



School of Mechanical and Manufacturing Engineering

The University of New South Wales

MMAN4020 - Practice Thesis B

Trimester 2 - 2019

Wear Tech (Health Group 2)

Date of Submission: 25th Aug, 2019

Student Name	Portfolio	Student Number
Chuhao (Martin) Cai	Data Communication	5057700
Aidan Cornwall	Power	5016188
Zachary Hamid	Data Analysis	5059915
Ping Li	Physical Housing	5060649
Calandra Lunardo	Data Analysis	5061090
Jonathan Stacey	Electrode Design	5061920

Consolidated Conclusive Statements

Electrode Design - Jonathan Stacey

The key purpose behind the electrode system was to prevent motion of the electrode relative to the skin and reduce interference. In the final design, a spring pushes the silicone-surrounded electrode against the skin and maintains the pressure to ensure the friction at the interface achieves these goals. The component and material selections were carried out such that the design life of the electrode subsystem exceeds the reasonably necessary time-frame for target user in India, lasting approximately 55 years. Furthermore, electrode spring system functions without additional electricity, making it indefinitely sustainable in terms of its energy usage. With its low overall cost and simple operation, the electrode system was found to be economically and technically feasible for application in India. If future development were to occur, it is recommended that the system be adapted for coping with ECG stress tests to further aid cardiologists in India.

Physical Housing - Ping Li

The physical housing portfolio is an essential component of the wearable device. It not only serves as the component that combines and connects all of the other physical components together, but is vital in the device achieving its aims and objectives. The vest design of the housing together with the electrode pin design allows the device to maximise its functionality through the flexibility of electrode placement, at the same time providing a secure fit for the electrode to produce optimum ECG signals. The selected materials of the housing such as polyester and polyester blend enables the device to be pleasant and easy for the patient to use while meeting all the criteria necessary to support the functionality of the device. And the relatively low cost and simple manufacturing of the housing design presents the device an opportunity to compete with existing options and adds real world value to the device, while the life cycle analysis offers insight into the environment impact of the design and presents a critical perspective, providing the device with chance future improvement.

Data Analysis I - Calandra Lunardo

The final diagnosis of the monitoring device depends on establishing a link between the health and technology aspect of this project scope. A simultaneous understanding of Coronary Artery Disease (CAD), the electrocardiogram (ECG) data representing CAD, and the 2-dimensional convolutional neural network (2D-CNN) as the method of data analysis is vital in returning a successful end output. This segment determined the possible output classifications of CAD for the 2D-CNN model to return a diagnosis of CAD. This final classification was determined based on the available ECG databases, the technical feasibility of training and creating a functional 2D-CNN, and the relevance of the diagnosis to the patients in India. With the final output classification identified, the appropriate input for the 2D-CNN could then be established. ECG datasets were sourced, extracted, processed, and plotted for feature analysis to distinguish and classify accordingly to create the final model of the ECG complex image to feed into the 2D-CNN. Since creating a working 2D-CNN was the end goal, a total of 19,565 ECG complex images were plotted. 8,924 stable ECG complex images were classified for the stable dataset and 10,641 unstable ECG complex images for the unstable dataset for the 2D-CNN to train with.

Data Analysis II - Zachary Hamid

A 2-dimensional convolutional neural network was designed and implemented for automated classification of 12-lead single-beat ECG images into the categories of stable and unstable. The network was found to achieve an accuracy of 99.18% with a sensitivity and specificity of 99.11% and 99.22% respectively on a testing set of 1,954 preprocessed 12-lead single-beat ECG images containing stable and unstable ECGs sourced from the Staff III and PTB databases. It is recommended that the final trained model be tested on new data not contained in the aforementioned databases in the future in order to ascertain a definitive conclusion that overfitting of data has not contributed to the obtained results.

Due to issues unrelated to technical feasibility, deployment to the Raspberry Pi Zero W was not possible in time for this report. The convolutional neural network was instead deployed to a Raspberry Pi 3B+ for determining performance of the network on low computing power devices, and additionally, obtain estimates for the possible performance on a Raspberry Pi Zero W. The network was found to be able to make a prediction within approximately

5.3 seconds on the Raspberry Pi 3B+ and therefore it was determined that this network could be feasibly implemented on the Raspberry Pi Zero W, and is something that should be considered for future work.

Data Communication - Chuhao (Martin) Cai

Remote medical treatment is closely related to data communication. It requires that the data should be exchanged securely, stably and efficiently. A concept of the Internet of Medical Things (IoMT) is implemented to improve remotely monitored health informatics. The IoMT, a combination of remote wearable diagnostic devices and applications, which can connect to associated health care system using networking technologies. The Cloud-connected diagnostic device not only can send the ECG data directly but also can save the data in the secure cloud environment. The advantage of using the cloud to store ECG data are zero-maintenance, high durability and security. In addition, the cloud is considered as an economical storage service which is only for \$0.02 per GB. Creating a smartphone-based user interface can greatly improve communication efficiency so that improving the treatment effect. To solve the urgent situation, it is suggested that sending push notification on smartphones or smartwatches rather than personal computers.

Power - Aidan Cornwall

Without a power source, the device would not be able to function, it is essentially the heart of the device. The power consumption of each component was calculated, as well as the energy required for a single reading. The most crucial and sensitive component of the wearable device was the Raspberry Pi as it requires a very stable 5 Volt DC supply and cannot operate with a voltage lower than 4.8 Volts. It was found that these computers can be powered by portable power banks as they output the optimal voltage for the Raspberry Pi. A suitable power bank was selected and was found to have enough energy capacity for 3 readings while being cheap and in a robust housing. The power bank can be charged by simply plugging a cable into a power socket, and requires almost zero maintenance. A screening life cycle analysis was performed of lithium ion batteries to narrow the focus for further studies.

Contents

1	Introduction	1
2	Electrode Design - Jonathan Stacey	3
2.1	Methodology	3
2.1.1	Medical informatics wearable project constraints	4
2.1.2	Literature review findings	4
2.2	Design objectives	5
2.2.1	No relative motion	6
2.2.2	Ergonomics	6
2.2.3	Correct positioning	6
2.3	Design concepts	7
2.3.1	Individual suction	8
2.3.2	Collective suction	8
2.3.3	Spring system	8
2.3.4	Spring system grip	8
2.3.5	Conductive foam	9
2.3.6	Elastic material	9
2.3.7	Skin clips	9
2.3.8	Lever suction	9
2.3.9	Protrusion	10
2.3.10	Design decision	10
2.4	Design development	11
2.5	Final design	17
2.5.1	Materials	19
2.6	Force analysis	19
2.7	Cost analysis	21
2.7.1	Material cost	21
2.7.2	Manufacturing cost	22
2.8	Life cycle analysis	24
2.8.1	Electrode	24

2.8.2	Spring	24
2.9	Future work	26
3	Physical Housing - Ping Li	27
3.1	Aim and Objective	27
3.2	Methodology	30
3.3	Literature Review Findings	32
3.4	Design	32
3.4.1	Design Concepts	32
3.5	Decision Process	40
3.6	Final Design	41
3.6.1	CAD Modelling	41
3.6.2	Materials Selection	44
3.6.3	Calculations	45
3.6.4	Material and Manufacturing Cost Analysis	46
3.6.5	Life Cycle Analysis	48
3.7	Conclusion	52
4	Data Analysis I - Calandra Lunardo	54
4.1	Aims & Objectives	54
4.2	Research: Understanding the ECG & its Features	55
4.2.1	The ECG Complex	55
4.2.2	The 12-Lead ECG	56
4.2.3	Coronary Artery Disease	58
4.2.4	ECG of Acute Coronary Syndrome	60
4.2.4.1	STEMI	60
4.2.4.2	NSTEMI	60
4.2.4.3	Unstable Angina	61
4.2.5	ECG of Stable CAD Patients	62
4.2.6	Ideal 2D-CNN Output Classifications	62
4.3	Approach: Obtaining ECG Databases	63
4.3.1	PhysioNet Database	63
4.3.1.1	PTB Diagnostic ECG Database	63

4.3.1.2	STAFF-III Database	65
4.3.2	Finalizing 2D-CNN Output Classification and Context of Device Use	65
4.3.2.1	Stable	66
4.3.2.2	Unstable	66
4.4	Methodology: Extracting & Processing ECG Signals from Databases	67
4.4.1	Extracting ECG Signals	68
4.4.1.1	Accessing ECG signal files	68
4.4.1.2	Downloading and storing the ECG signal into a matrix	68
4.4.1.3	Creating an ECG signal CSV file	69
4.4.2	Processing ECG Voltage Signals	71
4.4.2.1	Processing the ECG Signal for Beat Segmentation	71
4.4.2.2	Plotting ECG Image into PNG file	71
4.4.2.3	Image Processing	72
4.5	Results: ECG Image & Classification for 2D-CNN	72
4.5.1	Stable ECG Complex Image	73
4.5.2	Unstable ECG Complex Image	74
4.6	Future Work	74
5	Data Analysis II - Zachary Hamid	77
5.1	Aims & Objectives	77
5.2	Methodology	78
5.2.1	Development of a 2D CNN	78
5.2.1.1	Convolution layers	78
5.2.1.2	Pooling layers	80
5.2.1.3	Densely connected layers	81
5.2.1.4	Activation function	82
5.2.1.5	Regularisation	84
5.2.1.6	Loss function	86
5.2.1.7	Optimiser	88
5.2.2	Implementation and training of 2D CNN	89
5.2.2.1	Splitting the dataset	90
5.2.2.2	Deployment to Raspberry Pi	91
5.3	Final 2D CNN architecture	91

5.3.1	Technical hyperparameters	93
5.4	Results & Discussion	94
5.4.1	2D CNN performance	94
5.4.2	Raspberry Pi 3B+ performance	97
5.5	Future work	99
6	Data Communication - Chuhao (Martin) Cai	100
6.1	Aims and Objectives	100
6.2	Raspberry Pi Zero W Setup	102
6.2.1	Structure of the Approach	102
6.2.2	Methodology	103
6.2.2.1	Headless Raspberry Pi Zero W Setup	103
6.2.2.2	Wi-Fi Connection Test	105
6.2.2.3	Bluetooth Connection Test	106
6.2.2.4	Remote Control	106
6.2.3	Results and discussion	108
6.2.3.1	Headless Raspberry Pi Zero W Setup	108
6.2.3.2	Wi-Fi connection test	108
6.2.3.3	Bluetooth connection test	109
6.2.3.4	Remote control	110
6.3	Data Upload	111
6.3.1	Structure of the Approach	111
6.3.2	Methodology	112
6.3.2.1	MQTT protocol	112
6.3.2.2	Dropbox Cloud	114
6.3.2.3	Amazon S3 Bucket	116
6.3.3	Results and discussion	118
6.3.3.1	MQTT protocol	118
6.3.3.2	Dropbox Cloud	119
6.3.3.3	Amazon S3 Bucket	120
6.3.3.4	Comparison	120
6.4	User Interface	122
6.4.1	Structure of the Approach	122

6.4.2	Methodology	123
6.4.2.1	Push Notification	123
6.4.2.2	User Interface Design	125
6.4.3	Results and discussion	126
6.4.3.1	Push Notification	126
6.4.3.2	User Interface Design	127
6.5	Future Work	128
7	Power - Aidan Cornwall	129
7.1	Aim and Objectives	129
7.2	Methodology	129
7.3	Power Consumption	130
7.4	Feasibility of Previous Electricity Generation Methods	132
7.5	Rechargeable Battery Power Feasibility	132
7.6	How Batteries Work	133
7.7	Power Supply Options	135
7.7.1	Option 1	137
7.7.2	Option 2	139
7.7.3	Option 3	140
7.7.4	Chosen Power Supply	141
7.8	Life Cycle Analysis	142
7.8.1	Lithium Ion Batteries	142
7.8.2	Raspberry Pi	146
7.9	Future Work	147
8	Conclusion	148
9	References	149
9.1	Introduction	149
9.2	Electrode Design - Jonathan Stacey	149
9.3	Physical Housing - Ping Li	150
9.4	Data Analysis - Calandra Lunardo	150
9.5	Data Analysis - Zachary Hamid	152
9.6	Data Communication - Chuhao (Martin) Cai	154

9.7 Power - Aidan Cornwall	155
Appendices	161
A Spring Data Sheets	161
B Electrode Material Sources	163
C MATLAB Extraction Code	164
D Python Extraction & Image Processing Code	166
E Python Image Filtering Code	169
F 2D CNN Python Code	171
G 2D CNN Prediction Testing Python Code	177
H Raspberry Pi 3B+ Testing Output	181

List of Figures

1 Scope and usage process	2
2 Design flowchart	3
3 Electrode system concepts	7
4 Chosen concepts for development	11
5 Initial CAD design	12
6 CAD design 2	13
7 CAD design 3	14
8 CAD design 4	15
9 CAD design 5	16
10 CAD of final design	17
11 Final design in extended and compact positions	18
12 Free body diagram of an electrode at the surface of the skin	19
13 Project plan	26
14 Concept 1: Cover Vest	33

15	Electrode Pin Design - Expanded view	35
16	Concept 2: Holy shirt design	36
17	Concept3: Wire sash design	38
18	CAD render of final design	41
19	Strap and buckle (Wisdompro)	42
20	Base vest with electrode pins(Underarmour)	43
21	Belt clip	44
22	Normal ECG beat (Hampton, 2013)	55
23	12-lead normal ECG (Hampton, 2013)	58
24	ECG indicating STEMI (Hampton, 2013)	60
25	ECG indicating NSTEMI (Hampton, 2013)	61
26	Final image of sample stable ECG complex	73
27	Final image of sample unstable ECG complex	74
28	Convolution Process Example (Yamashita, Nishio, Do & Togashi, 2018) . .	79
29	Max vs. Average Pooling (kernel size: 2x2; stride: 2) (Hamid, 2019) . . .	81
30	Artificial Neuron (Ahire, 2018)	82
31	Neural Network (Wikipedia, 2019)	82
32	Effect of overfitting data (Despois, 2018)	84
33	Effect of dropout on a neural network (Srivastava et al., 2014)	85
34	Simplified visualisation of devised 2D CNN architecture	92
35	Technical visualisation of devised 2D CNN architecture	92
36	Validation accuracy and loss per epoch	94
37	Prediction outputs of CNN model on 30 image dataset on Raspberry Pi 3B+ .	98
38	Network of IMoT	100
39	Raspberry Pi Zero W	101
40	Structure - Raspberry Pi Zero W Setup	102
41	Green LED on Raspberry Pi Zero W	104
42	Set up Network on GNU nano	105
43	IP address for Raspberry Pi Zero W	105
44	Command for Opening Bluetooth in Raspberry Pi	106
45	Bluetooth Pairing	106
46	Booting Raspberry pi	107
47	Result for Raspberry Pi Zero W setup	108

48	ping Test from smartphone	109
49	ping Test from Raspberry Pi Zero W	109
50	Bluetooth connection showing with laptop ID	110
51	Connection Failed	110
52	Structure - Data Upload	111
53	Communication via MQTT protocol	112
54	Certificate and private secret key	113
55	Code - connecting to AWS MQTT client	113
56	AWSIoTMQTTClient SDK installation	114
57	Code - uploading ECG data	114
58	MQTT.fx configuration	115
59	Code - sending imagefile to Dropbox	116
60	Flowchart - sending image file to S3 Bucket	116
61	Add user to IAM	117
62	Attach Policy to S3 Bucket	117
63	Code - Seding image file to AWS S3 Bucket	118
64	Result sent by Python	118
65	Result showing on MQTT.fx	119
66	Result - test01.png showing in the Dropbox file 'patient01'	120
67	ecgpatient01.png showing in the AWS S3 Bucket	120
68	Structure - User Interface	122
69	Flowchart - Push notification	123
70	Code - sending emergency message to real-time database	124
71	Code - Allowing to read and write in database	124
72	User Interface developed by Users' needs	125
73	Emergency flag in FCM database	126
74	Push notification pop up	127
75	Conceptual Design - User Interface	127
76	Circuit Diagram	131
77	Rechargeable Battery Pack	137
78	Discharge Curve at Different Temperatures	138
79	Sparkfun Electronics Power Bank	139
80	USB Battery Pack for Raspberry Pi	140

81	General Life Cycle of Lithium Ion Batteries	142
82	General Life Cycle of a Raspberry Pi	146

List of Tables

1	Design objective comparison	5
2	Concept decision matrix	10
3	Improvements to design 1	12
4	Improvements to design 2	13
5	Improvements to design 3	14
6	Improvements to design 4	15
7	Improvements to design 5	16
8	Raw material costs	22
9	Manufacturing cost estimates	23
10	Total cost estimates	23
11	Pugh Matrix for concept selection	40
12	Raw material costs	47
13	Manufacturing cost estimates	47
14	Various classifications from PTB Diagnostic ECG Database	64
15	Comparison of AFs	83
16	Optimisation methods	88
17	Split of ECG images	91
18	Technical parameters of convolution layers of devised model	93
19	Technical parameters of densely connected neural layers of devised model	93
20	Technical parameters of max pooling layers of devised model	93
21	Performance of network on validation and test datasets	95
22	Confusion matrix of network on testing set	95
23	Network prediction times for 30 ECG images on Raspberry Pi 3B+	97
24	Confusion matrix of network on 30 image testing set on Raspberry Pi 3B+	98
25	Difference between Remote Control Connection and RealVNC	111
26	Comparison of three different methods of data uploading	121
27	Wattage of Components	130
28	Battery Pack Options	136

1 Introduction

The quality of life of individuals is largely determined by their health. Unfortunately, many developing regions of the world are under-resourced which results in a lack of access to reliable medical care and equipment. This inevitably causes the deterioration of the health of large demographics, culminating in large numbers of preventable deaths. A 2016 study by the World Health Organisation into the Indian health workforce uncovered that in the region of urban India 41.6% of doctors were inadequately qualified (Anand & Fan, 2016, p. 9). To make matters even more urgent, this same study showed that India has an extremely imbalanced doctor to patient ratio of only five doctors for every ten thousand patients. Accordingly, the prevalence of widespread health issues in this area needs to be combatted. Cardiovascular diseases are responsible for 25% of mortality in India, with the highest proportion of these being due to ischemic heart disease (IHD), also called coronary artery disease (Prabhakaran, Jeemon & Roy, 2016). IHD is the narrowing of heart arteries due to plaque build-up, which limits the flow of essential oxygenated blood to the heart. Symptoms can include chest pain and tightness, breathlessness, drowsiness or heart attacks in extreme cases, but fortunately early detection and monitoring can be carried out using electrocardiograms (ECGs).

“The human heart muscle contractions forming a heartbeat are preceded by the generation of electrical signals that cause heart cells to depolarize, a mechanism by which neighbouring cells in the heart become positively charged due to calcium ion influx. The electrical signals generated can be detected and measured by electrodes on the skin, which is the basic principle behind an ECG” (Stacey, 2019).

This report details the inception and development of a wearable device to monitor coronary artery disease, called the Ischemic Heart Tracker (IHT), to help manage the doctor to patient imbalance and improve the health of people in India. The overall design is a compression shirt containing custom ECG measuring and analysis technology, and the envisioned scope and use of this product is outlined in Figure 1 below.

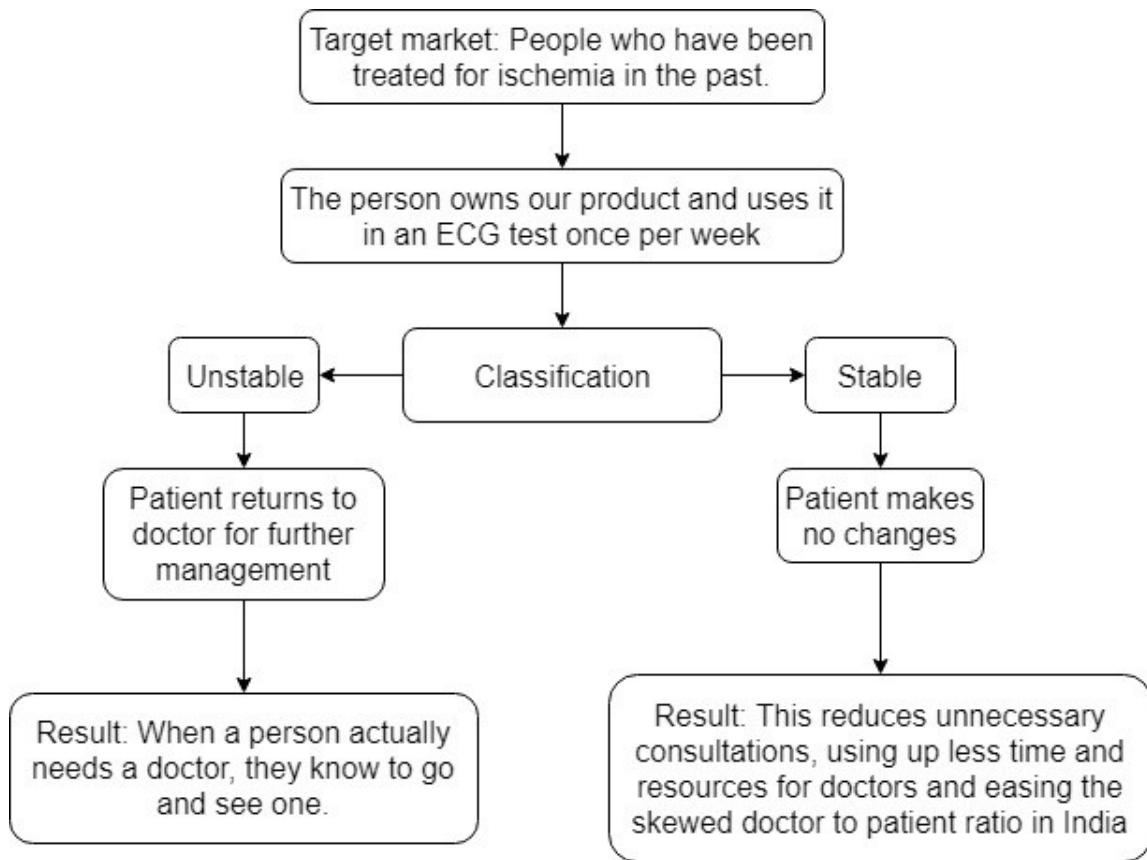


Figure 1: Scope and usage process

The approach with the IHT was to use a convolutional neural network (CNN) to automate the classification of 12-lead single-beat ECG images. An unstable classification means that the patient should consult their doctor, however a stable classification means they need not do so. This stable classification has the potential to free up the time of doctors by allowing patients to avoid unnecessary consultation and save money in the process.

2 Electrode Design - Jonathan Stacey

2.1 Methodology

To achieve the goal of creating electrodes capable of being incorporated into the Ischemic Heart Tracker, two sets of constraints were considered. The relationship between the design decisions and the constraints is shown in Figure 2 below.

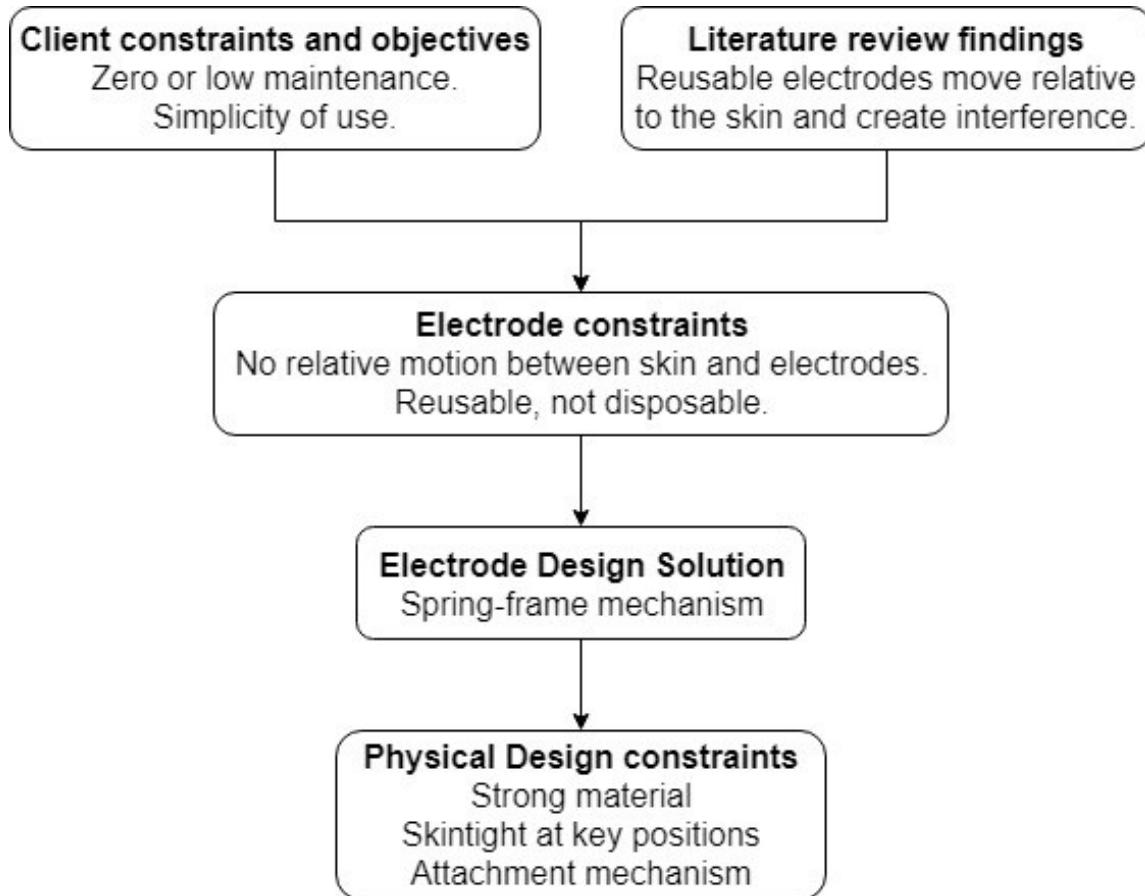


Figure 2: Design flowchart

2.1.1 Medical informatics wearable project constraints

The constraints provided were as follows:

1. Minimise unit cost
2. Simple or zero-maintenance requirement
3. Robustness
4. Simplicity that aids in use
5. Language and cultural adaptability

The constraints (2) and (4) were encompassed by the design objective; “Usable by the patient,” however the language constraint (5) was not applicable to this system as it is a small mechanical system requiring minimal interaction with the user.

2.1.2 Literature review findings

A literature review was carried out to determine the optimum quantity, placement and type of electrodes to use in this type of ECG. The primary decision was between disposable (wet) or reusable (dry) electrodes. It was found that dry electrodes would be more satisfactory for meeting the previous constraints C2 and C4:

Wet electrodes are effective and accurate despite their other drawbacks, but newer dry electrodes are superior in many ways even though they are still being researched, and in the future, they seem likely to dominate the market once they become cost effective because of their reusability and skin-friendliness. A wearable ECG vest or similar technology would therefore benefit from recognising the superiority of newer dry-electrodes arranged in the standard configuration over other electrode setups, however for stress testing the Mason-Likar placement would be preferable. (Stacey, 2019, p. 4)

2.2 Design objectives

The subsequent goals for the design of the electrode system were weighted according to their importance and are compared in Table 1.

Table 1: Design objective comparison

Design objective	Weighting score (1-3)
Important	
No relative motion	3
Usable by the patient	3
Correct positioning	3
Desirable	
Durable	2
Easily replaceable	2
Clear signal	2
Reduces interference	2
Easily moveable	2
Trivial	
Aesthetic form factor	1
Small	1

As a result of meeting the important requirements, some of the trivial and desirable objectives were naturally met, and as such the important objectives were explored more deeply in an earlier version of this report:

2.2.1 No relative motion

The movement of reusable electrodes is the greatest contributor to the interference and signal quality obtained from them (Dozio, Baba, Assambo, & Burke, 2007). These desirable constraints need to be minimised by restricting this motion, therefore for referencing purposes a coordinate axis will be defined as:

- X-axis: Around the body, parallel with the surface of the skin
- Y-axis: Along the length of the body from head to toe, parallel with the surface of the skin
- Z-axis: Directly out from the body, perpendicular to the surface of the skin

The motion of each electrode must be constrained along all axes to optimise the signal quality and reduce interference, although the motion is constrained by default in the negative z-direction by the skin.

2.2.2 Ergonomics

Disposable electrodes use sticky gel to attach directly to the skin, but they can cause skin irritation (Chen, et al., 2014) and would be very inconvenient if incorporated into a wearable ECG product. Therefore, sticky electrodes are surpassed by electrodes that use different mechanisms to restrict the motion along the skin.

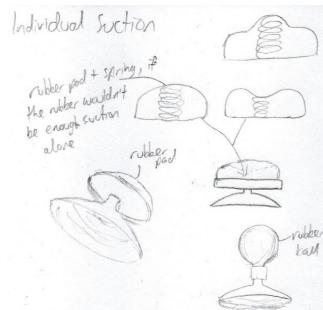
2.2.3 Correct positioning

The correct positioning of the electrodes is of paramount importance to obtain data from different regions of the heart; however, this goal is more dependent on the physical design than the electrode design. The impact this goal has on the design of the electrode is that the electrode position on the product must be changeable in some way.

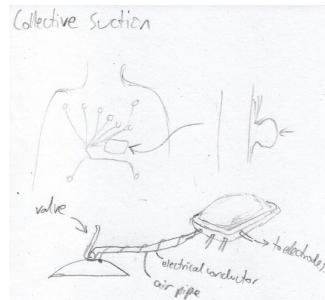
(Stacey, 2019, p. 4)

2.3 Design concepts

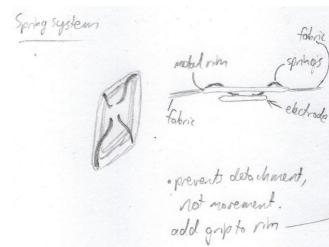
To meet these requirements and objectives, design concept sketches were created and explored, the some of which are shown in Figure 3 below.



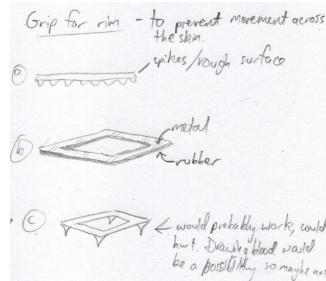
(a) Individual suction



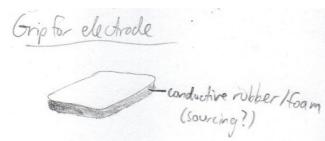
(b) Collective suction



(c) Spring system



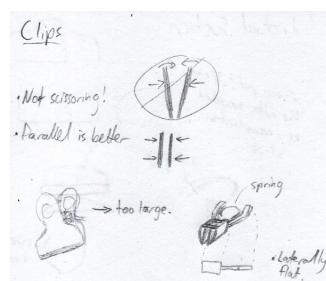
(d) Spring system grip



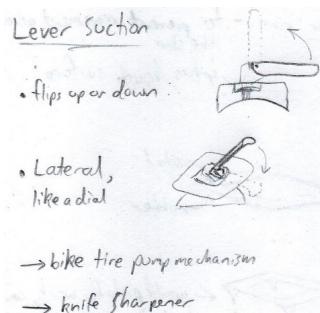
(e) Conductive foam



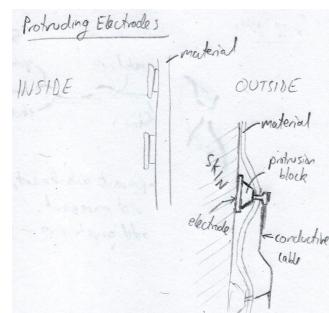
(f) Elastic material



(g) Skin clips



(h) Lever suction



(i) Protrusion

Figure 3: Electrode system concepts

Each of these concepts was evaluated in an earlier version of this report (Stacey, 2019) and the summaries of these are given below:

2.3.1 Individual suction

Shown in Figure 3a, in this design each slightly concave electrode has a rubber suction-creating ball or pad affixed above the electrode, similar to the typical reusable electrodes already on the market but with a flatter profile. The advantage is that it stays affixed, but the drawback is that this is a bulky design that may fall off.

2.3.2 Collective suction

Displayed above in Figure 3b, this concept is characterised by air tubes connecting all the concave electrodes to a central suction pad or balloon, allowing the suction force to all come from one place. One advantage of this idea is that the small, flat profile of the electrodes means they are less likely to fall. However, some disadvantages were the cumbersome design, and that this would have required valves at each electrode to enable them to be affixed one by one to the skin.

2.3.3 Spring system

Shown in Figure 3c, the spring system consists of a mechanism that exerts a force toward the skin to make the electrodes jut inwards from the material slightly. A benefit of this design is that it should prevent electrodes losing skin contact effectively. However, some downsides to this concept are that it does not prevent translation across the skin and requires close fitting/skin-tight material to be useful.

2.3.4 Spring system grip

As presented above in Figure 3d, this is a feature of the previous design to improve its function. The concept is a housing for the electrode with roughness, rubber or some other high friction mechanism, or spikes to prevent xy-plane motion. This is a useful feature because it can be used in conjunction with

other designs and has a flat profile, but it may not be sufficient for preventing relative motion if this is the only concept used to achieve this goal.

2.3.5 Conductive foam

Figure 3e shows a representation of a conductive foam or other soft material as the electrode contact. This has the advantage that it could prevent all relative motion from causing issues, however it is still a relatively new idea that may be expensive, underdeveloped or difficult to source.

2.3.6 Elastic material

As shown in Figure 3f, this is a physical design concept involving elastic material unified with the IHT in key areas or bands. This would help hold the electrodes in place, but might not work very well if a patient begins to move during their ECG.

2.3.7 Skin clips

Appearing as Figure 3g above, this idea was to use conductive clamps on the skin, and would completely fix the issue of relative movement. On the other hand, this would have had major shortcomings, including being unsanitary, needing more frequent replacement, and being very difficult for the patient to apply to themselves in the correct positioning if this were built into an item of clothing.

2.3.8 Lever suction

This concept, shown in Figure 3h, was to use a lever that pulls a flexible suction cup inwards when engaged. This would fix movement issues whilst maintaining a small form factor, but the suction may not be adequate. Similar to other designs, it could also be quite difficult to operate through a layer of clothing.

2.3.9 Protrusion

Figure 3i shows the protrusion idea, which is a simple geometric feature that could be used in conjunction with many other designs. In this idea, electrodes are not flush with the material of the product but instead stick inwards and are thus held against the skin. In addition to its compatibility, the goal of this design is to reduce motion in the z-direction, forcing the electrode to maintain its contact with the skin at all times. A limitation of this is that it requires the clothing material to be both skintight and reasonably strong, as well as to not move at all on the body.

(Stacey, 2019)

2.3.10 Design decision

Next, a decision matrix was used to determine which designs to develop further as shown in Table 2 below.

Table 2: Concept decision matrix

Design concept name	How well goals are met (0-5)									Weighted total
	Important			Desirable			Trivial			
Reusable adhesive	5	1	4	1	4	2	5	5	5	74
Individual suction	5	4	5	4	3	5	5	4	0	86
Collective suction	5	3	5	3	3	3	4	4	0	73
Spring system	4	5	5	5	3	5	4	4	4	92
Electrode grip	3	5	5	4	3	5	4	4	5	89
Elastic	3	5	4	4	5	1	3	3	5	73
Clips	5	3	4	4	4	5	5	5	3	89
Lever suction	5	3	5	4	3	5	5	4	3	88
Protruding electrode	2	5	5	5	5	3	2	2	3	77

The highest ranked concept was the spring system (blue), which was compatible with three of the other features conceived (light blue). The concept sketches for these four ideas are displayed together in Figure 4.

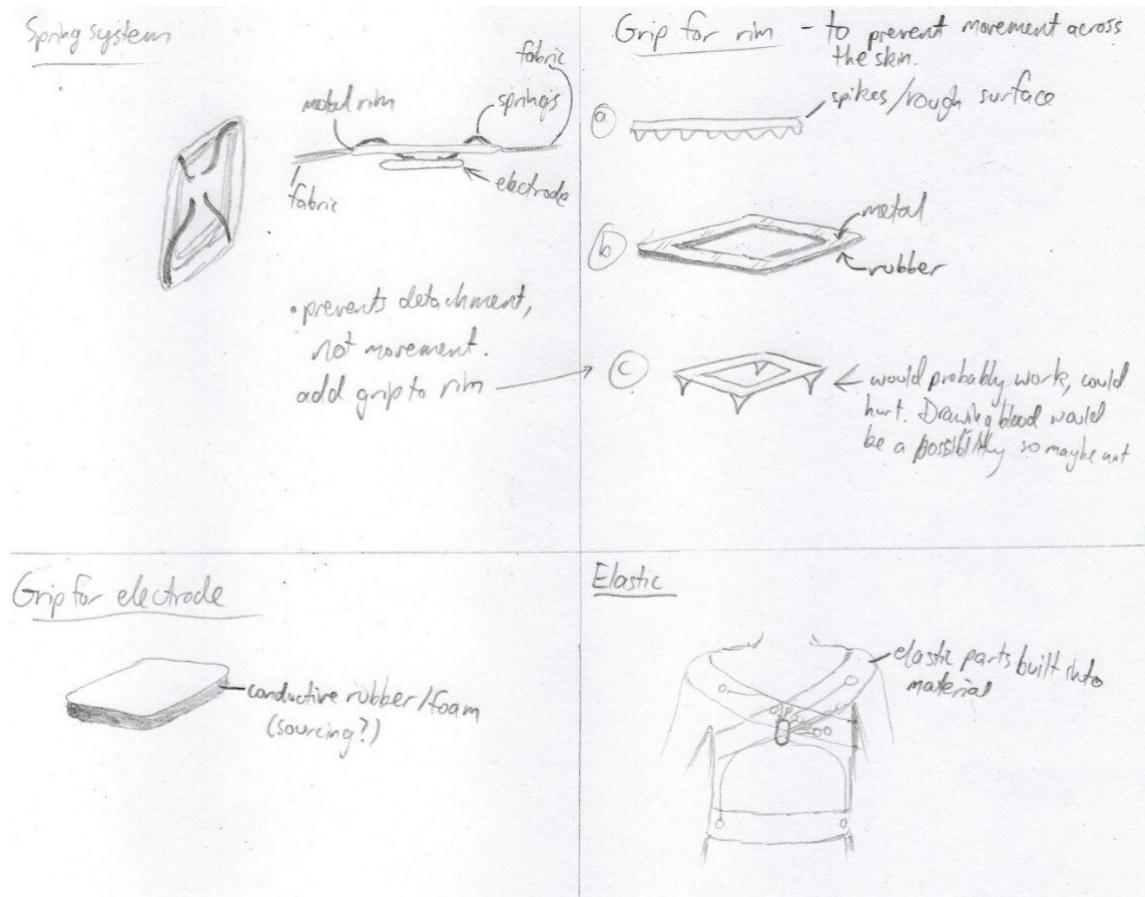


Figure 4: Chosen concepts for development

2.4 Design development

An iterative improvement approach was taken to create a more detailed model of the electrode subsystem. This subsystem aimed to include the four chosen concepts while still aiming to have a small form factor. Additionally, one of the physical design aspects was that this system needed a way to attach to the material of the compression shirt. The method selected was to base this mechanism on a lapel pin, therefore this was integrated into the electrode design from the beginning. Each iteration was evaluated and optimised for the subsequent model, beginning with the simple design shown in Figure 5 which made it easier to visualise the system, however many issues were immediately visible, as listed in Table 3.

