

# DIGITISED ECG MOBILE APP

---



Presented by:

Dimpho Sefora

Prepared for:

Dr. Yaaseen Martin

September 21, 2025

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfilment of the academic requirements for the degree of Electrical and Computer Engineering.

# Abstract

In this work we describe a novel Thesis Template, to be used by students in Electrical & Engineering at the University of Cape Town. This section entails the abstract of the document.

# Acknowledgments

# Plagiarism Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have the IEEE convention for citation and referencing. Each contribution to, and quotation in, this final year project report from the work(s) of other people, has been attributed and has been cited and referenced.
3. This final year project report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as there own work or part thereof.



.....  
Dimpho Sefora

September 21, 2025

# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgments</b>	<b>ii</b>
<b>Plagiarism Declaration</b>	<b>iii</b>
<b>Table of Contents</b>	<b>iv</b>
<b>List of Figures</b>	<b>vii</b>
<b>List of Tables</b>	<b>viii</b>
<b>Nomenclature</b>	<b>ix</b>
<b>Chapter 1: Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Problem Statement . . . . .	3
1.3 Objectives . . . . .	3
1.4 Scope and Limitations . . . . .	4
1.5 Thesis Outline . . . . .	4
<b>Chapter 2: Literature Review</b>	<b>6</b>
2.1 Introduction . . . . .	7
2.2 ECG Fundamentals . . . . .	7

2.2.1	ECG Signal Generation . . . . .	7
2.2.2	ECG Waveform Formation . . . . .	8
2.2.3	ECG leads . . . . .	9
2.3	Formats for Digitised ECGs . . . . .	10
2.3.1	Scanned PDF Format . . . . .	10
2.3.2	Manual Digitisation . . . . .	11
2.3.3	MATLAB/Python Digitisation . . . . .	11
2.4	Smartphone-based ECG Capturing . . . . .	12
2.4.1	Performance - Resolution (can it handle it) . . . . .	12
2.4.2	Accessibility and Cost-Effectiveness . . . . .	12
2.5	Digitising ECG Signals . . . . .	12
2.5.1	Artifact Removal . . . . .	13
2.5.2	Segmentation . . . . .	15
2.5.3	Background Removal and Signal Extraction . . . . .	15
2.5.4	Post-processing . . . . .	15
2.5.5	Signal Digitisation . . . . .	15
2.6	Conclusion . . . . .	15
<b>Chapter 3:</b>	<b>Methodology</b>	<b>17</b>
3.1	Overview of Approach . . . . .	17
3.2	System Requirements and Specifications . . . . .	19
3.2.1	Functional User Requirements . . . . .	20
3.2.2	Non-Functional User Requirements . . . . .	20
3.3	Tools and Technology Stack . . . . .	21
3.3.1	Development Tools and Specifications . . . . .	21
3.3.2	Technology Stack . . . . .	22
3.4	Image Processing Model . . . . .	25
3.5	Design and Prototyping . . . . .	26

3.5.1	Wireframes . . . . .	26
3.5.2	Iterations . . . . .	28
3.5.3	User Flow . . . . .	28
3.6	Validation, Benchmarking, and Optimisation . . . . .	30
3.7	Evaluation . . . . .	30
<b>Chapter 4:</b>	<b>Design</b>	<b>31</b>
4.1	System Design . . . . .	32
4.2	Hardware Design . . . . .	33
4.3	Software Design . . . . .	34
4.4	Implementation . . . . .	35
4.5	Integration or Test Rig . . . . .	36
<b>Chapter 5:</b>	<b>Results</b>	<b>37</b>
5.1	Results . . . . .	37
5.2	Tips for Results Figures . . . . .	38
5.3	Pictures and screenshots . . . . .	39
<b>Chapter 6:</b>	<b>Conclusions</b>	<b>41</b>
6.1	Conclusions . . . . .	41
6.2	Future Work . . . . .	41
<b>Bibliography</b>		<b>42</b>
<b>Appendix A: Supporting Data</b>		<b>49</b>
A.1	Lyrics to Soft Kitty . . . . .	49
<b>Appendix B: Satirical Support</b>		<b>50</b>

# List of Figures

2.1	Placeholder for ECG diagram . . . . .	9
3.1	Spiral Model of Development . . . . .	18
4.1	Example system level design illustration . . . . .	33
4.2	Example hardware level design illustration . . . . .	34
5.1	The correlation coefficient as a function of sample count . . . . .	39
5.2	Oscilloscope measurement showing physical line signals on both ends of a transmission line during master switch-over [25] . . . . .	39
5.3	An example image with custom scaling . . . . .	40
B.1	You must prepare to defend your thesis [26] . . . . .	50



# List of Tables

2.1	Luminance Correction Methods . . . . .	13
3.1	Functional User Requirements . . . . .	20
3.2	Non-Functional User Requirements . . . . .	20

# Nomenclature

**depolarization wave** the wave of electrical activity that spreads through the heart muscle, causing it to contract. [7](#)

**ECG** Electrocardiogram. [1](#)

**ROI** Region of Interest. [13](#)

# Chapter 1

## Introduction

Electrocardiograms are one of the first tests conducted when one is admitted to the hospital. They provide a wealth of information on the health of a patient. However, they continue to be produced in paper format. This means they must be stored for longevity in large storage rooms

### 1.1 Background

An electrocardiogram [ECG](#) is a technology that both measures and records the electrical signal patterns describing the rhythmic activity of the heart. These electrical pulses are what signal the heart's skeletal muscles to undergo ventricular contraction [1]. Irregularities from the expected patterns provide a wealth of information on the patient's condition.

Clinically, two ECG systems are in use: paper-recorded and computer-based. Paper ECGs remain the foundational format and are routinely requested as an initial diagnostic test [1]. Computer-based ECGs, first introduced in the 1960s [2], aimed to reduce the dependency on specialist interpretation, but continue to face issues of diagnostic reliability and digital record management [3]. Furthermore, due to significant technological and logistic barriers associated with accessibility and cost, the widespread adoption of these systems has been limited [4] [5]. Therefore, paper ECGs remain widely produced. However, their physical form presents a persistent limitation: secure and accessible storage for later interpretation. Addressing this challenge motivates the digitisation of paper ECGs into formats that can be efficiently stored and accessed, especially for the long-term treatment of cardiac patients [6].

Manual digitisation, which involves manually scanning or transcribing paper ECG tracings into digital systems, is a popular solution to this issue. Although this technique eventually makes it possible to save ECGs in electronically, it is expensive, time-consuming, and prone to errors, especially in environments with lots of patients [7].

Recent developments in mobile health technologies suggest a potential pathway to address this limitation. Smartphones, with their widespread availability and camera facilities, enable paper ECGs to be captured, digitised, and stored within mobile applications [8] [9]. This strategy creates a way to bridge the gap between affordable availability to digital ECG formats and possibly addressing the storage problem [10]. Mobile platforms can enhance patient record management and continuity of treatment by enabling the direct archiving and retrieval of ECG information on portable devices. However, while promising, these developments also highlight the need to explore how paper-recorded ECGs can be reliably digitised and integrated into mobile-accessible formats.

