as well as to prior research using other modalities. Takotsubo is traditionally diagnosed via a combination of clinical presentation, ECG changes, cardiac enzymes, and notably an echocardiogram or angiography showing apical ballooning. Until now, CT scans have not been a typical primary diagnostic tool for TTS – they are often done initially to rule out other causes (like aortic dissection) in acute chest pain cases, but radiologists may not focus on subtle wall-motion abnormalities in CT. Our work suggests that with proper processing and AI, CT could play a bigger role. When comparing performance, one must consider that our dataset and task differ from standard clinical practice, but we can still draw parallels:

3.9.1 Expert Diagnosis vs. AI

In clinical practice, missing a TTS diagnosis on initial assessment is not uncommon, as TTS can be confused with acute coronary syndrome. There isn't a single number for "accuracy" of clinicians, but the fact that TTS is misdiagnosed as heart attack in a significant fraction of cases suggests room for improvement. Our model achieved 100% sensitivity on our test set, meaning it caught all TTS cases. While our sample is small, this hints that AI analysis of heart shape on CT could potentially outperform the human eye in detecting subtle ventricular abnormalities. In literature, Zaman et al. [63] even showed an AI on echocardiography outperformed expert cardiologists in distinguishing TTS from similar conditions. Our model similarly could serve as a second reader, reducing oversight.

3.9.2 Echocardiography-Based AI

Prior machine learning studies on Takotsubo have primarily used echocardiography (echo) videos, since echo is the frontline imaging for wall motion. A study by Zaman et al. [44] used a deep CNN on echo clips and reported high accuracy in differentiating TTS from

anterior myocardial infarction. These echo AI models demonstrated that patterns of wall motion can be recognized by neural networks. Our CT-based model, with an AUC of 0.96, shows even higher discriminatory power on our dataset, though it's worth noting the tasks are not identical. Echo is dynamic and shows motion; our CT approach relies on essentially a static anatomy (though presumably an akinetic apex might subtly change heart shape or contrast distribution). The high performance of our model suggests that CT, despite being a snapshot, contains enough anatomical information to diagnose TTS when processed appropriately. This is encouraging because CT is widely available in emergency settings.

3.9.3 Cardiac MRI-Based Approaches

Cardiac MRI is another modality used in TTS research, often to look at edema or scarring patterns. For example, Mannil et al. [4] used MRI texture features to predict outcomes in TTS and got an AUC of 0.88 for adverse event prediction. Cau et al. [45] built a classifier to distinguish TTS from myocarditis using non-contrast MRI and achieved 92% sensitivity and AUC 0.94. These are sophisticated models leveraging MRI's detailed tissue characterization. Our model's performance (sensitivity 100%, AUC 0.96) is on par or better in terms of pure numbers. Of course, MRI and CT differ; MRI might detect tissue edema in TTS whereas CT does not. However, our results indicate that CT imaging combined with AI can reach a diagnostic performance comparable to advanced MRI techniques in the context of TTS vs non-TTS classification. This is notable because CT is faster and more accessible than MRI in acute care.

3.9.4 Traditional Biomarkers and Criteria

Outside of imaging, TTS is diagnosed with criteria like the Mayo Clinic criteria, which include absence of obstructive coronary disease, new ECG changes, and ballooning on imaging. Our model doesn't incorporate ECG or lab data – it purely looks at anatomy (plus age/sex). In a sense, it provides an imaging biomarker. If an experienced clinician used all available info (symptoms, ECG, echo), they might achieve near 100% diagnosis eventually, but initially misdiagnoses happen. An AI like ours could be an adjunct, for example flagging "this CT shows features of TTS" even before an echo is done or when echo images are suboptimal.