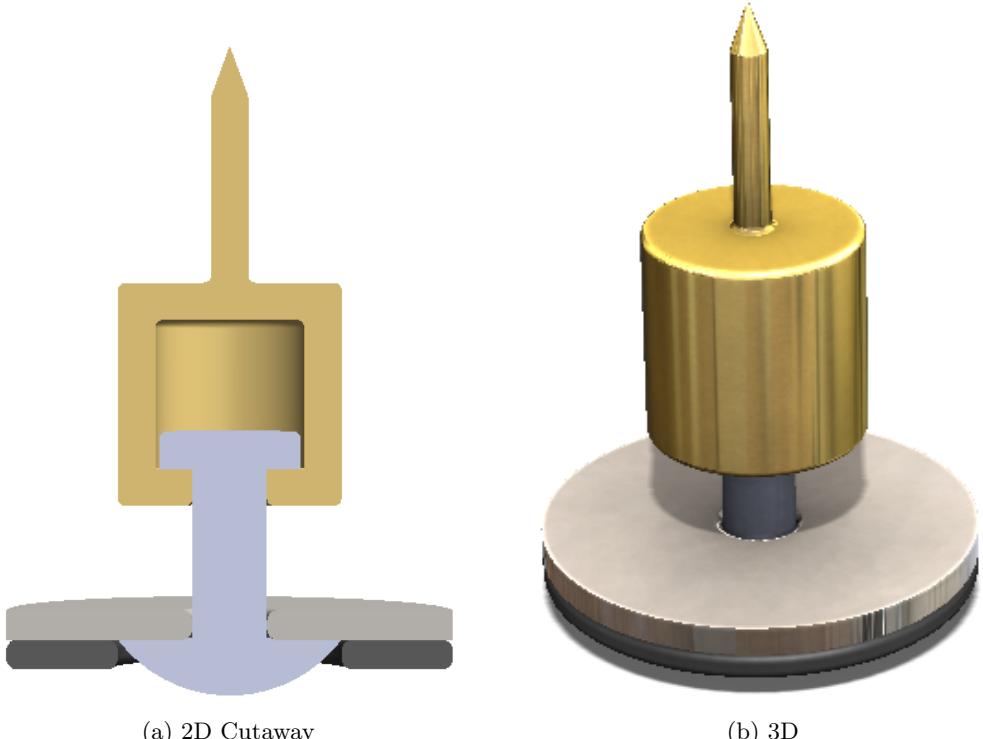


Figure 5: Initial CAD design

Table 3: Improvements to design 1

Problem	Solution
The system is required to fit between the skin and the material	Switch the position of the shaft into the silver electrode
Poor electrical signal because the electrode face provides a contact surface with the skin that is too small	Flatten the electrode contact surface
On medical ECG electrodes the conductive surface has a diameter of 10mm (Grassini, 2013)	Adjust dimensions
On medical ECG electrodes the adhesive surface has a diameter of 16mm (Grassini, 2013)	Adjust dimensions
The pin may not be long enough to adequately reach the clasp on the other side of the material	Adjust dimensions

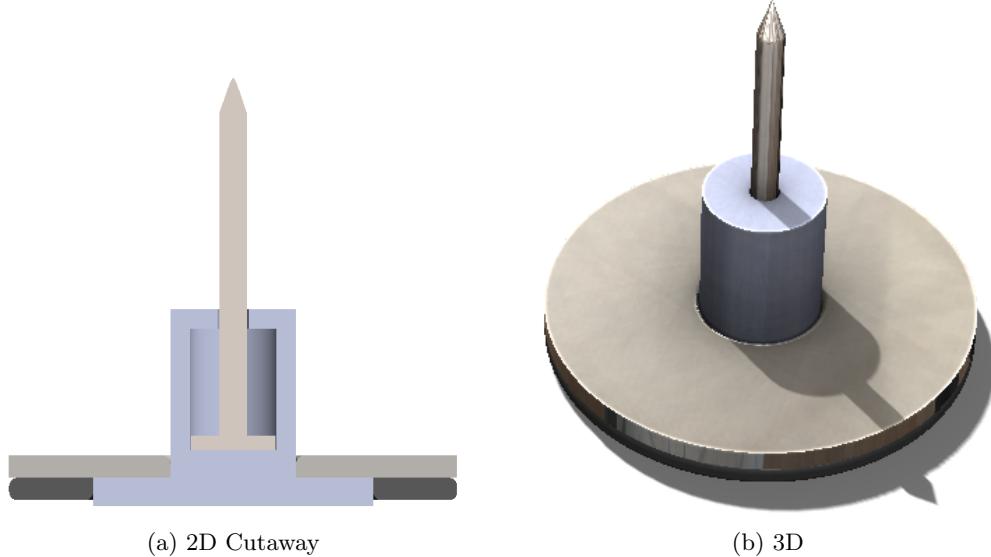


Figure 6: CAD design 2

The improvements from Table 3 were used to create the model shown in Figure 6. This version of the system is similar to the final design, however there were still some clear issues. One issue that was challenging to pinpoint was that there would be no leverage for the pin. If the clasp was attached to this iteration and the spring pushed against the pin, the clasp would move but the material would remain flush with the top surface of the electrode. This and other problems are shown in Table 4.

Table 4: Improvements to design 2

Problem	Solution
The shaft is not long enough because the length of the compressed spring was not considered	Extend the shaft down towards the skin
There is nothing for the pin to push against the material with, so the spring would just move the clasp out from the body	Create a surface attached to the pin for the material to rest against
Imprecise pin size	Adjust dimensions

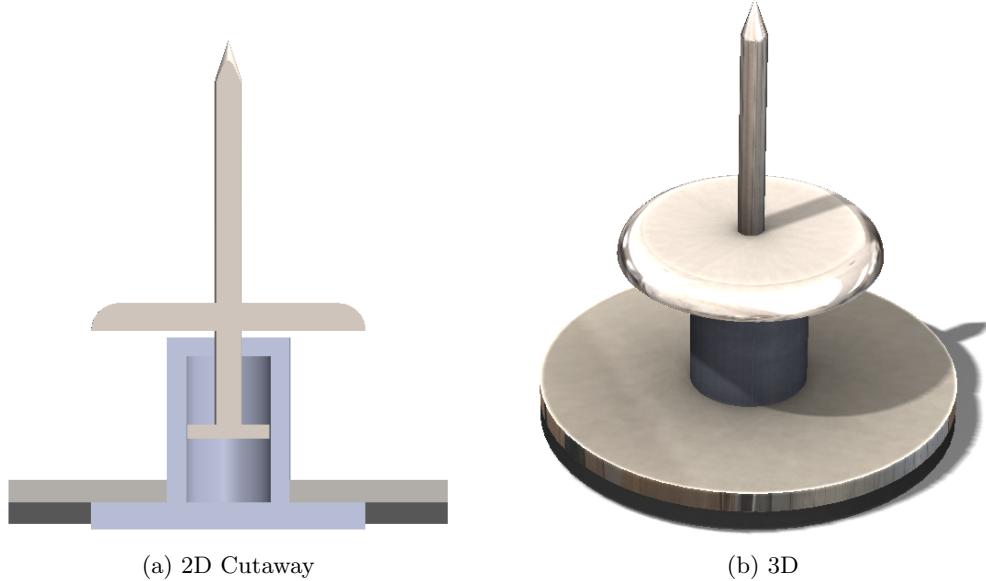


Figure 7: CAD design 3

The Computer Aided Design (CAD) displayed in Figure 7 above has had the obvious problems fixed, so a more detailed approach dictated the further design optimizations. Many of the changes made after this point were focused more on clarity, ease of use and specificity rather than on the basic structure of the system design. However, one more major improvement was required to allow the electrode to be held in the compact position until the Ischemic Heart Tracker was donned by the patient. The immediate changes made are shown in Table 5 below.

Table 5: Improvements to design 3

Problem	Solution
The shaft length is not tailored to the primary spring length	Choose primary spring, then adjust the shaft
The electrode protrudes too much, making the IHT difficult to don	Design a releasable mechanism to lock the electrode system in the compact position

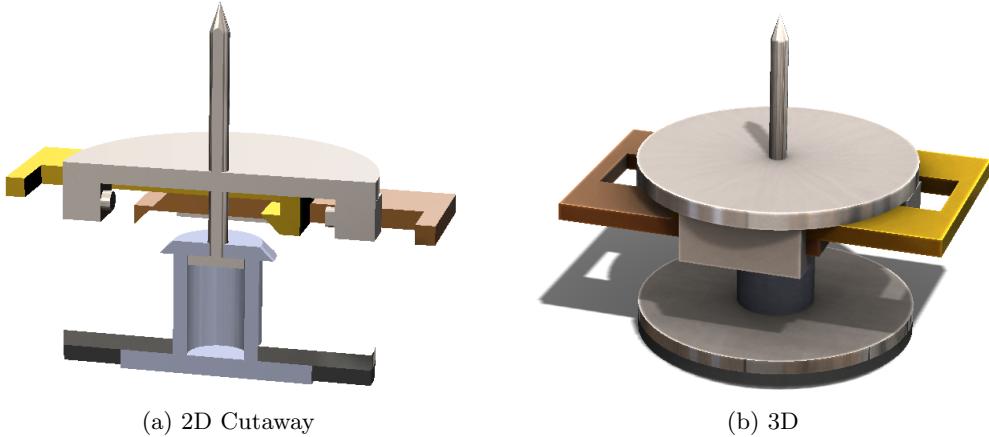


Figure 8: CAD design 4

As shown in yellow and brown in Figure 8 above, two extra components were added to the latch subsystem. These each slide along the brass rails (grey) and can hold the electrode (silver) in the compact position. This model only contains the initial CAD of this sub-assembly however, so the subsequent material designation and inclusion of more components according to Table 6 helped make the design clearer and more feasible.

Table 6: Improvements to design 4

Problem	Solution
Materials have not been assigned	Make informed material designations
Primary spring is missing from the CAD	Use the CAD from the spring supplier
The brown lower latch lacks strength on its left side	Double the thickness of the lower latch on the left side

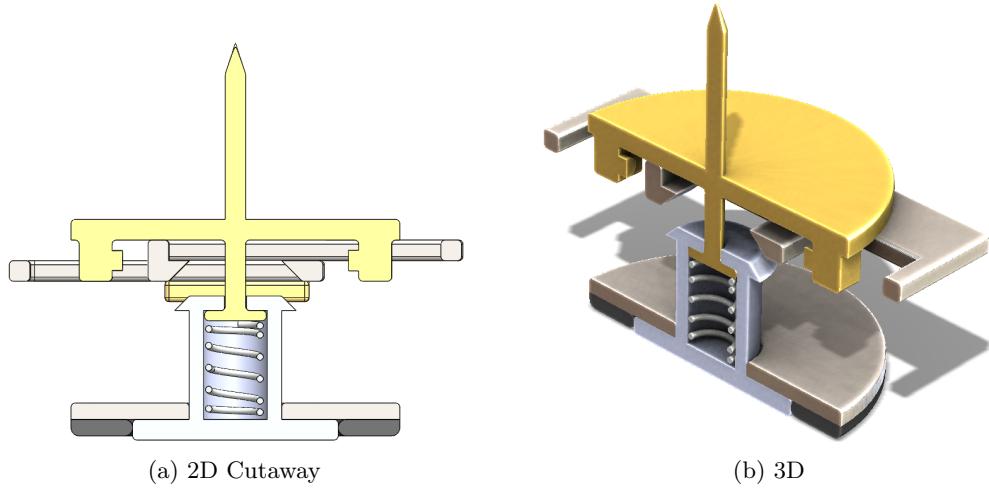


Figure 9: CAD design 5

Figure 9 shows an almost-complete version of the electrode assembly, with only a small number of changes to be made to prevent small errors such as the secondary springs falling from the system. The final iteration of optimizations is shown in Table 7.

Table 7: Improvements to design 5

Problem	Solution
Secondary springs are missing from the CAD	Use the CADs from the spring supplier
The right-hand secondary spring would slip off the system: The lower latch component does not have sufficient area for a secondary spring to connect with on the right-hand side of Figure 9	Extrude the latch component to create an adequate surface for the secondary spring to push against.
The secondary springs could fall out of the assembly: The latch components have no features to hold the secondary springs in place	Create thin dimples to hold the secondary springs in place

2.5 Final design

The final design is shown in Figure 10. In addition to the changes listed above, most sharp edges were filleted to enable the components to be die-cast more easily.

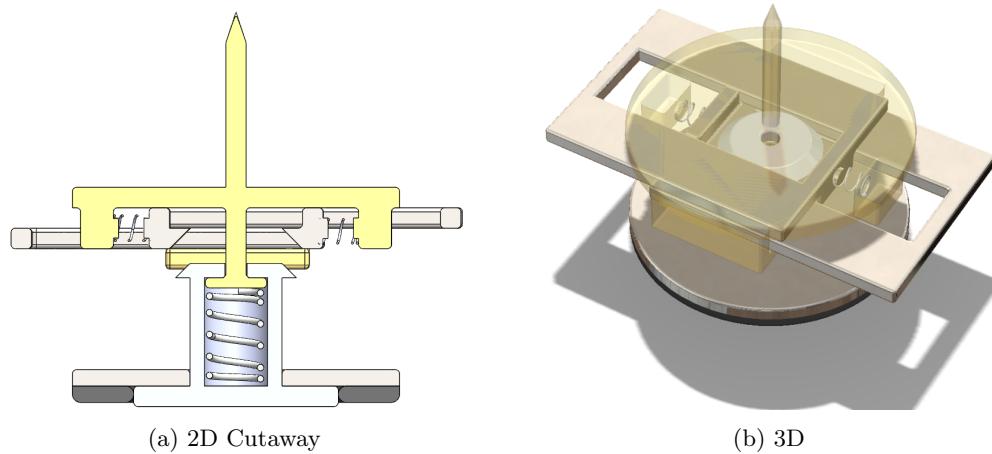


Figure 10: CAD of final design

The electrode system has two equilibrium states, namely *extended* and *compact*. The compact position is characterised by the primary spring being full compressed and the latch mechanism being engaged. This is how the electrodes are to be positioned whilst the patient is putting on or removing the IHT, to make the electrodes less likely to catch or snag on anything as they are moving. Once in place, the user can simply press on the two catch-releases, through the compression shirt material, to allow the primary spring to extend once more, pushing the electrode against the user's skin with a force of up to 9.61N (see Appendix A). These two positions are shown in different projections in Figure 11. Assuming full contact between the skin surface and the electrode, the distance between the skin and the material at the location of an electrode is only 8 mm when the electrode is in the compact position.

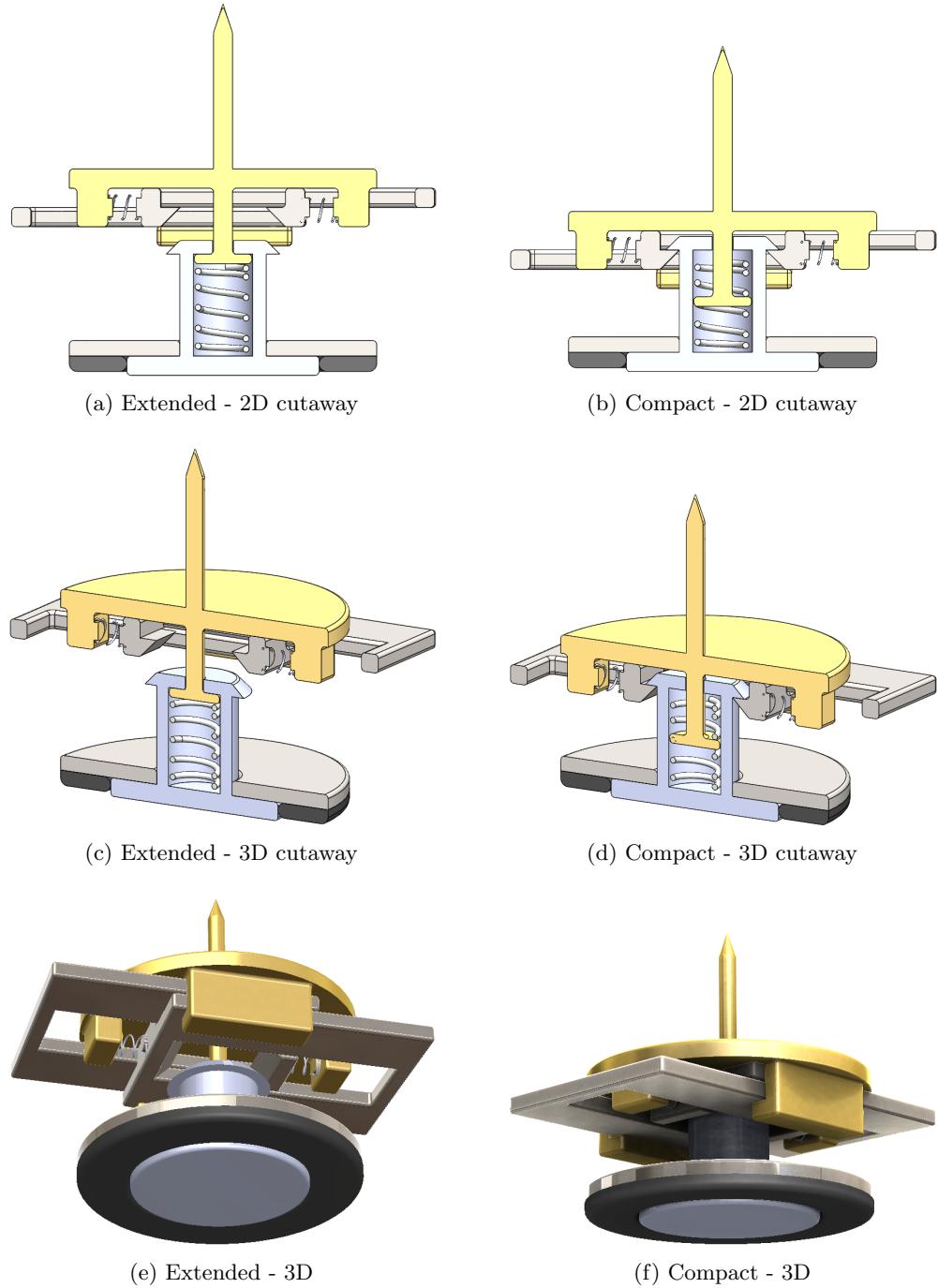


Figure 11: Final design in extended and compact positions

2.5.1 Materials

The electrode was designed to be composed of silver with a coating of silver chloride. This combination was chosen because it creates a superior electrode with high conduction and low interference (Tallgren, Vanhatalo, Kaila, & Voipio, 2005). The electrical signals are conducted through the silver to the brass pin. Composed of mostly copper, brass is a conductive alloy with high strength, making it suitable for being the base for all moving parts of the assembly. The signal is then transmitted via direct contact to the pin-clasp on the outside of the material where a connected wire receives the signal.

2.6 Force analysis

The goal of the primary spring was to exert a force on the electrode towards the skin of the patient to prevent the electrode from losing contact or moving relative to the skin even if the material were to shift or move. The electrode system and compression shirt will be in equilibrium before any motion occurs. Consequently, this short analysis assumes that the weight of the electrode is entirely supported by the compression shirt through the lapel-pin mechanism, and aims to determine the maximum lateral force across the skin that can be resisted by the spring-grip system. This calculation is based upon the free body diagram shown below in Figure 12.

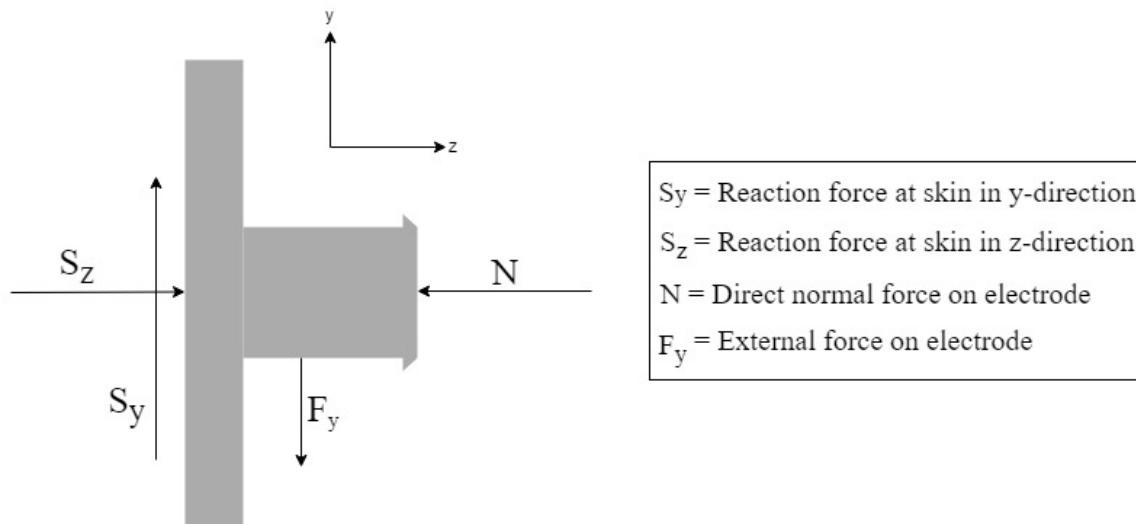


Figure 12: Free body diagram of an electrode at the surface of the skin

Sum of forces in the z-direction:

$$\begin{aligned}\sum F_z &= 0 \\ S_z &= N\end{aligned}\tag{1}$$

Sum of forces in the y-direction:

$$\begin{aligned}\sum F_y &= 0 \\ S_y - F_y &= 0 \\ S_y &= \mu_s N\end{aligned}\tag{2}$$

Hence:

$$\begin{aligned}\mu_s N - F_y &= 0 \\ F_y &= \mu_s N\end{aligned}\tag{3}$$

Making the assumption for this analysis that the force N in the z-direction is not from the material elasticity but from the primary spring alone, it follows that:

$$N_{max} = F_{spring, max}\tag{4}$$

The maximum force is delivered from the spring when it is fully compressed, which is equal to 9.61N (see Appendix A). As such, the maximum normal force N on the skin is also 9.61N. Therefore by combining Equation 2 and Equation 4, the maximum force lateral to the skin that can be countered by the frictional properties of the silicone is:

$$F_y = \mu_s F_{spring}\tag{5}$$

The coefficient of static friction μ_s is specific to the materials in contact. Experimentally, the coefficient of static friction between human skin and silicone has been calculated as 0.61 ± 0.21 (Zhang & Mak, 1999). This study took into account the variation in skin friction at different parts of the body, and whilst no tests were taken on the chest region the mean result can fairly be used. This introduces an uncertainty of 34.4% to the calculations, but as the analysis already excludes the pressure of the compressive shirt the analysis should still provide a reasonable approximation. Using this μ_s value gives:

$$\begin{aligned}F_{y, max} &= 0.61 \times 9.61 \\ F_{y, max} &= 5.86N\end{aligned}$$

This result can be applied as a vector of length 5.86 in any direction on the xy-plane, meaning it represents the maximum allowable force in any direction parallel to the skin, no just vertically. The Ischemic Heart Tracker is designed for stationary ECG tests, with little to no movement by the user during the ECG. However, a future application could be ECG stress tests which involve walking and running in accordance with the Bruce protocol (Queensland Health, 2012). The significance of the calculation result $F_{y, max} = 5.86N$ is connected with this potential future application, showing that the IHT could be used without drop-outs or interference provided the lateral force on the electrode does not exceed 5.86N. This occurs because below this threshold the electrode is held in place by the static friction created between the silicone and skin as a result of the direct force from the primary spring.

Another consideration to take into account is that the material of the compression shirt acts as a set of springs to hold the electrodes in position in all directions. This effect is being ignored in the z-direction as stated earlier to account for the uncertainty in the μ_s value, however in the xy-plane this characteristic should not be discounted. The isotropic linear elastic material generates a restorative force on any electrode that shifts from its natural position, ensuring that the placement of the electrodes remains consistent throughout each test.

2.7 Cost analysis

The cost of each component in the electrode system was determined by combining the raw material cost with the manufacturing cost. However, the springs and grip ring have no manufacturing cost given because they can be purchased already made at a relatively low cost.

2.7.1 Material cost

Using the Solidworks 2018 Mass Study evaluative feature, the exact volumes and masses of the assembly components were calculated. Combining these values with raw material cost information allowed an initial estimate to be made. The sources of the “Material Cost” data from Table 8 varied between components (see Appendix B) and is a non-applicable (N/A) field for the springs, but what this system needed to consider is that machined components (clasps and plate) require excess material to begin with. The updated masses and volumes

flowed on to slightly increase the costs, which is included in the values in Table 8.

Table 8: Raw material costs

Component	Material	Volume (mm ³)	Mass (g)	Material Cost	Cost/Unit
Plate	Chrome Stainless Steel	221.355	1.725	\$150/m ²	\$0.057
Grip Ring	Silicone	94.12	0.22	\$5.68/kg	\$0.001
Electrode	Silver	136.57	0.76	\$0.55/g	\$0.418
Pin	Brass	287.79	0.76	\$9/kg	\$0.007
Clasp 1	Stainless Steel	256.00	2.00	\$234/m ²	\$0.104
Clasp 2	Stainless Steel	257.66	2.00	\$234/m ²	\$0.104
Primary Spring	Alloy Steel SS	3.48	0.03	N/A	\$0.446
Secondary Spring 1	Alloy Steel SS	0.14	0.00	N/A	\$0.739
Secondary Spring 1	Alloy Steel SS	0.14	0.00	N/A	\$0.739
Total					\$2.615

2.7.2 Manufacturing cost

The manufacture of metal components can be carried out in several ways. The method is not important for the springs, because they can be bought pre-made, however the remaining components required more depth. The plate and clasps are simple to machine on a Computer Numerical Control (CNC) machine. Based on estimates from custompart.net, it is both high speed and low cost to use a CNC milling machine to create the clasps and a CNC turning machine to manufacture the plate. For more complex geometries however, the cost of using a CNC machine rose dramatically. Die-casting can be used to make precise components to very precise tolerances, however it is most applicable to large scale manufacture. Additionally, certain materials are more suitable than others for die-casting. Nevertheless, this project aimed to develop a realistic concept design only, therefore die-casting has been used for estimations, with the acknowledgement that certain drawbacks could become significant with changed conditions. If a smaller number of electrodes needed to be created, other methods should be considered. The cost estimates are displayed in Table 9 below.

Table 9: Manufacturing cost estimates

Component	Material	Manufacturing method	Manufacturing cost
Plate	Chrome stainless steel	Machining	\$0.02
Grip ring	Silicon	N/A	N/A
Electrode	Silver Chloride	Die cast	\$2.55
Pin	Brass	Die cast	\$2.96
Clasp 1	Stainless Steel	Machining	\$2.42
Clasp 2	Stainless Steel	Machining	\$2.42
Primary spring	Alloy Steel SS	N/A	N/A
Secondary spring 1	Alloy Steel SS	N/A	N/A
Secondary spring 2	Alloy Steel SS	N/A	N/A

By combining Table 8 and Table 9, the total cost estimate for each electrode was made. It was noted that a limitation of this method is that it failed to consider the initial costs involved with creating die-molds and ignored some overheads, however as a preliminary estimate it gave a reasonable approximation as shown in Table 10.

Table 10: Total cost estimates

Component	Material Cost	Manufacturing cost	Total Cost
Plate	\$0.057	\$0.020	\$0.077
Grip ring	\$0.001	N/A	\$0.001
Electrode	\$0.418	\$2.550	\$2.968
Pin	\$0.007	\$2.960	\$2.967
Clasp 1	\$0.104	\$2.420	\$2.524
Clasp 2	\$0.104	\$2.420	\$2.524
Primary spring	\$0.446	N/A	\$0.446
Secondary spring 1	\$0.739	N/A	\$0.739
Secondary spring 2	\$0.739	N/A	\$0.739
Electrode System	\$2.615	\$10.370	\$12.985

2.8 Life cycle analysis

2.8.1 Electrode

There is very little load placed on the basic components in this system apart from the primary spring, therefore the predominant concern was the continued function of all non-spring parts from a non-fatigue standpoint. The sanitary lifetime of the skin-contact electrode can be easily extended by cleaning the silver chloride surface. Without the presence of a corrosive agent, the silver chloride coating protects the electrode, keeping it largely inert and safe from decay over time. Therefore, the only potentially limiting component was the primary spring.

2.8.2 Spring

As the only parts to deform during the normal operation of the electrode system, the fatigue life of the springs needed to be considered. The primary spring is never lengthened, only compressed, therefore as a compression spring never undergoes plastic deformation. Using an endurance limit calculator from Advanced Mechanical Engineering Solutions (2019), the shear stress amplitude was determined to be $\tau_a = 515.17MPa$, while the midrange shear stress was calculated as $\tau_m = 640.99MPa$. Using the Goodman method for endurance limits, the shear endurance limit was found to be $S_{se} = 636.59MPa$. If the maximum shear stress on the spring never exceeds its endurance limit, the spring will last an infinite number of cycles. However, as shown in Equation 6, this was not the case:

$$\begin{aligned}\tau_{max} &= \tau_m + \tau_a & (6) \\ \tau_{max} &= 640.99 + 515.17 \\ &= 1156.16MPa\end{aligned}$$

Consequently the primary spring will not last an infinite number of cycles without failure, because $\tau_{max} > S_{se}$. The supplier stated that the "fatigue life of these springs ranges from 2,500 cycles to over a million cycles depending upon the load and size of the spring" (Lee Spring, 2017). As the supplier, their estimates were deemed valid for this analysis.

The IHT is designed for use once per week, and on any electrode the primary spring will be compressed fully exactly twice as the IHT is donned and removed. Additionally, each primary spring has been assumed to experience full compression an extra five times during each use as a conservative safety factor, despite there being no reason this should occur in a stationary ECG test. Next, the fatigue life of the spring was conservatively approximated to be 20,000 cycles in the absence of the appropriate S-N curve from fatigue testing. Accordingly, the estimated lifespan in years of the spring was calculated as:

$$\begin{aligned} Lifespan &= \frac{\text{fatigue life}}{(\text{cycles per use}) \times (\text{uses per year})} \\ &= \frac{20000}{(7) \times (52)} \\ &= 54.95 \text{ years} \end{aligned} \tag{7}$$

This estimation of 55 years shows that the spring endurance is more than adequate for the envisioned usage patterns. The World Health Organisation (2016) gave the life expectancy at birth in India as 67 years for males and 70 years for females. As a result, the IHT electrodes would last the average owner their entire life even if they began using it at age 15. Furthermore, the target market are not fully healthy individuals but those with a history of coronary artery disease, bringing their average life expectancy below the national average. Therefore, it is reasonable to conclude that the electrode system will not be the limiting factor in determining the product life of the Ischemic Heart Tracker.

2.9 Future work

Whilst the system as outlined above should be functional, there remain some project areas with development opportunity. The adaptability of this system to other ECG scenarios such as stress tests is one such area, for which a detailed investigation into stress test forces should be made. This would likely result in the redesigning of the electrode system slightly to accommodate increased motion and jolting, which could be countered by improved spring strength, decreased mass, optimised surface material or a more compact design. One of the greatest benefits of exploring this application is that the IHT could then be used for initial screening and diagnosis purposes. This would enable potential sufferers of Coronary Artery Disease to participate in a stress test within decreased time frame. Theoretically, it would also free up cardiologists to minister to their more critical patients. The gantt chart shown below in Figure 13 outlines how the design project was divided, and indicates that further development for stress-tests could potentially be accomplished within a similar span.

Tasks	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
Decide electrode type										
Write up decision process										
Write up use and testing outline										
Design or choose sticking mechanism										
Electrode attachment to vest design										
CAD of electrode and connected parts										
Develop locking mechanism										
Life cycle analysis										
Cost analysis										
Write up all parts										
Planned time										
Task completed										

Figure 13: Project plan

A limitation of the current system design is that the electrode would still protrude from the compression shirt a significant amount despite the capability to be locked in the compact position. This 8mm protrusion could make the IHT difficult to use, however there are several viable solutions. Short of a redesign, attaching the electrodes after putting the IHT on is a possibility. An alternate approach could be to use a material with woven-in conductive surfaces, almost dispelling the need for physically separate electrodes entirely. This would raise new design challenges however, and the strength and elasticity of the current solution appears adequate for its purpose, an area which is expanded upon by a deeper investigation into the IHT's physical housing design process below.

3 Physical Housing - Ping Li

3.1 Aim and Objective

The physical housing of the wearable device is a critical component of the overall device and greatly influences the practical functionality of the device. The housing serves as the component that connects all other portfolio and at the same time, determines how all the portfolios fit together to achieve the functions of the device. Thus, the aims and objectives of the physical housing must closely align with the overall aim and objective of the device. The overarching aim of the device as determined by the group, is outlined in the introduction. A shortened summary of the overall aim is “to provide a simplistic, affordable and less resource intensive alternative for CHD monitoring process for urban India. This is achieved through the use a wearable ECG device that can monitor a patient’s condition weekly”. This is based on the problem statement proposed in the CIOP Brief, where the problem statement states “Research and determine the number one health problem in an under-developed region of the world and design a remote wearable diagnostic device and associated system to allow remote medical treatment by doctors and allied health professionals, based in Australia.” (CIOP, 2018)

Therefore, based on the overall aim of the device set by the group as a whole with consideration of the CIOP, the objective of the physical housing are separated into 2 parts, Part A and Part B and is as follows:

Part A

- Durable
- Comfortable
- Easy to clean
- Cheap to produce
- Easy to manufacture

Part B

- Able to house all components (Electronics, power source, electrodes) efficiently
- Allow for re-positioning of electrodes for maximum effectiveness of ECG data
- Able to hold electrodes in place while patients are moving
- Able to cater to people of different size and shape
- Simple to operate

Part A focuses on the objective of the stand-alone properties of the housing, in terms of how the housing will perform by itself. From durability and ergonomics to material costing and ease of manufacturing. The durability and ergonomics are important as this is a reusable wearable device and therefore, must withstand repeated uses while remaining comfortable for the wearer. The material costing and manufacturing significantly influences the production cost of the device. And as the economic feasibility of the device must be kept in mind as the purpose of this device is provide a cheaper alternative to measuring ECG data as opposed to current available options (which is visiting a hospital in person). Consequently, the device must be designed to be cost effective when compared to current options.

The objectives in Part A will be addressed in the design process. Both through the design of the housing as well as the material selection.

Part B focuses on the objectives of how the physical housing interacts and synergises with the other portfolios. These objectives are directed at how the housing contribute to the functionality of the device, in particular with respect to the ECG data collection process, which is discussed in detail in the Introduction. These objectives form the majority of the design constraints of the physical housing design. The housing must first and foremost, be able to live up to its namesake, and that is to efficiently house the other physical components. It then must be able to allow for accurate positioning of the ECG electrodes and maintain that position over time, with movement from the patient. Finally, the ability of the device to be used by differently shaped people and the ease of operation of the device are also

important. As it is target at a very large and unspecific demographic and will be regularly used and without the supervision of a medical professional.

These objects in Part B will be addressed in the design of the housing and in collaboration with the other portfolios.

The completion of these objectives of the physical housing will be determined by the final design. The design will need to strive to achieve all of the listed objectives above but is understandable if not all can be due to limitations.

3.2 Methodology

As the scope of the portfolio of physical design for this project does not involve any experimentation or lab type testing as with a regular research thesis, the methodology section will only include the methodology involved in conceptualising of the design of the housing of the device.

1. Analyses of the group aim and how physical housing would impact the device

The aim of the group was first analysed to determine how the housing would fit in and contribute to the group objective. This involved closely examining how the proposed wearable device will solve the chosen health issue and the process of how the patient will be using the device.

This understanding of how the housing fits into the device then led to understanding how the housing will impact the overall device, what feature of the housing to must include and the order of importance of each of the features.

2. Determine the aims and objectives

Once an understanding of the functions of the housing was reached, and the features that the housing will need to include to make the device functional are then determined, on which are the aims and objectives of the housing component built. This aim and objective then guided both the literature review as well as the design concepts that followed, ensuring the final design of the physical housing will allow the device to function as needed and fit with the overall aim of the device.

3. Literature review

A literature review of existing options for the physical housing design and component layout was conducted. The aim of the literature review was to gain an understanding of current designs and methods, and to assist with the ideation of the design concepts. This literature review will ideally form as a baseline on which the design concepts can be built on and at the same time, show any design gap in design or methods in current designs.

4. Design concepts

Based off the Part B of the aims and objectives of the housing component determined earlier, and the literature review conducted, and three individual design concepts was conceptualised. These concepts should stay as align as closely as possible with the objectives and aim to pull knowledge from the literature review as well as match the intensions of the device. One of these ideas was developed into the final design.

5. Decision and analysis of final design

A decision matrix was utilised to determine the best design concept to move forward with. Once the final design was chosen, a materials and costing analysis was then conducted with Part A of the aims and objectives in consideration. This was accompanied by a Life Cycle Analysis of the housing. These analyses will guarantee the final design will satisfy all aims and objectives listed.

6. Reporting

The final stage of the methodology was to document the final design of the housing, the process in reaching this final design and any finds along the way in a report format. This individual section of the physical housing portfolio was combined with the other portfolios to form a comprehensive report about the final device.

Gantt Chart

Gantt chart was utilised to aid and keep track of progress mentioned in the methodology. This Gantt chart is included in the appendix section and covers the design, analysis and reporting stages of the methodology.

3.3 Literature Review Findings

A comprehensive Literature Review was conducted in Thesis A, a brief summary of the findings are presented below.

The literature review conducted during Thesis A and presented in the previous progress report indicated that there is no set standard for the layout of components and physical design of a wearable ECG device, but only the design and layout of the individual components. i.e. power, electronics. The findings although somewhat inconvenient in terms of providing specific directions on the physical design of the whole device, is able to provide a rough starting position for the design concepts. It was also able to establish that this device is in a brand-new category in terms of wearable devices and might be able to fill a gap in the current market. (P, Li. 2019)

3.4 Design

The design concepts are based off the aims and objectives listed in Section 3.1 above, as well as any finding from the literature review in Section 3.3.

3.4.1 Design Concepts

Concept 1: Cover vest and electrode pin design

This design presented in Figure 14 consists of a vest component and an electrode component. This design focuses on precision of the placement of electrodes, allowing for maximum flexibility and adaptability.