## 1.2 Problem Statement

The digitisation of paper-based electrocardiograms continues to present a challenge, as reliance on physical records causes inefficiencies in storage, retrieval, and long-term accessibility [8, 11]. Despite being widespread, manual digitisation is expensive, inefficient, and prone to errors [7] and existing mobile scanning tools are not designed for ECG waveforms, leading to distortion, misalignment, and loss of signal fidelity [12]. Current methods are inadequate for efficiently converting paper ECGs into accurate, analysis-ready digital formats, highlighting the need for reliable, mobile-accessible digitisation solutions.

## 1.3 Objectives

These objectives guide the progression of the research, ensuring that the work done is centred on the problem statement. The following objectives will be explored throughout the report:

1. Review literature on ECG signal structure, interpretation, and digitisation techniques.
2. Investigate existing image processing and signal processing methods for waveform extraction from photographs.
3. Design and develop a mobile application (Android or iOS) for accurate ECG strip digitisation.
4. Implement algorithms for noise reduction, de-warping, filtering, and artifact removal.

5. Store processed ECG signals in a standardised digital format.
6. Test and evaluate the accuracy and usability of the developed application.

## 1.4 Scope and Limitations

### Scope:

- Mobile application development for Android/iOS.
- Image capture, processing, and signal extraction for standard ECG paper strips.
- Local storage of digitised ECG in standard digital formats.

### Limitations:

- No integration with electronic health record (EHR) systems in this phase.
- Limited dataset for testing.
- Optimised for standard-format single-lead or multi-lead strips.
- No integration of deep-learning analysis and diagnosis of digitised data

## 1.5 Thesis Outline

- **Chapter 2:** Literature Review — ECG basics, digitisation methods, and relevant image/signal processing techniques.

- **Chapter 3:** Methodology and Design — approach, algorithms, and system architecture.
- **Chapter 4:** Implementation — development process and app features.
- **Chapter 5:** Results and Discussion — evaluation and performance.
- **Chapter 6:** Conclusion — summary of findings and future work.

# Chapter 2

## Literature Review

This chapter provides a comprehensive evaluation of literature relevant to the research topic. It begins with an introductory section outlining the motivation for the study, focusing on the need to further investigate mobile app development for digitising and storing ECG paper records. A brief overview of the theory behind ECG signal formation follows, providing the foundation for understanding its digitisation.

The review then adopts a funnel approach: first examining existing methods of ECG record digitisation, then evaluating the potential of mobile applications for this purpose through image capture, and finally considering techniques for extracting ECG waveforms from images. The chapter concludes with a summary of key findings, identifying gaps in current research and setting the stage for the methodology that follows.



## 2.1 Introduction

The electrocardiogram (ECG) has been a fundamental tool in medicine since 1901 [8]. Over the years, a vast number of ECG paper records have been produced. Consequently, researchers have devised methods to digitise these records in order to ensure long-term preservation, improve accessibility, and facilitate future evaluation and diagnosis. It is therefore important to investigate and assess these methods to understand their respective strengths and limitations.

With the growing use of mobile applications in healthcare [11], new opportunities for ECG digitisation have emerged. This enables investigation into whether ECG images captured with mobile devices can be effectively processed for digitisation.

## 2.2 ECG Fundamentals

Electrocardiogram signals show how the path of the [depolarization wave](#) - the flow of a group of positive electric charges - moves during each heartbeat [13]. Hearts have natural pacemakers that initiate electric signals that cause atrial and ventricular contraction [14]. Electrocardiograms are built to sense these natural signals and convert them into useful information that portray the heart's condition.

### 2.2.1 ECG Signal Generation

The electric activity of the heart is governed by the sinoatrial (SA) and atrioventricular (AV) nodes. The SA node initialises the impulses and the AV node introduces delays to their delivery to co-ordinate between arterial and ventricular contractions

for optimum heart functionality [14].

As the positive charges move through the heart's muscles, their negative resting state changes to a positive active state. This change in charge of the internal muscles affects a change to the external charge surrounding it [13]. This creates the potential difference measured by the electrocardiogram electrodes. If detected it is noted as a deflection on the ECG waveform. The bigger the potential difference, the larger the deflection [13].

(image explaining or showing ECG deflections - so they know what to look for in the actual trace)

### 2.2.2 ECG Waveform Formation

An ECG signal consists of five deflections that together form the PQRS complex, representing a single heartbeat and the complete cycle of myocardial contraction and relaxation [1].

The P wave reflects atrial depolarisation, initiating atrial contraction. It is typically a low-amplitude, gently sloped deflection due to the slower propagation of the depolarisation wave through atrial myocardial cells [15]. The QRS complex corresponds to ventricular depolarisation and the onset of ventricular contraction, with the R wave being the largest positive deflection [16]. The T wave represents ventricular repolarisation, preparing the ventricles for the next cardiac cycle [16].

## 2.2. ECG FUNDAMENTALS

---

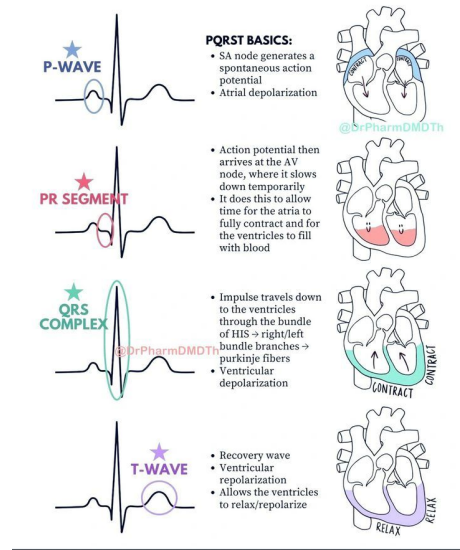


Figure 2.1: Placeholder for ECG diagram

The timing and amplitude of the PQRST wave are essential for patient health analysis [17]. Significant signal loss in the digitisation process would thus render the ECG useless for diagnostic purposes.

The grid is formed ... such that reading the waveforms can remain consistent throughout.

### 2.2.3 ECG leads

The direction of this wave is dependent on the leads. The leads are defined by the electrodes used. In a 12-lead ECG, there are 10 electrodes used to measure the heart's activity from different angle and planes. Each lead provides different and vital information about arrhythmias and myocardial infarction (heart attacks).

There are the limb electrodes places by the left arm and leg and the right arm.

Measurements between two of them, with the third being a reference, creates the I, II, and III dipoles leads. When a fourth electrode known as the reference or zero electrode is introduced, the augmented leads are created. This is a measurement from one of the limb electrodes to the reference electrodes. This creates one of the unilateral leads.

In 1934, it was noted that there were areas in the heart that needed to be measured to better detect myocardial infarctions. This gave birth to the precordial leads through the use of six electrodes placed around the precordial (chest area about and around the heart).

These make up the 12 leads that illustrate the different waveform patterns seen in the electrocardiogram trace. They are used to section the ECG into 12 important regions which can be segmented and analysed further. [\[1\]](#)

## 2.3 Formats for Digitised ECGs

In recent decades, multiple approaches have been developed for digitising and storing ECG records, ranging from the scanning of paper traces into PDF format to direct digitisation using computer applications such as MATLAB or Python.

### 2.3.1 Scanned PDF Format

The most straightforward method of digitising ECGs is by scanning paper records into PDF format. This approach makes data easily accessible across multiple devices, such as laptops and mobile phones. It also facilitates better organisation and storage of patient information, improving the efficiency of record retrieval and supporting

long-term preservation that physical paper cannot guarantee.