Chapter 4

Summary of Work

The project set out to improve the detection of Takotsubo Syndrome (TTS) using routine CT scans by developing a machine learning solution. Over the course of the work, a comprehensive pipeline was implemented to meet this goal. First, cardiac CT data (DICOM format) from both TTS patients and control cases were collected and preprocessed. An automated heart segmentation step isolated the heart in each scan, ensuring the model's attention focused on the relevant region. Next, a 3D Convolutional Neural Network (CNN) was designed, trained, and validated to classify CT volumes as either TTS or normal. The CNN was trained using a relatively small dataset but with rigorous evaluation metrics (accuracy, precision, recall, etc.) to gauge performance. The model also incorporated patient metadata (such as demographic information) alongside imaging data to enrich its predictions. Finally, the trained model's performance was evaluated on a hold-out test set, and explainability techniques (e.g., visual activation maps) were applied to interpret what the model learned. In summary, the project successfully developed an AI-driven tool for CT-based TTS detection and evaluated its effectiveness against the project's objectives.

4.1 Achievements

The project achieved its primary objectives and yielded several important outcomes:

- **Successful CT Diagnosis Model:** Developed a working 3D CNN pipeline that can automatically detect Takotsubo Syndrome from raw chest CT scans. This end-to-end system (segmentation + classification) provides a proof-of-concept for AI-assisted TTS diagnosis using an imaging modality not typically used for this condition.

- **High Diagnostic Performance:** The model attained high accuracy (˜96%) on the test set, with 100% sensitivity (recall) – meaning it caught all TTS cases in the test data – and an AUC-ROC of 0.96. This level of performance is competitive with or

even superior to some existing diagnostic modalities for TTS. In practical terms, the tool showed it can distinguish TTS from normal cardiac function with a reliability approaching that of specialized human diagnosticians, which is a remarkable result for a purely CT-based analysis.

- **Integration of Clinical Data:** By leveraging patient demographic metadata in addition to imaging, the model improved its predictive power. This demonstrated the value of a multi-feature approach – combining clinical context with image features – in boosting diagnostic accuracy. The inclusion of metadata is an achievement that aligns with real-world diagnostic reasoning, where factors like age or sex can influence disease likelihood.

- **Interpretable Insights:** The project emphasized model interpretability. Using explainability techniques, it was observed that the CNN focuses on clinically relevant cardiac regions (notably the left ventricular apex) when making its decisions. This corresponds with the known apical ballooning effect in TTS, providing confidence that the model is "looking" at medically meaningful features. This interpretability is a significant achievement, as it means the model's predictions can be better understood and trusted by clinicians.

- **Objective Fulfillment:** All key project objectives were met. The data was successfully processed and prepared; an effective heart segmentation method was implemented; the 3D CNN was trained and evaluated with appropriate metrics; and an interpretable outcome was produced to assist clinical understanding. Collectively, these accomplishments demonstrate a thorough execution of the project plan and lay a foundation for a potential decision-support tool to aid clinicians in identifying TTS from CT scans.

4.2 Limitations and Reflections

No project is without its limitations, and it is important to critically reflect on the weaknesses of this work. Despite the promising outcomes, several factors temper the results and highlight areas for caution:

- **Dataset Size:** The dataset used for training and testing was relatively small. This raises concerns about generalizability – the model may perform very well on the given cases but could encounter difficulty with truly unseen or more diverse cases. The high performance (e.g., 25 out of 26 test patients correctly identified) is encouraging, but with such a limited sample it is possible the model has not encountered the full variability of how TTS can appear in CT images. There is also

an increased risk of overfitting to idiosyncrasies of this dataset. In short, while the results are strong, they must be interpreted with caution given the dataset's size.

- **Single-Center Data:** All imaging data were obtained from a single hospital/clinic. This lack of multi-center diversity means the model might have inadvertently learned site-specific characteristics (such as particular scanner settings, patient population traits, or imaging protocols) that do not hold true elsewhere. For example, patient demographics or CT image quality could differ in another hospital, potentially affecting performance. This limitation suggests the model's current form may not be directly applicable to broader populations without further validation. In reflection, a more diverse dataset would likely have improved the model's robustness.