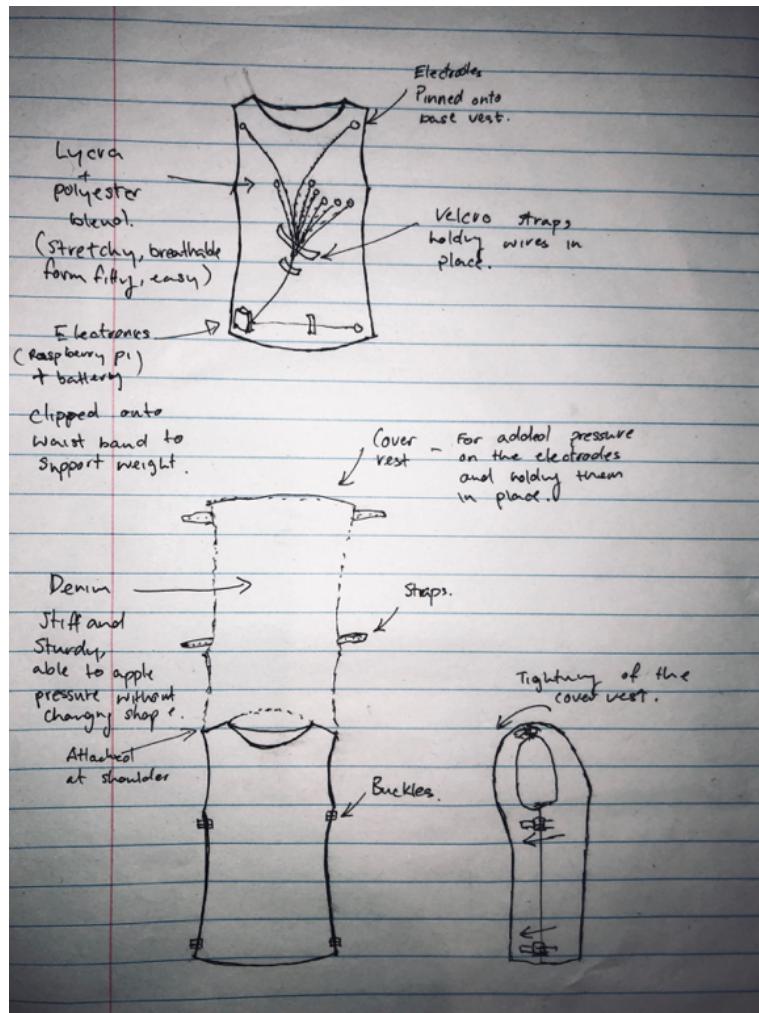


Figure 14: Concept 1: Cover Vest

The vest design consists of an inner skin-tight base vest made from a stretchy material (Similar to compression shirt by SKINS) with a none stretch cover vest (polyester) on top. This 2-layer design allows for the base layer to hold the position of the electrode pin in X - Y plane, while allowing the cover vest layer to apply pressure in the Z – plane. The skin tight base vest allows the electrode pins to be pinned onto the vest in virtually any location (in the X-Y plane, parallel to the patient's chest) and maintain that location. This process will be administered by a medical professional on initial consultation, to ensure proper

placement of electrodes. The none stretch cover vest is then placed over the electrodes on top of the chest and back and tightened down with the tension straps and buckles. The cover vest will apply pressure to the electrodes (in the Z-plane) and thus, holding it in place against any movement. In particular, neutralising the gravitational forces acting on the electrode pins. The connection wirings of the electrodes will be organised with Velcro straps on the base vest. These wirings are also covered up by the cover vest as an added safety precaution. The electronics component including the battery pack and Raspberry Pi will be located at the bottom of the vest. They will be located inside a plastic housing, which is able to be clipped to the belt or the waistband of the patient so that the vest would not have to take the weight of the electronics. The base vest material will be made from a cheap, comfortable, elastic and easy to clean materials. While the cover vest material will need to be none stretching and thus, able to apply pressure when tightened. The material ideally would be also breathable so the user does not over heat while moving.

Electrode Pin Design

The electrode pin design draws inspiration from traditional lapel pins, utilising the pin and locking mechanism.

As shown in the expanded view in Figure 15, the head of the pin is the electrode plate that touches the skin and conducts current for the ECG data. The design of the head is covered in the Section 2, the Electrode Design portfolio. The pin section of the electron pin is able to pierce the base vest fabric and continues to be locked in the pin locking mechanism. The locking mechanism has a simple spring-loaded release mechanism, allowing for secure fit of the electrode pin and quick lease for easy adjustment. It also has the electrode wire attached to the back of the pin which connects the electrode to the on-board electronics located at the base of the vest. This entire apparatus will experience pressure applied by the cover vest and be completely hidden from user one the positions of the electrodes are set up.

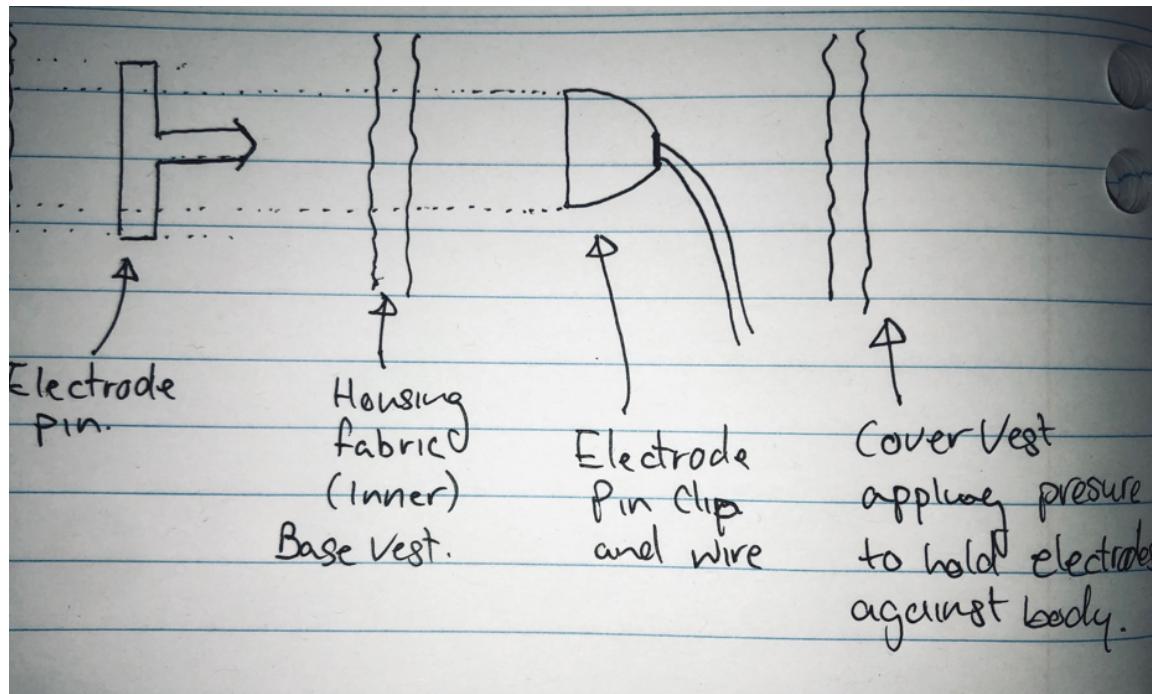


Figure 15: Electrode Pin Design - Expanded view

Concept 2: Hole shirt design

This design is based around one skin tight T-shirt with pre-existing button hole like openings sewed into the fabric, demonstrated in Figure 16. This design is focuses on simplicity of design but lacks precision in electrode placement and is not able to adapt to non-standard body shapes. It is able to be utilised by traditional electrodes.

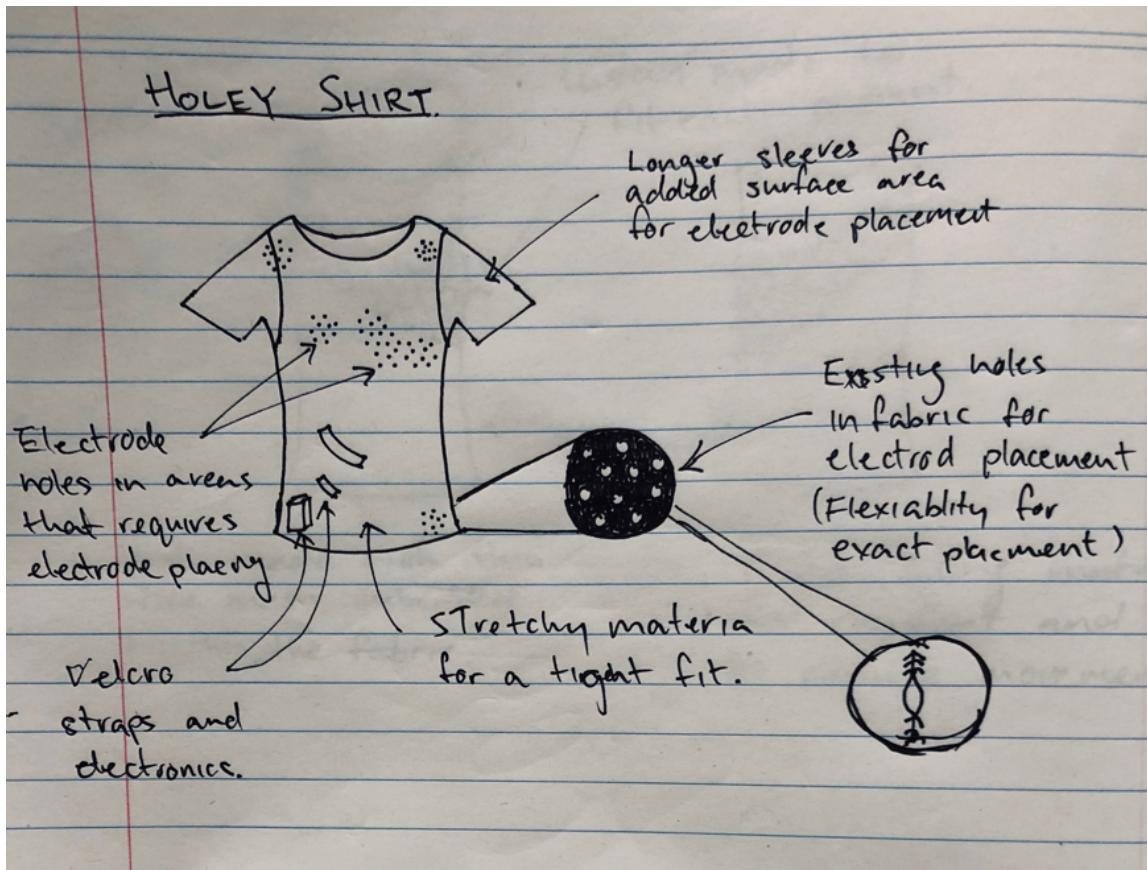


Figure 16: Concept 2: Holey shirt design

T-Shirt design

This design consists of one skin tight T-shirt with button like holes stitched into the material. The skin tight, form fitting material will allow the electrodes to be easily placed on the patient and offer some anti movement support for the electrodes once they are placed. The electrode holes placed in grid patterns and concentrated in the areas where electrodes are required to be placed. i.e. the left and right shoulder, around the heart and across from the heart. These holes will allow regular electrodes to feed through the hole and be attached to the patient. The holes are constructed similarly to traditional button holes where the opening is surrounded by stitching to prevent the material from ripping. The choice of a T-shirt over a vest design is to accommodate the larger shoulder area where the electrodes

might need to be placed. Upon initial consultation, the medical professional will place the electrodes in the hole location of best fit. The holes will then be marked with a fabric marker to help the patient identify in the future which hole to thread the electrode. The patient will need to remove the electrodes after each use and re-thread every time they wish to perform the ECG testing as they will unlikely be able to remove the T-shirt without detaching the wiring connecting to the electrodes. These electrodes can be regular reusable electrodes or disposable sticky electrodes. If a regular reusable none stick electrode is used, it will only be attached to the patient from the tension of the compressive T-shirt material. The patient will need to remain very still and be lying down during the ECG process. The patient will be afforded a lot more freedom to move if they decide to use a disposable sticky electrode. Both of these electrode methods present a large area of error for the placement of the electrode, more so the reusable none stick electrode. Similarly to the cover vest design, the battery and electronics will be placed on the bottom of the shirt and clipped on the waist band of the patient.

This design is extremely simple and can be used with traditional electrode options, but the opportunity for error is exceedingly large. This error initially stems from the placement of electrodes by the medical profession, who might not find the perfect hole to place the electrode, and is then compounded by the T-shirt acting only as electrode placement tool and allowing large amount of movement of electrode once the patient takes the T-shirt home.

Concept 3: Wire sash design

This design consists of a vest design with a wire mesh sash that wraps around the chest of patient, shown in Figure 17. This design will be used in conjunction with the electrode pin design. This design focuses on precision of placement and stability of the electrodes with movement.

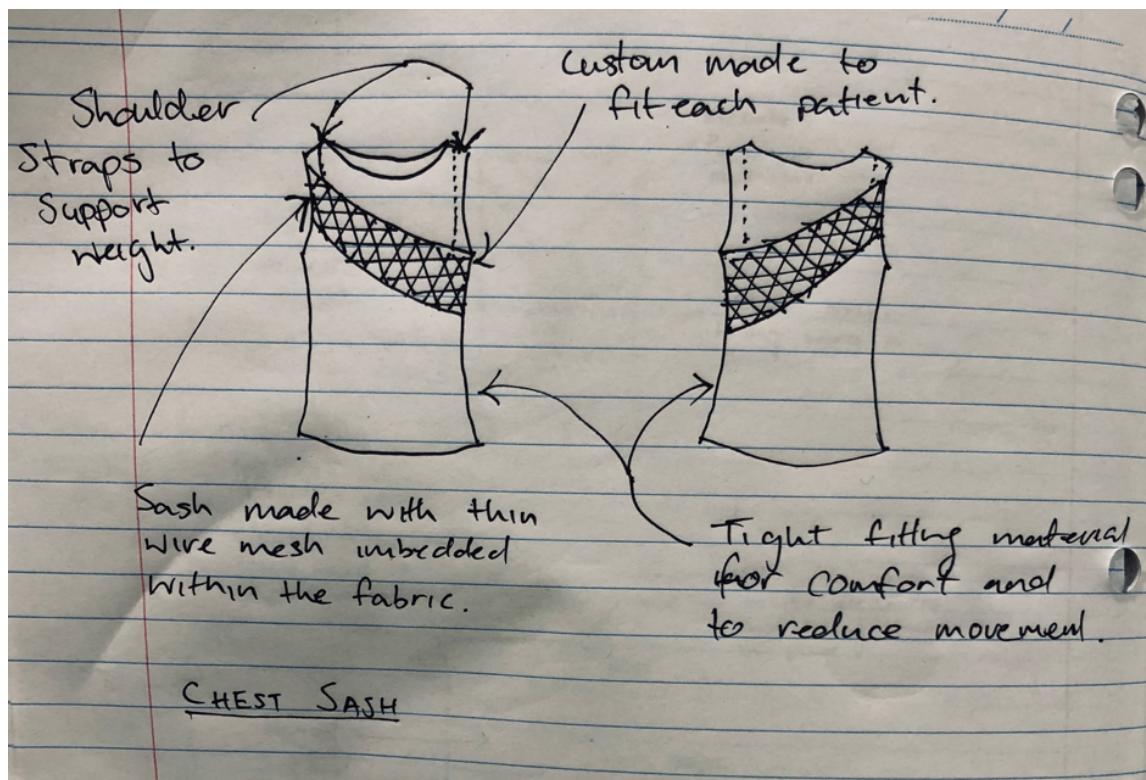


Figure 17: Concept3: Wire sash design

Wire Sash

This design is built around a wire mesh sash embedded within a tight-fitting vest. The wire sash wraps around the chest of the patient will provide structural support for the electrodes, fixating them in prepositioned locations even with the most vigorous movements. The wire mesh is made from thin metallic wires similar to the wires used in pipe cleaners. The wire mesh will take up the contouring shape of the chest and stay in that shape. This property will allow for a perfect alignment of the electrodes once they are initially placed onto the wire mesh, with very small amount of movement error. The wire mesh will also hold tension which will allow it to hold the electrodes in place. This wire mesh will have ample space in-between the wires to allow for the precise placement of the electrode pin. The mesh covers the major areas where the electrodes will be placed, leaving two electrodes, right shoulder and right hip without support. These two electrodes will only use the electrode

pin mechanism as support as it is not functional viable to insert wire mesh everywhere on the vest. As the wire mesh will carry some weight, it will need to be supported by the shoulder straps, which is stitched in to the material. Similar to the cover vest design and holy shirt design, the battery and electronics will be placed on the bottom of the shirt and clipped on the waist band of the patient.

Although this concept will provide very precise and solid placement of the electrodes, it is not ergonomic, being heavy and stiff to wear, as well as being difficult to clean. Because of the design of the electrode pins, the wire mesh might introduce unwanted conductors and influence readings from the ECG testing. The wire mesh will also cause wear and tear of the material it is housed in. These must be taken into consideration when selecting for the final design.

3.5 Decision Process

The selection of which design concept to progress with will be decided using a Pugh matrix, which will provide a quantitative differentiation between the designs. The criteria of the Pugh is based on the aims and objectives of the physical housing outline in the Aims and Objectives section. The weighting of each criteria is determined based on its importance to the device, also based on the aims and objectives.

Table 11: Pugh Matrix for concept selection

Evaluation Criteria	Weighting (1-5)	Concept 1 (Datum)	Concept 2	Concept 3
Aims and Objectives: Part A				
Durability	3	0	0	0
Comfort	3	0	0	-1
Easy to clean	3	0	1	-1
Cheap to manufacture	3	0	1	-1
Aims and Objectives: Part B				
Efficient containment of components	3	0	0	0
Freedom of electrode placement	5	0	-1	0
Maintains electrode placement	4	0	-1	1
Cater to different shapes and sized people	3	0	0	-1
Simple to operate	3	0	0	0
Weighted Score		0	-4	-7
Ranking		1	2	3

From the Pugh Matrix displayed in Table 11, it can be seen that the Cover vest design scored the highest of all the concepts. This is likely because this design is the most well balanced in terms of simplicity, freedom of electrode placement and ability to hold the electrodes in place. It can also be seen from the matrix that the weighting is leans heavier towards aims and objectives of Part B, this reflects the importance of synergy of the housing with the other components for the overall device's functionality.

3.6 Final Design

From the Pugh Matrix in Table 11, the final design is determined to be the cover vest design. CAD modelling was created for this design to illustrate the design and hand calculations were conducted to validate its functional requirements.

3.6.1 CAD Modelling

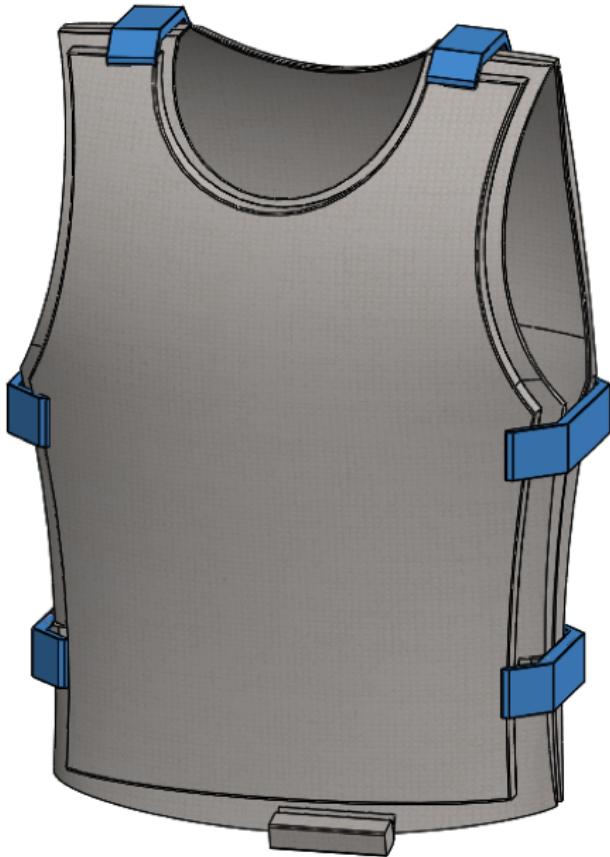


Figure 18: CAD render of final design

The above Figure 18 shows the cover vest when fully assembled on a patient. The base vest is resting on the patient's body, the electrode pins are pinned onto the vest at the locations

determined by the medical professional, each electrode is connected to the power storage device and on-board electronics through wires, the electronics and power storage is attached to the waist band of the patient and the cover vest is on top on the whole assembly.

The blue tension strap and buckle is holding the cover vest together and applying pressure to the electrode pins underneath and at the same time, the cover vest is keeping the electronic's wires covered. This strap and buckle is illustrated in the Figure 19. For simplicity and cost efficiency sake, a generic, pre-made strap and buckle will be used. The strap is adjustable length wise and will be stitched onto the cover vest.



Figure 19: Strap and buckle (Wisdompro)

Figure 20 shows the electrode pins pinned onto the vest. As the majority of the electrode pin is underneath of the vest, only the locking mechanism of the pin is showing over the material of the shirt.



Figure 20: Base vest with electrode pins(Underarmour)

Finally, Figure 21 displays the spring loaded clipping mechanism of the electronics component. The battery and on-board Raspberry Pi Zero is attached to the base vest (stitched onto the material), but to eliminate any unnecessary load being placed on the vest, the electronics will also be clipped onto the waist band or belt of the patient using this clip.

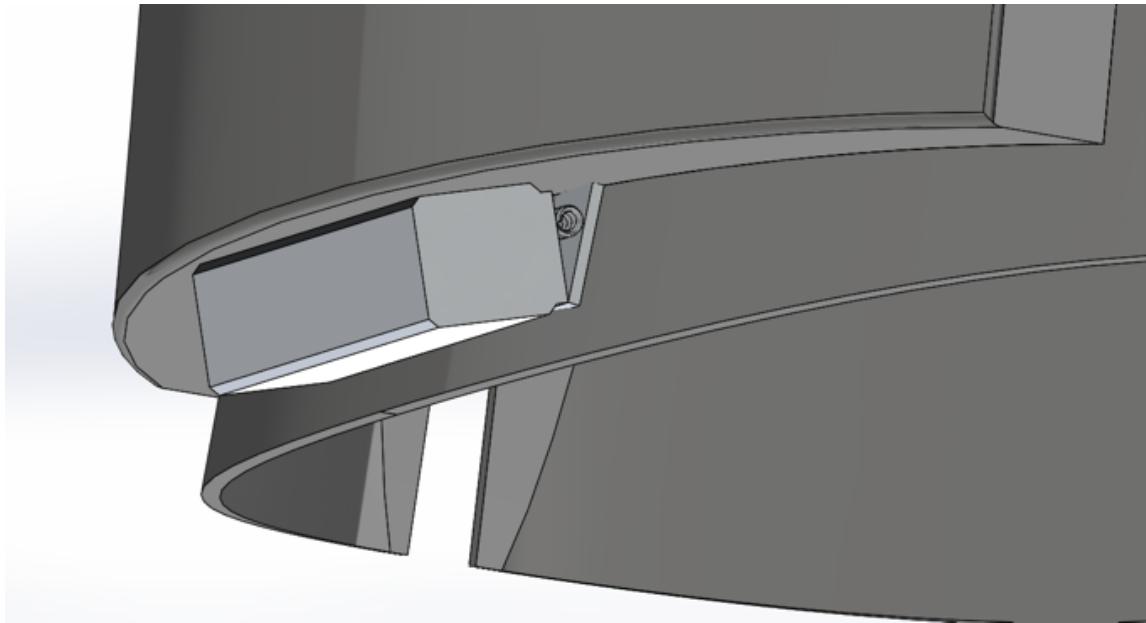


Figure 21: Belt clip

3.6.2 Materials Selection

The materials selected for the base vest is 90/10 polyester elastane blend while the cover vest is 100% polyester fabric.

The 90/10 polyester elastane is selected for its versatility and its ability to fit with the Part A of the Aims and Objective. This material is breathable, form fitting, durable, comfortable and cheap to manufacture. (M. Senthilkumar, 2011)

Polyester, or polyethylene terephthalate is a synthetic material of long-chain synthetic polymer formed from the reaction of a diol and an acid anhydride. (T. Costa, 2006). This fibre is strong due to its crystalline nature and prevents it from wrinkling. It is also hydrophobic as the lack of polarity prevents water molecules from entering the polymer system, meaning it is fast drying. This material however cannot stretch as the repeated strain will cause the polymer system to distort.

Elastane is also a synthetic material, composed of polyurethane. Elastane fibres are highly stretchable due to the formation of the polyurethane bonds (M. Senthikumar, 2011) and thus is commonly combined with other materials to produce sportswear. It is able to stretch 800% of its original length (T. Chen 2011), so even small amounts of elastane blended into

the polyester will pass on this stretchy property.

The addition of elastane in the polyester fabric dramatically improves the comfort and the stretchiness of the polyester, leading to excellent form fitting abilities and matches the aims and objectives of the housing. A 90/10 polyester elastane blend material is the most common combination as elastane is expensive and too higher elastane content will reduce some of the polyester's strength and breathability. The popularity of this blend also leads to lower cost.

A plain polyester is chosen for the cover vest as the function of the cover vest is to apply pressure to the electrode pins. This is because the strength and non-stretchy property of polyester is highly desirable. Polyester is also highly breathable and will not over heat the patient while the patient is wearing both of the layers, especially in the hot climate of India.

3.6.3 Calculations

Hand calculations were conducted to verify the feasibility of the design to perform as intended. These calculations are summarised in the following section, the actual calculations and all relevant data sheets are attached in the appendix of the report.

Forces of tension strap in cover vest

The two major forces the cover vest and tension straps will need to overcome is the gravitational forces acting on the electrode pins as well the spring force of the electrode pin.

Gravitational forces

The purpose of the cover vest is to apply pressure to the electrode pins in the z plane of device (the X-Y plane is parallel to the body, Z-plane is into the body) to counter the gravitational force acting downwards on the electrode pins. This pressure is applied through tightening of the 6 tension straps connecting the front cover vest and back cover vest.

The total weight of one electrode pin assembly is 11.61 grams. (6.61 g for the electrode pin and 5 g for the locking mechanism) The copper wire connecting the electrodes to the electronics is around 3g/m. The total wire consumption of the device is 8m. Thus, the total weight needed to overcome by the cover vest and tension straps is 163.32 grams or 1.6N of force.

Due to non-stretching properties of the polyester material of the cover vest, the cover vest

is assumed to be non-deforming. Thus, the forces required to generate by the patient on each tension strap to maintain the position of the electrode on the base vest is 0.26N, using sum of forces.

Spring loaded electrode pin

The electrode pin is loaded with a spring. In order for the electrode pin to function, this spring must be compressed. The force required to compress the spring is 5.86N per electrode pin, from here it can be determined using sum of forces that each tension strap will need to supply 11.72N to compress the springs within the electrode.

Total force: 11.98N

Force in spring loaded clip

The selected battery weighs in at 145g while the Raspberry Pi weighs in at 9g. the electronics totally 154g.

Calculations are conducted assuming worst case scenario where the patient clips the electronics to their belt (not possible with waistband) upside down. The material of the belt being leather and the spring-loaded clip has a PET plastic tip, the coefficient of friction is 0.35.

The forces required for the spring is 300N using sum of forces.

3.6.4 Material and Manufacturing Cost Analysis

Using average industry prices of materials and standard materials usage of vest, the following materials costing for one complete assembly of vest are displayed in Table 12. This is only the material cost and does not include any manufacturing costings.

Table 12: Raw material costs

Component	Material	Material usage	Cost per unit	Total Cost
Base vest	90/10 polyester elastane	1.65(m ²)	2.96/m ²	\$4.88
Cover vest	Polyester	1.5(m ²)	4.96/m ²	\$7.44
Electrode pin locking mechanism	Steel	59.52(g)	0.0014/g	\$0.08
Connection wires	Copper	8(m)	1.2/m	\$9.60
Tension straps	Polyester	1.2(m)	0.76/m	\$0.91
Buckles	PET plastic	149.04(g)	0.002/g	\$0.30
Belt clip	PET plastic	31.42(g)	0.002/g	\$0.06
Total				\$22.99

Table 12 only presents the raw material costs of the vest. The final cost of manufacturing of the housing may vary due to different machining costs. Table 13 incorporates the material cost and the estimated industry machining and manufacturing costs. To keep the cost of the device low, parts when able, are directly sourced instead of manufactured and also based on orders of 1000 items per order.

Table 13: Manufacturing cost estimates

Component	Quantity	Manufacturing and Material Cost	Total Cost
Base vest	1	\$7.03	\$7.03
Cover vest	1	\$9.59	\$9.59
Electrode pin locking mechanism	12	\$0.33	\$3.96
Connection wires	8m	\$9.60	\$9.60
Tension straps	1.2m	\$0.91	\$0.91
Buckles	6	\$0.13	\$0.78
Belt clip	1	\$1.06	\$1.06
Total			\$32.94

Of the above components, only the belt clip required special machining to manufacturing. All other component are either pre-made or in the case of the vests, can be produce without special machinery.

Cost Analysis Conclusion

The estimated cost of \$32.94 is very low for the housing of a wearable device here in Australia and is very reasonable and achievable to for a country like India. The low price is mostly due to the simplicity of the design, the limited amount of components of the design and the relatively cheap materials selected. This is ideal as this would allow the device to be competitive in the market and provide validation for its viability. Although it must be kept in mind that this is only an estimation based on online industry information and adjustment must be made for reality.

3.6.5 Life Cycle Analysis

Scope and Methods

This life cycle analysis was conducted with the intention of assessing the environmental impact of the physical housing vest design. Each component will be broken down to the base materials and assessed using the most severe impacts. This will form an assessment that might include the impact of the material itself, the impact created during the manufacturing process, the impact of the component during its usable life span and finally the impact of the component at the end of its life. A range of previous published studies and reports will be used to form the basis of this life cycle.

Base vest, cover vest and tension straps

The physically largest component of the housing are the base vest, cover vest and tension straps. These are made using polyester, with small amounts of elastane.

Polyester

The majority of both the base and cover vest is made from polyester, while the tension strap is 100% polyester. The vests are also the largest component of the housing. Thus, it will likely have the most impact on the life cycle of the housing. The Life cycle analysis of a polyester garment by G.G. Smith, R.H. Barker (G. Smith, 1995) was used as the main source of data and information as this study uses polyester in a very similar application. According

to Smith, the energy consumption of manufacturing of the material only accounts for 18% of the overall energy usage of the material with the rest coming from consumer laundry, both washing and drying of the garment. Of the manufacturing energy, the resin manufacturing and fabric production consumes the largest amount of energy. Resin manufacturing includes extracting the base material from oil and gas while the fabric production includes dying, knitting and finishing. In terms of solid waste produced by this component, the majority again comes from the consumer at around 66%. Of the manufacturing process, the fabric production creates the most waste but are often recycled immediately. 95% of the solid waste produced by the consumer is from municipal water treatment sludge created through the laundry process. The main airborne pollutants in the life cycle of polyester are nitrogen oxide, hydrocarbons, carbon di and mon oxides. The main cause of this from is once again, the energy generation process of the consumer laundry process.

Elastane

Only a small amount of the device contains elastane (10% of the base vest), but it's impact is still relevant for the LCA. The "Environmental assessment of Swedish fashion consumption" (R. Sandin, 2015) was used as a guild line for the analysis of elastane.

This report formed similar conclusion with Smith's study in that the majority of the impact is from the consumer laundry process. In the energy usage aspect, solid waste aspect and airborne aspect. One area this report highlights that is lacking from Smith's study, is the micro fibre contamination of the elastane, although this could be due the publishing dates of both study/report. As this report is much more recent than Smith's study, more of an emphasis might have been put on micro fibre issues.

Buckles and belt clip

The buckles and belt clip are made from PET plastic. Although this material is commonly used for making water bottles, it is also a very viable material for producing plastic buckles and plastic belt clips through due to the material availability. Although this is PET material has the same material composition as polyester described, it has a very different manufacturing process and thus, a different set of life cycle impact. This life cycle analysis is based on "Comparative Life cycle Assessment of Polylactic acid (PLA) and Polyethylene

terephthalate (PET)” (J. Sheng, 2011).

This study suggests similarly to polyester material above, the procurement of the raw materials greatly impacts the environment as the raw material used to produce PET plastic is petroleum hydrocarbon, or oil. But this study also focuses on the large amounts of travel PET plastic undergoes from refinery to consumer. This transportation causes additional environmental impact as the means of travel will mostly be powered by hydrocarbon, which produces significant air pollution.

The end life of the PET plastic is also extremely harmful as it is not bio degradable and is seldom recycled, leading to this material ending up in landfills. While residing in landfills, PET produces methane while also contaminating the land it is on.

Electrode pin locking mechanism and spring in belt clip

Steel

Both the locking mechanism and the spring in the belt clip will be made using steel. The life cycle analysis by the World Steel Association will be used as a source of data. (WSA, 2015)

The production of steel is very material and energy intensive, with the production of 1 tonne of steel requiring 1.4 tonne of iron ore, 0.8 tonne of coal and 0.3 tonne of lime stone. The impact on the environment from the depletion of these resources are significant. Although these resources although plentiful, are finite.

The production process itself has tremendous impact also, although in comparison, steel itself has actually relatively smaller impact than other common metals during production. It produces 2 kgCO₂/kg of steel, compared to 16 kgCO₂/kg for aluminum. Also, the steel industry has also taken steps to reduce waste with 96% of raw material being turned into products.

The re-usability and recyclability of steel also reduces the impact of this material. The amount of energy required to re-purpose steel is significantly lower than the initial energy required to produce steel from raw materials. So, the steel used in this device will likely be reused and thus, reducing waste. This allow the value and impact created with the initial production of the material to be beyond the intended purpose and life span.

All the impacts of the locking mechanism and spring are in the production and manufacturing,

as once the steel has been machined into the locking mechanism or spring, it has next to zero additional environmental impact.

Copper wire

Copper wires are used as connection wires from the electrode pin to the electronics. Reviewing “Copper life cycle assessment data for copper wire” (European Copper Institute, 2012) was able to give insight into the life cycle of this component.

In terms of environmental impact, it is very similar to that of steel, both highly material and energy intensive, but very little additional impact beyond production and manufacturing. One area in which copper exceeds steel is the reusability. As the value of copper is greater than the value of steel and has highly functional purposes this material will likely be recycled many of times. This greatly reduces the initial impact of production.

Life Cycle Analysis Conclusion

From the Life Cycle analysis, it can be seen that the largest component of the device, the two vest, creates the most impact during its laundry and washing process, producing solid waste and airborne waste. The PET plastic used in the buckle and belt clip although has no significant impact during usage, causes substantial damage during its end life, slowly breaking down in landfills which produces methane and contaminants the land. The steel and copper used in the electrode pin locking mechanism, spring and connection wire all have very high material and energy intensity during production, but is highly recyclable, thus greatly reduces its environmental impact.

3.7 Conclusion

The design of the physical housing is able to contribute to achieving the overarching goal of the device mentioned in the Section 3.1. It is able to do so by achieving the individual aims and objective in set out Parts A and B of Section 3.1. A cost and life cycle analysis were also conducted to validate and provide insight into the design

Through the design of the housing, the functional and collaborative objectives in Part B can be seen as achieved in full. The vest design is able to house all the components, maintain position of electrodes and allow flexibility in the placement of the electrodes. These allowed for a seamless connection to the rest of components as well as an accurate ECG data collecting process. Although this design is able to satisfy all the functional objectives, it can still be improved through increasing simplicity. As of now, the cover vest design has several detachable sub component, adding to the complexity. This can be improved with further analysis and through the building of a prototype.

The objectives in Part A were also addressed, through the material selection. The polyester and polyester blend selected for the vests are both able to support the design criteria of the cover vest design of being non stretching and form fitting, while at the same time adding to the ergonomics and ease of use aspects of the design. The remaining prefabricated materials were chosen to reduce the complexity and cost of the design. Based on the life cycle analysis, the majority of the component will cause an environmental impact, but this environmental impact will likely occur with any other material, perhaps only with a lesser severity. Less impactful materials can be chosen, but may affect the production of the cost, leading to a less feasible design.

This reduced cost (final cost being \$32.94) determined from the cost analysis is highly important as it provides the design, or at least the housing component of the design with a competitive advantage, allowing it to have an opportunity to be feasible in a real world situation, especially in the urban India setting. It must be noted that this cost is only an estimation of the device, with assumptions made based of internet prices. The accuracy of this could be improved by making contact and inquiries directly with suppliers and manufacturer to receive much more update and realistic data.

Finally the life cycle analysis was able offer insights into the environment impact of the design and presents a critical perspective. The analysis determined that the base vest and cover vest will create the largest environmental impact during the usage life span of the device, while the smaller components of the housing, the steel used for locking mechanism and spring, and PET plastic used for buckle and belt clip will produce the most amount of impact during the cradle and grave stages of its life cycle.

Moving forward, a prototype should be built for more in-depth analysis of the design. Market analysis should also be conducted to further verify the real world opportunities of this device.

Overall, this report is able to provide a good and viable design solution that is able to both deliver on the outcomes presented in the CIOP and satisfy the teams own set list of aims and objectives. The design and analysis has opportunity to improve with further work.

4 Data Analysis I - Calandra Lunardo

4.1 Aims & Objectives

The major objective of the role for data analysis was to understand the chosen method of data analysis, a 2D Convolutional Neural Network (2D-CNN), obtain the data representative of the chosen health problem, Coronary Artery Disease (CAD), and perform an accurate automated diagnosis from the obtained data. All this, whilst staying relevant towards the defined project scope. Due to the large scope of this role, it was shared with my peer, Zachary Hamid. This component of data analysis is focused on linking the health problem of CAD with the chosen method of data analysis utilizes a 2D-CNN.

The data analysis segment as a whole aims to fill in the gap in knowledge of application of a 2D-CNN by creating a functional and accurate 2D-CNN for classification of CAD. It was established with the given Health CIOP project scope in mind that a 2D-CNN was the most ideal method for automated electrocardiogram (ECG) data analysis. In brief, the ECG graph required analysis as a whole image in which the 2D-CNN was capable of conducting for image classification. Furthermore, 2D-CNN were already implemented for use in automated ECG classification of arrhythmia, an occurrence of an abnormal heart rhythm. Further research into ECG classification also concluded that the 2D-CNN yielded the highest accuracy of 99.05% and average sensitivity of 97.85% compared to other autonomous data analysis methods such as a 1-Dimensional Convolutional Neural Network (1D-CNN), Support Vector Machines, and K-Star Neural Networks with the same dataset to analyse and classify ECG data for arrhythmia . However, in light of such success in arrhythmia classification, there was a predominant lack of accessible and available research in the use of 2D-CNNs to classify CAD (Lunardo, 2019).

Due to the lack of available resources for 2D-CNN use for ECG classification of CAD, the first objective was to establish a clear understanding of both the chosen disease and how it was graphically represented in an ECG. This clear understanding would be vital in sorting the data according to its CAD diagnosis prior to training the 2D-CNN.

The next objective would then be to determine the final output classification the 2D-CNN

is meant to give. Upon understanding CAD in an ECG, possible classifications could then be determined along with the type of datasets needed to support the chosen classifications. Depending on the availability of usable datasets, the initial CAD classifications would then be refined again to make it feasible to create a training set for the 2D-CNN.

With the known ECG data classifications and available datasets, the objectives of extracting, processing, and sorting these datasets according to its CAD classification to create the training sets needed to train the 2D-CNN follows.

4.2 Research: Understanding the ECG & its Features

The 2D-CNN has established itself as a powerful tool in image classification for ECG images of arrhythmia (Lunardo, 2019). However, due to the lack of knowledge on the application of 2D-CNN for image classification for CAD, there is a need to first understand how the features of the ECG vary for CAD patients. Not only did this understanding help in determining the different classifications of ECGs for CAD and the validation of classification of ECGs in later stages, but also helped in sourcing the appropriate databases and datasets to extract from. The obtained ECG needs to be reflective of only the features of a patient undergoing a cardiac event specifically indicative of CAD.