However, this method has notable limitations. Since the ECG remains an image rather than raw signal data, it cannot be manipulated or re-analysed for further processing. Scanned records may contain physical defects such as creases, stains, or fading, particularly when dealing with older paper traces. These issues cannot be corrected during scanning, which reduces the reliability and quality assurance of the data. Thus, while efficient for archiving and accessibility, this approach is limited in its usefulness for future diagnostic and analytical applications.

### 2.3.2 Manual Digitisation

A more time-consuming approach to ECG digitisation involves manually recording the millivolt–time coordinates of the individual signal leads onto an electronic spreadsheet and entering them into software such as MATLAB or Python. Unlike scanning into PDF format, this method produces a true digital signal that can be manipulated, processed, and analysed with high accuracy. However, it is highly labour-intensive and susceptible to human error, particularly when large volumes of ECG records must be digitised.

### 2.3.3 MATLAB/Python Digitisation

Technological advancements, along with the development of Python and MATLAB libraries that integrate vision frameworks for object identification, have enabled the efficient extraction of signals from scanned ECG images. Algorithms can now automate the tracing of ECG waveforms into time-series coordinates, which can subsequently be plotted, analysed, and manipulated in digital form. The extracted data

are typically stored in CSV format, facilitating efficient organisation, retrieval, and downstream processing.

This approach offers several advantages: automated signal acquisition, reliable digital storage through CSV files, and straightforward visualisation within environments such as MATLAB, Python IDEs, and Excel. Despite these benefits, practical limitations remain. Such tools are generally deployed on laptops or desktop computers, which may not always be accessible in the fast-paced environment of a hospital or outside clinical settings when records are urgently required. Furthermore, most implementations perform best with scanned images and often show reduced reliability when applied to camera-captured inputs, limiting their adaptability in resource-constrained or less controlled settings.

## **2.4 Smartphone-based ECG Capturing**

### **2.4.1 Performance - Resolution (can it handle it)**

### **2.4.2 Accessibility and Cost-Effectiveness**

## **2.5 Digitising ECG Signals**

The fundamental concept of digitising ECG signals lies heavily within image processing. Generally, the process involves preparing the image through filtration and artifact removal, isolating the region of interest, particularly the 12 leads, through segmentation, and converting the extracted ECG waveform into a one-dimensional digital signal suitable. However, machine learning models have also been used to

achieve greater results in varying datasets.

### 2.5.1 Artifact Removal

The report written by Cuong et al. highlights how the performance of their model dropped by 60.87% when artifacts are present in the processed images [18], emphasising the need for their removal.

#### Shadow and Crease Removal

Shadows and creases represent the most disruptive artifacts in ECG image digitisation. Mishra et al. introduces the concept of luminance correction for shadow removal, given that they occur due to obscured light [19]. This method involves changing the luminance/brightness of the ROI without affecting the colour. This can be done in different colour spaces.

Table 2.1: Luminance Correction Methods

Space	Description	Strengths	Limitations	Cite
RGB	Adjust luminance in each channel	Requires no conversion	Can erode the colour	[20]
LAB	Corrects luminance separately from colour	Luminance and colour well separates	Can introduce noise	[20]
YCbCr	Corrects brightness separately from colour	Good luminance separation	Produces less uniform colour	[21]

### Noise Removal

**Mean Filters:** Within a pixel cluster of odd  $n \times n$  sizing, the middle pixel is replaced by the average of all their pixel values. It unfortunately serves to blur the image, removing important object detail [22].

**Median Filters:** These are better capable at removing salt and pepper (discrete) noise, while preserving edge details. However, it does tend to remove finer details that appear as outliers in a cluster of pixels [22].

**Gaussian Filters:** These better preserve finer details as compared to mean and median filters [22]. However, they perform poorly when applied to discrete noise.

**Wiener Filters:** In images that have noise caused by motion blur, these filter perform excellently. They are less effective for salt and pepper noise as compared to other linear filters [22].

Camera-captured images are littered with continuous random noise caused by the hardware involved. This makes filters that tackle Gaussian noise more appealing for the purpose of this project.

### Text Removal

ECG papers contain labels for each lead. These characters need to be removed before converting the extracted waveform into a 1-D signal.

**Optical Character Recognition:** Vision frameworks are used to identify text in images. The algorithms then use morphological operators to isolate the regions, and image impairments to reconstruct the area [23].

**Image Processing:** There are a couple of image processing approaches that can be used to remove text, with most using cropping



### 2.5.2 Segmentation

Image Processing Approach

Machine Learning Approach

### 2.5.3 Background Removal and Signal Extraction

Image Processing Approach

Machine Learning Approach

### 2.5.4 Post-processing

### 2.5.5 Signal Digitisation

## 2.6 Conclusion

- Summary of reviewed methods and gaps.
- Justification for proposed app.
- Key algorithmic challenges for this project.
- Broad description of subject
- Some relevant history
- Current implementations in industry

## 2.6. CONCLUSION

---

- New & Related Research on the subject

# Chapter 3

## Methodology and Design

This chapter details the design and engineering decisions taken to develop and eventually implement digitising ECG signals through mobile applications. These decisions are guided by the observations highlighted in the .

It will begin with a highlevel overview of the proposed approach for tackling the problem. It will then provide specific details on the methods and techniques used in the research, including any algorithms, tools, and technologies employed. The chapter will also discuss the rationale behind these choices, highlighting how they align with the research objectives and address the identified challenges.

### 3.1 Overview of Approach

The core principle of the proposed approach is iteration. With each new stage of development, earlier steps are revisited to ensure ongoing risk management and

### 3.1. OVERVIEW OF APPROACH

---

continuous improvement. This cyclical process mirrors the spiral model commonly applied in system and software design.

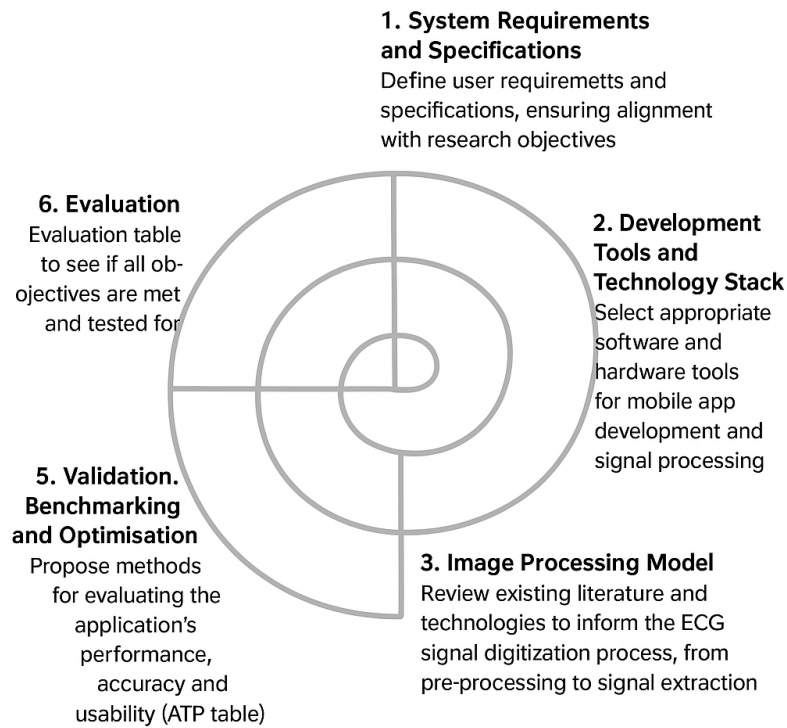


Figure 3.1: Spiral Model of Development

- 1. System Requirements and Specifications:** Define user requirements and specifications, ensuring alignment with research objectives
- 2. Development Tools and Technology Stack:** Select appropriate software and hardware tools for mobile app development and signal processing
- 3. Image Processing Model:** Review existing literature and technologies to inform the ECG signal digitisation process, from pre-processing to signal extraction