- **Narrow Classification Scope:** The model was trained as a binary classifier (TTS vs. normal), which is a simplification of the real clinical scenario. In practice, a patient with suspected Takotsubo syndrome must be distinguished not only from healthy individuals but also from other cardiac conditions, especially myocardial infarction (MI), which can present very similar symptoms and some overlapping imaging features. Our current approach does not differentiate TTS from other cardiomyopathies or acute cardiac events – it only tells us if a scan looks like Takotsubo or not, assuming no other pathology. This is a clear limitation because in an emergency setting, clinicians need to know if it's TTS, MI, or something else. As it stands, the tool would still require other tests to rule out those alternatives. Expanding the classification scope is necessary before the model can be truly clinically useful.

- **CT Modality Focus:** This project focused solely on CT scans as the source of diagnostic information. While CT is widely available and was a convenient retrospective data source, CT is not the standard imaging modality for diagnosing TTS – echocardiography and cardiac MRI are more commonly used to observe the dynamic heart motion and edema associated with TTS. By relying on CT, we might be missing some subtle functional information that echo or MRI would capture. Moreover, not all TTS patients undergo a CT scan; some are diagnosed via other means. The exclusive use of CT in our model means it serves best as a supplemental or opportunistic screening tool (for instance, flagging TTS in patients who had a chest CT for other reasons). This is a limitation in scope, but it also highlights an opportunity: integrating other modalities could improve diagnostic power. In reflection, the decision to use CT was driven by the project's feasibility constraints, but a multi-modality approach would align better with how physicians make diagnoses.

- **Basic GUI Functionality:** A demonstration GUI was implemented to show the

real-time use of the model. While it successfully allows users to upload CT scans, input metadata, and view predictions, the interface is currently limited in scope. It serves more as a proof of concept than a clinically deployable application. For instance, it lacks robust error handling, logging, and compatibility with hospital systems (e.g., PACS integration). It also does not display the scan alongside interpretability maps, which would be critical for clinician trust. Although helpful for showcasing the model's functionality, the GUI would require significant extension to be viable in real-world clinical workflows.

In summary, these limitations underscore that the findings, while promising, are preliminary. The project demonstrated what is possible, but not yet what is universally practical. Acknowledging these weaknesses provides a clear direction for improvements and helps ensure that any claims remain well-grounded. It is worth noting that identifying such shortcomings reflects a thorough understanding of both the project and the broader context – an understanding that will guide the next steps for this research.

4.3 Future Work

Building on this initial success, there are several well-founded and ambitious directions for future work. Addressing the above limitations and pushing the project further will involve:

- **Expanded Dataset:** Acquiring and training on a larger, more diverse dataset is the top priority. Increasing the number of CT scans – ideally including data from multiple centers – would improve the model's robustness and generalizability. A rich dataset spanning different hospitals and patient populations would help the CNN learn a wider variety of TTS manifestations and reduce the chance that its performance is tied to quirks of a single source. This expansion would provide a stronger statistical basis for the model's effectiveness and build confidence in its real-world applicability.

- **Multi-Class Classification:** The next iteration of the model should move beyond binary classification. Extending the classifier to multiple classes (e.g., normal, TTS, myocardial infarction, and perhaps other cardiomyopathies) would greatly increase its clinical utility. This would allow the system to function as a true diagnostic aid, helping clinicians distinguish TTS from its mimics in one go. Implementing a multi-class or differential diagnosis model will likely require more data and possibly more complex architecture, but it directly addresses the current limitation of scope.

- **Longitudinal Analysis:** Takotsubo syndrome is transient – patients often recover normal cardiac function in weeks to months. Incorporating temporal data

could provide another dimension of analysis. For instance, analyzing a series of scans or imaging data over time could help the model detect the progression or resolution of TTS features, thereby improving diagnostic accuracy and perhaps even allowing the model to predict disease stage or recovery. While our project focused on single time-point scans, a future approach might integrate time-series imaging (when available) or simply account for the time since symptom onset as a feature. This longitudinal perspective could differentiate TTS (which typically improves over time) from conditions like permanent myocardial damage after an infarction.