4.2.1 The ECG Complex

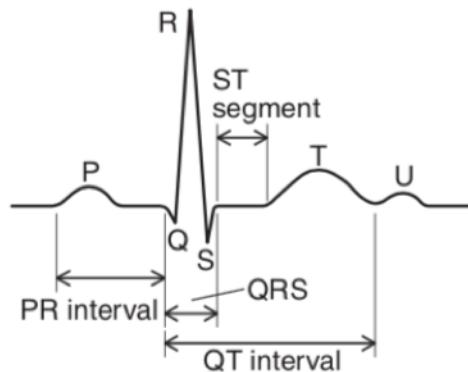


Figure 22: Normal ECG beat (Hampton, 2013)

The ECG is a diagnostic tool in understanding the cardiac rhythm of the human heart. The heart is a muscle which contracts in association to electrical changes known as ‘depolarization’. Electrodes are attached to the surface of the body to detect these electrical changes representative of the cardiac cycle otherwise known as the heart beat shown in Figure 22 and otherwise known as the ECG complex (Hampton, 2013).

A brief explanation of the ECG complex and its alphabetical landmarks defined by John R. Hampton’s *The ECG Made Easy*:

- The P-wave signals the beginning of the cardiac cycle. The heart muscle begins to contract from the upper chambers.
- The PR-interval occurs when contraction is being propagated down towards the bottom chambers of the heart.
- The QRS-complex corresponds to contraction of the bottom chambers of the heart. This contraction facilitates pumping of blood throughout the human body, hence the large deflection.
- The ST-segment should register as zero (flat) in an ECG of a healthy heart.
- The T-wave results from the heart muscle relaxing post contraction.

These alphabetically represented components will be continuously referred to in subsequent segments.

4.2.2 The 12-Lead ECG

WearTech decided on a 12-lead ECG which requires ten electrodes to be placed on the human body to create the ECG image. The ten electrodes comprise of six electrodes placed on the chest (V1, V2, V3, V4, V5, V6) and one on each of the four limbs (LA: Left Arm, RA: Right Arm, RL: Right Leg, LL: Left Leg). A 12-lead ECG was preferred because it is the clinical standard (Ashley & Niebauer, 2004), and because accurate CAD diagnosis requires cross analysis of multiple different leads, which will be further discussed in later segments.

The 12-leads representing the comparison of electrical activity in electrodes (Hampton, 2013).

- V1: V1 and average of (LA+RA+LL).
- V2: V2 and average of (LA+RA+LL).
- V3: V3 and average of (LA+RA+LL).
- V4: V4 and average of (LA+RA+LL).
- V5: V5 and average of (LA+RA+LL).
- V6: V6 and average of (LA+RA+LL).
- I: LA and RA.
- II: LL and RA.
- III: LL and LA.
- VR: RA and average of (LA + LL).
- VL: LA and average of (RA + LL).
- VF: LL and average of (LA + RA).

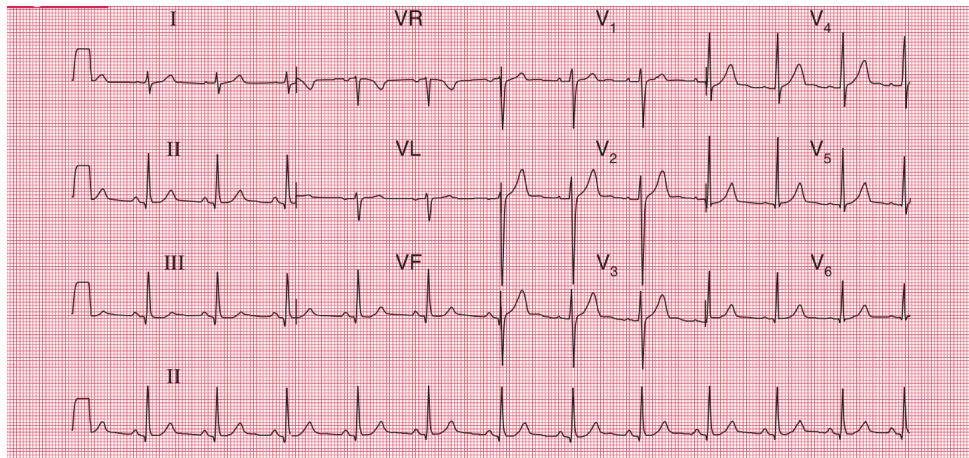


Figure 23: 12-lead normal ECG (Hampton, 2013)

The 12-lead ECG comprises of twelve different waveforms, as shown in Figure 23. Each lead represents the comparison in electrical signals in specific electrodes (Hampton, 2013). Since the electrodes are placed in different locations on the human body, each lead can be seen as depicting the electrical activity of the heart from different planes. This also explains why each lead varies in waveform. The deflections in each lead correspond to whether the electrical signal is moving towards or away from the heart. When the electrical signal is moving towards the electrode, the deflection in the waveform is positive. The deflection is negative when the electrical signal is moving away from that electrode.

4.2.3 Coronary Artery Disease

Coronary Artery Disease (CAD), the health problem WearTech chose to tackle, is mainly caused by atherosclerosis. Atherosclerosis is simply described as the build up of plaque inside the blood vessels, constricting the amount of blood flow to the heart. It is synonymous with Ischemic Heart Disease and Coronary Heart Disease. CAD is irreversible, but procedures such as angioplasty, stent implantation, or coronary artery bypass graft surgery (CABG) can be done in more serious cases. CAD should be monitored and managed to prevent further aggravation through a change in lifestyle, and stabilised through medication depending on the stage and extent of atherosclerosis.

Severe cases of CAD where blockages in coronary arteries are significant may result in

Acute Coronary Syndrome (ACS) (Kapit, Macey, Meisami, 2000).

There are three main types of ACS and patients undergoing any case of ACS should seek medical help:

- ST Segment Elevation Myocardial Infarction (STEMI).

STEMI is what most people think of when they hear the term "heart attack". STEMI is the most serious type of heart attack, where there is a long interruption to the blood supply. This is caused by a total blockage of the coronary artery, which can cause extensive damage to a large area of the heart (Hampton, 2013).

- Non-ST Segment Elevation Myocardial Infarction (NSTEMI).

NSTEMI can be less serious than a STEMI. This is because the supply of blood to the heart may be only partially, rather than completely, blocked. As a result, a smaller section of the heart may be damaged. However, NSTEMI is still regarded as a serious medical emergency. Without treatment, it can progress to serious heart damage or STEMI (Hampton, 2013).

- Unstable Angina.

Unstable angina is the least serious type of ACS. However, like NSTEMI, it is still regarded as a medical emergency as it can also progress to serious heart damage or STEMI. In unstable angina, the blood supply to the heart is still seriously restricted, but there is no permanent damage, so the heart muscle is preserved .

CAD can also result in the occurrence of angina pectoris or stable angina. Stable angina is a condition denoted by chest discomfort during strenuous activities. Stable angina in general is not usually a condition of significant concern unless it develops further into unstable angina. As long as a patient with CAD is not undergoing and/or have not recently experienced ACS, they can be classified as stable and should not be of significant concern given that they have been cleared by their medical professional.

These categories of ACS and stable CAD set the basis of determining the final output classifications of the 2D-CNN.

4.2.4 ECG of Acute Coronary Syndrome

After understanding CAD and its effects on the human body, a more technical approach of understanding CAD on an ECG was the next focus (Herring & Paterson, 2006).

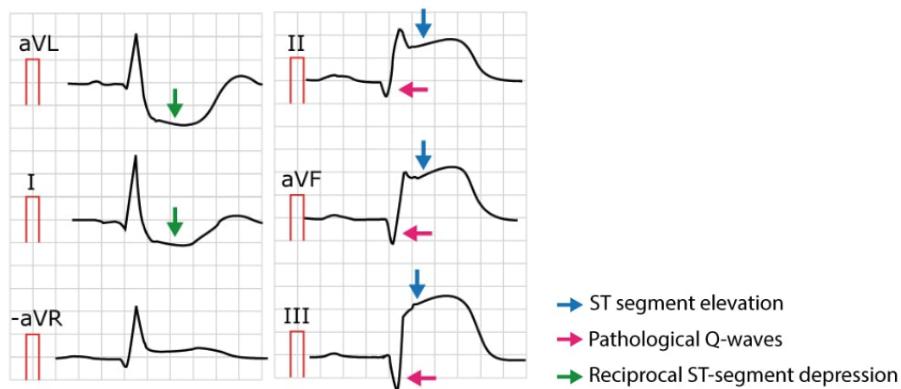


Figure 24: ECG indicating STEMI (Hampton, 2013)

4.2.4.1 STEMI Patients undergoing STEMI will have elevated ST-segments in their ECG leads corresponding to the part of the heart that is damaged as labelled by the blue arrow in Figure 24. This means that the patient should seek immediate emergency help because they are experiencing a heart attack (Herring & Paterson, 2006).

The ST-elevation must exist in at least 2 anatomically contiguous leads to be STEMI. Anatomically contiguous leads mean they could be above, below, or beside each other in the ECG image (Herring & Paterson, 2006). Sometimes, STEMI is accompanied by ST segment depressions in leads which detect from the opposite angle of the leads showing the elevation (Hampton, 2013). After a variable time, usually within a day or so, the ST segments return to the baseline, the T waves in the affected leads become inverted, and Q waves develop. Once Q-waves and inverted T-waves have developed following infarction, these ECG changes are usually permanent (Hampton, 2013).

4.2.4.2 NSTEMI NSTEMI is defined by ST-segment depression and accompanied with either a flat T-wave or T-wave inversion. The leads displaying ST-segment depressions does not necessarily reflect the location of the blockage, and hence the ECG is unable to

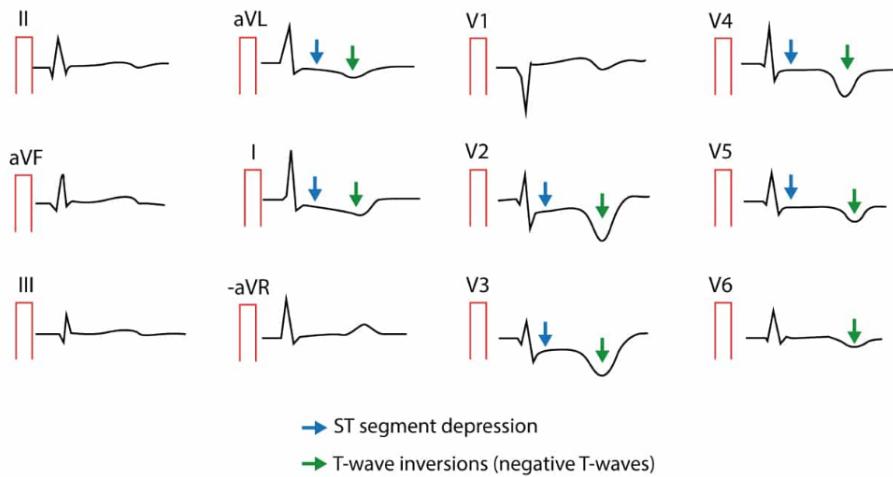


Figure 25: ECG indicating NSTEMI (Hampton, 2013)

localize blockage unlike for STEMI. It is important to note that it is the ST-segment depression that reflects the cardiac event. Q-waves however do not develop in the case of NSTEMI.

4.2.4.3 Unstable Angina In an ECG depicting unstable angina, changes in the ST-segment would have occurred. This means ST-segment elevation or depression would be the first determining factor. Followed by testing the troponin levels of the patient. If the troponin level is normal, then it will be categorized as unstable angina. However, if the troponin levels are elevated, it would then be categorized as either STEMI or NSTEMI depending on the ST-segment. Detecting the level of troponins in a patient would require a blood test, making it unfeasible to detect solely based on an ECG image (Herring & Paterson, 2006).

However, because unstable angina is still a case of ACS and is a concern, albeit not an urgent one, detection of any change in ST-segment should be the first determining factor when reading an ECG (Herring & Paterson, 2006).

4.2.5 ECG of Stable CAD Patients

Stable angina in an ECG can only be depicted in the occurrence of a stress test. A stress test is when the ECG of the patient is taken during exercise. If stable angina is present during the stress test, ST-segment depressions should be seen (Herring & Paterson, 2006) (Hampton, 2013).

Otherwise, given that patients with prior occurrences of ACS eventually have their ECG return to normal, it could be assumed that stable patients have a normal and stable ECG.

4.2.6 Ideal 2D-CNN Output Classifications

Based on previous segments and the understanding of how the different classifications of CAD affected the ECG differently, it seemed feasible to have a 2D-CNN return an output of all the available classifications aside from unstable angina as a blood test determining the patient's troponin levels were needed.

ST-T changes in an ECG normalizes within days or weeks depending on the severity of the infarction. However, the QRS changes are mostly permanent, particularly the Q-wave. Isolated T-wave inversions occur only after the cardiac event has surmounted. This means that in all case scenarios of detecting ACS, the ST-segment is the main feature of the ECG to look at, followed by which lead the change occurs in.

All the possible classifications of CAD are listed:

1. Stable
2. Stable Angina
3. NSTEMI
4. STEMI

Subsequent segments would proceed to search for available databases representative of the ECGs of these classifications to make training the 2D-CNN possible.

4.3 Approach: Obtaining ECG Databases

In continuation to having a concept of the feasible CAD classifications the 2D-CNN should return, a thorough search for available ECG databases was conducted. Upon understanding the vital features the ECG image has to contain for accurate diagnosis of CAD classification, a list of requirements for the database was developed:

1. All ECG databases has to be 12-lead.
2. ECG has to be specific to CAD.
3. Dataset size of each classification has to be sufficiently large for accurate 2D-CNN training.
4. Dataset size for each classification has to be balanced for accurate 2D-CNN training.
5. Background information on the patient is ideal to assist in identifying the appropriate classification.

4.3.1 PhysioNet Database

PhysioNet is a website managed by the members of Massachusetts Institute of Technology (MIT) Laboratory for Computational Physiology and was established in 1999 with the support of the United States' National Institute of Health (NIH). PhysioNet at large provides free access to collections of physiological and clinical data and related open-source software. Research from Thesis A established that previously built 2D-CNNs for arrhythmia extracted datasets from PhysioNet.

Among the 35 databases available on PhysioNet, the PTB Diagnostic ECG Database and the STAFF-III Database were the only databases that met the criteria listed. The biggest limiting factor was obtaining 12-lead ECGs.

4.3.1.1 PTB Diagnostic ECG Database The database contained 549 records from 290 patients and each patient was represented by one to five records. Each patient record consists of 15 simultaneously measured signals: the conventional 12-leads and three additional Frank's lead ECGs represented as VX, VY, and VZ. The Frank's lead ECGs, however, was omitted from use because it was not represented in clinical standard

ECGs and would not be in line with the STAFF-III Database ECGs, creating discrepancies in training datasets.

In this database, out of the 290 patient records, 22 were undefined and 52 were healthy control patient records each with more than two minutes of ECG data. This would be the main resource for the healthy ECG classification dataset. The remaining 216 consisted of patients with different types of CAD associated diagnosis as listed below in Table 14.

Diagnostic class	Number of patient records
Myocardial infarction (STEMI/NSTEMI)	148
Cardiomyopathy/Heart failure	18
Bundle branch block	15
Dysrhythmia	14
Myocardial hypertrophy	7
Valvular heart disease	6
Myocarditis	4
Miscellaneous	4

Table 14: Various classifications from PTB Diagnostic ECG Database

Upon reviewing the STEMI and NSTEMI ECGs from the 148 patient records as shown in Table 14, there was an imbalance of data within the classification, whereby a significant portion of the ECGs did not contain any form of ST-segment changes. After comparing the ECGs with that of the patient profile, it was indicative because the cardiac episode had occurred a few weeks prior to when the ECG was taken. This was supported with remaining T-wave inversions in a lot of the patient ECGs, meaning that the cardiac episode had occurred in the past and the ECG was in the process of stabilizing. As a result, there were only a handful of patient records containing obvious ST-segment elevation and depression.

The remaining diagnostic classes shown in Table 14 had some ECGs indicative of ACS too,

but mostly had ECGs reflective of a past cardiac episode.

The 52 healthy controls' ECGs were looked through and controls with isolated T-wave inversion removed from the classification. Albeit isolated T-wave inversion not being indicative ACS, it could still mean that the patient had suffered from a previous cardiac episode and was unaware. The T-wave would be the last remaining traces of indicating the occurrence in an ECG image. Hence, to err on the side of caution, it was more ideal to remove such patients from the list of healthy controls.

4.3.1.2 STAFF-III Database This database contains ECG recordings from 104 patient records, whom were undergoing a coronary balloon angioplasty procedure. This is a procedure performed for CAD patients by inserting a long, thin tube called a catheter, with a small balloon tip into the blocked artery. The balloon is inflated at the blockage site in the artery to flatten or compress the plaque against the artery wall, hence relieving the blockage.

Out of the 104 records of patients with occlusions, 35 of the patients had previous myocardial infarction. An occlusion in an artery is an occurrence of ACS, and this means that the patient requiring a coronary balloon angioplasty procedure should have ECGs indicative of ST-segment changes, be it STEMI or NSTEMI. This database would prove useful in increasing the variability of the dataset for the STEMI and NSTEMI classifications for training the 2D-CNN.

The ECG recordings reflect pre-inflation ECG and balloon inflation. The mean inflation time was 4 minutes and 23 seconds, ranging from 1 minute 30 seconds to 9 minutes and 54 seconds. Hence, the data extracted should be before the 1 minute and 30 second mark to ensure no ECG data indication stabilization is obtained.

4.3.2 Finalizing 2D-CNN Output Classification and Context of Device Use

Based on the limited availability of datasets for the ideal classifications previously listed, the final classification that was for the 2D-CNN to be feasibly trained for had to be less specific. Furthermore, there was no appropriate datasets indicative of stable angina in

an ECG stress test, hence the classification could not be represented. As a result, the final decision was to place all ECGs displaying any symptoms of ACS into a category of 'Unstable'. All other ECGs displaying a healthy record would then be 'Stable'.

A patient is first cleared by their medical professional to be stable, be it with the help of medication or medical procedures. Whereas unstable means that the patient is in need of medical assistance and a follow up on their condition is needed.

The resulting 2D-CNN after training should be able to output with a high accuracy whether the patient is stable or not. In the context that the 2D-CNN returns a stable classification, the patient would then not have a need to seek medical help and continue with regular weekly monitoring via the device or the moment there is any chest discomfort or pain.

The weekly monitoring is the maximum time a patient should check their ECG because the occurrence of an ACS cardiac event might have occurred without the patient feeling any symptoms or discomfort. As previously mentioned, the ST-segment changes of the ECG would stabilise in a few hours to a few days after a cardiac event, but the ECG would have remaining T-wave inversion, hence the this feature was included as an unstable classification and the patient should seek medical help.

The main concept is for the patient to be accurately aware of their cardiac status and only schedule a medical appointment the moment their condition is no longer stable.

4.3.2.1 Stable As mentioned, the lack of available datasets for stable angina meant that its classification could not be independently represented. As a result, the ECGs of stable angina and a healthy ECG was classified together under the umbrella of being stable.

This whole dataset would comprise of the available healthy ECG data which was mostly obtained from the PTB Diagnostic ECG Database and the 2D-CNN trained on this dataset.

4.3.2.2 Unstable Even though data sets were available for both STEMI and NSTEMI where the ST-segment showed obvious changes in the ECG, these ECGs were a minority in the two databases. Justifiably, patients do not receive immediate help and often have

prescribed medication to stabilise the ST-segment changes. The imbalance in size of the datasets would pose a challenge in training the 2D-CNN and may result in an overfit or underfit. Furthermore, because unstable angina is unable to be specifically classified through an ECG image without the additional blood test for normal troponin levels, the 2D-CNN would not be possible to be trained solely on ECG imaging to classify it. However, since the first step of detection for unstable angina is the change in ST-segment, just like STEMI and NSTEMI, the three classifications were grouped as one body of being unstable. Regardless which occurrence or diagnosis of ACS, the patient would require medical help, allowing for this single unstable classification to hold valid as a case for concern.

Listed are the graphical conditions that would group the input ECG as unstable:

- ST-segment changes in the form of either elevation or depression.
- and/or pathological Q-wave.
- and/or traces of T-wave inversion.

It is necessary that each patient monitors their ECG at a weekly minimum by getting it analysed via the device. It would be ideal to detect a patient as stable only during a cardiac event which would be mainly represented by ST-segment changes. However, there are many cases where patients do not feel any pain, or only a slight discomfort and not identify it as a cardiac event occurring. As previously mentioned, the ST-segment changes of the ECG would stabilise in a few hours to a few days after a cardiac event. As a result, before the ECG continues to stabilise and the condition becomes less identifiable on the ECG over time, justifying the need for the patient to have a weekly check to see if any form of ACS had occurred without them knowing and be able to have it treated (Hampton, 2013).

4.4 Methodology: Extracting & Processing ECG Signals from Databases

The PTB Diagnostic ECG Database and the STAFF-III Database both helmed from the PhysioNet Database.

4.4.1 Extracting ECG Signals

The PTB Diagnostic ECG Database and the STAFF-III Database both helmed from the PhysioNet Database, meaning that the similar formatting of data would allow for extraction methods to be applied to both databases.

The first step was to download the PhysioNet WaveForm DataBase (WFDB) Toolbox for MATLAB. The WFDB toolbox contains a collection of functions for reading, writing, and processing physiologic signals and time series used by PhysioBank databases. This includes the ECG signals from the PTB Diagnostic ECG Database and the STAFF-III Database.

Both PhysioNet databases were sorted according to the patient data with their own directory. Some patients would have multiple ECG signal records while some only had one. Therefore, the task at hand was for the script to access each patient directory and read each ECG signal, and extract the stipulated period of signal recording into a comma-separated values (CSV) file (.csv). The CSV file should contain the ECG voltage signal of each individual lead in order of how the signal would be portrayed in the next stage of processing the signal.

A MATLAB script was written to do the extraction and is presented in the appendix. The following paragraphs gives a brief insight to what the written code does.

4.4.1.1 Accessing ECG signal files Each PhysioNet database had its own directory, followed by its own data directory, and its individual patient directory. Each patient directory then contained the ECG signal data represented in the form of a data (.dat) file and header (.hea) file. This was performed by the MATLAB *urlread* function to access the specific directory.

4.4.1.2 Downloading and storing the ECG signal into a matrix Upon accessing the directory the ECG signal header and data file was contained in, the MATLAB WFDB Toolbox *rdsamp* function would then download the two files for that particular ECG signal for the specific number of samples, N. The number of samples, N, represented the duration of the ECG signal at each millisecond instance of time. This meant that a sample size N of 1000 would extract 1000 milliseconds of ECG signal data (Naumann Silva, 2014).

The header file contained The data file contained
The *rdsamp* function returned the following in the written code:

1. ***ecg***: NxM matrix of M signals with each signal being N samples long. This was where all the voltage signals were stored for the stipulated number of samples, N.
2. ***tm***: Nx1 vector of doubles representing the sampling intervals.

The PTB Diagnostic ECG Database and the STAFF-III Database both had its own order of the 12-lead signals of the ECG. However, since the 2D-CNN would be identifying and classifying based on image features, the image of the ECG had to have the same consistent order of 12-lead arrangement for successful classification to be plausible. Hence, rearrangement of the matrix containing the ECG for each database was conducted in the order of lead V1, V2, V3, V4, V5, V6, I, II, III, VR, VL, and VF.

Furthermore, the VR, VL, and VF leads had to be calculated from the I and II leads as shown in Equation 8, 9, and 10 respectively.

$$VR = -\frac{1}{2}I - \frac{1}{2}II \quad (8)$$

$$VL = I - \frac{1}{2}II \quad (9)$$

$$VF = II - \frac{1}{2}I \quad (10)$$

4.4.1.3 Creating an ECG signal CSV file The previous step saw the creation and storing of data into a matrix in the order wanted. This step saw the creation of a CSV file using the MATLAB *writematrix* function.

Each ECG signal recording would have its own CSV file which contained data for the stipulated amount of time defined in the rdsamp function input.

The first row of the CSV file represents each instance of time in millisecond increments. The following rows then represented the signals of each voltage beat in order of V1, V2, V3,

V4, V5, V6, I, II, III, VR, VL, and VF. Hence, each column represents the ECG signal voltage for each of the 12-leads at that millisecond instance in time.

The order and format of the CSV file has to be consistent throughout because the subsequent step of processing would be done solely on the CSV file data.

4.4.2 Processing ECG Voltage Signals

This processing stage utilized a Python script of code to transform the CSV file of the ECG signals created in the previous segment into individual ECG complex images ready for input into the 2D-CNN for training and testing. This script would automatically access the CSV files and run the process on all the CSV files in that directory. The data in the CSV file was first extracted into a matrix in the Python script so that further processing can be conducted.

4.4.2.1 Processing the ECG Signal for Beat Segmentation The next step was to utilize the *bioecg* function from the installed BioSPPy library and the Christov Segmentation method to perform beat segmentation for the matrix of signals. This means breaking up the ECG into its individual beats. Each ECG would have different signal duration, and varying lengths. Hence, it would not be feasible or accurate for the 2D-CNN to train and recognise the features of an ECG given the lack of consistency. Hence, the necessity of analysing a single ECG complex (Christov, 2004).

The Christov Segmentation method was chosen because of its mathematical straightforwardness and high accuracy with sensitivity of 99.74% and specificity of 99.65% with the additional RR-interval analysis component. The main feature of the Christov Segmentation method detects the QRS complex of each beat, which is the largest deflection of the single heart beat. The RR-interval analysis detects the R-peaks of the signal in lead-II as it is the most stable lead. The signal interval between R-peaks was then halved to create the segmented beat (Christov, 2004).

4.4.2.2 Plotting ECG Image into PNG file The ECG complex was then plotted into a Portable Network Graphic (PNG) File (.png). A PNG file format was chosen because it employs a loss-less compression algorithm to ensure that the file quality does not change over time. Furthermore, a PNG file format is more ideal for created images, which applies for this context of use.

Furthermore, the bordering frame of the image was removed. Since the ECG signals will comprise of single lines, removal of similar features such as a line border would be preferable

to speed up computational time.

4.4.2.3 Image Processing In order to minimize the data size of each ECG complex image such that computational time is minimized, a few steps were taken for image processing to make the image as small as possible within visual reason.

The first step being to crop the image and remove the unnecessary white spaces from the left and right edges of the ECG complex. This step would not affect the signal component of the image.

Next was to convert the image into grey-scale such that the image matrix was further compressed due to the removal of the red green and blue (rgb) channels of the image.

Following, the image was down-sampled by 25% while maintaining the height and width ratio.

Finally, the image was also filtered to make the pixel dimensions less sharp. Initially, the dimensions were at 342 width by 2521 height and then filtered to achieve 184 width by 904 height in pixels. This filter was added using the Python cv2 library and was able to reduce the image size from 44KB to 9KB without affecting the image quality to the visual eye.

4.5 Results: ECG Image & Classification for 2D-CNN

The final image of the ECG complex was 184 by 904 pixels. A final manual re-classification of the images was also performed to ensure that each ECG complex in the dataset was representative of its stable or unstable classification. This was to ensure that the datasets would be as accurate as possible for the 2D-CNN to train with.

4.5.1 Stable ECG Complex Image

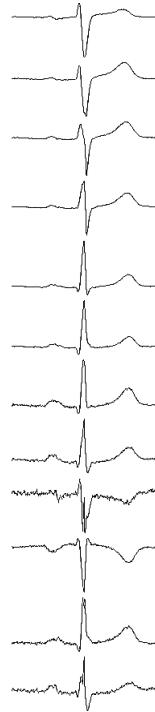


Figure 26: Final image of sample stable ECG complex

Figure 26 represents a sample image obtained from the 8,924 images of stable ECG dataset ready to be used for training of the 2D-CNN.

4.5.2 Unstable ECG Complex Image

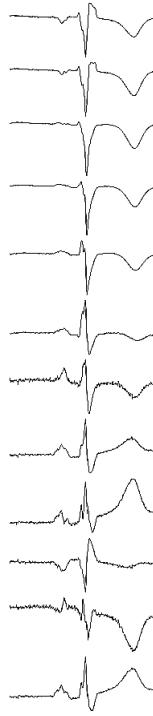


Figure 27: Final image of sample unstable ECG complex

Figure 26 represents a sample image obtained from the 10,641 images of unstable ECG dataset ready to be used for training of the 2D-CNN.

4.6 Future Work

There is a lot of potential for further improvement of this segment of data analysis. The biggest constraint was obtaining not just the appropriate dataset for each classification, but also a dataset sufficiently large such that accurate training for the 2D-CNN model could be performed. The current classification condenses the five ideal ECG classifications for CAD into 2 bodies of stable and unstable instead. Any future work could be focused on successfully performing these five classifications of healthy, stable angina, unstable angina, NSTEMI, and STEMI such that better monitoring and diagnosis of treatment could be performed at home.

One of the benefits of this device is that it is possible to make it self improving. Once the 2D-CNN is implemented and detecting and classifying ECGs, the database of ECGs would expand tremendously. This means that there would be a larger number of ECGs that could be represented, expanding the dataset for each ideal classification. Retraining could be performed with little tweaking to the 2D-CNN code.

To tackle diagnosing stable angina, an ECG performed during exercise (stress test) would be required. Furthermore, no additional electrodes or physical changes to the device will be needed, hence it is fairly feasible to be done. A website should be created for easy access to instructions on how to exercise during a stress test. Also, because a stress test induces stress on the heart, there might be a possibility of NSTEMI or STEMI occurring due to the test. As a result, additional live detection features of myocardial infarction should be added. This would be the challenging segment as the data communication and the 2D-CNN diagnosing would then have to be instantaneous to be able to tell the patient to stop exercising immediately.

As for diagnosing unstable angina, perhaps a small blood test feature could be added to the device to detect the troponin levels of the patient. This should be done when a ST-segment change in the ECG is detected. However, regardless of the result of the troponin levels, the patient would still need to seek medical help due to the fact that any change in ST-segment indicates acute coronary syndrome. Hence the necessity of having an unstable angina classification is not really necessary, but could be implemented nonetheless.

As for NSTEMI and STEMI, a larger dataset of ECGs indicating ST-segment changes would prove most useful in supporting these classifications.

Also, this device could potentially be personally tailored to the patient's ECG for even more accurate analysis. Each patient would have slightly different stable ECGs, and their own stable ECG data could then be used as the stable dataset to train their own 2D-CNN. Understandably, there will be limitations in computing memory as training a 2D-CNN for each individual patient will take more time and cost than a generic model. However, it is a method that can be undertaken to create a tailored user experience, further alleviating the

strain on local specialist medical professionals.

5 Data Analysis II - Zachary Hamid

5.1 Aims & Objectives

The primary aim of the entire data analysis portfolio was to understand and implement a pipeline of performing automated classification of raw ECG data. This pipeline involved obtaining datasets that were appropriate and representative of the problem of classifying patients into stable or unstable categories based on ischemic heart disease. This was followed by processing and filtering this data to ensure it was appropriate for an automated analysis solution. Then finally developing a method for automated classification of this data. Due to the large scope of this role, it was shared with my peer, Calandra Lunardo. The result of this is a reduced scope for this section of the report that focuses on the last stage of the aforementioned pipeline.

The method of automated classification was to involve the development of a 2-dimensional convolutional neural network by understanding and comparing the current techniques used for developing these networks and employing methods relevant for this scope of research. This network was to be applied to 12-lead single-beat ECG images that are processed from raw ECG data through methods detailed in the previous section of this portfolio to determine if a patient has a stable or unstable ECG reading. The convolutional neural network had to attain a high accuracy and correctly predict classifications a majority of the time. This is because the network is designed to be deployed into a medical field and misdiagnosis can be severely consequential.

A secondary aim for this section of the portfolio was to demonstrate technical feasibility of automated classification using this convolutional neural network solution on a Raspberry Pi Zero W, as this was the computing module of choice for conceptual development of the device proposed in this report due to its low cost, compact size and ease of prototyping, making it ideal for use in this application. Due to this decision, there were restrictions placed on the network in that it had to be computationally performant to conform to the relatively low processing power of the Raspberry Pi Zero W, obtaining predictions within a reasonable timeframe.

5.2 Methodology

In order to achieve the aims and objectives for this section of the data analysis portfolio, intensive research had to be undertaken regarding the various aspects that comprise a convolutional neural network in order to establish appropriate design decisions for the network architecture that were relevant to the task of 12-lead, single-beat ECG image classification on a Raspberry Pi. As part of these design decisions, considerations needed to be made regarding computational feasibility and that due to time and cost constraints, network tuning wasn't possible.

Once all design decisions were made, a finalised architecture for the network had to be developed and subsequently implemented in code. Following this implementation, the available preprocessed dataset, from work conducted by Calandra, needed to be separated and prepared appropriately for training and testing the network. This training and testing process then had to be performed using a high-performance computing solution. Finally, the network had to be deployed to a Raspberry Pi for performance testing.

5.2.1 Development of a 2D CNN

5.2.1.1 Convolution layers There is one central component that separates a convolutional neural network from any other machine learning model and that is the convolution layers. The primary purpose of these convolution layers is to perform feature extraction of the input passed through it and generate a set of feature maps. These feature maps are, in essence, abstract representations of salient features of the input, for example corners or edges. These maps are created by applying what is known as a convolution to this input which involves a kernel sliding over the input array and performing a linear operation on it based on the values present in the kernel. The output of this operation is then passed through the activation function layer which is detailed in section 5.2.1.4 to determine what information gets passed through to successive layers. The values present in the kernel are learnable parameters which means they are iteratively updated during the training process in order to obtain better feature maps that more accurately depict important parts of the image for classification.

There are a few parameters relating to this layer that must be set manually. These

parameters are known as *hyperparameters* because they are not learned during the training process. These hyperparameters are the size of the kernel (size of the window to perform operations), the number of kernels (the number of feature maps to obtain, affecting the output size of this layer), and the kernel stride (how far to slide the kernel over input between each operation) (Yamashita, Nishio, Do & Togashi, 2018). In most instances, the kernel size is either 3x3, 5x5, or 7x7, with the kernel stride set to 1. Since changes in very small neighbourhoods in the ECG data are important for classification, for the purposes of this research the smallest kernel size of 3x3 and a stride of 1 were chosen. An example of the convolution process is shown in Figure 28 below.

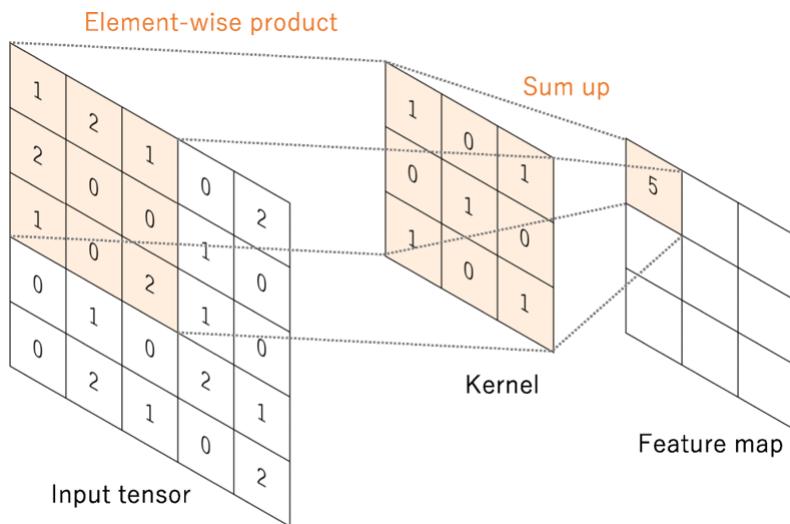


Figure 28: Convolution Process Example (Yamashita, Nishio, Do & Togashi, 2018)

Regarding the number of groups of convolution layers plus max pooling layers, the choice was motivated by pre-existing network configurations proposed by Simonyan & Zisserman (2014) that detail networks of increasing complexity and their performance. Due to computational limitations that needed to be considered, and since the network was planned to be deployed to a Raspberry Pi Zero W, the final decision was to reduce both the number of groups and number of kernels by half from the configuration B network shown in their research. This configuration was chosen as the motivation since it produced a very good error percentage to network size ratio. Additional motivation for this is due to the fact that with an input size of 224x224 the networks that were proposed by Simonyan & Zisserman still all had over

100 million parameters. With an input size of 184x904, as is the case for the ECG images used in this project, this would mean the number of parameters would become technically impractical and not computationally feasible.

In addition to these chosen parameters, the number of convolution layers was also an important part of the network that had to be considered. Since pooling layers (detailed in section 5.2.1.2) were to be used, there will be information loss in the network. To help counteract this, it was decided that two successive convolution layers would be used before each pooling layer. This helps alleviate the potentially problematic information loss by building up a better representation of the data before it becomes spatially reduced from pooling.

5.2.1.2 Pooling layers In almost all convolutional neural networks, there exists pooling layers that are present after convolution layers. The primary function of this layer is to reduce the complexity of the data being passed through the network, and consequently, reducing required computational effort. The benefits that this layer provides are local positional invariance, meaning shifting of the subject in the input data won't affect classification as well as reducing overfitting due to forcing information loss through this process of data complexity reduction (detailed in section 5.2.1.5).

Similar to the convolution layer, the pooling layer involves a kernel that slides over the data that is passed in with the kernel size and stride provided as hyperparameters, though there is no option for more than one kernel as there is with convolution layers. There are two main implementations of the pooling layer that were researched; these are average pooling and max pooling.

- **Average pooling** takes the average value within the kernel and passes this to the output data.
- **Max pooling** takes the maximum value within the kernel and passes this to the output data.

An example of the effect this has on downsampling is if a kernel size of 2x2 was chosen with a stride of 2 (this is standard for most CNN architectures), then the output data would be

a quarter of the size of the input data as shown in Figure 29 below. Though it should be noted if the data is 3D, the depth of the data will not be downsampled. Although the data being passed into the input of the network is 2D, convolution layers build up a 3D array of these 2D inputs, in which the depth represents each feature map. Therefore, each individual feature map will be downsampled but the number of feature maps will remain the same.



Figure 29: Max vs. Average Pooling (kernel size: 2x2; stride: 2) (Hamid, 2019)

The most commonly used method of pooling is max pooling, and this is the method chosen to be used for the 2D CNN model. Max pooling has a certain desirability over average pooling due to the fact that it is more sensitive to salient features, this is an important feature since the ECG input data involves lots of white space that provides no information surrounding important edge-like features. If average pooling was to be used instead, the white space would have an impact on the information that is passed through the network which is undesirable behaviour for this particular problem.

5.2.1.3 Densely connected layers The main purpose of the densely connected layers is to provide a mapping from the feature maps generated by the convolution layers to an output representing a predicted classification. The term densely connected comes from the fact that all neurons in any given layer are connected to all neurons in the next and previous densely connected layers as well as all inputs into the first layer. The input into the initial densely connected layer is a 1-dimensional flattened representation of the final feature map volume generated by the convolution and pooling layers. This information passes through all the densely connected layers through linear functions that are learned during training, as well as activation functions for each layer, to a final activation layer that determines the predicted classification (Yamashita, Nishio, Do & Togashi, 2018).