4. **Design and Prototyping:** Create iterative versions of the mobile application to test core functionalities, using wireframes for illustration
5. **Validation, Benchmarking, and Optimisation:** Propose methods for evaluating the application's performance, accuracy, and usability (ATP table)
6. **Evaluation:** Evaluation table to see if all objectives are met and tested for

## 3.2 System Requirements and Specifications

The objectives of the research are clear: design a mobile application (Android or iOS) that can (1) facilitate the capturing of ECG paper records, (2) process the captured images to extract the ECG waveform, and (3) store the digitised ECG signals in a standardised digital format. The following user and functional requirements and specifications are derived from these objectives.

### 3.2.1 Functional User Requirements

Table 3.1: Functional User Requirements

#no.	Requirement	Description
FR01	Camera Access	Application must be able to access the mobile's camera
FR02	Image Import	Access photo library for image upload
FR03	Pre-processing	Must facilitate cleaning and aligning of image for signal extraction
FR04	Signal extraction	Detect signal and extract it in pixel coordinates
FR05	Reconstruction	Convert pixel coordinates to amplitude-time series
FR06	Data storage	Facilitate storing of digitised data in external device or cloud in CSV format
FR07	Visualisation	Display interactive ECG waveform

### 3.2.2 Non-Functional User Requirements

Table 3.2: Non-Functional User Requirements

#no.	Requirement	Description
NFR01	Performance	Image capture and signal extraction should complete within acceptable time ( $< 5$ sec)
NFR02	Accuracy	Maintain reliable signal extraction across various ECG paper qualities
NFR03	Usability	Interface must be simple, clear, and intuitive for users with minimal training

NFR04	Reliability	Handle errors gracefully and allow retry; ensure data persists between sessions
NFR05	Compatibility	Support Android or iOS devices with standard camera specifications
NFR06	Security	Store data securely on device; optionally encrypt sensitive information

## 3.3 Tools and Technology Stack

Defining the development environment for any software project is crucial, as it influences the efficiency of the development process, the quality of the final product, and the ease of maintenance. This section outlines the chosen tools and technologies for developing the mobile application for digitising ECG signals. Notice that the chosen operating system is iOS because of its comprehensive set of libraries and frameworks for image processing through Xcode.

### 3.3.1 Development Tools and Specifications

- **Xcode:** The integrated development environment (IDE) for macOS, used for developing software for iOS. It provides a suite of tools for designing user interfaces, writing code, and debugging applications.
- **Apple Developer:** A platform that provides resources and tools for iOS app development, including access to beta software, advanced app capabilities, and app analytics. This is essential for deploying to a physical iOS device for testing through Testflight.

- **Simulator:** A tool within Xcode that allows developers to test and debug iOS applications on a Mac without needing a physical device. It simulates various iPhone and iPad models and iOS versions.
- **iPhone 13:** A physical device used for testing the application in real-world conditions, ensuring that it performs as expected on actual hardware and observing real user interactions.
- **VMware Workstation Pro:** A virtualisation software used to run macOS on a Windows machine, enabling the use of Xcode and other macOS-specific tools without needing a dedicated Mac. It provides a controlled environment for iOS development and testing. The specific macOS installed is Sequoia, running on a Mac Mini 4. For compatibility with the Windows PC, the virtual machine is allocated 4GB RAM, 4 CPU cores and 128GB of disk space.

#### 3.3.2 Technology Stack

**iOS SDK** The iOS SDK is Apple’s official development toolkit for building applications on iPhones and iPads. It provides access to the operating system’s APIs, device hardware (camera, sensors, storage), and system-level services. This research project requires close integration with the iPhone camera for ECG capture, efficient on-device signal processing, and secure file handling. The iOS SDK ensures native access to all necessary features while maintaining compliance with Apple’s ecosystem.

It provides a complete and unified set of frameworks for image processing, file management, and graphics rendering, while being optimized for Apple hardware, ensuring smooth performance and efficient energy usage. It offers long-term reliability and stability since it is fully supported by Apple.



- **Security:** Applications built with the iOS SDK run in a sandboxed environment, preventing unauthorized access to other apps or system files. It supports data encryption at rest and in transit, aligning with healthcare data protection needs (e.g., HIPAA compliance). It requires that all apps be **digitally signed**, which ensures authenticity and protects users from tampered or malicious code.
- **Performance:** The SDK provides frameworks (e.g., Core Image, Accelerate) that are hardware-accelerated using the CPU, GPU, and Neural Engine. It ensures real-time responsiveness, critical for ECG pre-processing and visualization tasks. The native compilation reduces overhead compared to cross-platform solutions, leading to faster execution and lower battery drain.
- **Integration:** The SDK seamlessly integrates with device features such as the camera, storage, sensors, and security mechanisms. Furthermore, it provides compatibility with higher-level frameworks such as Core Image (image processing), Vision (computer vision), and CloudKit (cloud storage). It as well ensures consistent user experience by adhering to iOS UI/UX design standards.

#### Core iOS Frameworks and Libraries

- **Core Image:** Apple’s image processing framework designed for filtering, noise reduction, normalization, and geometric corrections such as dewarping ECG scans. It uses GPU acceleration, allowing fast and energy-efficient processing on mobile hardware. Core Image can work in parallel with Vision to prepare images for further feature extraction.
- **Vision Framework:** A high-level computer vision framework for feature detection, object recognition, and image alignment. Vision is useful for detecting

ECG grid lines and aligning the paper image before signal extraction. It complements Core Image by handling structural analysis while Core Image focuses on pixel-level filtering.

- **Accelerate Framework:** A low-level framework that provides optimized mathematical, signal processing, and digital signal processing (DSP) functions. It uses vectorization and hardware acceleration (CPU and GPU) to speed up operations. This is crucial for transforming images into ECG signals, and it often works in sequence after preprocessing by Core Image and Vision.
- **Foundation Framework:** A system-level framework that provides essential services such as data structures, file management, and I/O operations. It is used for generating and managing CSV files that store the extracted ECG time-series data. Its advantage lies in efficiency and reliability when handling structured data within the iOS environment.
- **Core Graphics / Quartz 2D:** A two-dimensional rendering engine that allows vector-based and pixel-accurate drawing. It is used in DigECG to visualize ECG waveforms on the app's interface. This framework complements Accelerate, where Accelerate computes the signal data and Core Graphics ensures precise visual representation.