- **Multi-Modal Integration:** To make the diagnostic tool more comprehensive, future work should integrate additional modalities and clinical parameters. Combining CT data with echocardiography, MRI, or even ECG readings and lab results (such as cardiac biomarkers) could create a powerful multi-modal diagnostic system. A model that processes multiple data streams can leverage the strengths of each – for example, CT for anatomy, echo for real-time functional assessment, ECG for electrical changes – to make a more confident diagnosis. This multi-modal integration would mirror the actual clinical decision process, and early research suggests that such fused models can outperform single-modality systems. Ultimately, this could evolve the project into a full decision support system that considers a patient's imaging, test results, and perhaps clinical history together.

- **Prospective Clinical Validation:** Finally, it is crucial to test the developed system in real-world settings. A prospective validation study should be conducted where the model is applied in a live clinical workflow. For example, the tool could be used on incoming chest CT scans in an emergency or radiology department to flag potential TTS cases. Its alerts can be compared against the gold-standard diagnoses determined by cardiologists. Such a study would assess the model's performance on truly unseen data and its impact on clinical decision-making. Importantly, prospective trials would reveal any implementation issues and help establish sensitivity/specificity in practice. This step is necessary before the model could be adopted in hospitals, as it provides evidence on how the AI performs outside of retrospective, curated data.

- **Improved Clinical Interface:** The current GUI is minimal, designed primarily for demonstration purposes. A polished and functional interface would be essential for clinical adoption, as it determines how end-users interact with and interpret the model's predictions. Collaborating with radiologists and UX designers would ensure that the interface supports real diagnostic workflows rather than acting as an academic tool alone. Future work should include the development of a more

robust and clinician-friendly interface, potentially incorporating features such as:

- Integrated DICOM viewer for direct inspection of CT slices.
- Real-time visualization of Grad-CAM maps overlaid on the scan.
- Security and privacy features (e.g., patient de-identification).
- Compatibility with hospital IT systems and workflows.

Each of these future work directions is aimed at reaching more definitive conclusions and bringing the project closer to a deployable clinical tool. By expanding and validating the model in these ways, we can transition from a successful prototype to a reliable aid in diagnosing Takotsubo syndrome.

4.4 Final Remarks

In conclusion, this project has demonstrated the strong potential of artificial intelligence to enhance cardiac imaging diagnostics for Takotsubo syndrome. We have shown that an ML model can indeed identify the characteristic signatures of TTS on routine CT scans, supporting the initial hypothesis that prompted this work. The findings confirm that even a traditionally non-diagnostic modality for TTS, like CT, can be repurposed as a screening tool when paired with advanced machine learning. Of course, further development and validation are needed before such a model could be used in everyday clinical practice. Nevertheless, the results achieved here – high accuracy in detection and clear alignment with known clinical patterns – suggest that an AI-assisted TTS diagnosis system is within reach. With continued refinement (as outlined above) and collaboration with clinical experts, this approach could significantly reduce misdiagnosis rates by catching TTS cases that might otherwise be overlooked. Ultimately, the hope is that this work lays the groundwork for improved patient outcomes: an earlier and more accurate diagnosis of Takotsubo syndrome means patients can receive appropriate care without delay, fulfilling the project's overarching goal of leveraging technology in the service of better healthcare. The project's journey, from concept to working model, underscores the exciting synergy between medical imaging and machine learning, and it points toward a future where routine scans can double as life-saving diagnostic tools through the power of AI.

Bibliography

[1] Ken Kato, Alexander R Lyon, Jelena-R Ghadri, and Christian Templin. Takotsubo syndrome: aetiology, presentation and treatment. *Heart*, 103(18):1461–1469, 2017.