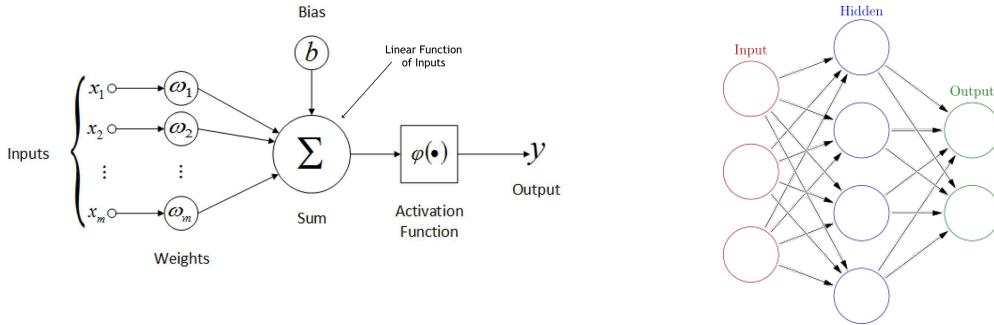


Figure 30: Artificial Neuron (Ahire, 2018) Figure 31: Neural Network (Wikipedia, 2019)

There were several considerations involved in determining both the number of layers and the size of the layers for this section of the network. Firstly, shallower (i.e. less densely connected layers) networks are in general less accurate than deeper networks but faster to train, whereas wider networks (i.e. layers with a large number of neurons in each layer) are more accurate than skinnier networks (Basha, Dubey, Pulabaigari & Mukherjee, 2019). Secondly, the number of learnable parameters was a technical constraint that affected training time, and therefore had to be balanced. An initial choice for the number of neurons in each layer was 1024, though due to the sheer size of the data being passed through, this resulted in 340 million parameters that needed to be learned for the first dense layer alone. This was deemed to be too impractical for training in a reasonable amount of time and it would also increase the probability of the model overfitting the data (a concept that is explained in section 5.2.1.5). Due to this, the number of neurons in each layer was reduced to 256, resulting in approximately 68 million parameters for the first layer, meaning the model was much more feasible to train. Finally, only two densely connected layers were chosen since there was a high number of convolutional layers already so combining this with a deep densely connected layer network would increase the risk of overfitting the data and losing network performance.

5.2.1.4 Activation function The activation function (AF) of both the convolution and densely connected layers of a CNN determine how information gets passed from the output of one layer to the input of the next. The choice of activation function can have a significant effect on not only the accuracy of the network but also the training time. The table below outlines the various advantages and disadvantages of each AF.

Table 15: Comparison of AFs

AF	Advantages	Disadvantages
Sigmoid	<ul style="list-style-type: none"> Smooth gradient Output values bound between 0 and 1 Clear predictions 	<ul style="list-style-type: none"> Vanishing gradient Outputs not zero centered Computationally expensive
tanh	<ul style="list-style-type: none"> Zero centered Like Sigmoid 	<ul style="list-style-type: none"> Like Sigmoid
ReLU	<ul style="list-style-type: none"> Computationally cheap 	<ul style="list-style-type: none"> Dying Neuron phenomenon
Leaky ReLU	<ul style="list-style-type: none"> Prevents Dying Neuron phenomenon Like ReLU 	<ul style="list-style-type: none"> Results not consistent

Nwankpa et al. (2018) performed an extensive analysis of various activation functions that are in regular use for deep learning problems and comparatively determined their performance. The Rectified Linear Unit (ReLU) activation function was the most widely used function for deep learning applications and has state-of-the-art performance at the time of writing. Additionally, ReLU has proven to allow faster network training and offer improved performance and generalisation over the sigmoid and tanh functions. This is because it represents a nearly linear function and thus it is easy to optimise using gradient-descent methods that are explained in section 5.2.1.7 (Hamid, 2019). Exactly linear functions are undesirable for deep learning as it results in the entire network collapsing down to a single linear function of the input, removing any potential for non-linear modelling. However, although ReLU represents a *nearly* linear function, it is still non-linear and is therefore appropriate to use.

Unfortunately, there is a widely known issue with this AF known as the "Dying Neuron" phenomenon wherein under certain conditions, some neurons or cells within a network will become inactive resulting in no data being able to pass through them. This effectively renders that section of the network useless. The solution to this is to use a revised version of this AF called the "Leaky ReLU" that is shown in Table 15.

For the convolution and densely connected layers in the network, the most desirable advan-

tage for Leaky ReLU is that it is computationally cheap. Since incredibly large amounts of data pass through these layers, it is more important that it is computationally feasible for prediction and training to be performed, especially if the final network is to be deployed on a low compute power device such as the Raspberry Pi Zero W. The disadvantage that is shown here is actually a positive for the particular application of interest since inconsistent results helps to add slight noise to the network and provide a means of reducing overfitting and so Leaky ReLU was the AF of choice for all layers except the max pooling (no AF) and final layers.

Previously, the chosen activation function for the final layer of the network was Softmax. The original motivation for this is because Softmax was most suitable for multi-class probabilistic prediction. This was appropriate since initially the ECG data was to be classified into 3 different categories. However, with a scope reduction due to inadequate representative data, it was decided that only two categories would be used. Due to this decision, the most appropriate activation function for the final layer became the Sigmoid function due to its bounding between 0 and 1, clear predictions and smooth gradient (Nwankpa et al., 2018).

5.2.1.5 Regularisation One of the most problematic occurrences in machine learning is the overfitting of a model to a particular dataset. Overfitting means that a model has been trained to match the training dataset *too closely*, and as a result, generality of the model is lost. When a model doesn't generalise well, it means that the model will not perform effectively on any new data that was not part of the dataset that it was trained on.



Figure 32: Effect of overfitting data (Despois, 2018)

There are many factors that can cause overfitting, such as high model complexity (Hawkins, 2004), lack of data, data that is not entirely representative of the population that the model is trying to generalise to, or simply training for too long on the same data. Since the architecture that was to be devised was not able to be tuned later, a certain degree of

over-engineering was involved in order to make the model as robust as possible from the beginning. Additionally, there was a concern regarding lack of diversity in the available dataset, since these were only taken from two different databases and a few hundred patients. This meant the final model was quite complex and as a result, measures to combat overfitting had to be employed. This is the purpose of regularisation techniques and while generally only one or two techniques are used by most networks, three unique techniques were employed into the final network for this project so that each technique can contribute less aggressively than usual and allow for more diversity in overfitting risk reduction. These three techniques are dropout, L1/L2 regularisation and batch normalisation.

Dropout: This method of regularisation combats overfitting by ignoring or "dropping out" an input from the previous layer with some probability p during the training process, where p is another hyperparameter that needs to be manually chosen. The purpose of ignoring a portion of the data is to simulate sparsity, and introduce noise into the network to prevent the model from training on the actual data too closely (Srivastava et al., 2014). The effect of this can be seen in Figure 33 below.

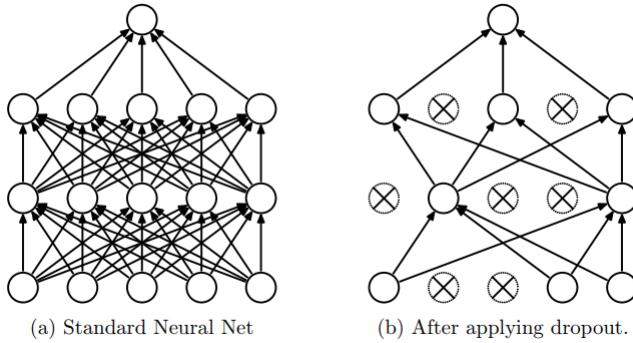


Figure 33: Effect of dropout on a neural network (Srivastava et al., 2014)

It was discovered by Baldi et al. (2013) that a dropout value of 0.5 obtains maximum regularisation. However, since dropout is being applied alongside other regularisation techniques in this case, a slightly less aggressive value of 0.4 was chosen instead.

L1/L2 Regularisation: L1 and L2 regularisation work by adding a cost to the weights of the densely connected layers of a network, where this cost scales with the weights themselves

and some value λ and so the larger the weight, the higher the cost. The result of this is a discouragement of learning an overly complex model which therefore reduces the risk of overfitting.

Ng. (2004) and Mazilu & Iria (2011) compared the performance of L1 and L2 regularisation and discovered that in the presence of many irrelevant features, L1 regularisation significantly outperformed L2 regularisation. L1 regularisation also possesses the benefit of automated feature selection by forcefully shrinking the weights of irrelevant features to 0, as compared to L2 regularisation where it would just shrink these weights some value close to, but not equal to 0 (therefore these features would still contribute to the network) (Moore & DeNero, 2011). Since the input data that is being dealt with involves a large amount of white space (hence, irrelevant features), it is inevitable that sparsity in the final feature maps will be passed through to the densely connected layers of the network. Therefore, L1 regularisation was the method employed in the network with a low value of $\lambda = 0.001$ so this regularisation isn't aggressive, as it is being employed alongside other regularisation techniques.

Batch Normalisation: This is a method that was developed for the purpose of speeding up training of deep learning networks as well as reducing overfitting. In some cases, it can entirely eliminate the need for other regularisation techniques but due to the complexity of the proposed network shown in section 5.3 and limitations on dataset availability, this method was implemented alongside other techniques. Batch normalisation works by reducing something known as *Internal Covariate Shift* which is defined as the change in the distribution of network activations due to the change in network parameters during training. By fixing the distribution of the layer inputs as the training progresses, the training speed is increased, an additional benefit is that it allows for higher learning rates without issues of getting trapped in local minima. Higher learning rates are beneficial since they allow for quick convergence in parameter learning, however before batch normalisation this came at the risk of the learned parameters becoming stuck in local minima and thus the global optimum is never achieved (Ioffe & Szegedy, 2015). It is a method that has become standard-use in industry for deep learning networks and so it is a method that was employed for the final CNN architecture (Hamid, 2019).

5.2.1.6 Loss function In machine learning, the loss function is a function that serves to represent a cost associated with making a prediction, and for simplicity can be seen as a representation of the certainty of a prediction. The process of learning involves attempting to minimise this loss function, and thereby consequently increasing the certainty of correct predictions. There are many loss functions available in machine learning, however, many of these are inappropriate for use with a convolutional neural network.

The chosen loss function for this network was binary cross-entropy, the reason this loss function was chosen was not for performance reasons as the previous design decisions were, but because it provides an intuitive method for understanding the performance of the network and how the loss relates to predictive confidence. Cross-entropy is used to quantify the difference between two probability distributions, the "true" probability distribution of the classes and the models predicted distribution. The "true" distribution that this loss function attempts to match is a probability of 0 for the incorrect class and 1 for the correct class. The equation for cross-entropy for binary classification is given as:

$$H = -(y * \log_e(p) + (1 - y) * \log_e(1 - p)) \quad (11)$$

Where H is the binary cross-entropy loss, p is predicted probability of class A, and y is true probability of class A. An example on the intuitive nature of this loss function is given below:

If the correct class is class B, then we have the "true" distribution of:

$$Pr(A) = 0.0, Pr(B) = 1.0$$

If the network outputs a prediction of:

$$Pr(A) = 0.3, Pr(B) = 0.7$$

Then the cross-entropy is calculated as:

$$H = -(0.0 * \log_e(0.3) + 1.0 * \log_e(0.7)) = 0.357$$

Then, if the network outputs a worse prediction of:

$$Pr(A) = 0.9, Pr(B) = 0.1 \implies H = -(0.0 * \log_e(0.9) + 1.0 * \log_e(0.1)) = 2.303$$

As shown above, a larger loss results, since the model is very incorrect about the prediction. Since the loss is only contributed to by the predicted probability of the correct class, once the loss is obtained, it is very easy to solve for this probability to determine how close a models predictions were and this will be useful for demonstrating model performance.

5.2.1.7 Optimiser The purpose of an optimiser in machine learning is to minimise the aforementioned loss function and, in the case of a CNN, find an optimal solution for the weights of a network that give this minimum loss. It can be thought of as the method of travelling down a hill to the bottom, which represents a minimum of the cost function with different methods taking larger or smaller steps in a certain direction based on various factors. Choosing an appropriate optimiser is important as it can have a significant effect on the solution that is found by the network during training as well as the training time itself. There are two categories of optimisers: non-adaptive and adaptive. Non-adaptive methods are those in which the size of the step, known as the *learning rate*, taken in any given direction in an attempt to minimize the loss function is a constant value. Conversely, adaptive methods possess the ability to vary this step size.

There are countless implementations of both non-adaptive and adaptive methods, however many of these are mostly only suitable for very specific applications that were not relevant for this CNN and so only the most common and relevant implementations are shown in Table 16 below:

Table 16: Optimisation methods

Non-adaptive methods	Adaptive methods
Batch Gradient Descent	Adaptive Gradient (AdaGrad)
Stochastic Gradient Descent (SGD)	RMSProp
	Adaptive Moment Estimation (Adam)

Batch Gradient Descent involves updating network weights only after all the training data has passed through once. The result of this is slow training time, and infeasibility for large datasets since calculations are performed on all data points at once. Since the CNN for this project was trained on approximately 14,000 184x904 images, this was not an option. In comparison, Stochastic Gradient Descent (SGD) updates network weights after N data

points have passed through, where N is known as the *batch size*, and so the number of data points is not an issue regarding computational feasibility. By default, SGD was the only option of the non-adaptive methods that could be considered.

Comparing the adaptive methods, Kingma & Ba (2014) showed in their research proposing the Adam optimiser method, that it empirically outperformed the other adaptive methods listed above. This is due to the fact that the Adam method is based on both AdaGrad and RMSProp and employs their advantages whilst also improving on their shortcomings. The most desirable advantage that Adam possesses stems from the AdaGrad method, in that it is efficient and performant for sparse gradients (Duchi et al., 2010). The reason this is desirable is because the input data that is being dealt with by the network involves large areas of white space that possess no information that is useful to the network. Due to the AdaGrad properties present in Adam, these gradients would cease to impact the loss function minimisation, thereby increasing performance. It is unnecessary to compare AdaGrad and RMSProp in detail since it has already been proven that they are outperformed by Adam and so this was determined to be the most optimal adaptive method.

Research conducted by Wilson et al. (2017) concluded that the Stochastic Gradient Descent method does in fact obtain greater accuracy compared to adaptive methods with *optimal tuning* of the learning rate. However, it can be an arduous process to obtain this optimally tuned learning rate, and most importantly, tuning of the network was not feasible for this project due to time and cost constraints. Consequently, Adam was almost guaranteed to produce higher performance, and hence, this was the optimiser employed for the network.

5.2.2 Implementation and training of 2D CNN

Once the final design decisions were made, the architecture was developed and implemented through the use of Keras (Chollet, 2015), which is a Python-based neural network library that specialises in convolutional neural networks. The code that was written for this implementation can be seen in Appendix F. Since the network was trained on such a large dataset of high-resolution images, implementation of a high performance computing solution for this training process was necessary. The most suitable solution was determined to be deploying the model to a virtual machine instance created on Google Cloud's Compute

Engine, utilising 4 NVIDIA Tesla V100 GPUs. This was important for this task since the network being trained is based around performing operations on images, and GPUs specialise in parallelisable workloads. However, since renting this hardware is expensive, it introduced cost constraints and prevented multiple iterations of network architecture tweaking from being possible. The original hardware that was chosen to train the network was 24 Intel Xeon CPUs, however this resulted in a training time of approximately 100 minutes per epoch (where an epoch is simply passing all training data through the network once), due to the inability for the CPUs to efficiently parallelise the workload. After employing the 4 GPUs, the training time was significantly reduced to just over 1 minute per epoch, making it feasible to train for several hundred epochs in a reasonable amount of time. Early stopping was employed to the training process which stops model training if a particular performance metric hasn't improved for a certain number of epochs, this works to prevent overfitting so that the model doesn't keep training even when the performance isn't improving.

5.2.2.1 Splitting the dataset

In order to obtain an effective performance measure for how well the model proposed in sections 5.3 and 5.3.1 suits the task of automated analysis of ECG data, the available preprocessed ECG image dataset as a result of the work conducted by Calandra had to be separated into training, validation and testing sets. The training set contained the actual data that will be used to train the network itself.

Generally, a validation set would only be necessary in cases where a search of different models is to be performed, using the predictive performance on the validation set as a heuristic for determining the most optimal model. However, due to the computational complexity involved in training a CNN, this automated space search is generally not employed and instead, replaced by manual parameter tuning. Due to time and cost constraints, this manual tuning was not possible and so the sole purpose of this validation set was to provide an independent dataset for performance analysis to determine the best network weights. The reason a validation set is more desirable for network performance estimation and selection over the training set, is there is a bias of the model towards the training set since this is the data that the model is trying to fit to. This means that performance on the training set is not indicative of performance on data outside of this dataset.

There is also inherent bias involved in this process of selection using the validation set however, since this set is being used to find the best model, and so models that suit this dataset more will be chosen, regardless of whether this model would suit new data outside of this dataset. Due to this, a third and final set, the testing set, had to be used to obtain the most realistic performance metrics for the final network, as free from bias as possible. The 19,535 ECG images were split into sets with a ratio of 10% for testing, and of the remaining 90%, an 80:20 split was utilised for training and validation, respectively. The exact number of ECG images in each dataset is shown in Table 17 below.

Table 17: Split of ECG images

Set		
Training	Validation	Testing
14,064	3,517	1,954

Before sorting the data into these three sets, however, it had to be repeatedly randomly shuffled in order to both remove external selection bias (if one was to sort the data manually into the three sets), and ensure a high likelihood of having a 50/50 balance of stable and unstable data in each of the three datasets. This balance is important to ensure the model doesn't bias itself towards one class over another by simply being exposed to more data from that class.

5.2.2.2 Deployment to Raspberry Pi

In addition to the split of the dataset shown in Table 17 above, 30 ECG images were completely removed from the dataset and set aside, 15 from each category. This was in order to test how long the model would take to make a prediction on a Raspberry Pi, in similar conditions to what could be expected if the product was to be used. The correct environment was set up on the Raspberry Pi and the trained network was then deployed and predictions were performed on these 30 ECG images.

5.3 Final 2D CNN architecture

From the numerous design decisions that were outlined in section 5.2.1, the final 2D convolutional neural network architecture was built and is shown in Figure 34 and 35 below.

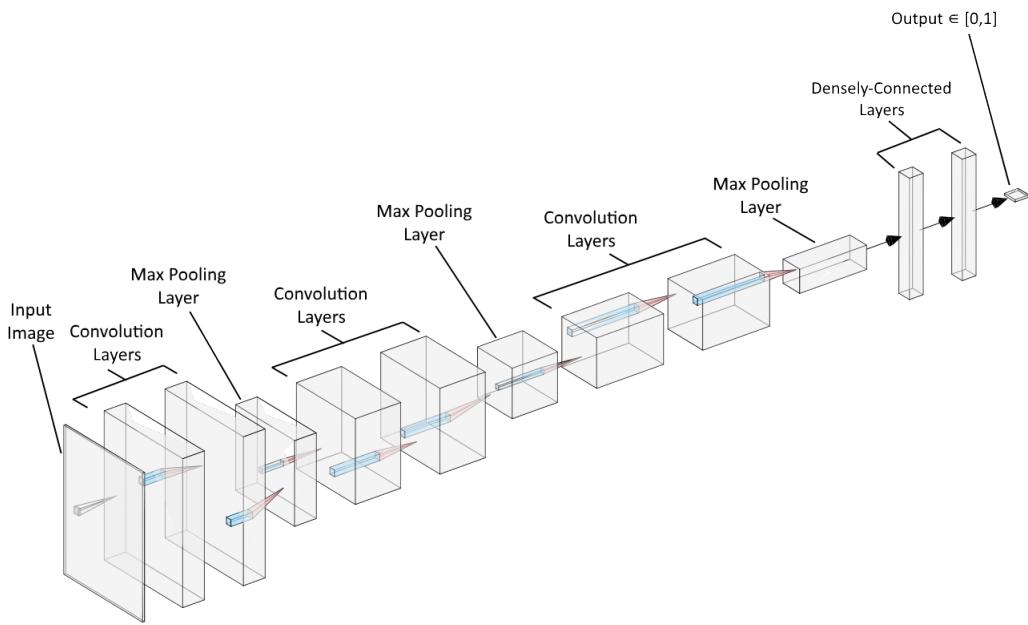


Figure 34: Simplified visualisation of devised 2D CNN architecture

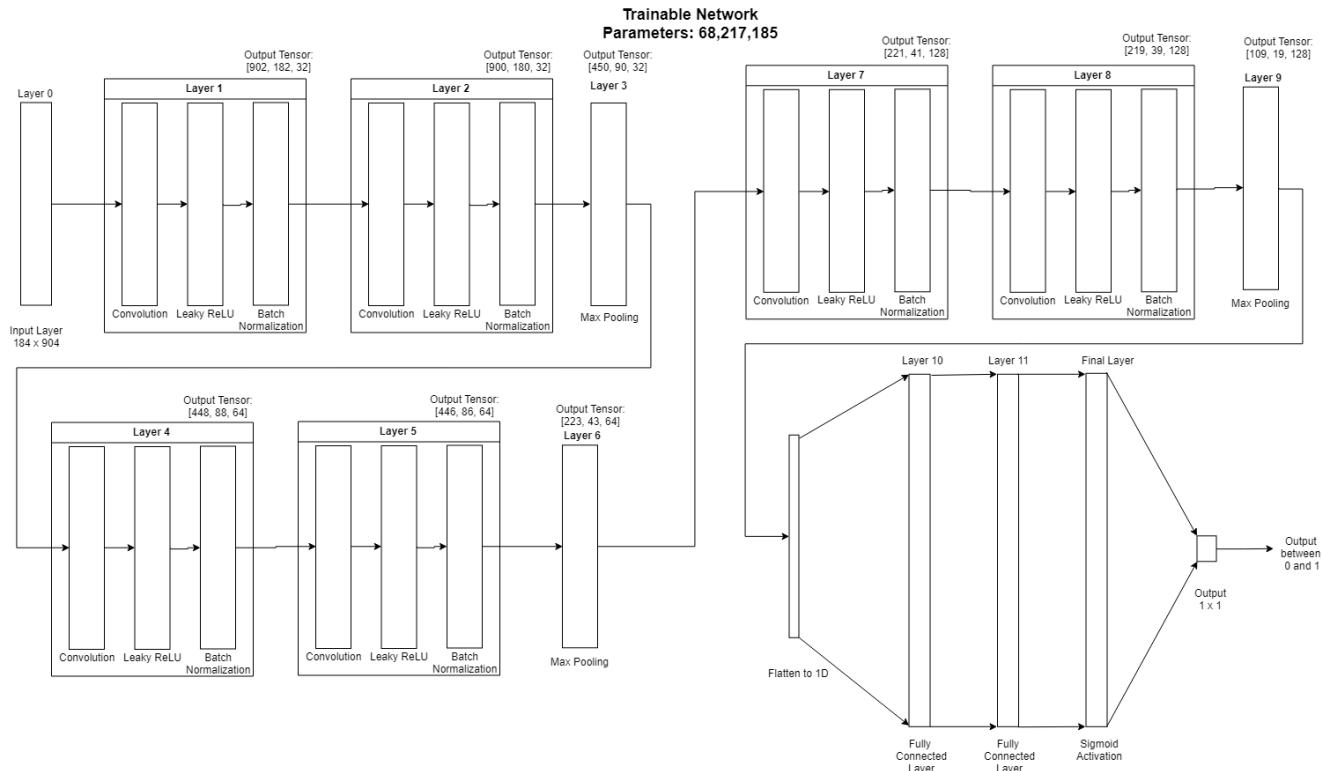


Figure 35: Technical visualisation of devised 2D CNN architecture

5.3.1 Technical hyperparameters

The technical details involving the exact implementation of the layers of the network based on the considerations detailed in section 5.2.1 can be seen in Tables 18, 19, and 20 below.

Table 18: Technical parameters of convolution layers of devised model

Layer #	# of Kernels	Kernel Stride	Kernel Size	Batch Normalisation?	Dropout?	Activation Function
1, 2	32	1	3x3	Y	N	Leaky ReLU ($\alpha = 0.1$)
4, 5	64	1	3x3	Y	N	Leaky ReLU ($\alpha = 0.1$)
7, 8	128	1	3x3	Y	N	Leaky ReLU ($\alpha = 0.1$)

Table 19: Technical parameters of densely connected neural layers of devised model

Layer #	# of Neurons	Batch Normalisation?	Dropout?	Activation Function	Regularisation?
10	256	Y	Y (P = 0.4)	Leaky ReLU ($\alpha = 0.1$)	Y (L1; $\lambda = 0.001$)
11	256	Y	Y (P = 0.4)	Leaky ReLU ($\alpha = 0.1$)	Y (L1; $\lambda = 0.001$)
12 (Output)	1	N	N	Sigmoid	N

Table 20: Technical parameters of max pooling layers of devised model

Layer #	Kernel Stride	Kernel Size	Dropout?	Batch Normalisation?
3	2	2x2	N	N
6	2	2x2	N	N
9	2	2x2	N	N

5.4 Results & Discussion

5.4.1 2D CNN performance

The CNN shown in section 5.3 above was trained on the training dataset for 1000 epochs, with the validation set of data passed through the network after each epoch to obtain measures of the network accuracy and loss. Due to early stopping on the validation loss, the model stopped training after approximately epoch 460 and a graph outlining the validation set predictive performance of the network is shown in Figure 36 below.

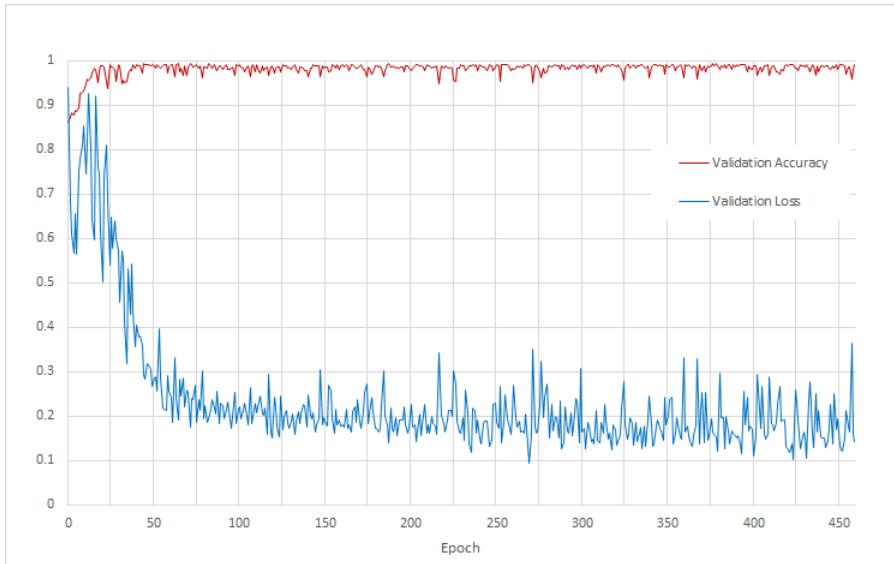


Figure 36: Validation accuracy and loss per epoch

Although it's intuitive to refer to the accuracy to determine how well the model performs, the metric that was used to evaluate the best model was the validation loss. This is because the loss is more informative than accuracy when analysing network performance. To understand why this is the case, consider two models M_1 and M_2 . The validation set consists of 4 images, with 2 images belonging to class 0 and 2 images belonging to class 1. If a model makes a prediction <0.5 it belongs to class 0, otherwise it belongs to class 1. M_1 predicts all of these classes correctly, but with predictions of [0.4], [0.3], [0.6], [0.65], and M_2 also predicts all classes correctly but with predictions of [0.0001], [0.002], [0.9993], [0.9924]. While both models have an accuracy of 100%, it is clear that M_2 represents a

model that is more confident in its predictions (thereby having a lower loss) and would be more robust and generalise better to new or noisy data.

From the results obtained in Figure 36 above, it can be seen that the lowest validation loss that was achieved was at epoch 285. Following this discovery, the testing set of data was applied at this epoch to obtain final performance metrics of the network as detailed in Table 21 below:

Table 21: Performance of network on validation and test datasets

Set	Loss	Accuracy (%)	Sensitivity (%)	Specificity (%)
Validation	0.09353	99.0333	99.1218	98.9566
Testing	0.08457	99.1812	99.1138	99.2182

The validation set metrics are included for convenience but only the testing set results should be looked at as the true performance of the network.

In order to obtain a greater insight into its efficacy, a confusion matrix of the network on the testing set is given in Table 22 below. This matrix outlines the true negative (TN), true positive (TP), false negative (FN), and false positive (FP) classification results of the testing set.

Table 22: Confusion matrix of network on testing set

		Predicted Class: Stable	Predicted Class: Unstable	
Actual Class: Stable	TN = 988		FP = 8	996
Actual Class: Unstable	FN = 9		TP = 949	958
		997	957	

A confusion matrix serves as a way to describe not only the errors being made by a model but also, and more importantly, the *types* of errors being made. This can assist in giving insight to what the model might be struggling with. The FN and FP fields of the table

above provide information on whether or not the model has biased classification accuracy towards the stable or unstable class. It can be seen that FN and FP are almost equal, and so it can be concluded that the model does not seem to contain any bias towards a particular class. This could be attributed to the care taken to ensure that the testing set was shuffled and contained an equal distribution of data for both classes and therefore the model had equal exposure to data from both classes (Wei & Drunbrack, 2013).

As explained in section 5.2.1.6, the binary cross-entropy loss function was chosen specifically due to the intuitivity it lends towards interpreting the loss of the network. Since only probability predictions of the true class contributed to the loss, Equation 11 could be applied to the testing loss in Table 21, and an average estimate for the predictive confidence of the model was obtained as shown below:

$$\bar{H} = -(\log_e(\Pr(\text{Correct Class})_{avg})) = 0.08457$$

$$\implies \Pr(\text{Correct Class})_{avg} = e^{-0.08457} = 0.9189$$

This means that on average, a predicted probability of the true class of 92% was made by the network. For example, if the true class was unstable, the model would, on average, give a 92% probability for the class being unstable. This demonstrates that the 2D CNN model has high confidence on the correct predictions that it makes.

The results obtained in Table 21 and 22 demonstrate a promising outcome for the potential of the developed 2D CNN model to be successfully applied to 12-lead single-beat ECG images to perform accurate automated classification. However, due to the nature of deep learning, the possibility that these accurate results are due to the data being overfit is one that cannot be ignored. A significant number of measures were taken to prevent this in both the design considerations of the network architecture, as well as the separation of data into three separate shuffled, distinct datasets. Due to this, the likelihood of overfitting is reasonably low, but it is not something that can be unequivocally ruled out, or even definitively tested for due to lack of access to data outside of the Staff III (Martínez et al., 2017) and PTB (Bousseljot et al., 1995) databases. Therefore, no certain generalisations should be made for the performance of this network on data outside of these two databases. If this network was to achieve similar performance results on data that was obtained from

new patients outside of these two databases, then the performance could be defined as being more conclusive.

5.4.2 Raspberry Pi 3B+ performance

Originally, the network was planned to be tested on the Raspberry Pi Zero W. However, due to technical issues regarding library configuration errors when combining Tensorflow (the backend library of Keras) with the Raspberry Pi Zero W architecture, this was not possible to achieve in time. In lieu of this, the network was instead tested on a Raspberry Pi 3B+, and ballpark estimates for the potential computational performance on the Raspberry Pi Zero W were approximated since the computational capabilities of the Raspberry Pi 3B+ are not significantly different. The 30 images that were set aside from the original dataset were run on the Raspberry Pi 3B+ and the time it took to make the predictions was measured and detailed in Table 23 below, output for this test can be seen in Appendix 23.

Table 23: Network prediction times for 30 ECG images on Raspberry Pi 3B+

Avg Prediction Time (s)	Total Prediction Time (s)
5.34 ± 1.02	160.2

This average prediction time proves that it is perfectly technologically feasible for the CNN model to perform automated analysis on the Raspberry Pi 3B+, and due to reasonably insignificant differences in computational capability, consequently hypothesises that this conclusion could also extend to the Raspberry Pi Zero W with future work regarding resolution of the previously mentioned technical issue.

A confusion matrix detailing the predictive performance of the model on the 30 image test set is shown below. However, it should be noted that 30 images is not a sufficient dataset size to draw any overall conclusions about network performance from and the main purpose of this testing set was for testing how quickly the Raspberry Pi 3B+ was able to perform successive predictions.

Table 24: Confusion matrix of network on 30 image testing set on Raspberry Pi 3B+

	Predicted Class: Stable	Predicted Class: Unstable	
Actual Class: Stable	TN = 14	FP = 1	15
Actual Class: Unstable	FN = 0	TP = 15	15
14		16	

The exact predictions made by the network compared to the true class of the images is shown in Figure 37 below.

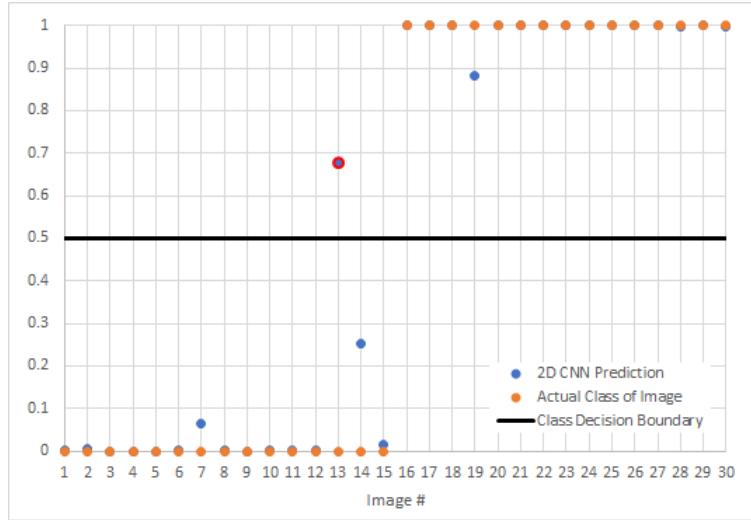


Figure 37: Prediction outputs of CNN model on 30 image dataset on Raspberry Pi 3B+

Although general conclusions cannot be drawn from such a small sample size, the output predictions of the network can still give some insight into potential problems. The single incorrect classification shown in Table 24 (false positive) is outlined in red in the figure above. This classification as well as the other reasonably uncertain classifications for images 7, 14 and 19 are all interestingly from the data in the Staff III database. This could imply that some of this data was not correctly classified in the training data, or contains noise that isn't present in the PTB database. Outside of these few images, the rest of the classifications were made correctly with very high certainty.

5.5 Future work

The most significant limitation that had the potential to be problematic for the performance of the network was a lack of availability of diverse data, since the data that was used to train the network was sourced from just two databases. With the deployment of the device proposed in this report, there would effectively be access to a boundless and ever-growing database of professionally examined and classified ECGs that can be used to periodically improve the model through further training and exposure to ECG data obtained under different conditions. This is due to the data communication infrastructure that is proposed by Martin below. This examination process would be performed by Australian medical professionals in which they would apply a diagnosis to the ECG data, thereby allowing it to be used for training.

With this additional data, there would be a reduced likelihood of the data being overfit, and the model would become more robust and accurate due to exposure to data obtained by a wider variety of patients. This essentially creates a self-improvement cycle for network performance. As mentioned previously, due to time constraints, network parameter tweaking wasn't a possibility, so it is very likely, and in fact almost guaranteed, that optimal network performance hasn't been achieved. There are many ways that this performance can be improved such as:

- Replacing the Adam optimiser with the Stochastic Gradient Descent optimiser with an optimally tuned learning rate
- Reducing complexity of the network to improve prediction time, and perhaps even allow near real-time prediction rates as well as reduce possibility of overfitting data
- Tweaking hyperparameters to maximise network accuracy
- Resolve library incompatibility issues to allow for the network to be run on the Raspberry Pi Zero W

6 Data Communication - Chuhao (Martin) Cai

6.1 Aims and Objectives

The CIOP requires to design a remote wearable diagnostic device, helping to ease the effects of coronary heart disease (CHD) in India. It is essential to ensure that the data can be successfully transferred and finally sent to doctors. Therefore, data communication will focus on designing a smooth and stable network for the wearable device. A concept of the Internet of Medical Things (IoMT) was proposed and will be implemented to the wearable device. Internet of things, a connected infrastructure of medical devices, which allows the wearable device to upload data to the Cloud centre and sent to the Australian doctors.

The concept of data communication had been updated as Figure 38 shown below,

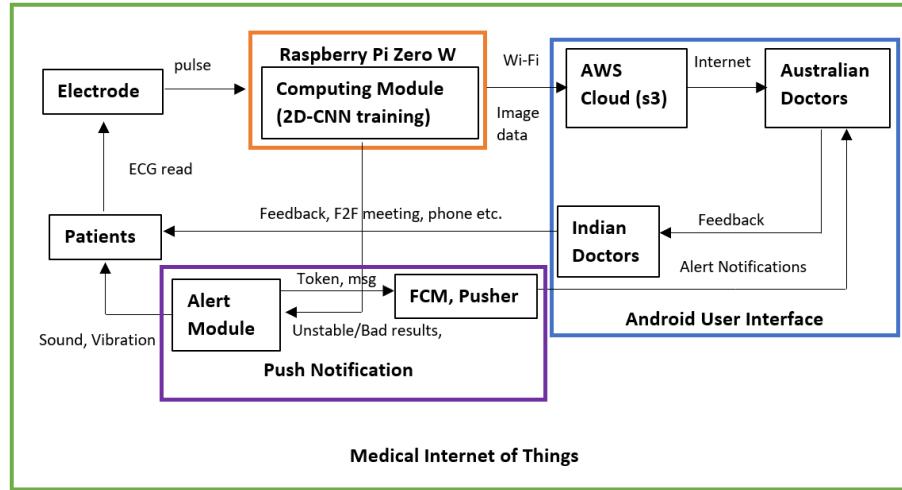


Figure 38: Network of IMoT

The electrode will collect raw data from the ECG stress test, converting into pulse and inputting to the medical device Raspberry Pi Zero W. The procedure also is considered as Internal Communication while external Communication built based on the Internet of Medical Things. The image data is generated by 2D-CNN training module as mentioned in the Data Analysis section, Raspberry Pi Zero W will upload the data as image files to Amazon Web Service (AWS) s3 bucket via Wi-Fi. By using Android-based User Interface, Australian doctors receive data directly from the S3 Bucket through the internet. After

reviewing the results and diagnose the cases, Australian doctors send feedback message and communicate to Indian doctor on the Android application. Finally, the Indian doctor will have a face to face meeting with the patient. When the device generates bad results, an alert module will be activated to warn the user. On the other hand, Raspberry Pi Zero W will send an emergency message to Firebase Cloud Messaging (FCM), generating push notification to an android application so that both Indian doctors and Australian doctors will be noticed in the first place.

Raspberry Pi Zero W was chosen as the ideal wearable device and IoMT project since it is low cost, light to carry and robustness quality. Raspberry Pi Zero W has the connectivity with Bluetooth 4.1, wireless LAN, it can run as a micro-computer unit independently when the network is working that is the user who wearing the device can upload the data easily to Australian doctor without plugging into laptop or network adapter.