#### External Tools for Benchmarking and Evaluation

- **OpenCV:** An open-source computer vision library that provides a wide range of algorithms for image segmentation, edge detection, and noise removal. OpenCV is advantageous for prototyping and experimentation, as it offers robust tools not natively available in iOS. In DigECG, it was used in parallel with Core

Image and Vision during development to validate preprocessing accuracy and ensure robustness under varied conditions.

- **MATLAB:** A high-level scientific computing environment with specialized toolboxes for biomedical signal processing and analysis. MATLAB was used to benchmark ECG extraction accuracy against ground truth datasets, and to apply advanced filtering and spectral analysis. It complements OpenCV by focusing on quantitative evaluation of the extracted signals, ensuring that image-to-signal conversion methods in the app are scientifically validated.

#### Cloud Integration

- **CloudKit:** Apple’s cloud service framework for storage and synchronization across devices. CloudKit provides end-to-end encryption and secure authentication, making it suitable for sensitive healthcare data. For DigECG, it is planned as an extension to store digitized ECG files securely, ensuring data is not only stored locally but also accessible across devices for patient monitoring and medical record keeping. This complements local CSV file storage by adding reliability and redundancy through cloud backup.

## 3.4 Image Processing Model

In this section, the observations from the literature will be used to draw up and choose a model for the image processing required for digitising the ECG signal.

### 3.5 Design and Prototyping

Prototyping bridges the conceptualisation of the app and its realisation. Low-fidelity wireframes are used to outline structure, navigation, and features. Iterative refinements ensure usability and alignment with requirements.

#### 3.5.1 Wireframes

##### Home / Dashboard Wireframe

**Purpose:** Navigation hub.

**Features:**

- Upload/scan ECG image.
- Access recent digitised signals.
- Settings (signal scaling, export formats).

##### Image Pre-processing Screen Wireframe

**Purpose:** Present uploaded ECG scan and allow preprocessing.

**Features:**

- Toggle grid removal.
- Noise reduction preview.

- Region-of-interest selection.
- Undo/redo controls.

#### **Signal Extraction Screen Wireframe**

**Purpose:** Display detected ECG trace.

**Features:**

- Overlay of raw scan and extracted waveform.
- Baseline correction adjustments.
- Manual correction tools (erase/add points).

#### **Signal Reconstruction & Visualisation Wireframe**

**Purpose:** Display reconstructed digital ECG waveform

**Features:**

- Interactive graph (zoom, pan, scale).
- Lead/time scaling indicators (mm/s, mV).
- Resampling and export options.

### **Export / Results Screen Wireframe**

**Purpose:** Final storage and sharing stage

**Features:**

- Export options (CSV, JSON, MATLAB, PDF).
- Share via email/cloud.
- Comparison view (digitised vs raw scan).

### **3.5.2 Iterations**

The initial prototype offered only ...

Iterations added:

- Preprocessing toggles (grid/noise removal).
- Manual correction tools for overlapping artifacts.
- Interactive waveform viewer (zoom, pan, scaling).
- Multiple export formats.

### **3.5.3 User Flow**

**Home Screen**

- Case 1: Upload or capture ECG.

- Case 2: Access recent digitised signals.
- Case 3: Open settings for scale and export configuration.

#### **Pre-processing Screen**

- Case 1: Apply automatic grid removal. (maybe remove in other iterations - can be a base)
- Case 2: Apply artifact filtering.
- Case 3: Select ECG region of interest or entire ECG.

#### **Signal Extraction Screen**

- Case 1: View automatically extracted waveform.
- Case 2: Manually adjust trace (add/remove points).
- Case 3: Re-run extraction with different thresholds.

#### **Signal Reconstruction & Visualisation**

- Case 1: View digitised ECG in interactive graph.
- Case 2: Compare digitised vs raw waveform overlay.
- Case 3: Adjust scaling (25 mm/s, 10 mm/mV).

#### **Export Screen**

- Case 1: Save waveform as CSV.
- Case 2: Export as PDF overlay.
- Case 3: Share via cloud/EHR system.

## **3.6 Validation, Benchmarking, and Optimisation**

## **3.7 Evaluation**



# Chapter 4

## Application Development

The Design is, as the name suggests, about the prototype or system you designed in order to achieve investigation or development goals of your research objective. The Design chapter is something that is typically found in engineering theses, hence our inclusion of that chapter. The scope and complexity of this chapter (or associated design chapters) depends on the level of the project: obviously a BSc final year project is going to be smaller scale and less complicated than a MSc project.

Commonly, systems that are built nowadays, and this relates especially to computer engineering or mechatronics types projects (but is relevant to many electrical engineering project too), involve multiple aspects. Typically: (a) the System Level; (b) the Hardware Level and (c) the Software Level. In addition, there may be considerations for the environment and/or ‘test rigs’ (i.e., the infrastructure that may need to be set up around the system under test in order to perform the testing, and the test rig may in itself be a complicated system that needs thorough explanation.)

### 4.1 System Design

As mentioned, the system you are designing may have multiple parts, both of the prototype and its surrounding test rig (you could call this ‘System Level Design’ if you prefer, or something more accurate for your particular project). The system level section of the design aims to explain what these big pieces or subsystems are that you are going to develop. Often, for embedded systems particularly, the design is divided into a front-end and a back-end.

The front-end provides the point of interaction with other systems and/or the user. A graphic user interface (GUI) may be part of the front-end (depending on the design) ... or the front-end might be signal conditioning and sampling electronics that then feeds into a front-end processor (e.g., FPGA) and further on into the system (e.g., towards back-end processing stages and storage). The user interface and GUI might be more in the back-end in some designs, e.g., a website services which the user or other programs connect to.

*Note:* It is usually imperative to have a clear and easy-to-follow diagram (e.g., Fig. 4.1) to illustrate the system level and to refer to in your explanations for this section.

The sections that follow the System Level depends on what your system involves. We have provided an example here of a system that has some Hardware Level aspects, some Software Level aspects and considerations for Integration (in this case setting up a test rig).

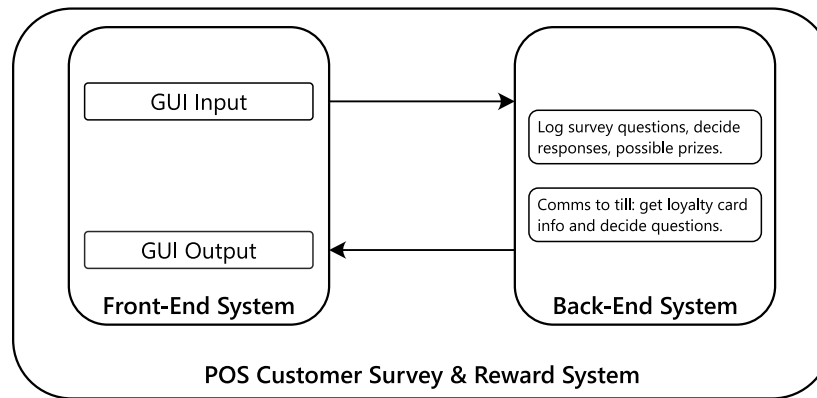


Figure 4.1: Example system level design illustration

## 4.2 Hardware Design

The Hardware Design sections include significant details on your hardware design, PCB considerations, hardware interfacing and connections and power considerations, among other aspects that are specific to the hardware concerns of your prototype or system being built. By ‘significant details’ it is suggested that you do not need to go into excessive minutiae of the design – if you are building a significant piece of hardware largely from scratch, then you probably need a good amount of details to explain your choices etc. You can also use an appendix in which to park information that may be providing extraneous detail that you think is nevertheless needed but is causing the write-up to become too bulky.