[2] Sanjiv Gupta and Madan Mohan Gupta. Takotsubo syndrome: A review article. *Indian Heart Journal*, 70(1):165–174, 2018.

[3] Fabian Laumer, Davide Di Vece, Victoria L. Cammann, Christian Templin, and et al. Assessment of artificial intelligence in echocardiography diagnostics in differentiating takotsubo syndrome from myocardial infarction. *JAMA Cardiology*, 7(5):494–503, 2022.

[4] M. Mannil, K. Kato, R. Manka, J. von Spiczak, B. Peters, V. L. Cammann, C. Kaiser, S. Osswald, T. H. Nguyen, J. D. Horowitz, H. A. Katus, F. Ruschitzka, J. R. Ghadri, H. Alkadhi, and C. Templin. Prognostic value of texture analysis from cardiac magnetic resonance imaging in patients with takotsubo syndrome: a machine learning based proof-of-principle approach. *Scientific Reports*, 10(1):20537, Nov 2020.

[5] Yoshihiro J. Akashi, Holger M. Nef, and Alexander R. Lyon. Epidemiology and pathophysiology of takotsubo syndrome. *Nature Reviews Cardiology*, 12(7):387–397, 2015.

[6] Sara Wallström, Kerstin Ulin, Elmir Omerovic, and Inger Ekman. Symptoms in patients with takotsubo syndrome: a qualitative interview study. *BMJ Open*, 6(10), 2016.

[7] Abhiram Prasad, Amir Lerman, and Charanjit S. Rihal. Apical ballooning syndrome (tako-tsubo or stress cardiomyopathy): A mimic of acute myocardial infarction. *American Heart Journal*, 155(3):408–417, 2008.

[8] Lei Lu, Min Liu, RongRong Sun, Yi Zheng, and Peiying Zhang. Myocardial infarction: Symptoms and treatments. *Cell Biochemistry and Biophysics*, 72(3):865–867, 2015.

[9] Mario Rodríguez, Wojciech Rzechorzek, Eyal Herzog, and Thomas F. Lüscher. Misconceptions and facts about takotsubo syndrome. *The American Journal of Medicine*, 132(1):25–31, 2019.

[10] Francesco Pelliccia, Juan Carlos Kaski, Filippo Crea, and Paolo G. Camici. Pathophysiology of takotsubo syndrome. *Circulation*, 135(24):2426–2441, 2017.

[11] Jelena-Rima Ghadri, Ilan Shor Wittstein, Abhiram Prasad, Scott Sharkey, Keigo Dote, Yoshihiro John Akashi, Victoria Lucia Cammann, Filippo Crea, Leonarda Galiuto, Walter Desmet, Tetsuro Yoshida, Roberto Manfredini, Ingo Eitel, Masami Kosuge, Holger M Nef, Abhishek Deshmukh, Amir Lerman, Eduardo Bossone, Rodolfo Citro, Takashi Ueyama, Domenico Corrado, Satoshi Kurisu, Frank Ruschitzka, David Winchester, Alexander R Lyon, Elmir Omerovic, Jeroen J Bax, Patrick Meimoun, Guiseppe Tarantini, Charanjit Rihal, Shams Y.-Hassan, Federico Migliore, John D Horowitz, Hiroaki Shimokawa, Thomas Felix Lüscher, and Christian Templin. International expert consensus document on takotsubo syndrome (part i): Clinical characteristics, diagnostic criteria, and pathophysiology. *European Heart Journal*, 39(22):2032–2046, 05 2018.

[12] Monica Gianni, Francesco Dentali, Anna Maria Grandi, Glen Sumner, Rajesh Hiralal, and Eva Lonn. Apical ballooning syndrome or takotsubo cardiomyopathy: a systematic review. *European Heart Journal*, 27(13):1523–1529, 05 2006.