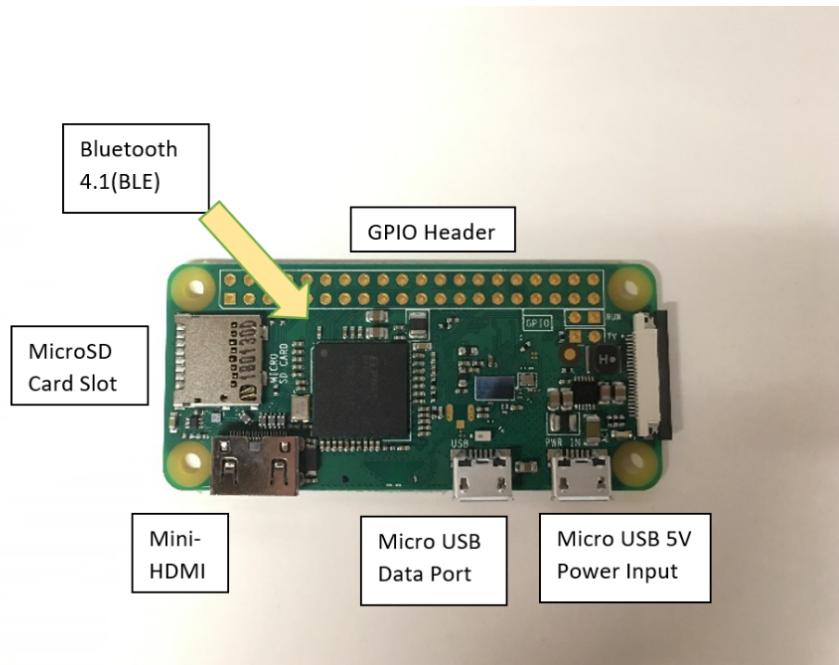


Figure 39: Raspberry Pi Zero W

Debian Buster Version 4.19 was recommended to be implemented in Raspberry Pi Zero W as an image. The differences between Debian Stretch and Debian Buster is that the Debian Buster security changes. The Debian Buster is harder to hack and its better choice

to protect users' privacy. The device will be tested in various fields. Connection test can determine the stability of the device, uploading test indicates the technical feasibility of each uploading method and Push notification test shows the robustness of the device.

To minimise the unit cost, using \$9 Raspberry Pi Zero W and \$9 8GB SanDisk micro SD Card (JB Hi-Fi,2019) spend less than \$ 20 in total. Budget for Amazon S3 Bucket is \$0.02 per GB storage which is economical service.(Cloud Storage Pricing — S3 Pricing by Region, 2019) Using Cloud centre to store the ECG data meets the zero-maintenance requirement. Moreover, Amazon Web Service is considered as multi-functional, long durability, high-security web service to save ECG data. The Android-based user interface allows that the Australian doctors can communicate with Indian doctors through live chat which is the simplicity that aids in use. Lastly, the language for user interface offers two different language version which is English (as default) and Hindi.

6.2 Raspberry Pi Zero W Setup

6.2.1 Structure of the Approach

This chapter is revised version of Interim report. (Cai.C.H, 2019) The approach of testing the Raspberry Pi Zero W device combines with two types of connection test, which were the Wi-Fi connection test and Bluetooth Connection test. Controlling the device remotely by using two different Application to test and verify the connectivity.The outline has been shown in the figure below,

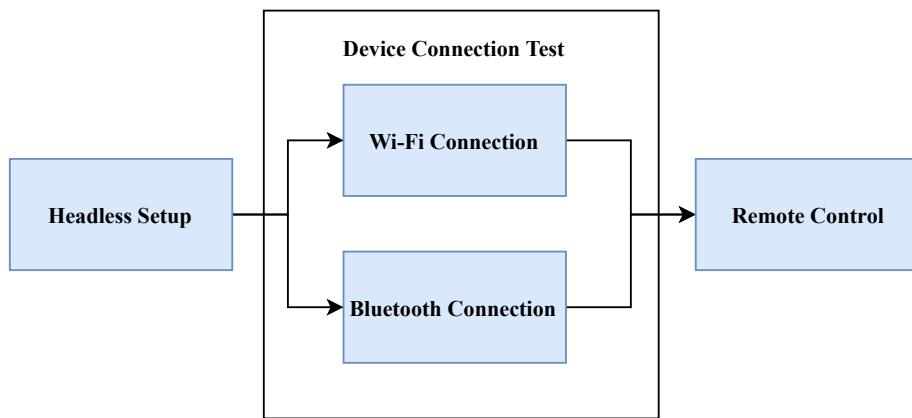


Figure 40: Structure - Raspberry Pi Zero W Setup

To test the connectivity of the device, firstly, a headless setup is necessary. Writing Debian Buster Version 4.19 image in an SD card and setting up Ethernet Gadget in a personal computer to connect the device via SSH. After connecting to the Raspberry Pi, the wireless connection test can be conducted. The following sections will be explained the methodologies to test two different types of wireless connection, and finally using VNC to verify the connectivity of the device.

The following contributions were made in this chapter:

- The Raspberry Pi Zero W has been set up with an 8GB SD card and connected via SSH.
- Wi-Fi connection test has been conducted by setting up network configuration. The Ping test result showed connectivity by observing the reaction time of the test. The lower reaction time and connectivity are positively correlated.
- Bluetooth connection test has been conducted by pairing with the Bluetooth address. However, the unknown error in “Connection failed – No usable services on this device” occurred when trying to connect to the laptop.

6.2.2 Methodology

6.2.2.1 Headless Raspberry Pi Zero W Setup

Raspberry Pi Zero W as a smart single-board computer, is required to set up by installing operating system images. The Raspbian Buster with desktop and recommended software based on Debian Buster Version 4.19 was chosen to implement to the Raspberry Pi Zero W. Writing a Raspberry Pi operating system image on SD card. The minimum recommended card size by Raspberry Pi Official is 8 GB, and the physical size of the SD card requires MicroSD cards. (“SD cards - Raspberry Pi Documentation”, 2018)

Before writing the operating system into the MicroSD card, it needs to be formatted by Formatter software. SD Card Formatter software was used to format the MicroSD cards quickly. After card formatting, the Debian Buster Version 4.19 can be written into MicroSD cards by Win32DiskImager software.

Next step is to set up with Ethernet Gadget. The Raspberry Pi Zero W can get access through the laptop via the Ethernet Gadget. To set up Ethernet Gadget, adding command “dtoverlay=dwc2” as the last line in the document config.txt that is in the

MicroSD card post-burn. Saving the config.txt file and adding the command “modules-load=dwc2,g_ether” between the commands “rootwait” and “quiet”. (“Linux commands - Raspberry Pi Documentation”, 2019) Laptop sometimes cannot identify the Raspberry Pi Zero W as Ethernet/RNDIS Gadget. It is suggested that installing Bonjour, Apple’s implementation of zero-configuration networking can offer service discovery, and hence, the Ethernet Gadget can be detected. (“Turning your Raspberry PI Zero into a USB Gadget”, 2019)

The final step was to enable the SSH, a cryptographic network protocol for operating network service securely (Ylonen & Lonwick, 2006), it can be enabled by creating a new in MicroSD card named “ssh” (“SSH (Secure Shell) - Raspberry Pi Documentation”, 2019). After that, it is able to log into the Raspberry Pi Zero W on PuTTY, a free and open-source terminal emulator network file transfer software.

The device cannot be immediately connected. A green LED on the Raspberry Pi will flash for about ten seconds. When the green LED was stable and did not flash for a few seconds, which mean SSH can access it. The green LED will be shown below,

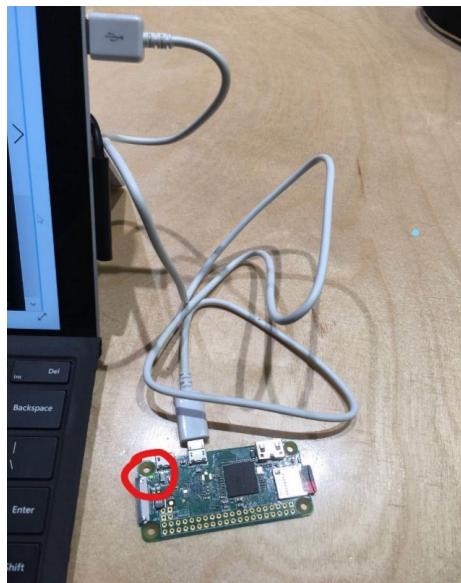


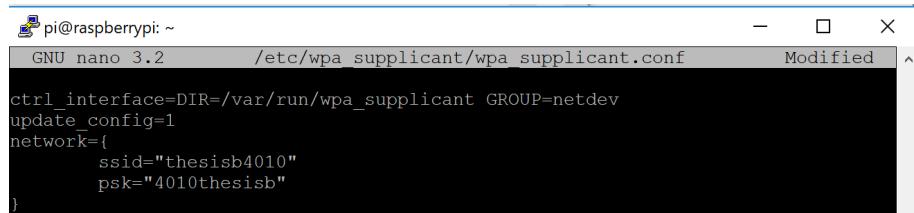
Figure 41: Green LED on Raspberry Pi Zero W

Logging in the account “pi” with default password “raspberry” and check whether the Raspberry Pi Zero W has the new network device called usb0. Usb0 can be verified by using

command “sudo ifconfig” on the PuTTY terminal. (“Setting WiFi up via the command line - Raspberry Pi Documentation”, 2019) If usb0 is shown on the terminal, which means the computer or laptop is successfully connected to the Raspberry Pi Zero W. It is ready for the wireless connection test.

6.2.2.2 Wi-Fi Connection Test

Issuing the command “sudo iwlist wlan0 scan” will list the Wi-Fi nearby. The network can be added to Raspberry Pi Zero W through “WPA-supplicant”. Editing the configuration file in nano by sending command “sudo nano /etc/wpa_supplicant/wpa_supplicant.conf”. In the nano editor, adding the script in the bottom,



```
pi@raspberrypi: ~
GNU nano 3.2          /etc/wpa_supplicant/wpa_supplicant.conf      Modified
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
    ssid="thesisib4010"
    psk="4010thesisb"
}
```

Figure 42: Set up Network on GNU nano

In this case, a personal hotspot was set up by smartphone to test the connectivity of the network. The Raspbian also supports to set up multiple network configuration for wireless connectivity. Simply adding the same format as the command above. Sending command “sudo ifconfig” on PuTTY terminal, so that acquired the IP address.



```
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 172.20.10.3  netmask 255.255.255.240  broadcast 172.20.10.15
        ether b8:27:eb:03:b0:0e  txqueuelen 1000  (Ethernet)
          RX packets 39220  bytes 20354293 (19.4 MiB)
          RX errors 0  dropped 0  overruns 0  frame 0
          TX packets 19001  bytes 4963294 (4.7 MiB)
          TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
pi@raspberrypi:~ $
```

Figure 43: IP address for Raspberry Pi Zero W

Ping test was implemented to the IP address to test the connection via Ping application. Another way to test the connectivity was to test on the terminal by ”sudo ping (server name)”.

Ping, such a networking utility was written by Mike Muuss in 1983, can be used to test

the reachability of an IP address network. (Flack, 1999) The method was inspired by the principle of echolocation, using timed IP/ICMP ECHO_REQUEST and ECHO_REPLY packets to measure the “distance” to the target machine that is testing network connection status.

6.2.2.3 Bluetooth Connection Test

Although the project team decided to use Bluetooth as a data communication method no longer, the test had been conducted and will be demonstrated in this section. Bluetooth pairing on Raspberry Pi Zero W required to use command “bluetoothctl” to open the Bluetooth switch and show the address of nearby devices. Then turn on Bluetooth agent with following instruction “agent on”, “default-agent”. Finally, “scan on” to list the available devices nearby. Pairing with a specific address can finish the Bluetooth connection. However, data cannot be sent via this method.

```
pi@raspberrypi:~ $ bluetoothctl
Agent registered
[bluetooth]# agent on
Agent is already registered
[bluetooth]# default-agent
Default agent request successful
[bluetooth]#
```

Figure 44: Command for Opening Bluetooth in Raspberry Pi

```
[thesisb4010]# pair 70:81:EB:C2:9C:8E
Attempting to pair with 70:81:EB:C2:9C:8E
Request confirmation
[agent] Confirm passkey 979467 (yes/no) : yes
```

Figure 45: Bluetooth Pairing

6.2.2.4 Remote Control

Using Remote Desktop Connection can be a more straightforward way to test the wireless connectivity of the Raspberry Pi Zero W. By implementing this method, configuration and preferences can be changed on the laptop efficiently.

Firstly, adding the network in the wpa_supplicant configuration same as the process in Wi-Fi connection test. When the network “thesisb4010” set up, run “ifconfig” command to check the raspberry pi IP address. Then, running “sudo apt-get install tightvncserver” and

“sudo apt-get install xrdp” command so that the laptop can get access to the device server. Open Remote Desktop Connection and connect with the IP address gathered previously. A warning will show up when connecting to the device. For testing the remote control, this warning can be ignored.

Then the server will require to have a username and password. Logging on with the pi as username and raspberry as password. Waiting a few seconds, Raspberry Pi Zero W will load the full desktop interface.(“Getting Started with Raspberry Pi Zero W”, 2019)

Another simple way to remote Raspberry Pi Zero W was using the RealVNC, a software providing remote access which consists of a server and client application for the Virtual Network Computing protocol to control another computer’s screen, can be more stable and more features than Remote Desktop Connection.

This method required to enable VNC Server on Raspberry Pi, the first step was to install the RealVNC server at the command line, issuing the command on the terminal.

Then enabling VNC Server at the Raspberry Pi Zero W configuration using “raspi-config”. The configuration has shown as the following the figure, Navigate to Interfacing Option and enable VNC.

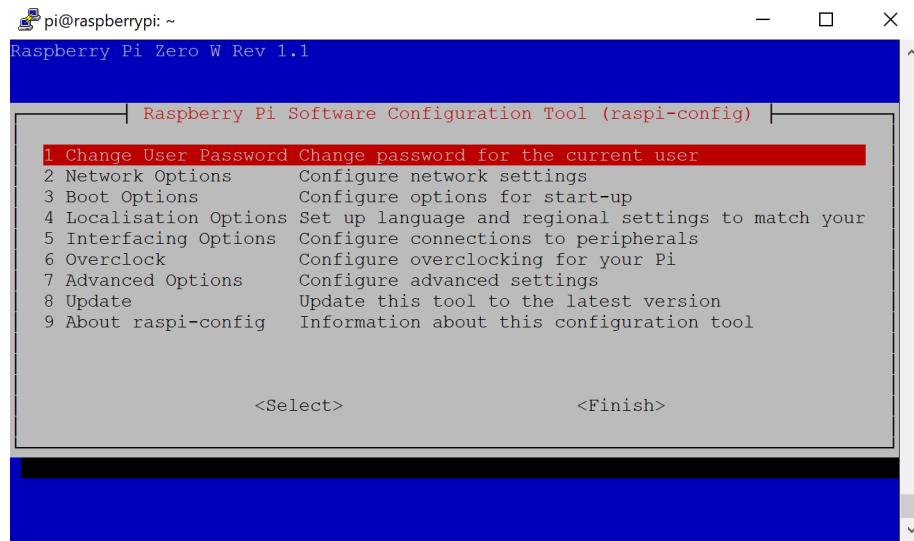


Figure 46: Booting Raspberry pi

Applying IP address to RealVNC Viewer and logging with a username can get access into Raspberry Pi Zero W.(“VNC (Virtual Network Computing) - Raspberry Pi Documentation”,

2019) However, the viewer might unable to show the desktop of the Raspberry Pi since the resolution was not matching. It can be solved by editing the resolution setting in the raspberry pi configuration – advanced options – resolution.

6.2.3 Results and discussion

6.2.3.1 Headless Raspberry Pi Zero W Setup

The results of headless Raspberry Pi Zero W set up has shown in the terminal below,

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jul  9 09:18:48 2019

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@raspberrypi:~ $
```

Figure 47: Result for Raspberry Pi Zero W setup

Terminal showed the Raspberry Pi Zero W connected to the laptop via SSH successfully and the “pi@raspberrypi” in green means the local address of Raspberry Pi. The first two lines represented the information of the username and secret password. The information of the operating system also has shown on the terminal which the system was running in Linux raspberry pi version 4.19. In order to keep the account secure, system reminded the user to change the default password.

6.2.3.2 Wi-Fi connection test

Two results for Wi-Fi connection test is showing as following, The results illustrated that the Raspberry Pi Zero W successfully connected to Wi-Fi hotspot provided by the smartphone. The result shows on figure 16 denote the ping test from the smartphone to the Raspberry Pi Zero W. Another result shows on figure 17 represent the ping test from the Raspberry Pi to the Google server. Both results proved that Raspberry Pi Zero W enable to connect Wi-Fi and to use the hotspot network. Moreover, another finding was that by comparing the reaction times of the ping test, reaction speed for the ping test between smartphone and raspberry pi was faster than the pint test between Google server and raspberry pi. The difference may be caused by the ”distance” between two IP address.

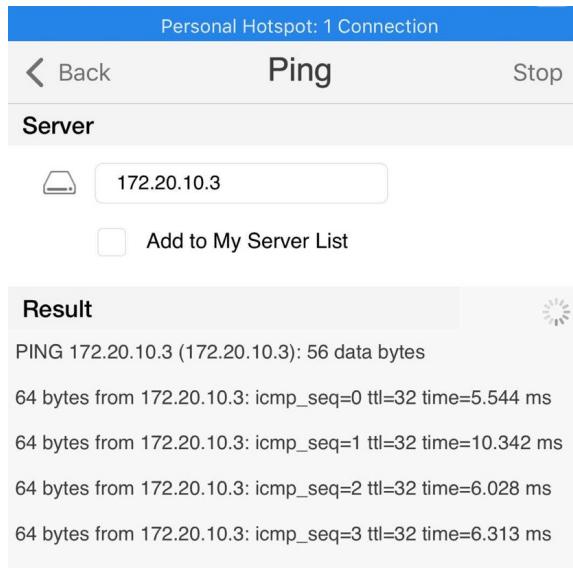


Figure 48: ping Test from smartphone

```

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Jul 12 00:28:57 2019
pi@raspberrypi: ~ ping www.google.com
PING www.google.com (216.58.196.132) 56(84) bytes of data.
64 bytes from syd15s04-in-f4.1e100.net (216.58.196.132): icmp_seq=1 ttl=54 time=
25.8 ms
64 bytes from syd15s04-in-f4.1e100.net (216.58.196.132): icmp_seq=2 ttl=54 time=
38.4 ms
64 bytes from syd15s04-in-f4.1e100.net (216.58.196.132): icmp_seq=3 ttl=54 time=
30.6 ms
64 bytes from syd15s04-in-f4.1e100.net (216.58.196.132): icmp_seq=4 ttl=54 time=32.4 ms
64 bytes from syd15s04-in-f4.1e100.net (216.58.196.132): icmp_seq=5 ttl=54 time=51.0 ms
64 bytes from syd15s04-in-f4.1e100.net (216.58.196.132): icmp_seq=6 ttl=54 time=49.2 ms
64 bytes from syd15s04-in-f4.1e100.net (216.58.196.132): icmp_seq=7 ttl=54 time=30.4 ms

```

Figure 49: ping Test from Raspberry Pi Zero W

The finding proved Raspberry Pi Zero W enable to use as the device to upload data on Cloud server via Wi-Fi connectivity.

6.2.3.3 Bluetooth connection test

The results for Bluetooth connection test will be demonstrated below, Results presented the laptop has been pairing with raspberry pi successfully, it can be concluded that Raspberry Pi Zero W enable to connect another device via Bluetooth Low Energy.

However, when trying to connect Raspberry Pi to a laptop, an error with "Connection failed – No usable services on this device" occurred. Therefore, for further Bluetooth test, the

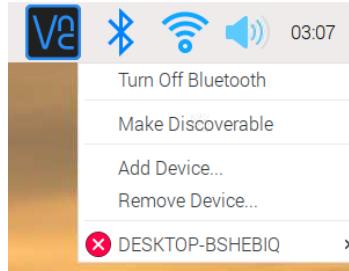


Figure 50: Bluetooth connection showing with laptop ID

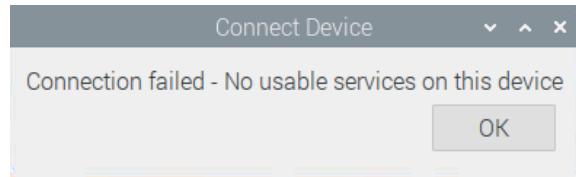


Figure 51: Connection Failed

Bluetooth service should be implemented into the Raspberry Pi Zero W. Since the project group decided to no longer use Bluetooth, the further Bluetooth test will be stopped.

6.2.3.4 Remote control

By comparing two remote control method, Remote Control Connection was extremely unstable when connecting via the SSH, although the software is free. RealVNC is much more stable than Remote Control Connection. In VNC viewer, the desktop can refresh with frequency 60 Hz that is more sensitive than the original software. Besides, RealVNC provides its cloud centre for users to upload files data. However, the drawback of software offers with a free trial for 30 days, which means the users have to pay for it. Therefore, for economic reason, using RealVNC is only for the current test.

Table 25: Difference between Remote Control Connection and RealVNC

	Advantage	Disadvantage
Remote Control Connection	Free	Delayed, Unstable, less function
RealVNC	Robust, Multi-functional, Sensitive, Cloud Centre	The software needs to be paid

6.3 Data Upload

6.3.1 Structure of the Approach

The figure below presents the combination of existing approaches to data uploading that is sending and saving to the Cloud Centre and comparison with three different upload method.

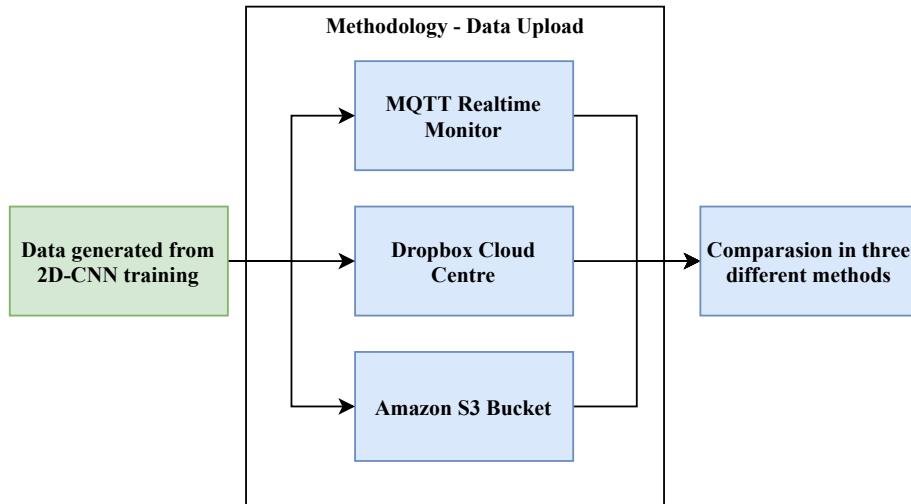


Figure 52: Structure - Data Upload

As mentioned on the Internet of Medical Things, Data will be generated from the 2D – CNN training and waiting to be transmitted to Cloud. In this chapter, three different methods were implemented to data upload which is using MQTT Protocol for Realtime monitoring, directly uploading data to Dropbox Cloud with a secret key and saving data into Amazon S3 Bucket by using AWS SDK for Python. Furthermore, the comparison in

three different methods for data upload is discussed in this chapter.

The following contributions were made in this chapter:

- Sending data to MQTT Broker using MQTT Protocol and receiving the data from the Broker.
- Uploading image data to Dropbox Cloud service
- Uploading image data to Amazon S3 Bucket
- Comparison of three different data upload methods

6.3.2 Methodology

6.3.2.1 MQTT protocol

MQTT stands for Message Queuing Telemetry Transport, is a device-to-device connectivity protocol ("MQTT", 2019), an extremely lightweight publish and subscribe messaging transport. It is designed for low network bandwidth connections with a remote location. The working principle has shown in the figure below.

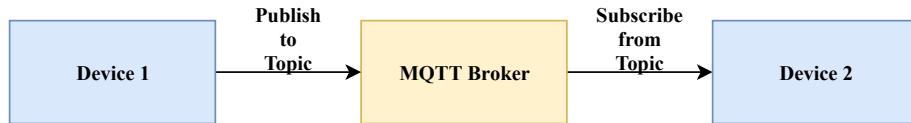


Figure 53: Communication via MQTT protocol

By using the MQTT protocol, the device can publish a message to a specific topic. When a publisher has a message to distribute, it will send the message to the connected MQTT broker such as Amazon Web Services. If there is another device subscribes the topic, the message will be sent to the device directly. However, the topic will be discarded if the topic has no current subscribers. MQTT is such an ideal protocol for the device has the bandwidth limitation because of its small size, low power usage. Therefore, it is feasible to implement the connectivity protocol to Raspberry Pi Zero W.

When the data has analysed and generated, the data message can be uploaded to the specific topic. After that, doctors can easily read and review the data from their device such as smartphone, personal computer in real-time. The methodology of implementing MQTT protocol to Raspberry Pi will be presented in the following sections.

Publishing data to MQTT Broker, one of the methods has been chosen was using AWS IOT to create an AWS IoT MQTT client. AWS IoT service is such a platform which allow users to connect with devices to the internet to exchange and process data securely. Furthermore, AWS also provided the service that such as Elastic search which allows users to visualise the data on the graph. Therefore, Using AWS IOT service is an ideal choice to monitor the ECG data.

After creating an AWS thing on the IoT core service, it is required to apply for a certificate and a policy. The figure below is showing the certificate and private secret key which allowing Raspberry Pi to get access to the AWS MQTT broker.

	VeriSign-Class 3-Public-Primary-Certification-Authority-G5.pem	7/08/2019 12:30 PM	Text Document	2 KB
	8556b624d8-private.pem.key	7/08/2019 12:28 PM	KEY File	2 KB
	8556b624d8-certificate.pem.crt	7/08/2019 12:28 PM	Security Certificate	2 KB

Figure 54: Certificate and private secret key

AWS IoT Security Model is using public-key cryptography to ensure the message to be securely transferred. (Corbett, 2017) The cryptography has a public key which allows the message to be encrypted and a private key allows the message to be decrypted. The user who with the public key can check the signature and read the original message. The user with the private key can be the only one who can read the encrypted message. To send the message securely, a device certificate which is embedded with the public key also is required. Finally, an X.509 certificate is needed to prove the identity of the users and can be used to exchanges encrypted message with AWS IoT securely. It will be given to a Certificate Authority (CA). The certificates and private key are showing in Fig, as mentioned before. The certificates and private key are saved in the Raspberry Pi and be used to connect with the AWS MQTT client. The figure below shows the way to utilise the certificates and private key.

```
from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTClient #Import from AWS-IoT Library
import time#To create delay
from datetime import date, datetime #To get date and time

myMQTTClient = AWSIoTMQTTClient("")#Initialise the Client
myMQTTClient.configureEndpoint("a6miz8bcyngjx-ats.iot.ap-southeast-2.amazonaws.com", 8883)
myMQTTClient.configureCredentials("/home/pi/Desktop/Amazon_Root_CA_1.pem", "/home/pi/Desktop/a642fc3af2-private.pem.key", "/home/pi/Desktop/a642fc3af2-certificate.pem.crt")
```

Figure 55: Code - connecting to AWS MQTT client

Import AWSIoTMQTTClient library into python code, it is required to install the software development kit(SDK) of AWS IoT in Raspberry Pi.

```

pi@raspberrypi:~ $ git clone https://github.com/aws/aws-iot-device-sdk-python.git
Cloning into 'aws-iot-device-sdk-python'...
remote: Counting objects: 208, done.
remote: Total 208 (delta 0), reused 0 (delta 0), pack-reused 208
Receiving objects: 100% (208/208), 153.38 KiB | 87.00 KiB/s, done.
Resolving deltas: 100% (80/80), done.

```

Figure 56: AWSIoTMQTTClient SDK installation

Raspberry Pi now can get access and send message to AWS IoT cloud client. The ECG data can be sent by the python code, which runs on the Raspberry Pi, and the figure below shows the while loop, which used to send ECG data to AWS client. The method can be tested by uploading sample ECG data instead of real-time data.

```

while 1: #Infinite Loop
    now = datetime.utcnow() #get date and time
    current_time = now.strftime('%Y-%m-%dT%H:%M:%SZ') #get current time in string format
    time.sleep(2)
    payload = '{ "timestamp": "' + current_time + '", "Value": ' + ECGreadings + ' }'
    print(payload) #print payload for reference
    myMQTTClient.publish("test", payload, 0) #publish the payload

```

Figure 57: Code - uploading ECG data

MQTT.fx can be used to test AWS IoT things. The figure below shows the configuration of connection profiles.

When the device is trying to send a data message to the MQTT broker, the data is encrypted over Transport Layer Security (TLS), which is designed to provide communication security over a network. (Dierks & Rescorla, 2008) For MQTT, AWS IoT security mechanisms uses TLS client authentication to identify a different device.

6.3.2.2 Dropbox Cloud

The image data generated by 2-D CNN training can be uploaded on Dropbox Cloud centre. The method installs Dropbox Python SDK for API v2 and integrates Dropbox libraries into Raspberry Pi Zero W.

On the Dropbox side, an app is required to create on the DBX platform. DBX platform can integrate the function such as file storage, sharing, previews, and search. There is App key and App secret for users to send file or message to Dropbox Cloud. In the Application, multiple users (up to 500) can be added in and participating to send image files on Dropbox

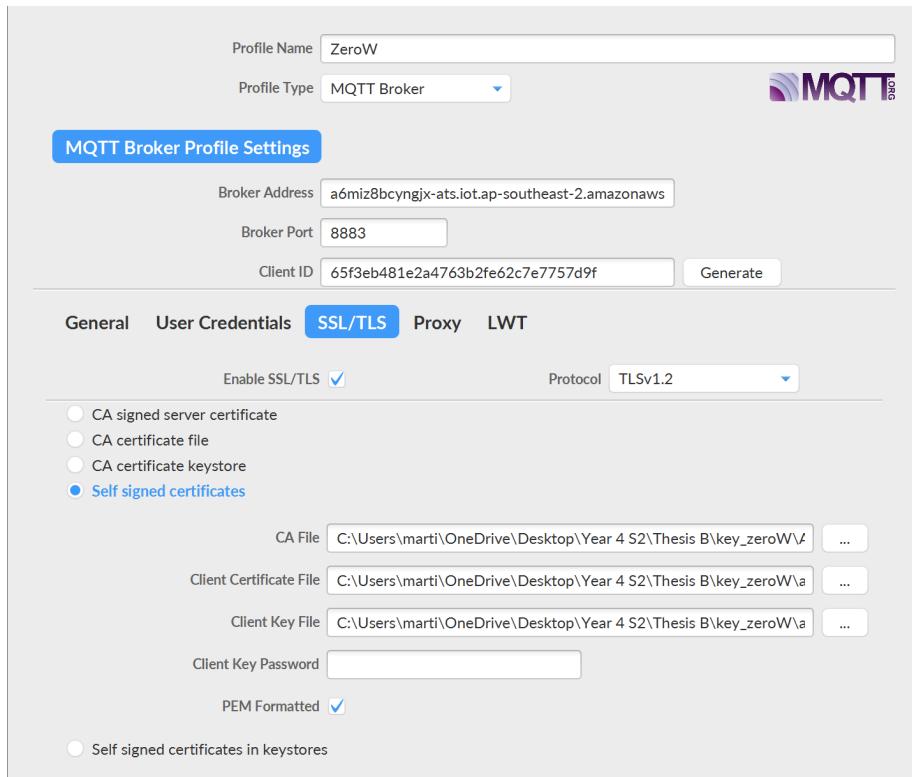


Figure 58: MQTT.fx configuration

at the same time.

For the App authentication, Dropbox only requires the user to offer App's app key and secret. When offering the secret and key for authentication, the HTTP username and password are replaced by the app key and secret. Another simple way to make calls to Application is to generate an access token for the Dropbox account through the App Console. This method can easily allow other users to get access to the Dropbox service through standard OAuth flow.

The process of OAuth flow for authorising the user with Dropbox is that an authorisation code will be sent to the App which has been approved. After that, the App will exchange the authorisation code for an access token which allows accessing the Dropbox via API.

The figure presents the python code running on Raspberry Pi to upload the image data to Dropbox Cloud. One of the examples was given from the Data analysis section. The key with 'Q25V2DMNnQ0AAAAAAAAAAZRGDELzTOze-1m4BL5u9fkpBl8iWoEzc75NL_b9FAq6Z' is

```

with open("test.png", "rb") as f:
    dbx = dropbox.Dropbox('Q25V2DMNnQ0AAAAAAAve5FJYbD2ou99uC30Kfsv0pCknP6MW3jd3btPNfWARZB')
    dbx.files_upload(f.read(), '/patient01/test.png')

```

Figure 59: Code - sending imagefile to Dropbox

the access token which can be used to access the Dropbox Cloud client via API. Furthermore, file folders will be created for a different patient. For example, file 'patient01' will only save the ECG data for the patient whose number of identities is '01'.

6.3.2.3 Amazon S3 Bucket

Amazon S3 is an object storage service which can use it to store and protect data for a range of use cases. Amazon S3 has high durability of an object because Amazon can provide the service such as secure access permission, cross-region replication, regularly tested back-up to prevent the data from malicious or accidental deletion.

Uploading data or file to Amazon S3 require a bucket, where the data can be stored securely and durably. The name of the Bucket must be unique since the name is shared across all existing bucket names in Amazon S3. To back-up or restore all data types, AWS offer the service such as Amazon S3 Glacier and S3 Glacier Deep Archive, which are designed to deliver high durability and have capabilities to meet compliance requirements. Therefore, using AWS can efficiently prevent patients from data loss.



Figure 60: Flowchart - sending image file to S3 Bucket

AWS requires the access key and secret key to get access to S3 Bucket. It is different from Dropbox access token since the AWS offers the identity service named AWS Identity and Access Management (IAM). IAM is a web service to create users and manage the respective access, and the web service can control who is authenticated and authorised to use resources.

In the IAM service, users can be created, AWS will generate access key ID and Secret access key. However, the key has no permission to read or upload files to S3 bucket since AWS requires to attach the permission policies to users. There are three ways to attach

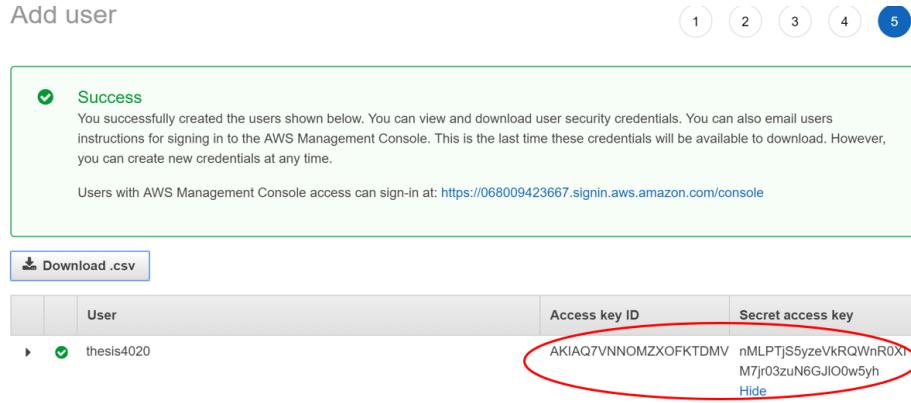


Figure 61: Add user to IAM

access policy, adding a user to a group with the access policy, copying permissions from an existing user, or attaching existing policies directly.

The screenshot shows the 'Attach Policy' section. It includes a filter for 'Policy Type' set to 's3' and a table with four results. The first row, 'AmazonS3FullAccess', has a checked checkbox and is highlighted with a red box. The other three rows are unselected.

Policy Name	Attached Entities	Creation Time	Edited Time
<input checked="" type="checkbox"/> AmazonS3FullAccess	1	2015-02-07 05:40 UTC+1000	2015-02-07 05:40 UTC+1000
<input type="checkbox"/> AmazonDMSRedshiftS3Role	0	2016-04-21 03:05 UTC+1000	2019-07-09 04:19 UTC+1000
<input type="checkbox"/> AmazonS3ReadOnlyAccess	0	2015-02-07 05:40 UTC+1000	2015-02-07 05:40 UTC+1000
<input type="checkbox"/> QuickSightAccessForS3Storage	0	2017-06-13 04:18 UTC+1000	2017-07-21 10:02 UTC+1000

Figure 62: Attach Policy to S3 Bucket

The simplest way is to create a group with `AmazonS3FullAccess` which provides full access to all buckets via the AWS Management Console. Once the policy is added into the group, users can join in the group and shared the access permission with other users in the same group. Therefore, every patient has a unique access key can get access and upload data to AWS S3 bucket. After installing Boto3 SDK into Raspberry Pi and then the image data can be uploaded directly by using the function "Upload_file" of the boto3 library. The figure below has shown the example code for sending the test image file to S3 bucket.