*Note:* Even if your project is entirely software, you may still want to have a Hardware section to explain what platform and related components you were using; such information can help others to recreate your experiments, which are a desirable property of a good thesis. If you are doing software performance tests you would need to provide characteristics of the platform, thus another reason for having a hardware section (but if the hardware section is just there for platform specs, then you can

keep the section pretty short, likely not needing more than a page).

It's generally a good idea to include a block diagram and or schematics at this point. Do not simply have a text-heavy discussion of what parts were used with a detailed schematic and photo of the hardware device that was built (doing so would offload the explanations and logical progression from design to PCB to the examiner to figure out, which is certainly not an advisable approach even if it saves much space).

A representative block diagram, which provides a clear explanation of a specific piece of a design, is presented in Fig. 4.2. This figure was drawn in [InkScape](#) [24]. When you want to import an InkScape figure (SVG format) into L<sup>A</sup>T<sub>E</sub>X, simply save it to PDF (use the drawing extents as the media box area) and include the figure. This template includes a `'make figures'` target meant for this purpose.

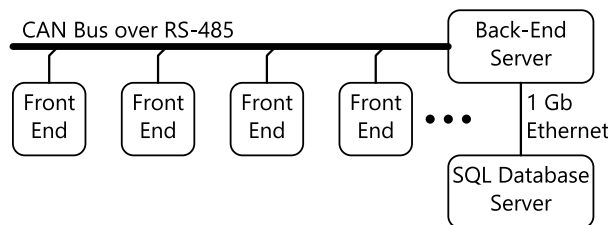


Figure 4.2: Example hardware level design illustration

## 4.3 Software Design

The software design section should go from the high-level design aspects, using for example block diagrams or UML class diagrams to explain the main parts of the system, and then going into details of specific operations or algorithms using some or a combination of, for example, pseudocode, UML state charts, UML activity diagrams, flow charts, or other appropriate figures to help the explanations.

You might decide to have some actual code (usually not more than code snippets, i.e., not whole programs) in the Software section, or you might decide to put such details into an implementation section (since the code is something that carries the design into an instantiated implementation).

Things you may have in the Software section include the following:

- Software designs drawings (e.g., block diagrams, UML diagrams, etc.)
- Algorithms (maybe in pseudocode, or actual code such as MATLAB)
- Code snippets (where relevant, used to illustrate how you went from algorithm, or an element of the software design, to executable implemented code)
- Implementation and development methods (for example specific software tools that had to be installed, scripts to run, parameters to use; but remember that some of this particular item may be better placed in the methodology – particularly if it relates to choices that were made earlier on or even before development started.)

## 4.4 Implementation

For a project that is largely hardware based, the implementation section is sometimes rather short, providing photos of the system and explaining some tips and methods on how it was put together (e.g., solutions that were learned for how to solder on parts effectively, or through the implementation experience realising parts that need to be handled with special care etc.) For a project that involves hardware and software, this section could include both tips on getting the hardware together, as well as details about implementing the software.

## 4.5 Integration or Test Rig

Some research projects require the development of surrounding infrastructure or a suitably conditioned or prepared environment in order to carry out the testing. This may involve developing some sort of test rig into which the prototype is placed or coupled so that testing can be performed on it. As a simple example, consider a vibration measuring device. If you want to test it in the lab, which has concrete floors but you want to test it for a range of flooring types, it may be necessary to build one or more test rigs that will provide the needed characteristics in order to test the product in a sufficiently authentic situation.

The integration section may alternatively, or in addition to the above point, explain how different subsystems of the system constructed are connected up. For example, this section might be used to explain the different ways to connect up a system that combines some software on a PC, a complicated DSP platform, and perhaps separate front-end conditioning circuitry, in order to complete experiments to test the achievement of different sub-objectives of the project.

# Chapter 5

## Results

### 5.1 Results

- Describe the experimental setup (ie. Hardware)
- Describe your experiments
- Describe your results

This chapter should generally present your finding in the form of figures, tables, equations, code listings, etc. It is for presenting and discussing your findings, which can split into sections if the experiment has multiple parts, or stages. You might also want to have a ‘Discussion’ or ‘Discussion of Results’ chapter, which may focus on either more detailed discussion of particular results, or more comprehensive discussion of the results and system performance as a whole. If you have a Discussion of Results section, then you do not want to discuss too much about the specific results

in this chapter and rather move the main discussion or reflective considerations of these to the Discussion of Results chapter.

However, and this is an important reminder, ensure that you do have some text of discussion for your results, to take the reader through your results and the figures you may provide. Make sure not to just put one figure after another without any attempt to explain the sequencing and what is being shown and perhaps some key issues in the figures presented (a monolithic sequence of figure after figure without any attempt at explanation will get undesirable responses from examiners).

If you have an experiments section, it is often useful to have a clear connection of experiment section corresponding to a result section showing results for that test. Examiners often appreciate that sort of clear and consistent structure that is easy to follow. For example, if you have section 5.1 as Modular Testing, then you could correspond to 6.1 Modular Testing Results (the two sections should be cross-linked with `\label` and `\ref`, in both directions)

## 5.2 Tips for Results Figures

Include good quality graphs (see Fig. 5.1 for an example).

An easy way to obtain more space for an article is to use wide, flat figures, such as Fig. 5.2.



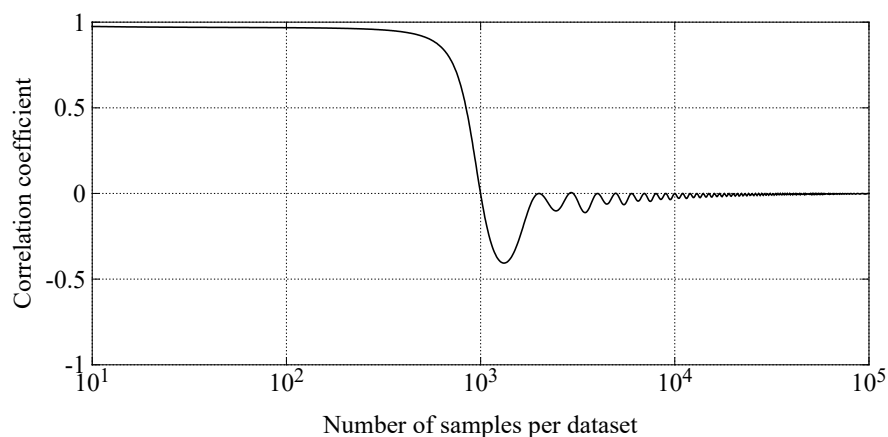


Figure 5.1: The correlation coefficient as a function of sample count

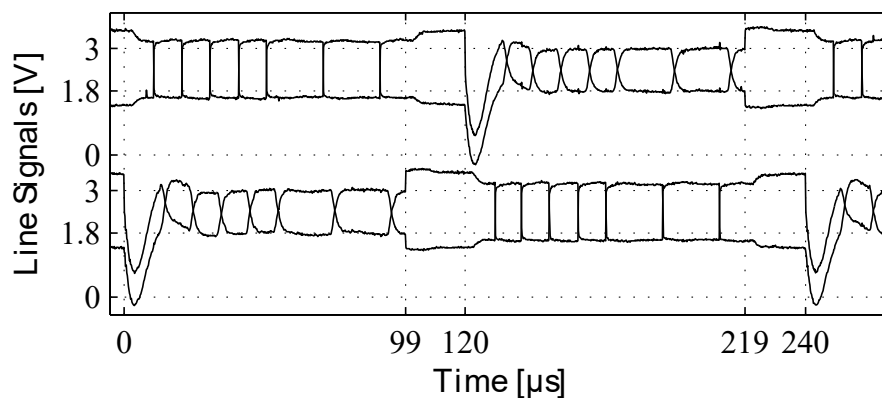


Figure 5.2: Oscilloscope measurement showing physical line signals on both ends of a transmission line during master switch-over [25]