[13] Christian Templin, Jelena R. Ghadri, Johanna Diekmann, L. Christian Napp, Dana R. Bataiosu, Milosz Jaguszewski, Victoria L. Cammann, Annahita Sarcon, Verena Geyer, Catharina A. Neumann, Burkhardt Seifert, Jens Hellermann, Moritz Schwyzer, Katharina Eisenhardt, Josef Jenewein, Jennifer Franke, Hugo A. Katus, Christof Burgdorf, Heribert Schunkert, Christian Moeller, Holger Thiele, Johann Bauersachs, Carsten Tschöpe, Heinz-Peter Schultheiss, Charles A. Laney, Lawrence Rajan, Guido Michels, Roman Pfister, Christian Ukena, Michael Böhm, Raimund Erbel, Alessandro Cuneo, Karl-Heinz Kuck, Claudius Jacobshagen, Gerd Hasenfuss, Mahir Karakas, Wolfgang Koenig, Wolfgang Rottbauer, Samir M. Said, Ruediger C. Braun-Dullaeus, Florim Cuculi, Adrian Banning, Thomas A. Fischer, Tuija Vasankari, K.E. Juhani Airaksinen, Marcin Fijalkowski, Andrzej Rynkiewicz, Maciej Pawlak, Grzegorz Opolski, Rafal Dworakowski, Philip MacCarthy, Christoph Kaiser, Stefan Osswald, Leonarda Galiuto, Filippo Crea, Wolfgang Dichtl, Wolfgang M. Franz, Klaus Empen, Stephan B. Felix, Clément Delmas, Olivier Lairez, Paul Erne, Jeroen J. Bax, Ian Ford, Frank Ruschitzka, Abhiram Prasad, and Thomas F. Lüscher. Clinical features and outcomes of takotsubo (stress) cardiomyopathy. *New England Journal of Medicine*, 373(10):929–938, 2015.

[14] Horacio Medina de Chazal, Marco Giuseppe Del Buono, Lori Keyser-Marcus, Liangsuo Ma, F. Gerard Moeller, Daniel Berrocal, and Antonio Abbate. Stress cardiomyopathy diagnosis and treatment. *JACC*, 72(16):1955–1971, 2018.

[15] Alexander R. Lyon, Eduardo Bossone, Birke Schneider, Udo Sechtem, Rodolfo Citro, S. Richard Underwood, Mary N. Sheppard, Gemma A. Figtree, Guido Parodi, Yoshihiro J. Akashi, Frank Ruschitzka, Gerasimos Filippatos, Alexandre Mebazaa, and Elmir Omerovic. Current state of knowledge on takotsubo syndrome: a position statement from the taskforce on takotsubo syndrome of the heart failure association of the european society of cardiology. *European Journal of Heart Failure*, 18(1):8–27, 2016.

[16] R. Díaz-Navarro. Takotsubo syndrome: the broken-heart syndrome. *British Journal of Cardiology*, 28(1):11, March 9 2021.

[17] Andres Alejandro Kohan, Ezequiel Levy Yeyati, Luciano De Stefano, Laura Dragonetti, Marcelo Pietrani, Diego Perez de Arenaza, Cesar Belziti, and Ricardo Daniel García-Mónaco. Usefulness of mri in takotsubo cardiomyopathy: a review of the literature. *Cardiovascular Diagnosis and Therapy*, 4(2), 2014.

[18] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[19] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud AA Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen AW Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, volume 25, pages 1097–1105. Curran Associates, Inc., 2012.

[21] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, 2016.

[22] Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19:221–248, 2017.

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, volume 32, pages 8024–8035, 2019. Available at `https://pytorch.org`.

[24] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016.

[25] Chen Chen, Chen Qin, Huaqi Qiu, Giacomo Tarroni, Jinming Duan, Wenjia Bai, and Daniel Rueckert. Deep learning for cardiac image segmentation: A review. *Frontiers in Cardiovascular Medicine*, 7:25, Mar 2020. This article is part of the Research Topic "Current and Future Role of Artificial Intelligence in Cardiac Imaging".