```

import boto3
from botocore.exceptions import NoCredentialsError

ACCESS_KEY = 'AKIA07VNNOMZXOFKTDMV'
SECRET_KEY = 'nMLPTjS5yzeVkrQWnR0XIM7jr03zuN6GJl00w5yh'

def upload_to_aws(local_file, bucket, s3_file):
    s3 = boto3.client('s3', aws_access_key_id=ACCESS_KEY,
                      aws_secret_access_key=SECRET_KEY)

    try:
        s3.upload_file(local_file, bucket, s3_file)
        print("Upload Successful")
        return True
    except FileNotFoundError:
        print("The file was not found")
        return False
    except NoCredentialsError:
        print("Credentials not available")
        return False

uploaded = upload_to_aws('test.png', 'ecgpatient01', 'testimage01.png')

```

Figure 63: Code - Seding image file to AWS S3 Bucket

6.3.3 Results and discussion

6.3.3.1 MQTT protocol

The data upload results by using the MQTT protocol shown in the figure below,

```

>>> %Run test.py
MQTT Client connection success!
[{"timestamp": "2019-08-19T13:00:28Z", "Value": 0.03},
 {"timestamp": "2019-08-19T13:00:28Z", "Value": 0.02625},
 {"timestamp": "2019-08-19T13:00:28Z", "Value": 0.028125},
 {"timestamp": "2019-08-19T13:00:28Z", "Value": 0.03625},
 {"timestamp": "2019-08-19T13:00:28Z", "Value": 0.03375},
 ...]

```

Figure 64: Result sent by Python

It can be obtained that the python code has connected to AWS MQTT Client successfully. The ECG value collected from the electrodes was uploaded to the client in real-time. Furthermore, the timestamp is recorded when the data is uploading.

To verify whether the client receives the ECG data, open the MQTT.fx application and subscribe to the topic 'test' to check if there is any message is sent. The figure shows that the message has been sent to the client success in the same format as the message sent by Raspberry Pi.

Overall, using the MQTT protocol can send message data to other devices efficiently in real-time. Therefore, a doctor using a laptop or smartphone can quickly review the result

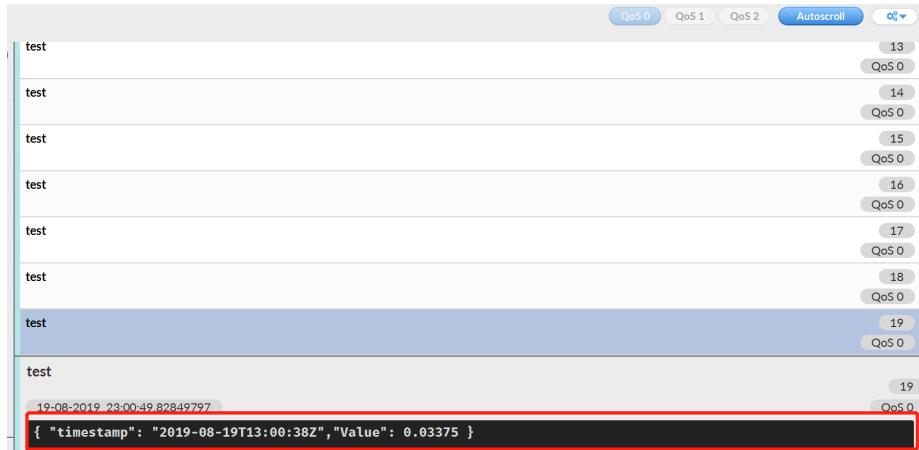


Figure 65: Result showing on MQTT.fx

and diagnose. When the patient is doing the ECG stress test, the doctors who subscribe to the topic can receive the real-time data from the AWS MQTT broker. The data can be visualised by using the AWS Elastic search. However, a doctor might diagnose multiple patients at the same time, which mean the doctor require to subscribe to multiple topics. Besides, MQTT message will not be saved if there is no subscriber at the moment; the doctor who disconnects to the topic due to the network issue will not receive the subsequent data.

6.3.3.2 Dropbox Cloud

By running the python code 'uploadtest' in Raspberry Pi. It can be obtained that the stress test image file has been uploaded to Dropbox successfully. The image file generated by the 2D CNN training module was uploaded and saved to the file folder 'patient01'. Login in to Dropbox Cloud to verify if the Dropbox receives the image file. The figure shows that the image file 'test.png' has been sent to the 'Dropbox-*i*patient01' successfully. Different folders will be created for every patient and named a 'patient name' + 'patient number'.

To diagnose coronary heart disease, the doctor can log in the Dropbox and quickly review the stress test results in the specific folders. Besides, the data will be saved in the folders so that the history of the results can also be checked and analysed.

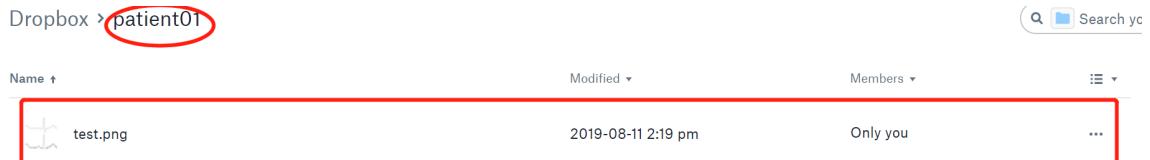


Figure 66: Result - test01.png showing in the Dropbox file 'patient01'

6.3.3.3 Amazon S3 Bucket

By running the python code 'uploadtestAWS.py' in Raspberry Pi. It can be obtained that the stress test image file has been uploaded to Amazon S3 Bucket successfully. The image file generated by the 2D CNN training module was uploaded and saved to the Bucket 'ecgpatient01'.

Accessing AWS Management Console S3 service can verify if the image was uploaded in the Bucket. The figure shows that the image file 'testimage01.png' has been sent to the bucket 'ecgpatient01' successfully. Unique Bucket will be created for every patient and named as 'ecgpatient' + 'patient number'.

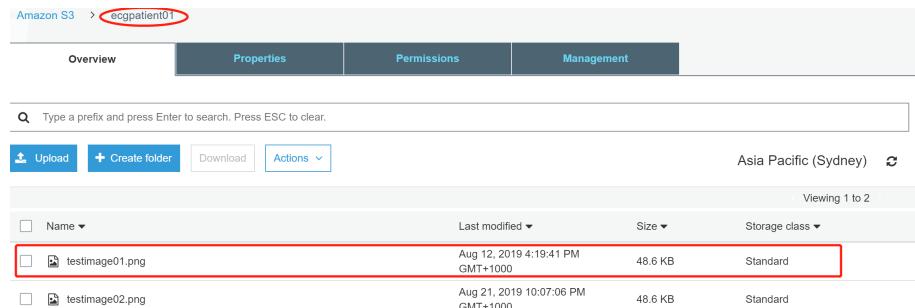


Figure 67: ecgpatient01.png showing in the AWS S3 Bucket

Doctors can get access to the AWS Management Console and collect patients' test image from the S3 Bucket. The method is like uploading an image file to Dropbox. However, to prevent the data from malicious or accidental deletion, AWS offers the service such as cross-region replication, versioning, and regularly tested back-up. Hence, it is considered that test image file saving in S3 bucket is durable and stable.

6.3.3.4 Comparison

The table below shows the comparison of three different methods of data uploading,

Table 26: Comparison of three different methods of data uploading

Method	Advantage	Disadvantage
MQTT	<ul style="list-style-type: none"> - Real-time monitor - Lightweight - Low Cost 	<ul style="list-style-type: none"> - Unstable - No storage - Message data
Dropbox	<ul style="list-style-type: none"> - Simple to use - Image data - Storage 	<ul style="list-style-type: none"> - Safety issue - No Backup plan
Amazon S3	<ul style="list-style-type: none"> - Image data - Safe - Storage - Multi-Funtion - High durability 	<ul style="list-style-type: none"> - Complex - Need to pay after 12 months trial

By comparing three methods of data uploading, although using AWS IoT MQTT broker for sending real-time message to doctors is lightweight and low cost, the connection is unstable, and the message will not be saved during the ECG stress test. Besides, the message data need to be visualised so that the doctors can diagnose the disease. For using the Dropbox platform, the image data can be easily uploaded by sharing the access token to the patients' device. The image files can be saved in the Cloud and doctors are allowed to review the history data of the patients. However, it is considered that using the access token is unsafe, and there is no back-up plan for saving patients' data. When it comes to Amazon S3, it can not only send and save the image files into the Bucket but also has abundant and robustness service such as Identity and Access Management(IAM) to keep the data in safe, S3 Glacier and S3 Glacier Deep Archive for long-term archive and digital preservation, S3 Cross-Region Replication to back-up the patients data. The doctor is required to learn how to manage files and users in AWS Management Console, hence a crash course of AWS for doctors is needed.

In conclusion, although Amazon S3 require to pay after the 12 months free trial, Amazon S3 with multi-function, high durability and security, it is considered as the best choice for uploading data at this stage.

6.4 User Interface

6.4.1 Structure of the Approach

The figure below presents the combination of existing approaches to data uploading that is sending and saving to the Cloud Centre and comparison with three different upload method.

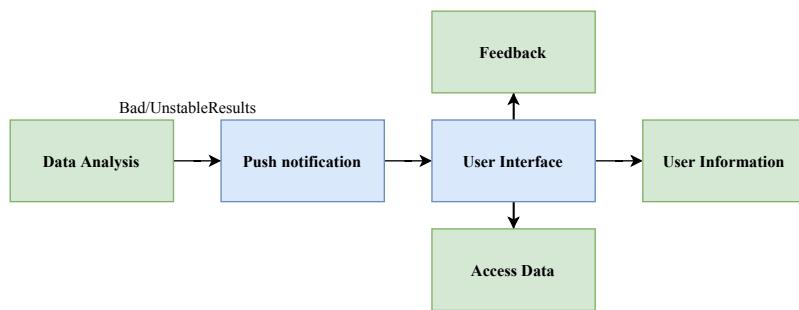


Figure 68: Structure - User Interface

As mentioned on the Internet of Medical Things, the alert module is activated as the bad results detected, and Raspberry Pi Zero W will send urgent medical attention to doctors. In this chapter, a method using Firebase Cloud Messaging (FCM) and PUSHER will be implemented, and a web-based python code was used to monitor the emergency status. The push notification is sent to the App client finally, a user interface is required to create and receive the notification. Besides, the user interface should meet all the users' requirements such as feedback, access data and information management. The design of a user interface based on Android will be explained in the User Interface Design section.

The following contributions were made in this chapter:

- Creating real-time database in the Firebase Clouding Messaging
- Monitoring the emergency status by using web-based python code. If the emergency flag is 1 in real-time database, it notify push notification server
- Sending emergency message to FCM with secret key.
- Sending push notification via PUSHER. PUSHER receives App token from FCM, FCM is provided device token from the app client

- Creating an android application to receive the push notification.

6.4.2 Methodology

6.4.2.1 Push Notification

When the Raspberry Pi detects the unstable or bad results, the alert system will be activated. On the one hand, the device will alert the patients with vibration and sound. On the other hand, the device will send a push notification to the doctors. Doctors will receive the emergency notification, reviewing the ECG results diagnose in the first place. Push notification, an event-based mechanism, when the event happens, it is sent from the remote servers push events to smartphone client apps. ("Push Notification Mechanisms for Pervasive Smartphone Applications - IEEE Journals & Magazine", 2014) The following section will explain the push notification mechanism and how can it be implemented into the Raspberry Pi Zero W.

The following figure shows the flow chart of sending a push notification,

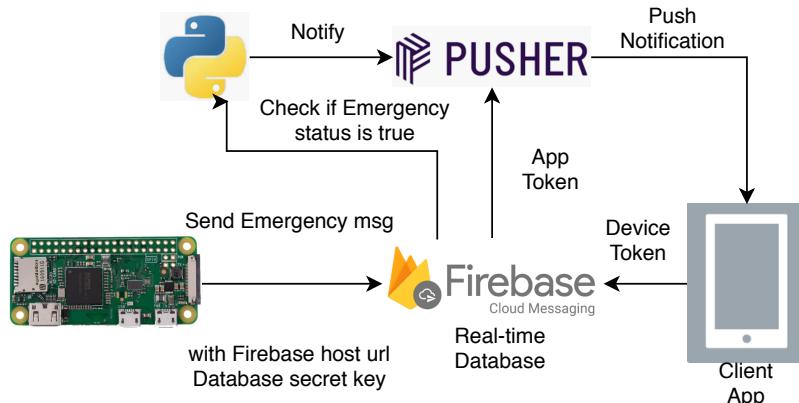


Figure 69: Flowchart - Push notification

Firstly, the client app will send the device token to Firebase Cloud Messaging (FCM). Push token is a unique key for an app-device combination, which ensures the notification is delivered to a specific device. In FCM, an app token is generated and sent to a third-party push service Pusher. As the unstable results are detected, Raspberry Pi activates the alert module and send an emergency message to the real-time database FCM with the Firebase host URL and database secret key. The emergency message, a status flag which indicates patient in an urgent situation. Web-Based python code is to monitor the emergency status

in FCM. It notifies Pusher to send a push notification when the emergency flag is true in the database. At this moment, a push notification pops up in the doctors' smartphone via the medical client application. The medical client application is designed as a user interface to create a communication platform between Indian doctors and Australian doctors. The user interface will be explained in the last sections of the chapter.

First step is to send device token to the Firebase Clouding Messaging, a client app can be created and designed by the Android Studio, the IDE for Android operating system and built on JetBrains' IntelliJ IDEA software. Sending emergency message is required to connect firebase real-time database with database secret and database link. The method is using python code in Raspberry Pi and firebase SDK is installed by the command ‘pip3 install python-firebase’. The error occurred when import the library ‘firebase’, the reason was because the ‘from .async import process_pool’ was incompatible with python 3.7. The way to solve the problem is rename the ‘async.py’ file in the firebase SDK package. The command is sent by using the firebase function ‘post()’, and sample code has been shown below.

```
firebase = firebase.FirebaseApplication('https://spartan-proxy-246223.firebaseio.com/')
firebase.post('/flag', {'Emergency':1})
```

Figure 70: Code - sending emergency message to real-time database

In this case, for the convenience of testing, the rule for accessing to FCM Realtime Database is changed to public.