## 5.3 Pictures and screenshots

When you include screenshots, pdfL<sup>A</sup>T<sub>E</sub>X supports JPG and PNG file formats. PNG is preferred for screenshots, as it is a loss-less format. JPG is preferred for photos, as it results in a smaller file size. It's generally a good idea to resize photos (not screenshots) to be no more than 300 dpi, in order to reduce file size. **Never change**

**the aspect ratio of screenshots and pictures!**

Fig. 5.3 shows an example of a PDF image with custom scaling.



Figure 5.3: An example image with custom scaling

Make sure to always use the best quality image possible. Use JPEG for photos, PNG for screenshots and PDF (scalable vector graphics) for everything else. JPEG is lossy, but good for photos, whereas PNG is lossless and good for images with large areas of solid colour.



(a) JPEG



(b) PNG



(c) SVG

# Chapter 6

## Conclusions

### 6.1 Conclusions

- Restate the problem. State the novel solution.
- Summarize what has been accomplished
- Summarize any limitations
- What worked really well and has a big impact?

### 6.2 Future Work

- How do you hope to continue work on this topic?
- Are there possible extensions?
- What are some improvements that could be made?

# Bibliography

- [1] M. AlGhatrif and J. Lindsay, “A brief review: history to understand fundamentals of electrocardiography,” *Journal of Community Hospital Internal Medicine Perspectives*, vol. 2, no. 1, Apr. 2012.
- [2] V. Medved, S. Medved, and I. Kovac, “Critical appraisal of surface electromyography (semg) as a taught subject and clinical tool in medicine and kinesiology,” *Frontiers in Neurology*, vol. 11, 2020. [Online]. Available: <https://api.semanticscholar.org/CorpusID:225062621>
- [3] J. Heaney, J. M. Buick, M. U. Hadi, and N. Soin, “Internet of things-based ecg and vitals healthcare monitoring system,” *Micromachines*, vol. 13, 2022. [Online]. Available: <https://api.semanticscholar.org/CorpusID:254440129>
- [4] B. Rubbo, N. K. Fitzpatrick, S. C. Denaxas, M. Daskalopoulou, N. Yu, R. S. Patel, and H. Hemingway, “Use of electronic health records to ascertain, validate and phenotype acute myocardial infarction: A systematic review and recommendations.” *International journal of cardiology*, vol. 187, pp. 705–11, 2015. [Online]. Available: <https://api.semanticscholar.org/CorpusID:13961345>
- [5] J. Qiu, J. Zhu, S. Liu, W. J. Han, J. Zhang, C. Duan, M. Rosenberg, E. Liu, D. Weber, and D. Zhao, “Automated cardiovascular record retrieval

- by multimodal learning between electrocardiogram and clinical report,” in *ML4H@NeurIPS*, 2023. [Online]. Available: <https://api.semanticscholar.org/CorpusID:265302359>
- [6] L. Ravichandran, C. Harless, A. J. Shah, C. A. Wick, J. H. McClellan, and S. Tridandapani, “Novel tool for complete digitization of paper electrocardiography data,” *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 1, 2013. [Online]. Available: <https://api.semanticscholar.org/CorpusID:1664058>
- [7] Z. J. Eapen *et al.*, “Defining a mobile health roadmap for cardiovascular care integration,” *Journal of the American Heart Association*, vol. 5, no. 11, 2016.
- [8] B. Martínez-Pérez, I. de la Torre-Díez, M. López-Coronado, and J. Herreros-González, “Mobile apps in cardiology: Review,” *JMIR mHealth and uHealth*, vol. 1, no. 2, p. e15, 2013.
- [9] H. Wu, K. H. K. Patel, X. Li, B. Zhang, C. Galazis, N. Bajaj, A. Sau, X. Shi, L. Sun, Y. Tao, H. Al-Qaysi, L. Tarusan, N. Yasmin, N. Grewal, G. Kapoor, J. W. Waks, D. B. Kramer, N. S. Peters, and F. S. Ng, “A fully-automated paper ecg digitisation algorithm using deep learning,” *Scientific Reports*, vol. 12, 2022.
- [10] Y. Liu, A. A. Abbasi, A. Aghaei, A. Abbasi, A. Mosavi, S. Shamshirband, and M. A. A. Al-qaness, “A mobile cloud-based ehealth scheme,” *arXiv preprint arXiv:2004.11842*, 2020, preprint. [Online]. Available: <https://arxiv.org/abs/2004.11842>
- [11] S. R. Steinhubl, E. D. Muse, and E. J. Topol, “Can mobile health technologies transform health care?” *Journal of the American Medical Association (JAMA)*, vol. 310, no. 22, pp. 2395–2396, 2013.

- [12] M. V. McConnell *et al.*, “Mobile health advances in physical activity, fitness, and cardiovascular health,” *Journal of the American College of Cardiology*, vol. 71, no. 23, pp. 2693–2704, 2018.
- [13] Osmosis from Elsevier, “Electrocardiography (ecg/ekg) – basics,” <https://www.youtube.com/watch?v=xIZQRjkwV9Q>, 2025, YouTube video.
- [14] Ignite Healthwise, LLC Staff, “Sa node and av node,” <https://myhealth.alberta.ca/Health/pages/conditions.aspx?hwid=sts14215>, MyHealth.Alberta.ca, 2024, current as of October 24, 2024.
- [15] C. O’Reilly, S. D. R. Oruganti, D. Tilwani, and J. Bradshaw, “Model-driven analysis of ecg using reinforcement learning,” *Bioengineering*, vol. 10, 2023. [Online]. Available: <https://api.semanticscholar.org/CorpusID:259268489>
- [16] J. O. Prima, B. S. Pamungkas, Nugraha, and Suprijanto, “Polyaniline as novel polymer materials for dry electrode- based electrocardiography (ecg),” *Jurnal Elektronika dan Telekomunikasi*, 2018. [Online]. Available: <https://api.semanticscholar.org/CorpusID:139344133>
- [17] A. Sinha, T. Aljrees, S. K. Pandey, A. Kumar, P. Banerjee, B. Kumar, K. U. Singh, T. Singh, and P. Jha, “Semi-supervised clustering-based dana algorithm for data gathering and disease detection in healthcare wireless sensor networks (wsn),” *Sensors (Basel, Switzerland)*, vol. 24, 2023. [Online]. Available: <https://api.semanticscholar.org/CorpusID:266381207>
- [18] C. V. Nguyen, H. X. Nguyen, D. Anh, and C. Do, “Vindigitizer: An image processing approach to digitize paper ecg records,” in *2024 Computing in Cardiology Conference*, 2024.
- [19] S. Mishra, G. Khatwani, R. Patil, D. Sapariya, V. Shah, D. Parmar, S. Dinesh, P. Daphal, and N. Mehendale, “Ecg paper record digitization