[26] Paul Suetens. *Fundamentals of Medical Imaging*. Cambridge University Press, 3rd edition, 2017.

[27] Dzung L Pham, Chenyang Xu, and Jerry L Prince. Current methods in medical image segmentation. *Annual Review of Biomedical Engineering*, 2:315–337, 2000.

[28] S Alnasser, M Abdulaal, A Maiter, et al. Advancements in cardiac structures segmentation: a comprehensive systematic review of deep learning in ct imaging. *European Radiology*, 2024.

[29] Xiahai Zhuang. Challenges and methodologies of fully automatic whole heart segmentation: A review. *Journal of Healthcare Engineering*, 4(3):981729, 2013.

[30] Carlos G Fonseca, Malte Backhaus, David A Bluemke, et al. The cardiac atlas project-an imaging database for computational modeling and statistical atlases of the heart. *Bioinformatics*, 27(16):2288–2295, 2011.

[31] S Kazemifar, D Dey, P Slomka, and R Nakanishi. Automated vascular segmentation in quantitative coronary ct angiography. *Journal of Imaging*, 1(1):1–9, 2014.

[32] K. U. Juergens, H. Seifarth, F. Range, S. Wienbeck, M. Wenker, W. Heindel, and R. Fischbach. Automated threshold-based 3d segmentation versus short-axis planimetry for assessment of global left ventricular function with dual-source mdct. *AJR. American Journal of Roentgenology*, 190(2):308–314, Feb 2008.

[33] Barba-J. Leiner, Boris Escalante-Ramírez, and Enrique Vallejo. Comparative study of variational and level set approaches for shape extraction in cardiac CT images. In Jorge Brieva and Boris Escalante-Ramírez, editors, *IX International Seminar on Medical Information Processing and Analysis*, volume 8922, page 89220Y. International Society for Optics and Photonics, SPIE, 2013.

[34] X. Chen, J. Jiang, and X. Zhang. Automatic 3d coronary artery segmentation based on local region active contour model. *Journal of Thoracic Disease*, 16(4):2563–2579, Apr 2024.

[35] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[36] Sebastian Ordas, Estanislao Oubel, Rubén Leta, Francesc Carreras, and Alejandro Frangi. A statistical shape model of the heart and its application to model-based segmentation. *Proceedings of SPIE - The International Society for Optical Engineering*, 6511, 03 2007.

[37] Yuchen Lei, Tonghe Wang, Xenia Dong, Shun Tian, Yijiang Liu, Hong Mao, Walter J. Curran, Hongmei K. Shu, Tian Liu, and Xiaofeng Yang. Mri classification using semantic random forest with auto-context model. *Quantitative Imaging in Medicine and Surgery*, 11(12):4753–4766, Dec 2021.

[38] Antonio Criminisi, Ender Konukoglu, and Jamie Shotton. Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning. 2011.

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241. Springer, 2015.

[40] Olivier Bernard, Alexandre Lalande, Clément Zotti, et al. 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, 2018.

[41] Jakob Wasserthal, Martin Meyer, Hans-Christian Breit, et al. Totalsegmentator: Robust segmentation of 104 anatomic structures in ct images. *Radiology: Artificial Intelligence*, 5(5):e220284, 2023.

[42] Fabian Isensee, Paul F Jaeger, Simon AA Kohl, et al. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2):203–211, 2021.

[43] F. Laumer, D. Di Vece, V. L. Cammann, M. Würdinger, V. Petkova, M. Schönberger, A. Schönberger, J. C. Mercier, D. Niederseer, B. Seifert, M. Schwyzer, R. Burkholz, L. Corinzia, A. S. Becker, F. Scherff, S. Brouwers, A. P. Pazhenkottil, S. Dougoud, M. Messerli, F. C. Tanner, T. Fischer, V. Delgado, P. C. Schulze, C. Hauck, L. S. Maier, H. Nguyen, S. Y. Surikow, J. Horowitz, K. Liu, R. Citro, J. Bax, F. Ruschitzka, J. R. Ghadri, J. M. Buhmann, and C. Templin. Assessment of artificial intelligence in echocardiography diagnostics in differentiating takotsubo syndrome from myocardial infarction. *JAMA Cardiology*, 7(5):494–503, May 2022.

[44] F. Zaman, N. Isom, A. Chang, Y. G. Wang, A. Abdelhamid, A. Khan, M. Makan, M. Abdelghany, X. Wu, and K. Liu. Deep learning from atrioventricular plane displacement in patients with takotsubo syndrome: lighting up the black-box. *European Heart Journal - Digital Health*, 5(2):134–143, Dec 2023.

[45] Riccardo Cau, Francesco Pisu, Jasjit S. Suri, Lorenzo Mannelli, Mariano Scaglione, Salvatore Masala, and Luca Saba. Artificial intelligence applications in cardiovascular magnetic resonance imaging: Are we on the path to avoiding the administration of contrast media? *Diagnostics*, 13(12), 2023.

[46] Davide Chicco and Giuseppe Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21:6, 2020.

[47] Masato Shimizu, Makoto Suzuki, Hiroyuki Fujii, Shigeki Kimura, Mitsuhiro Nishizaki, and Tetsuo Sasano. Machine learning of microvolt-level 12-lead electrocardiogram can help distinguish takotsubo syndrome and acute anterior myocardial infarction. *Cardiovascular Digital Health Journal*, 3(4):179–188, 2022.

[48] Michael Owusu-Adjei, Joseph B. Hayfron-Acquah, Timothy Frimpong, and Ganiu Abdul-Salaam. Imbalanced class distribution and performance evaluation metrics: A systematic review of prediction accuracy for determining model performance in healthcare systems. *PLOS Digital Health*, 2(11):e0000290, 2023.

[49] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009.

[50] David M. W. Powers. Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2011.

[51] Douglas G. Altman and J. Martin Bland. Diagnostic tests 1: sensitivity and specificity. *BMJ*, 308:1552, 1994.

[52] Anthony K. Akobeng. Understanding diagnostic tests 1: sensitivity, specificity and predictive values. *Acta Paediatrica*, 96(3):338–341, 2007.

[53] M. H. Zweig and G. Campbell. Receiver-operating characteristic (roc) plots: a fundamental evaluation tool in clinical medicine. *Clinical Chemistry*, 39(4):561–577, 1993.

[54] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[55] Andrew P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[56] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3):e0118432, 2015.

[57] Andreas Holzinger, Chris Biemann, Constantinos S Pattichis, and Douglas B Kell. What do we need to build explainable ai systems for the medical domain? *arXiv preprint arXiv:1712.09923*, 2017.

[58] Leilani H Gilpin, David Bau, Bianca Zoran, Ayesha Bajwa, Aviv Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89, 2018.

[59] Ery Tjoa and Cuntai Guan. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE Transactions on Neural Networks and Learning Systems*, 32(11):4793–4813, 2020.

[60] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*, 2017.

[61] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.

[62] Nishanth T. Arun, Nathan Gaw, Praveer Singh, Ken Hsuan-kan Chang, Mehak Aggarwal, Bryan L. Spangelo Hong Chen, Katharina V. Hoebel, Sharut Gupta, Jay Patel, Mishka Gidwani, Julius Adebayo, Matthew D. Li, and Jayashree Kalpathy-Cramer. Assessing the trustworthiness of saliency maps for localizing abnormalities in medical imaging. *Radiology: Artificial Intelligence*, 3(6):e200267, 2021.

[63] Fahim Zaman, Rakesh Ponnapureddy, Yi G. Wang, Kan Liu, and et al. Spatio-temporal hybrid neural networks reduce erroneous human 'judgement calls' in the diagnosis of takotsubo syndrome. *EClinicalMedicine*, 40:101115, 2021.