- and diagnosis using deep learning,” *Journal of Medical and Biological Engineering*, vol. 41, no. 4, pp. 422–432, Aug. 2021. [Online]. Available: <https://doi.org/10.1007/s40846-021-00632-0>
- [20] H.-T. Yang, G. Nan, M. Lin, F. Chao, Y. Shen, K. Li, and R. Ji, “Lab-net: Lab color-space oriented lightweight network for shadow removal,” *ArXiv*, vol. abs/2208.13039, 2022. [Online]. Available: <https://api.semanticscholar.org/CorpusID:251903420>
- [21] J. Wang, X. Li, L. Hui, and J. Yang, “Stacked conditional generative adversarial networks for jointly learning shadow detection and shadow removal,” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1788–1797, 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:413731>
- [22] N. Chauhan and B.-J. Choi, “Performance analysis of denoising algorithms for human brain image,” *Int. J. Fuzzy Log. Intell. Syst.*, vol. 18, pp. 175–181, 2018. [Online]. Available: <https://api.semanticscholar.org/CorpusID:69356704>
- [23] S. Ganesh, P. T. Bhatti, M. Alkhalaf, S. Gupta, A. J. Shah, and S. Tridandapani, “Combining optical character recognition with paper ecg digitization,” *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 9, 2021. [Online]. Available: <https://api.semanticscholar.org/CorpusID:235755938>
- [24] “InkScape Website,” <http://www.inkscape.org/>.
- [25] J. Taylor and J. G. Hoole, “Robust Protocol for Sending Synchronisation Pulse and RS-232 Communication over Single Low Quality Twisted Pair Cable,” in *Proceeding of ICIT*. Taiwan: IEEE, Mar. 2016.
- [26] “xkcd: Thesis defense,” <https://xkcd.com/1403/>, (Accessed on 10/20/2017).
-

- [27] “Big bang - warm kitty, soft kitty (sheldon’s lullaby sick song) instrumental version lyrics — metrolyrics,” <http://www.metrolyrics.com/warm-kitty-soft-kitty-sheldons-lullaby-sick-song-instrumental-version-lyrics-big-bang.html?ModPagespeed=noscript>, (Accessed on 10/20/2017).
- [28] M. Shaw, “Writing good software engineering research papers,” in *Software Engineering, 2003. Proceedings. 25th International Conference on.* IEEE, 2003, pp. 726–736.
- [29] B. Paltridge, “Thesis and dissertation writing: an examination of published advice and actual practice,” *English for Specific Purposes*, vol. 21, no. 2, pp. 125–143, 2002.
- [30] Y. Li, Q. Qu, M. Wang, L. Yu, J. Wang, L. Shen, and K. He, “Deep learning for digitizing highly noisy paper-based ecg records,” *Computers in Biology and Medicine*, vol. 127, p. 104077, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S001048252030408X>
- [31] H. Tun, W. K. Moe, and Z. M. Naing, “Analysis on conversion process from paper record ecg to computer based ecg,” *Applied Bionics and Biomechanics*, vol. 1, pp. 69–81, sep 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:196033483>
- [32] B. Martínez-Pérez, I. de la Torre-Díez, M. López-Coronado, and B. Sainz-De-Abaño, “Mobile apps in cardiology: review,” *JMIR mHealth and uHealth*, vol. 2, no. 1, p. e1, 2014.
- [33] U. Utsha *et al.*, “A smartphone application for beat-by-beat ecg signal (visualization),” *Unspecified (science direct abstract)*, 2024, article focused on a smart-health app for ECG visualization and monitoring. [Online]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S2352648324000011>



- [34] K. Fatema, S. Montaha, M. A. H. Rony, S. Azam, M. Z. Hasan, and M. Jonkman, “A robust framework combining image processing and deep learning hybrid model to classify cardiovascular diseases using a limited number of paper-based complex ecg images,” *Biomedicines*, vol. 10, 2022. [Online]. Available: <https://api.semanticscholar.org/CorpusID:253427401>
- [35] K. K. Shivashankara, Deepanshi, A. M. Shervedani, G. D. Clifford, M. A. Reyna, and R. Sameni, “Ecg-image-kit: a synthetic image generation toolbox to facilitate deep learning-based electrocardiogram digitization,” *Physiological Measurement*, vol. 45, 2023. [Online]. Available: <https://api.semanticscholar.org/CorpusID:259342491>
- [36] Interactive Biology, “The ultimate cardiac cycle video – most comprehensive on youtube!” <https://www.youtube.com/watch?v=toZI5997hz8>, 2025, youtube video.
- [37] U. Eco, *How to write a thesis*. MIT Press, 2015.
- [38] I. Graessler, J. Hentze, and T. Bruckmann, “[V-models for interdisciplinary systems engineering](#),” in *DS 92: Proceedings of the DESIGN 2018 15th International Design Conference*, 2018, pp. 747–756.
- [39] W. Royce, “The Software Lifecycle Model (Waterfall Model),” in *Proceedings of WESTCON*, Aug. 1070.
- [40] B. W. Boehm, “[A spiral model of software development and enhancement](#),” *Computer*, vol. 21, no. 5, pp. 61–72, May 1988.
- [41] T. Oetiker, H. Partl, I. Hyna, and E. Schlegl, “[The Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>](#),” <https://tobi.oetiker.ch/lshort/lshort.pdf>, Jul. 2015, version 5.05.

- [42] “IEEE Conference Paper Templates,” [http://www.ieee.org/conferences\\_events/conferences/publishing/templates.html](http://www.ieee.org/conferences_events/conferences/publishing/templates.html).
- [43] A. Baboon, B. Charles, D. Ester, and F. Generalson, “An Amazing Title,” Their Not-so-awesome University, Technical Report, Apr. 1492.
- [44] B. van der Zander, J. Sundermeyer, and T. Hoffmann, “[TeXstudio – A L<sup>A</sup>T<sub>E</sub>X Editor](https://www.texstudio.org/),” <https://www.texstudio.org/>.
- [45] “Voluntary Page and Overlength Article Charges,” <http://www.ieee.org/advertisement/2012vpcopc.pdf>, 2014.

# Appendix A

## Supporting Data

Appendix sections are where you can place large figures, data tables, and spinets of code. Use appendices to your benefit to keep the body of your thesis concise!

The lyrics found below are for your enjoyment, but also serve an important role in demonstrating latex syntax for formatting text and in-text citations.

### A.1 Lyrics to Soft Kitty

Soft kitty, warm kitty  
Little ball of fur  
Happy kitty, sleepy kitty  
Purr, purr, purr

This has been brought to you by Sheldon Cooper [\[27\]](#)

# Appendix B

## Satirical Support

This section provides some comic relief. In addition, it serves as an example of how to insert an image into your thesis with proper caption, label and citation.



Figure B.1: You must prepare to defend your thesis [26]