```
{
  /* Visit https://firebase.google.com/docs/database/security to learn more about se
  "rules": {
    ".read": true,
    ".write": true
  }
}
```

Figure 71: Code - Allowing to read and write in database

When the emergency status is updated, it is time to notify PUSHER server. However, sending a notification to the app client, the PUSHER server requires an app token from the FCM. FCM can get the device token from the App. In this case, device token can

be accessed by creating a new class which extends ‘FirebaseInstancedService’ in Android Studio. Calling ‘getToken’ within onTokenRefresh can get updated instance ID token. After that, app token will be available in the FCM and sent to PUSHER. To use push notification server, it is required that adding the ‘google-service’, ‘firebase-messaging’ and ‘push-notificaiton-android’ dependency in the App project Grandle in Android Studio. Finally, import PUSHER SDK will allow the App to register and subscribe to the Push Notification. After setting up PUSHER, web-based python code can send alert notification to PUSHER server as the emergency status is 1.

6.4.2.2 User Interface Design

Designing an associated system which allows remote medical treatment by Indian doctors and Australian doctors require to clarify user requirements. Hence, users' needs are the first thing to be declared. According to the users' needs and develop such a system or product which is satisfied with the requirements. Functional requirements, which implement to enable users to accomplish the user's needs, can be used to develop a final product. (“Functional and Nonfunctional Requirements: Specification and Types”, 2018) The figure below shows the relationship between the users' need and functional requirements.

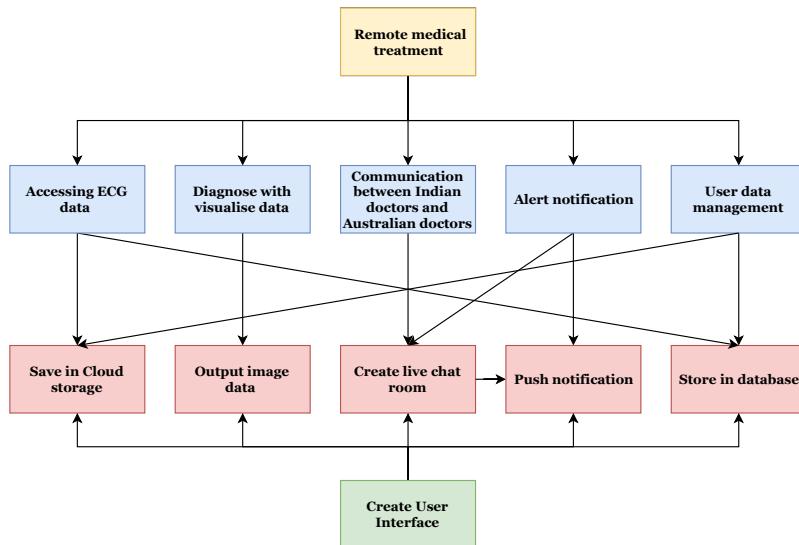


Figure 72: User Interface developed by Users' needs

Designing a user interface is satisfied with the requirement of remote medical treatment. In

the User Interface, doctors can collect ECG data from the Cloud or database. Moreover, Cloud and database offer a platform which allows users to modify the ECG data and user information. When a terrible or unstable result is detected from the patient, doctors should be notified emergency message as soon as possible. Push notification service can let the doctors know in the first place when the patients need urgent medical attention. The function of showing image data in the user interface can help the doctors to diagnose the disease. According to the patients' condition, Australian doctors communicate with Indian doctors, it can be solved by using the live chat function in the user interface. Besides, the functional requirement of push notification can also be implemented into the live chat between Indian doctors and Australian doctors. In conclusion, the concept has shown that the functional requirements have been satisfied with each of the users' needs. Therefore, the design of a multi-functional user interface perfectly implements Remote medical treatment.

6.4.3 Results and discussion

6.4.3.1 Push Notification

Then the emergency flag can be sent to FCM directly to the database URL.



Figure 73: Emergency flag in FCM database

The emergency flag has changed to 1 which mean the alert module has been activated, the web-based application will notify PUSHER server and publishing the emergency notification. However, the result didn't display the information of the patient so that the doctors can not clarify which patient needs urgent care. The header which indicates the patient identity must be added in the database. Moreover, emergency status can be represented by the emergency level, for example, level 1 to 3. The higher emergency level indicates the results were more deviation from the normal value. The improvement of emergency status should be discussed with the developer of data analysis in the future. As the figure shows below, the push notification pops up on the notification centre.

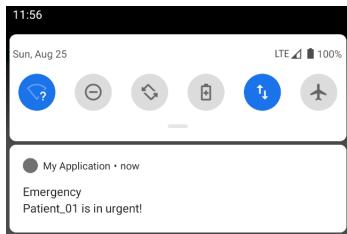


Figure 74: Push notification pop up

The advantage of sending push notification on smartphone is that doctors can notice the emergency message at any time. To improve the function of push notification, it is suggested that ECG stress test result should be sent with push notification at the same time.

6.4.3.2 User Interface Design

The figure below has shown the simple user interface on the Android smartphone.

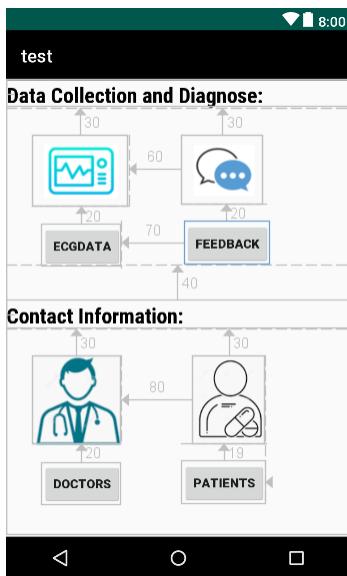


Figure 75: Conceptual Design - User Interface

Four main sections are included in the user interface when the stress test is finished Australian doctors can collect and review the image data through the button 'ECGDATA'. The button is linked to AWS S3 Bucket, the Android application download the image file

from the S3 Bucket and show it in another layout. ‘FEEDBACK’ button offers a live chat platform which allows the Australian doctors to communicate with the Indian doctors. Furthermore, the user interface also offers contact information of doctors and patients and the information data will be securely saved in the database. The user interface is currently in beta, it only meets the basic needs of the users. It is suggested that developer should publish three types of versions which are for patients, Indian doctors and Australian doctors. Authentication is also necessary that each user need an access key and secret. Therefore, a login system is considered to build in the future. Moreover, to adapt to all users’ devices, it is decided to bring the user interface beyond Android to PC and iOS platform.

6.5 Future Work

In the future, the improvement in coronary heart disease in Indian will be measured by analysing the ECG stress data in Cloud and database. To make full use of the resources of the Cloud service, it is suggested to implement other robustness cloud service in Amazon such as elasticsearch, CloudWatch and Kinesis Firehose stream etc. Secondly, developing extra language version such as Bengali based on the region-specific feasibility analysis. Besides, the user interface on IOS and PC should also be developed. A smartwatch might be a bright choice for communication. Finally, A local database server such as SQL needs to be created to manage users’ data and users’ documentation.

From the existing work, the results of uploading data and sending push notification can only be executed once the Raspberry Pi Zero W is controlled remotely. Therefore, the device is required to be tested independently with power supply, and the portable power bank mentioned in the power section will be chosen as a power supply for the device.

In conclusion, there is still much work to be developed to demonstrate that the network system developed in this portfolio should adopt in practical remote treatment. Last but not least, Raspberry Pi Zero W has shown the great potential on the Internet of Medical Things for remote medical treatment.

7 Power - Aidan Cornwall

7.1 Aim and Objectives

This section of the report explores the technical aspects of the power source of the Ischemic Heart Tracker (IHT) that monitors the heart of a stabilised patient that suffered from coronary artery disease. Calculations of the total power consumption of the components have been performed and a suitable power source has been justified. A life cycle analysis has been performed on the chosen power source and the Raspberry Pi which is used as an onboard computer.

7.2 Methodology

Research into the power consumption of the components was performed while also researching possible methods used to minimise power consumption. The research performed in the literature review identified an indefinitely sustainable power source suitable for the current project and proved to be promising, however, manufacturing processes are still experimental and not practical for the application. The usage process and the operation of the device has been finalised and has made the previously researched power source to become unfeasible. So, the feasibility of rechargeable batteries powering the device has been established and has been chosen as the power source of the IHT. The power capacity of one reading has been calculated and a suitable product has been selected. A “cradle to the grave” life cycle analysis has been performed to identify any environmental or other associated impacts the device has.

7.3 Power Consumption

To understand the power consumed by the device, the wearable needs to be broken down into its components, these components are fundamental to the operation of the device and are all necessary for the usage process. The Raspberry Pi Zero W is used in this report as it was the chosen computer as it was small and was theoretically capable of operating the program that would determine if the condition of the patient was stable or unstable. Unfortunately, some difficulties emerged and the program was experiencing some library related issues. The use of the raspberry Pi was to show that a low power and low performing computer could operate the program, so in practice, a Raspberry Pi 3B+ was able to operate smoothly. In Table 27, a list of components to be used and their respective wattages have been provided.

Table 27: Wattage of Components

Component	Voltage	Amperes	Power
Raspberry Pi Zero W <small>(Raspberry Pi, 2019)</small>	5 V	1.2 A	6 W
10 Dry Electrodes <small>(Plux, 2019)</small>	Maximum 3.5 V	0.17 mA	0.595mW x 10 Electrodes = 5.95 mW
Vibration Motor (Coin Vibrator) <small>(Baolong Electronic Group, 2018)</small>	3 V	Maximum 85 mA	0.225 W
Speaker <small>(Jaycar Electronics, 2019)</small>	1.414 V	0.176 A	0.25 W
	Total		6.5 W

A total of 6.5 Watts is needed to power the device, however, the figure obtained is a theoretical value, so to accommodate for any losses, an appropriate total would be close to 7 Watts. A complication when calculating the power of the components is that the electrodes, vibration motor and the speaker all use alternating current, except the Raspberry Pi which uses direct current. The electrical losses may appear to be quite high, although when converting electricity from alternating current to direct current, or vice versa, the losses will be heavily dependent on the quality of inverters or rectifiers which will have high costs, so to account for this, and to keep costs low, a higher power rating will be assumed.

The most important components of the device are the Raspberry Pi Zero W, and the electrodes, unfortunately, these components require different types of currents, alternating and direct. Since the electrodes, vibration motor and speaker use alternating current, these components must be analysed separately from the Raspberry Pi. An experimental circuit diagram of the system is shown in Figure 76.

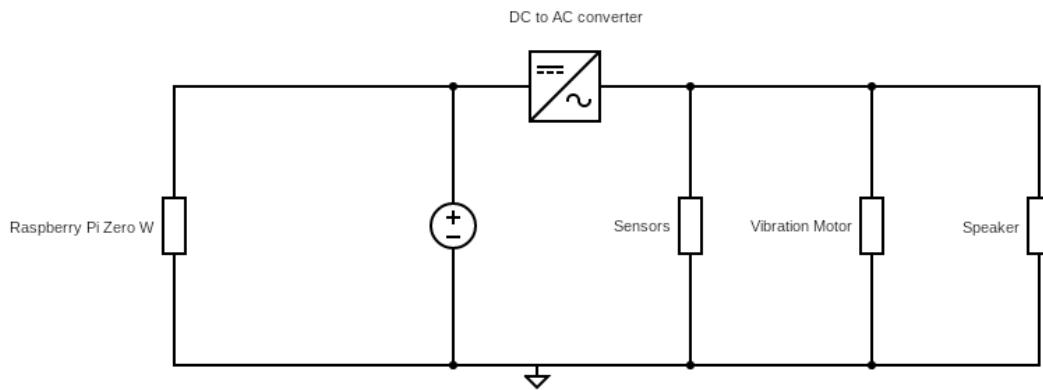


Figure 76: Circuit Diagram

Each component is placed in parallel so that a consistent 5 Volts is delivered across the branches, each branch will have other components such as resistors and capacitors so that the major components receive the suitable voltage and current as listed in Table 27. (Bosnjak, A., et al. 2017).

7.4 Feasibility of Previous Electricity Generation Methods

In the literature review (Cornwall, A., 2019), the triboelectric nanogenerator (TENG) was shown to be more appropriate to the project than the thermal energy harvester, this was due to the simple integration into clothing, higher power output, and the thermal energy harvester's heavy dependence on the difference between air and skin temperature (Leonov, V., et al. 2007). These methods of electricity generation would have complied with Outcome 3 of the CIOP, as the system would be indefinitely sustainable in terms of its energy usage as it would have generated electricity as the user moved during a stress test (Seung, W. et al 2015). As the project evolved, the idea of having the wearer perform a stress test had been agreed among the team to increase the risk associated with wearing the device as the wearer could suffer a heart attack. The project now involves the user sitting as the device performs a test. Therefore, the TENG would be extremely impractical to use as there will be no movement from the user for the TENG to harvest and convert into electricity. Another reason the TENG will not be considered as the power source for this project is that the TENG is in its early stages of development and requires complex and costly manufacturing methods. Additionally, the device will need approximately 7 Watts of electricity, and current research into this energy harvesting method have not produced this amount of electricity and this is the primary reason for not continuing research into this power generation method.

7.5 Rechargeable Battery Power Feasibility

The feasibility of the previous methods of generating electricity have been shown to be inadequate, so a more practical approach has been taken. While researching wearable, commercially available products, they all use a form of rechargeable battery that is sealed inside the device. Rechargeable batteries have been chosen to power the IHT since they are small, can produce a high, consistent power output for long periods of time and can be recharged without replacement for an extremely long time. Recharging these batteries will be as simple as plugging them into a wall, just like a mobile phone, therefore meeting the CIOP Constraint 2 of simple or zero-maintenance required, in terms of the power supply. Determining the amount of power required for the system is very important and introduces many variables. The capacity of the battery will affect the physical dimensions, weight, and cost, all of which are crucial in the usability and economic feasibility of the wearable device.

For the entire device to function, the Raspberry Pi needs to function, it requires a 5 Volt DC power supply, however, it is able to operate on an very stable 4.8V supply (Raspberry Pi, 2019), this is the lower threshold and the voltage should not be lower otherwise the Raspberry Pi will shut down.

To calculate the estimated capacity of the battery, first, the length of usage must be calculated. The Raspberry Pi is estimated to run for 30 minutes for each reading, this includes taking the reading via the electrodes and computational time. Calculations to find the minimum capacity for a single test is as follows:

$$30 \text{ minutes} = 0.5 \text{ hours}$$

$$0.5 \text{ hours} \times 7 \text{ Watts} = 3.5 \text{ Watt hours}$$

$$3.5 \text{ Watt hours} @ 5 \text{ Volts} = 700 \text{ milliampere hours}$$

There are multiple methods to create a battery pack that meets these specifications, the most important specification that needs to be met is that the Raspberry Pi is powered, it requires a minimum of 4.8 Volts. This can be achieved by simply connecting 4 rechargeable batteries in series, as standard, AA rechargeable batteries are generally accepted to have a 1.2 Volt rating (Battery Stuff, 2019). Standard AA rechargeable batteries would have a large enough capacity to power the device, as a single battery has 2300mAh (Energizer Recharge, 2019), which is capable of powering 3 complete tests. Another method is to buy previously combined battery packs that give a rating of 4.8 Volts at 3000mAh (Fluke, 2019). Additionally, they may also be powered by portable mobile phone chargers or power banks as they output 5 Volts continuously but can be very expensive.

7.6 How Batteries Work

A battery is essentially a container full of chemicals that can produce electrons. Chemical reactions that produce electrons are called electrochemical reactions, the battery takes advantage of these types of reactions and transfers the electrons from the negative terminal to the more positive terminal, generally, via a circuit (Brain, M. et al., 2000). The fundamental components of a battery are the two electrodes which are connected by an ionically conductive material called an electrolyte. The two electrodes have different chemical compositions and therefore different chemical potentials. When these electrodes are connected

via an external device, or circuit, electrons instantly flow from the more negative to the more positive potential. The electrolyte is responsible for the transportation of ions which maintains a charge balance, the electrical energy can be utilised by an external circuit. In rechargeable batteries, a larger voltage applied in the opposite direction can cause the battery to recharge (Armand, M. et al., 2008).

The amount of electrical energy that a battery can deliver is a function of the cell's voltage and capacity, which are dependent on the chemistry of the system. Another important parameter is power, which is significantly dependant on the chemical composition of the battery, but also partially dependant on the battery's engineering. The chemical reaction inside the battery influences the speed of the electron transfer as the speed of the reaction is related to the number of electrons flowing out of the terminal, this is known as current. However, the power of a battery is not linear, all batteries have a maximum current they can produce, for example, a 500 mAh battery cannot produce 30, 000 mA for one second because there is no way for the battery's chemical reactions to happen that fast. At higher currents, batteries can produce large amounts of heat which is extremely inefficient and a waste of energy. These factors need to be considered at very high currents, but over a normal range of use, the milliamp-hour ratings are generally regarded as linear (Brain, M. et al., 2000).

As a battery discharges, the internal electrochemical process results in the transfer of ions from one electrode to the other, through the electrolyte. When the battery is charged the, the process is reversed and the ions travel in the opposite direction. During charging and discharging, each electrode goes through a chemical reaction which generates these ions at one electrode and consumes the ions at the opposite electrode. How well this process is performed has a significant impact on the overall performance of the battery (Cope, R.C. et al., 1999). The electrolyte solution has an initial concentration of ions to support the chemical reaction and provides a medium for the transfer of ions, the rate and uniformity the ions move from one electrode to the other has a significant impact on the performance of the battery. The chemical reaction rate at the electrode which consumes ions is limited by the concentration of the ions at the surface. This concentration is linked to the ability of the ions able to move through the electrolyte. If the ion concentration across the surface of an electrode is uneven, the chemical reaction rate will not be uniform, leading to the formation

of dendrites on the electrode. Dendrites are a growth of material from the electrode, if these formations are not addressed, they can result in a dead battery. Consequently, ensuring a high uniform ion concentration at the electrode consuming ions will maximise performance.

Another factor of battery performance is related to the metallic structure of the electrodes, under extended low current conditions, the slow chemical reaction can cause relatively large metallic crystals to form. These larger metallic crystals reduce the surface area of the electrode, causing a potential drop in the overall battery capacity and an increase in internal resistance. This increase in internal resistance decreases the voltage output of the battery (Cope, R.C. et al., 1999).

The electrodes of the battery are a defining feature and heavily influences the properties and voltage outputs of these batteries. Typical modern day batteries include a variety of chemistries:

- Alkaline battery – Commonly used in Duracell and Energizer batteries, the electrodes are zinc and manganese oxide, with an alkaline electrolyte
- Lead acid battery – Used in automobiles, the electrodes are made of lead and lead-oxide with a strong acidic electrolyte, this battery is rechargeable
- Lithium-ion battery – very good power to weight ratio, this is often found in laptops and cell phones, this battery is rechargeable (Brain, M., et al. 2000)

7.7 Power Supply Options

While searching for suitable products that fit the needs of the IHT, criteria had to be satisfied so that the product would power the other components of the device. The most important criterion is that it would have to power the Raspberry Pi, this means that the power supply must be direct current and output approximately 5 Volts. Rechargeable batteries were chosen to be the source of power and lithium-ion batteries were chosen to be the primary battery of research due to the light weight, high, consistent power output, accessibility and availability on the market. Expensive, high quality products will not be considered to minimise unit costs as the price jump does not justify a slightly better performing battery, products will also need to be available in bulk as multiple units will be

manufactured. Other serious points of consideration were the simplicity of use, whether detaching the power pack was simple or required tools, and language constraints, however, the manufacturing company are expected to follow international standards and the local doctor assigning the device to the patient is expected to instruct the patient on how to use it.

A configuration of the standard AA batteries has been determined to be an unpractical power source since standard rechargeable batteries have an accepted value of 1.2 Volts, when 4 of these batteries are connected in series, they have a total of 4.8 Volts. This is the lower threshold of the Raspberry Pi's operating voltage, any lower and the Raspberry Pi will shut down. The discharge curve of these batteries show the actual voltage compared to the capacity of the battery, the general curve of these AA batteries will drop below the stated voltage toward the end of their capacity, this is not ideal as the voltage will drop below 4.8 Volts causing the Raspberry Pi to have insufficient power and to shut down earlier than expected (Yamaura, J. et al., 1993)(Energizer Data, 2019). Yamaura,J.'s paper is from 1993, present day rechargeable batteries have become more modernised, however, similar data has been shown in Energizer's data sheet, and is more recent. The options considered below were chosen based on these criteria and the information available on these specific products.

Table 28: Battery Pack Options

Option	Battery Pack	Price	Power Capacity	Source
1	5 V Li-Ion Battery Pack Rechargeable	\$2.90	2400 mAh	Alibaba.com
2	Sparkfun Lithium Ion Battery Pack	\$8.95	2500 mAh	Sparkfun Electronics
3	USB Battery Pack for Raspberry Pi	\$47.40	4000 mAh	Core Electronics

7.7.1 Option 1



Figure 77: Rechargeable Battery Pack

Option 1 is a pack of standard sized lithium ion batteries that have been connected in series and have been packaged so that they output 5 Volts, they are manufactured in Guangdong, China and are mass produced, resulting in an extremely cheap price. Since they are manufactured in such huge numbers, they have a questionable build quality, the charge and discharge curves on their website are blurry so their claims are suspicious. The site claims to have 1000 cycles, which is large, but not uncommon in lithium ion batteries. The power capacity of the battery compared to cycle times does not even have the correct rating and the charge and discharge curves show the incorrect voltage output, so it is assumed that the curve is standard among the manufacturer's products. The discharge curve of the battery, in Figure 2 shows that the voltage output falls quickly and stabilises at various voltages according to the temperature, this is not ideal as the target region, urban India, can experience extremely warm temperatures, and having a fluctuating battery voltage will be extremely impactful on the usability of the IHT (Alibaba Shenzhen Eastar

Battery Co, 2019).

DISCHARGE CURVE AT DIFFERENT TEMPERATURE

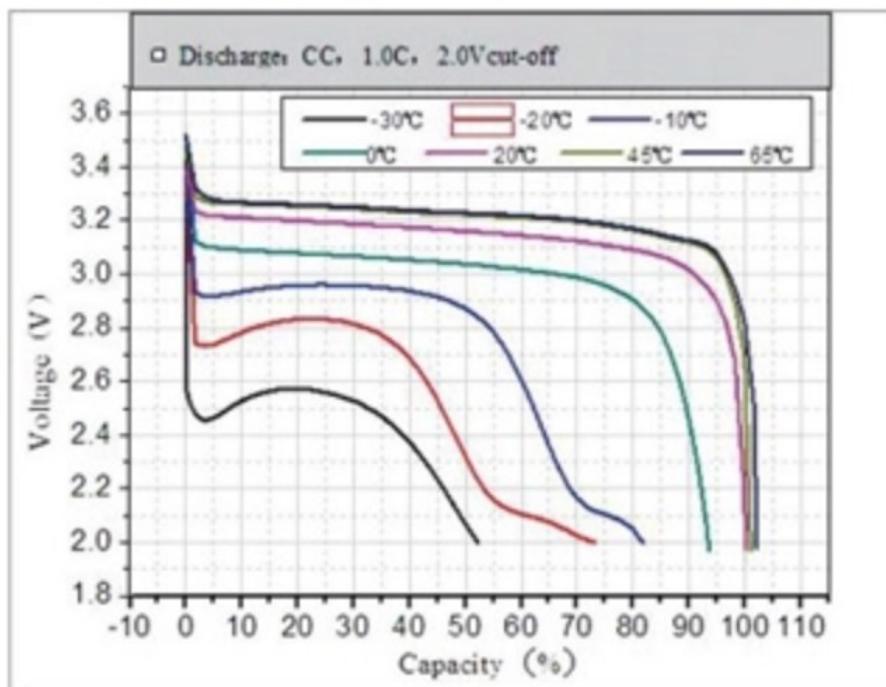


Figure 78: Discharge Curve at Different Temperatures

7.7.2 Option 2



Figure 79: Sparkfun Electronics Power Bank

Option 2 is a portable power bank, it is originally used to charge mobile phones and other handheld accessories. Power banks generally output 5 Volts, making them ideal to power a Raspberry Pi. This particular power bank outputs 5 Volts with a capacity of 2500 mAh, making it suitable to power 3 complete readings. Although the power bank has a slow charge time of 3-7 hours, this was not crucial to the operation of the IHT, since the IHT is estimated to only be used once a week. The power bank has a very robust housing made of an aluminium alloy and has a small form factor allowing it to be easily kept out of the way. While this particular power bank has an average sized capacity and cycle life, it makes up for these factors with safety features such as short circuit, overload and under voltage protection, as well as an ability to supply a low amperage, which is ideal for the Raspberry

Pi since it only draws current for the active components. While the power bank is 145 grams, which is quite heavy, it is estimated that the majority of the weight comes from the robust housing, but overall, it is still relatively cheap (Sparkfun, 2019).

7.7.3 Option 3



Figure 80: USB Battery Pack for Raspberry Pi

This option is considered a medium sized power bank on the market, it is light for its dimensions with a weight of 126.7 grams and has a large capacity of 4000 mAh, which is capable of providing power for 5 complete readings on a single charge. Just like option 2, this power bank can also supply a low amperage, making it suitable to power a Raspberry Pi. Two major drawbacks of this product are the size, and the price. This device is quite bulky and would be difficult to implement into the vest, and for the price of the device, it would be a difficult compromise (Core Electronics, 2019).

7.7.4 Chosen Power Supply

Ultimately, Option 2 was chosen to power the device, its small form factor, price and safety features were ultimately what made this option favourable. This component was cheap relative to the rest of the market and the ability to supply low amperage makes this product ideal to power the Raspberry Pi. The safety features are a major advantage over the other options as they prolong the life of the battery pack while also keeping the user safe. The robust, aluminium alloy housing of the power bank makes this option more favourable over the others as it prolongs the use of the power bank and reduces the risk of damage. The small form factor and solid design makes it a clear choice over the other power sources while also being simple to integrate into the IHT.

7.8 Life Cycle Analysis

A Life Cycle Analysis (LCA) is a study that assesses the potential environmental aspects and impacts associated with a product, process or service. An analysis encompassing the full cycle life is often called the “cradle to the grave” approach. This means that the product will be assessed from raw material extraction through to the final waste processing (Jonna M. Fry, et al. 2010) (B Corporation, 2008). A screening LCA can help identify the most significant contributors to environmental or other impacts, which can then be used to narrow the focus for further study. A full LCA can contribute to supporting marketing claims about specific products or services that claim to minimise environmental impacts (PWC, 2010). In this section of the report, a screening LCA of lithium ion batteries and the Raspberry Pi will be performed to narrow the focus for further studies.

7.8.1 Lithium Ion Batteries

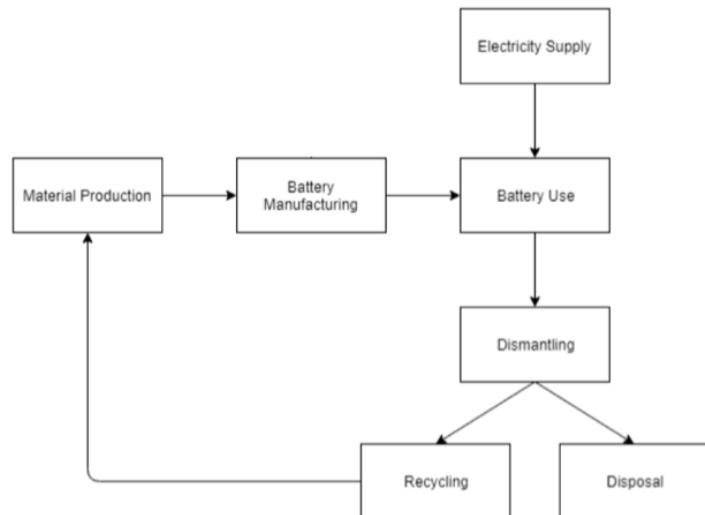


Figure 81: General Life Cycle of Lithium Ion Batteries

The first stage of a cradle to the grave life cycle analysis is the cradle, also known as the origins of the product or better known as the raw materials in which it is made

from. For lithium ion batteries, the raw material are mined in northern Chile and found in the form of lithium carbonate, a purple-grey coloured stone (Ebenerger, A. et al., 2005).

When the materials arrive at the manufacturing plant, they are mixed with binder powder, solvent and additives in a chemical vessel and pumped into a coating machine to eventually make the cathode. The coating machines produces a paste that is spread to a thickness of approximately 200 to 250 micrometres on both sides of an aluminium foil and then rolled and dried. The coated foil is then calendered to make the thickness more uniform and then trimmed to the correct width. The calandering process involves the foil passing between multiple pairs of calendering rollers at high temperatures and pressures. To manufacture the anode, graphite undergoes a similar process to the cathode, but it is spread on a copper foil and trimmed to width. Any trimmed materials are recycled and reused. The anode, separator and cathode layers are wound up and inserted into cylindrical or rectangular cases. These cells are then filled with electrolyte, the composition of the electrolyte and separator is dependent on the manufacturing company so a definitive composition is difficult to analyse. Insulators, seals, valves and safety devices are attached and the cells are crimped shut. The battery cells are fabricated in a completely discharged state and undergo a process of being charged, left on a test stand for several days and discharged. This process is repeated four times and is performed using a “cycler” to verify the quality of the product. Cells are fitted with electronic circuits to control charging and discharging and packed for transport (Gaines, L. et al., 2011)(Sullivan, J. et al., 2010).

Although the Ozawa, K.’s paper is from 1994, the degradation of the lithium ion batteries were found to vary from 10-20% after 500 cycles. It also found that lithium cobalt oxide (LiCoO_2) is very stable and has minimal risk to being overcharged since it is a very stable compound. Lithium is an extremely reactive metal, however, since the lithium is in a non-metallic state, these batteries are extremely safe against abuse. The batteries made by Sony, the first company to use these types of batteries commercially, are safe against abuse such as crushing and nail penetration. With a current breaker device and a vent mechanism provided for an abnormal rise of internal pressure, these batteries are safe even under overcharge conditions since the charge-discharge cycling is caused by the formation of a thin, non-active film on the surface of the carbon anode (Ozawa, K., 1994). Other sources suggested that lithium-ion batteries typically have life cycles from 500 to 1200

cycles (Electropaedia, 2005), although an advancement of technology may have been a large factor in the increased cycle life. The progress of technology is validated since a 2006 paper received results of approximately 2000 cycles (Ning, G. et al., 2006). It was found that typical life cycles of 1000 cycles at 80% depth of discharge (DOD). 100% DOD is often considered to be the life of a battery, meaning it has discharged 100% or lost 100% of its full charge, so 80% DOD is defined as a battery that has lost 80% of its charge (Guena, T., & Leblanc, P., 2006). When using lithium ion batteries over extended periods and charge cycles, the batteries begin to experience power fade. Power fade is when the maximum power of the battery slowly decreases. So, an increase of power fade means that the output power is decreasing. It was found that power fading does not originate from a single cause but from various processes and their interactions, chemistry of the system such as the active materials, electrode design, electrolyte composition and impurities. Ageing mechanics occur at the electrodes, however, the anode and cathode interact very differently to ageing. At the cathode, phase transitions and structural changes in the bulk material heavily influence ageing, while changes in the anode material are considered to be of only minor importance (Vetter, J. et al., 2005). Power fade can be effected by multiple factors but while under continuous conditions, such as temperature, the percent power fade was found to increase at a linear rate (Wright, R. B. et al., 2003).

At the end of the battery's life, batteries can be collected through special government collections or battery recycling bins. They are dismantled and sent to pyrometallurgical and hydrometallurgical treatments (Marques, P. et al, 2019). Recycling can recover many materials at different production stages, all the way from basic building blocks of battery manufacturing and fabrication to battery-grade materials. At one extreme side of recycling is the smelting process that recovers basic elements or salts. These are operational on a large scale and can take just about any input, including different battery chemistries or mixed feed. Smelting takes place at a high temperature, and organics, including the electrolyte and carbon anodes, are burned as fuel or reductant. The valuable metals, cobalt and nickel are recovered and sent to be refined so that the product is suitable for any use. Cobalt and Nickel currently produce high returns in the secondary marketplace. The other materials, including lithium, are contained in the slag, which is now used as an additive in concrete. The lithium could be recovered by using a hydrometallurgical process, if justified by price or regulations (Gaines, L. et al., 2011). Some companies claim that using recycled

cobalt reduces production energy for lithium cobalt oxide by 70%. Since cobalt recycling produces such high returns, some companies do not see the economic advantage of recycling lithium, although, the industry has concerns about the reduction of cobalt use in lithium ion batteries, if the secondary market prices continue to decrease, and lithium market prices increase, then many companies may invest in the recycling of lithium.

At the other extreme, a non-destructive method of recovering battery-grade materials have been demonstrated. This process requires an extremely uniform feed, because impurities in the feed jeopardizes the quality of the product. The components are separated by a variety of physical and chemical processes, and all active materials and metals can be recovered. Only the separator is unlikely to be usable, because its form cannot be retained. This alternative approach to battery recycling is a low-temperature process with a low energy requirement. A majority of the original energy and processing required to produce battery-grade materials from raw materials is saved. The recycling of lithium ion battery materials potentially reduces the material production energy by as much as 50%. If battery active materials can be recycled in forms suitable for reuse with minimal processing, this percentage reduction could be even higher, as considerable energy is committed to making them from raw materials (Gaines, L. et al., 2011). In 2002, the carbon footprint of lithium ion batteries is approximately 70kg of CO₂ per KWh (Armand, M. et al., 2008). An experimental process to recycle cobalt and lithium from batteries is currently under investigation, batteries are first dismantled and separated from their plastic and metal housings, the cells, which contain the chemicals used in the electrochemical reactions are then crushed and introduced to a leaching reagent. Experimental, environmentally friendly leaching reagents have been analysed and recover 90% of cobalt and 100% of lithium (Li, L. et al., 2010) and another environmentally friendly leaching reagent recovered 70% of cobalt and 99.1% of lithium (Jha, M.K. et al., 2013). These new recycling methods are more environmentally friendly than smelting and could prove to be very beneficial economically and environmentally in the future.

7.8.2 Raspberry Pi

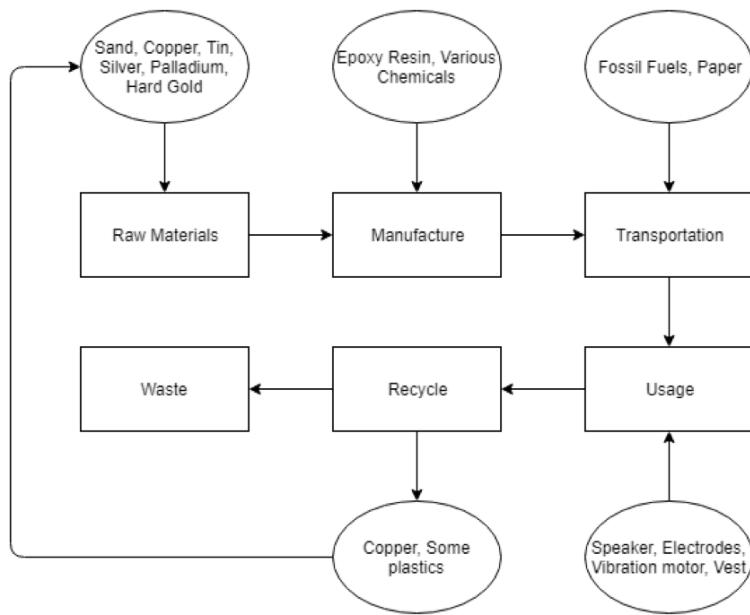


Figure 82: General Life Cycle of a Raspberry Pi

The raw materials that make up a Raspberry Pi consist of sand, copper, tin, silver, palladium and hard gold, these materials are present in almost every circuit board on the market. These raw materials are then processed with epoxy resin and to make fibreglass and create the circuit board and the conductive pathways of the circuit board are made of copper. Once the board has been created, a photosensitive material called a photo-resist is applied to both sides of the board, ultraviolet light is used so that the board can be handled safely in visible light. Next, the board is electroplated with tin, the tin only attaches to the copper, this is to protect the copper against tarnishing and presents a better surface to solder on. Finally, the components of the board can be soldered on to the board.

The most important component on the circuit board is the CPU, the CPU is made from silicon dioxide found in sand, the silicon is extracted from the sand and is purified in

multiple stages until it has reached Electronic Grade Silicon. The silicon of this grade is so pure that for every 1 billion silicon atoms, there is a possibility of just one atom being foreign. A large crystal of silicon is grown and sliced with an extremely thin saw into silicon slices called wafers. These are then polished and a laser is used to cut the wafer into chips.

Once the Raspberry Pi is assembled and packaged, they are then transported by plane to consumers. A wide variety of components can be connected to the Raspberry Pi, such as a camera, microphone, mice and LEDs. Since the circuit board has no moving parts, it is very uncommon for the board to break during regular use such as wear and tear. The connectors and ports may become damaged if they are frequently used, but can be replaced simply by soldering.

When the Raspberry Pi is disposed of, it can be recycled in a similar way to batteries, there are special collections just for electronics. Just like any computer or electronic device, they contain a various number of circuit boards, these computers can be stripped down of all the internals and the copper can be recovered through refining processes. The copper can be completely recycled and used in other devices or for completely different uses. Most of the plastics used in frames or for decoration are disposed of in land fill (Design Life-Cycle, 2016).

7.9 Future Work

A possible improvement to this study could be the implementation of other components in the life cycle analysis so that the cycle life of the components can be compared, and an investigation into the components that will fail first. Another improvement would be the inclusion of energy usage and the environmental impacts of each stage of the life cycle analysis. Regarding the power source, an improvement on the chosen product would be to find a lighter product with a similar robust housing and a larger capacity without a large price trade off.

8 Conclusion

Many developing regions of the world struggle when it comes to reliable medical care and equipment, especially with cardiovascular diseases. The Ischemic Heart Tracker aims to aid doctors in these struggling regions by reducing the need for patients to visit, and thereby allowing the doctor to attend other patients. The Ischemic Heart Tracker is a wearable device worn as a double layered vest. The first layer allows the electrodes to rest on the patient and the second layer ensures all the components are secure. The electrodes attach in a similar way to a lapel pin and are positioned specifically for each patient. They are arranged in an array to feed signals to an on-board computer with an electrocardiograph of the heart. The computer utilises a 2D CNN to analyse the ECG and will determine whether the user's heart is stable or unstable and will recommend actions based on the result of the reading. The readings are sent to Australian based health professionals to review the ECGs and the results of the device. As the ECG readings are sent to health professionals, a database of these readings will grow, allowing for retraining and increased accuracy of the 2D CNN over time and with possibly more classifications of diagnosis. The IHT was designed with the Raspberry Pi Zero W as the on-board computer, however, during testing, some unexpected difficulties occurred regarding incompatibilities between the Raspberry Pi Zero W architecture and the Tensorflow library. Fortunately, the network capabilities and CNN were tested and found to be operational on a Raspberry Pi 3B+, a newer model of Raspberry Pi.

A larger database with more classifications to identify different cardiovascular diseases could be an improvement in the future. A retrained 2D CNN able to distinguish multiple cardiovascular diseases would further reduce the patient load per doctor. Another recommended improvement of this device would be to incorporate the use of a stress test, this would allow the device to diagnose patients and the ECG database would continue to grow, allowing for stable angina diagnosis. Only slight physical modifications would be required to incorporate a stress test feature, such as upgrading the device to become more interactive, a larger battery and more secure mechanisms for the electrodes. The 2D CNN would have to be retrained but can be simply done through a software update.

9 References

9.1 Introduction

Anand, S., Fan, V. (2016). The Health Workforce in India [Ebook] (pp. 9, 17, 20). Geneva: World Health Organization. Retrieved from <https://www.who.int/hrh/resources/16058healthworkforceIndia.pdf>

Prabhakaran, D., Jeemon, P., Roy, A. (2016). Cardiovascular Diseases in India. *Circulation*, 133(16), 16051620. doi: 10.1161/circulationaha.114.008729

Stacey, J. (2019). ECG electrodes. UNSW

9.2 Electrode Design - Jonathan Stacey

Stacey, J. (2019). ECG electrodes. UNSW

Dozio, R., Baba, A., Assambo, C., & Burke, M. J. (2007). Time Based Measurement of the Impedance of the Skin-Electrode Interface for Dry Electrode ECG Recording. 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. doi:10.1109/tembs.2007.4353463

Chen, Y., Beeck, M. D., Vanderheyden, L., Carrette, E., Miha jlović, V., Vanstreels, K., . . . Hoof, C. V. (2014). Soft, Comfortable Polymer Dry Electrodes for High Quality ECG and EEG Recording. *Sensors*, 14(12), 23758–23780. doi:10.3390/s141223758

Stacey, J. (2019). Electrode Design. UNSW

Grassini, S. (2013). Electrochemical impedance spectroscopy (EIS) for the in-situ analysis of metallic heritage artefacts. *Corrosion and Conservation of Cultural Heritage Metallic Artefacts*, 347–367. doi: 10.1533/9781782421573.4.347

Tallgren, P., Vanhatalo, S., Kaila, K., & Voipio, J. (2005). Evaluation of commercially available electrodes and gels for recording of slow EEG potentials. *Clinical Neurophysiology*, 116(4), 799–806. doi: 10.1016/j.clinph.2004.10.001

Zhang, M, & Mak, A. F. T. (1999). In vivo friction properties of human skin . *Prosthetics and Orthotics International*, 23, 1–7. doi: 10.3109/03093649909071625

Queensland Health. (2012). Exercise Stress Testing Cardiac Sciences. Queensland Government.

Stress analysis of compression springs for fatigue loading. Retrieved from <https://www.mechanicsweb.info/MechanicalSprings/CompressionSpringFatigueDesign.aspx>

(2019, August 24). India. Retrieved from <https://www.who.int/countries/ind/en/>

9.3 Physical Housing - Ping Li

(2012). Retrieved 23 August 2019, from https://copperalliance.eu/uploads/2018/01/lca_copper_wire_2707.pdf

(2015). Retrieved 23 August 2019, from <https://www.worldsteel.org/en/dam/jcr:00892d89-551e-42d9-ae68-abdbd3b507a1/Steel+in+the+circular+economy---A+life+cycle+perspective.pdf>

(2019). Retrieved 23 August 2019, from <http://mistrafuturefashion.com/wp-content/uploads/2015/06/Environmental-assessment-of-Swedish-fashion-consumption-LCA.pdf>

Chen, T. (2011). Specialist Yarn and Fabric Structures Developments and Applications.

Costa, T., Feitor, M., Alves, C., Freire, P., & de Bezerra, C. (2006). Effects of gas composition during plasma modification of polyester fabrics. Journal Of Materials Processing Technology, 173(1), 40-43. doi: 10.1016/j.jmatprotec.2005.11.008

Senthilkumar, M. (2011). Elastane fabrics—A tool for stretch applications in sports.

Shen, J. (2011). Retrieved 23 August 2019, from https://nature.berkeley.edu/classes/es196/projects/2011final/ShenJ_2011.pdf

Smith, G., & Barker, R. (1995). Life cycle analysis of a polyester garment. Resources, Conservation And Recycling, 14(3-4), 233-249. doi: 10.1016/0921-3449(95)00019-f

9.4 Data Analysis - Calandra Lunardo

Lunardo, C. (2019) Automated classification of Electrocardiogram Data for Monitoring of Coronary Heart Disease. UNSW

- Lunardo, C. (2019). Data Analysis Interim Report. UNSW
- Hampton, John. (2013). The ECG Made Easy E-Book. London: Churchill Livingstone.
- Ashley EA, Niebauer J. Cardiology Explained. London: Remedica; 2004. Chapter 3, Conquering the ECG. Available from: <https://www.ncbi.nlm.nih.gov/books/NBK2214/>
- Ginghina, C., Ungureanu, C., Vladaia, A., Popescu, B. A., & Jurcut, R. (2009). The electrocardiographic profile of patients with angina pectoris. Journal of medicine and life, 2(1), 80–91.
- Rawshani, A. (2019). Classification of Acute Coronary Syndromes (ACS) Acute Myocardial Infarction (AMI) – ECG learning. Clinical ECG Interpretation. Available at: <https://ecgwaves.com/ecg-topic/acute-coronary-syndromes-acs-myocardial-infarction-ami>
- Kapit. Macey. Meisami (2000). The physiology Coloring Book. Addison Wesley Longman.
- Moody GB, The PhysioNet / Computers in Cardiology Challenge 2008: T-Wave Alternans. Computers in Cardiology 35:505-508; 2008.
- Martinez JP, Pahlm O, Ringborn M, Warren S, Laguna P, Sornmo L. The STAFF III Database: ECGs Recorded During Acutely Induced Myocardial Ischemia. Comput Cardiol, 44:266–133, 2017.
- Bousseljot R, Kreiseler D, Schnabel, A. Nutzung der EKG-Signaldatenbank CARDIODAT der PTB über das Internet. Biomedizinische Technik, Band 40, Ergänzungsband 1 (1995) S 317
- Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals (2003). Circulation. 101(23):e215-e220.
- Christov I. I. (2004). Real time electrocardiogram QRS detection using combined adaptive threshold. Biomedical engineering online, 3(1), 28. doi:10.1186/1475-925X-3-28
- T. Naumann and I. Silva, "Scaling the PhysioNet WFDB Toolbox for MATLAB and Octave," Computing in Cardiology 2014, Cambridge, MA, 2014, pp. 161-164.

9.5 Data Analysis - Zachary Hamid

Hamid, Z. (2019). Data Analysis Interim Report. UNSW

Yamashita, R., Nishio, M., Do, R., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights Into Imaging*, 9(4), 611-629. doi: 10.1007/s13244-018-0639-9

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. Retrieved from <http://arxiv.org/abs/1409.1556>

Ahire, J. (2018). The Artificial Neural Networks Handbook. Retrieved 23 August 2019, from <https://medium.com/@jayeshbahire/the-artificial-neural-networks-handbook-part-4-d2087d1f583e>

Artificial neural network. (2019). Retrieved 23 August 2019, from https://en.wikipedia.org/wiki/Artificial_neural_network

Basha, S., Dubey, S., Pulabaigari, V., & Mukherjee, S. (2019). Impact of Fully Connected Layers on Performance of Convolutional Neural Networks for Image Classification. Corr, abs/1902.02771. Retrieved from <http://arxiv.org/abs/1902.02771>

Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation Functions: Comparison of trends in Practice and Research for Deep Learning. Corr, abs/1811.03378. Retrieved from <http://arxiv.org/abs/1811.03378>

Despois, J. (2018). Memorizing is not learning!. Retrieved 23 August 2019, from <https://hackernoon.com/memorizing-is-not-learning-6-tricks-to-prevent-overfitting-in-machine-learning-820b091dc42>

Hawkins, D. M. (2004, January). The Problem of Overfitting. *Journal of Chemical Information and Computer Sciences*. <https://doi.org/10.1021/ci0342472>

Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R. (2014) Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15, 1, 1929-1958.

Baldi, P., & Sadowski, P. (2013). Understanding dropout. In *Advances in Neural Information Processing Systems*. Neural information processing systems foundation.

- Ng, A. Y. (2004). Feature selection, L1 vs. L2 regularization, and rotational invariance. In Proceedings, Twenty-First International Conference on Machine Learning, ICML 2004 (pp. 615–622).
- Mazilu, S., & Iria, J. (2011). L 1 vs. L 2 regularization in text classification when learning from labeled features. In Proceedings - 10th International Conference on Machine Learning and Applications, ICMLA 2011 (Vol. 1, pp. 166–171). <https://doi.org/10.1109/ICMLA.2011.85>
- Moore, R., & DeNero, J. (2011). L1 and L2 Regularization for Multiclass Hinge Loss Models. Proceedings of Symposium on Machine Learning in Speech and Natural Language Processing, 1–5. Retrieved from http://denero.org/content/pubs/mlslp11_moore_regularization.pdf
- Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Corr, abs/1502.03167. Retrieved from <http://arxiv.org/abs/1502.03167>
- Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. Retrieved from <http://arxiv.org/abs/1412.6980>
- Duchi, J., Hazan, E., & Singer, Y. (2010). Adaptive subgradient methods for online learning and stochastic optimization. In COLT 2010 - The 23rd Conference on Learning Theory (pp. 257–269).
- Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., & Recht, B. (2017). The marginal value of adaptive gradient methods in machine learning. In Advances in Neural Information Processing Systems (Vol. 2017-December, pp. 4149–4159). Neural information processing systems foundation.
- Chollet, F. (2015). Keras (Version 2.2.5) [Python].
- Wei, Q., & Dunbrack, R. L. (2013). The Role of Balanced Training and Testing Data Sets for Binary Classifiers in Bioinformatics. PLoS ONE, 8(7). <https://doi.org/10.1371/journal.pone.0067863>

Martínez, J. P., Pahlm, O., Ringborn, M., Warren, S., Laguna, P., & Sörnmo, L. (2017).

The STAFF III Database: ECGs recorded during acutely induced myocardial ischemia. In Computing in Cardiology (Vol. 44, pp. 1–4). IEEE Computer Society.
<https://doi.org/10.22489/CinC.2017.266-133>

Bousseljot, R., Kreiseler, D., & Schnabel, A. (1995). Nutzung der EKG-Signalldatenbank CARDIODAT der PTB über das Internet. Biomedizinische Technik / Biomedical Engineering, 40(s1), 317-318.

9.6 Data Communication - Chuhao (Martin) Cai

Cloud Storage Pricing — S3 Pricing by Region — Amazon Simple Storage Service. (2019).

Retrieved 25 August 2019, from <https://aws.amazon.com/s3/pricing/>

Cai, C.H. (2019). Data Communication Interim Report. UNSW

SD cards - Raspberry Pi Documentation. (2019). Retrieved from <https://www.raspberrypi.org/documentation/installation/sd-cards.md>

Linux commands - Raspberry Pi Documentation. (2019). Retrieved from <https://www.raspberrypi.org/documentation/linux/usage/commands.md>

Turning your Raspberry PI Zero into a USB Gadget. (2019). Retrieved from <https://learn.adafruit.com/turning-your-raspberry-pi-zero-into-a-usb-gadget/ethernet-gadget>

Ylonen, T., & Lonvick, C. (2006). The secure shell (SSH) protocol architecture.

SSH (Secure Shell) - Raspberry Pi Documentation. (2019). Retrieved from <https://www.raspberrypi.org/documentation/remote-access/ssh/>

Setting WiFi up via the command line - Raspberry Pi Documentation. (2019). Retrieved from <https://www.raspberrypi.org/documentation/configuration/wireless/wireless-cli.md>

Flack, M. (1999). The Story about Ping. Retrieved from <https://ftp.arl.army.mil/~mike/ping.html>

Getting Started with Raspberry Pi Zero W. (2019). Retrieved from <https://tutorial.cytron.io/2017/05/02/getting-started-raspberry-pi-zero-w/>

VNC (Virtual Network Computing) - Raspberry Pi Documentation. (2019). Retrieved from <https://www.raspberrypi.org/documentation/remote-access/vnc/>

MQTT. (2019). Retrieved 24 August 2019, from <http://mqtt.org/>

Corbett, N. (2017). Understanding the AWS IoT Security Model [Blog]. Retrieved from <https://aws.amazon.com/blogs/iot/understanding-the-aws-iot-security-model/>

Dierks, T., & Rescorla, E. (2008). The Transport Layer Security (TLS) Protocol Version 1.2. Retrieved 24 August 2019, from <https://tools.ietf.org/html/rfc5246>

Warren, I., Meads, A., Srirama, S., Weerasinghe, T., & Paniagua, C. (2014). Push notification mechanisms for pervasive smartphone applications. *IEEE Pervasive Computing*, 13(2), 61-71.

Functional and Nonfunctional Requirements: Specification and Types. (2018). Retrieved 24 August 2019, from <https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-and-types/>

9.7 Power - Aidan Cornwall

Bosnjak, A., Kennedy, A., Linares, P., Borges, M., McLaughlin, J., & Escalona, O. J. (2017, July). Performance assessment of dry electrodes for wearable long term cardiac rhythm monitoring: skin-electrode impedance spectroscopy. In 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 1861-1864). IEEE.

Gruetzmann, Anna, Hansen, Stefan, & Müller, Jörg. (2007). Novel dry electrodes for ECG monitoring. *Physiological Measurement.*, 28(11), 1375–1390. <https://doi.org/10.1088/0967-3334/28/11/005>

Grimnes, S. (1983). Impedance measurement of individual skin surface electrodes. *Medical & Biological Engineering & Computing.*, 21(6), 750–755. <https://doi.org/10.1007/BF02464038>

- Yu, X., Boehm, A., Neu, W., Venema, B., Marx, N., Leonhardt, S., & Teichmann, D. (2017). A wearable 12-lead ECG T-shirt with textile electrodes for unobtrusive long-term monitoring—Evaluation of an ongoing clinical trial. In EMBC & NBC 2017(pp. 703-706). Springer, Singapore.
- Joutsen, A. S., Kaappa, E. S., Karinsalo, T. J., & Vanhala, J. (2017). Dry electrode sizes in recording ECG and heart rate in wearable applications. In EMBC & NBC 2017 (pp. 735-738). Springer, Singapore.
- Meziane, N., Webster, J. G., Attari, M., & Nimunkar, A. J. (2013). Dry electrodes for electrocardiography. *Physiological measurement*, 34(9), R47.
- Raspberry Pi (2019). FAQs. Retrieved From:<https://www.raspberrypi.org/documentation/faqs/#power>
- Plux (2019). Electrocardiography (ECG) Sensor. Retrieved From: https://plux.info/barebone-sensors/10-electrocardiography-ecg-sensor.html?search_query=electrode+ecg&results=92
- Baolong Electronic Group (2018). BVM0720 Small Coin Vibration Motor. Retrieved From: <https://www.vibrationmotors.com/product-catalogue/coin-vibration-motors/c0720b015f>
- Jaycar Electronics (2019). 27mm All purpose replacement speaker. Retrieved From: <https://www.jaycar.com.au/27mm-all-purpose-replacement-speaker/p/AS3002>
- Raspberry Pi (2019). FAQs Power. Retrieved From: <https://www.raspberrypi.org/documentation/faqs/#pi-power>
- Battery Stuff (2019). Battery Bank Tutorial. Retrieved From: <https://www.batterystuff.com/kb/articles/battery-articles/battery-bank-tutorial.html>
- Fluke (2019). BP120MH Battery Pack. Retrieved From: <https://www.fluke.com/en-us/product/accessories/batteries/fluke-bp120mh#fbid=jIENSCPze1m?techspecs>
- Energizer Recharge (2019). Rechargeable Batteries. Retrieved From: <https://www.energizer.com/batteries/energizer-rechargeable-batteries>

- Cornwall, A (2019). Literature Review Thesis A (Power Supply).
- Leonov, V., Torfs, T., Fiorini, P., & Van Hoof, C. (2007). Thermoelectric converters of human warmth for self-powered wireless sensor nodes. *IEEE Sensors Journal*, 7(5), 650-657.
- Seung, W., Gupta, M. K., Lee, K. Y., Shin, K. S., Lee, J. H., Kim, T. Y., ... & Kim, S. W. (2015). Nanopatterned textile-based wearable triboelectric nanogenerator. *ACS nano*, 9(4), 3501-3509.
- Brain, M., Bryant, C. W., & Pumphrey, C. (2000). How batteries work. How Stuff Works, 1.
- Van Noorden, R. (2014). The rechargeable revolution: A better battery. *Nature News*, 507(7490), 26.
- Cope, R. C., & Podrazhansky, Y. (1999, January). The art of battery charging. In Fourteenth Annual Battery Conference on Applications and Advances. Proceedings of the Conference (Cat. No. 99TH8371) (pp. 233-235). IEEE.
- Armand, M., & Tarascon, J. M. (2008). Building better batteries. *nature*, 451(7179), 652.
- Yoshio, M., Brodd, R. J., & Kozawa, A. (2009). Lithium-ion batteries (Vol. 1). New York: Springer.
- Nazri, G. A., & Pistoia, G. (Eds.). (2008). Lithium batteries: science and technology. Springer Science & Business Media.
- Alibaba Shenzhen Eastar Battery Co, (2019). Factory Price 5v li-ion battery Pack Lithium ion rechargeable battery. Retrieved From: https://www.alibaba.com/product-detail/Factory-Price-5v-li-ion-battery_1901042808.html?spm=a2700.7724857.normalList.141.48d049cfdq39rZ
- Sparkfun (2019). Lithium Ion Battery Pack -2.5Ah(USB) Retrieved From: <https://www.sparkfun.com/products/14367>
- Core Electronics (2019). USB Battery Pack for Raspberry Pi-4000mAh-5V@1A. Retrieved From: <https://core-electronics.com.au/usb-battery-pack-for-raspberry-pi-4400mah-5v-at-1a.html>

- Energizer Data (2019). Energizer NH15-2300 (HR6) Data Sheet. Retrieved From: <http://data.energizer.com/pdfs/nh15-2300.pdf>
- Yamaura, J., Ozaki, Y., Morita, A., & Ohta, A. (1993). High voltage, rechargeable lithium batteries using newly-developed carbon for negative electrode material. *Journal of power sources*, 43(1-3), 233-239.
- Armand, M., & Tarascon, J. M. (2008). Building better batteries. *nature*, 451(7179), 652.
- Jonna Meyhoff Fry, Bryan Hartlin, Erika Wallén and Simon Aumônier (2010). Life cycle assessment of example packaging systems for milk. Retrieved From: <http://www.wrap.org.uk/sites/files/wrap/Final%20Report%20Retail%202010.pdf>
- B Corporation (2008). B Resource Guide: Conducting a Life Cycle Assessment (LCA). Retrieved From:http://nbis.org/nbisresources/life_cycle_assessment_thinking/guide_life_cycle_assessment_bcorp.pdf
- PWC (2010). Life Cycle Assessment and Forest Products: A White Paper. Retrieved From:<https://www.pwc.com/gx/en/forest-paper-packaging/pdf/fpac-lca-white-paper.pdf>
- Ebensperger, A., Maxwell, P., & Moscoso, C. (2005). The lithium industry: its recent evolution and future prospects. *Resources Policy*, 30(3), 218-231.
- Gaines, L., Sullivan, J., Burnham, A., & Belharouak, I. (2011, January). Life-cycle analysis for lithium-ion battery production and recycling. In *Transportation Research Board 90th Annual Meeting*, Washington, DC (pp. 23-27).
- Sullivan, J. L., & Gaines, L. (2010). A review of battery life-cycle analysis: state of knowledge and critical needs (No. ANL/ESD/10-7). Argonne National Lab.(ANL), Argonne, IL (United States).
- Ozawa, K. (1994). Lithium-ion rechargeable batteries with LiCoO₂ and carbon electrodes: the LiCoO₂/C system. *Solid State Ionics*, 69(3-4), 212-221.
- Electropaedia (2005). Battery Life (and Death). Retrieved From: <https://www.mpoweruk.com/life.htm>

- Ning, G., White, R. E., & Popov, B. N. (2006). A generalized cycle life model of rechargeable Li-ion batteries. *Electrochimica acta*, 51(10), 2012-2022.
- Vetter, J., Novák, P., Wagner, M. R., Veit, C., Möller, K. C., Besenhard, J. O., ... & Hammouche, A. (2005). Ageing mechanisms in lithium-ion batteries. *Journal of power sources*, 147(1-2), 269-281.
- Guena, T., & Leblanc, P. (2006, September). How depth of discharge affects the cycle life of lithium-metal-polymer batteries. In INTELEC 06-Twenty-Eighth International Telecommunications Energy Conference (pp. 1-8). IEEE.
- Marques, P., Garcia, R., Kulay, L., & Freire, F. (2019). Comparative life cycle assessment of lithium-ion batteries for electric vehicles addressing capacity fade. *Journal of Cleaner Production*, 229, 787-794.
- Wright, R. B., Christophersen, J. P., Motloch, C. G., Belt, J. R., Ho, C. D., Battaglia, V. S., ... & Sutula, R. A. (2003). Power fade and capacity fade resulting from cycle-life testing of advanced technology development program lithium-ion batteries. *Journal of Power Sources*, 119, 865-869.
- Li, L., Ge, J., Chen, R., Wu, F., Chen, S., & Zhang, X. (2010). Environmental friendly leaching reagent for cobalt and lithium recovery from spent lithium-ion batteries. *Waste Management*, 30(12), 2615-2621.
- Jha, M. K., Kumari, A., Jha, A. K., Kumar, V., Hait, J., & Pandey, B. D. (2013). Recovery of lithium and cobalt from waste lithium ion batteries of mobile phone. *Waste management*, 33(9), 1890-1897.
- Boyden, A., Soo, V. K., & Doolan, M. (2016). The environmental impacts of recycling portable lithium-ion batteries. *Procedia CIRP*, 48, 188-193.
- Contestabile, M., Panero, S., & Scrosati, B. (2001). A laboratory-scale lithium-ion battery recycling process. *Journal of Power Sources*, 92(1-2), 65-69.
- Lain, M. J. (2001). Recycling of lithium ion cells and batteries. *Journal of Power Sources*, 97, 736-738.

Dearborn, S. (2005). Charging Li-ion batteries for maximum run times. Power Electronics Technology, 31(4), 40-49.

Design Life-Cycle, (2016). Life Cycle of the Raspberry Pi A+. Retrieved From:<http://www.designlife-cycle.com/raspberry-pi>

Appendices

A Spring Data Sheets

Part Number

LC 016A 0 S

Description	Specification
Spring Type:	Standard Compression Series (inch)
Outside Diameter (mm):	3.05
Hole Diameter (mm):	3.18
Rod Diameter (mm):	null
Free Length (mm):	4.78
Rate (N):	3.735
Solid Height (mm):	2.21
Wire Diameter (mm):	0.41
Material:	Stainless Steel
Load at Solid Height (N):	9.61
Ends:	Squared and Ground
Direction of Wind:	Factory Option
Active Coils:	3.41
Total Coils:	5.41
Finish:	Passivate per ASTM A967

Retrieved from <https://www.leespring.com/compression-springs>

Part Number

CBM010B 01 E

Description	Specification
Spring Type:	Bantam™ Mini Series (metric)
Outside Diameter (mm):	1.32
Hole Diameter (mm):	1.5
Rod Diameter (mm):	null
Free Length (mm):	2
Rate (N):	0.241
Solid Height (mm):	0.53
Wire Diameter (mm):	0.1
Material:	Elgiloy
Load at Solid Height (N):	0.35
Ends:	Squared
Direction of Wind:	Factory Option
Active Coils:	2.36
Total Coils:	4.36
Finish:	None

Retrieved from <https://www.leespring.com/compression-springs>

B Electrode Material Sources

Component	Material Cost	Source	Hyperlink
Plate	\$150/m ²	All Things Stainless, Australia	http://allthingsstainless.com.au/virtuemart/stainless-steel-sheet/304-0.9mm-satin-stainless-sheet-per-square-metre.htm
Grip Ring	\$5.68/kg	Alibaba.com	https://www.alibaba.com/product-detail/price-of-silicone-rubber-per-kg_60777670999.html?spm=a2700.7724857.normalList.30.3e0a114bDwzmGe
Electrode	\$0.55/g	Money Metals Exchange	https://www.moneymetals.com/precious-metals-charts/silver-price
Pin	\$9/kg	Alibaba.com	https://www.alibaba.com/product-detail/brass-ingot-manufacturer-hot-sale-cheap_62058859944.html?spm=a2700.7724857.normalList.8.449d2789clWqez&s=p
Clasp 1	\$234/m ²	All Things Stainless, Australia	http://allthingsstainless.com.au/virtuemart/stainless-steel-sheet/304-2.0mm-satin-sheet-per-square-metre.html
Clasp 2	\$234/m ²	All Things Stainless, Australia	http://allthingsstainless.com.au/virtuemart/stainless-steel-sheet/304-2.0mm-satin-sheet-per-square-metre.html
Primary Spring	\$0.446/unit	Lee Spring	https://www.leespring.com/compression-springs?search=LC016A0S
Secondary Spring 1	\$0.739/unit	Lee Spring	https://www.leespring.com/compression-springs?search=CBM010B01E
Secondary Spring 1	\$0.739/unit	Lee Spring	https://www.leespring.com/compression-springs?search=CBM010B01E

C MATLAB Extraction Code

```
databases = ["staffiii"; "ptbdb"];
selected_database = 2;

[data, status] = urlread(char(strcat("http://physionet.org/physiobank/database/", databases(selected_database), "/CONTROLS")));
if (status ~= 1)
    disp("Error: Couldn't connect to PhysioNet Database!");
end

records = regexp(data, '\n', 'split');

records_trim = records;
if (any(contains(records, '/)))
    records_trim = extractAfter(records, '/');
end

disp(size(records));

%(size(records,2)-1)
for i=52:81
    % Get the signals
    [ecg, Fs, tm] = rdsamp(char(strcat(databases(selected_database), '/', records(i))), [], 100000);
    %%for staffiii database it's a 12-lead ECG with signal channels V1-V6, I,
    %%II, III
    %%AVR, AVL and AVF must be calculated
    %%https://pdfs.semanticscholar.org/8160/8b62b6efb007d112b438655dd2c897759fb1.pdf
    %%AVR = -1/2 * I - 1/2 * II
    %%AVL = I - 1/2 * II
    %%AVF = II - 1/2 * I
    avr = -0.5 * ecg(:,1) - 0.5 * ecg(:,2);
    avl = ecg(:,1) - 0.5 * ecg(:,2);
    avf = ecg(:,2) - 0.5 * ecg(:,1);
    fprintf("Got ECG number %i!\n", i);
    matrix = tm';
    matrix = [matrix; ecg(:,7)];
    matrix = [matrix; ecg(:,8)];
    matrix = [matrix; ecg(:,9)];
```

```
matrix = [matrix; ecg(:,10)'];
matrix = [matrix; ecg(:,11)'];
matrix = [matrix; ecg(:,12)'];
matrix = [matrix; ecg(:,1)'];
matrix = [matrix; ecg(:,2)'];
matrix = [matrix; ecg(:,3)'];
matrix = [matrix; avr'];
matrix = [matrix; avl'];
matrix = [matrix; avf'];

% Must have format:
% V1 - V6, I, II, III, aVR, aVL, aVF
filename = strcat(databases(selected_database), "_", records_trim(i), ".csv");
writematrix(matrix, filename, 'Encoding', 'UTF-8');
end
```

D Python Extraction & Image Processing Code

```
import matplotlib.pyplot as plt
import numpy as np
import os
from os import walk
import cv2
import csv
import math

import biosppy.signals.ecg as bioecg

data_dir = "data/ptbdb/"
image_dir = "data/images/"

num_leads = 12

# Function to plot on a subplot with necessary parameters
def plot(x,y,i):
    plt.subplot(num_leads,1,i)
    plt.plot(x,y, 'k', linewidth=0.5)
    plt.xticks([]), plt.yticks([])
    # Stop lines connected axis tick marks being shown
    for s in plt.gca().spines.values():
        s.set_visible(False)
    return

# For all directories that aren't images (that is, all separate .csv database
directories)
for (dirpath, dirnames, filenames) in walk(data_dir):
    if (dirpath.split("/") [1].split("\\\\") [0] != "images"):
        if filenames:
            for file in filenames:
                # Open each .csv file within a given directory and process it
                print("opening ", file)
                reader = csv.reader(open(dirpath + "/" + file, newline='\n'), delimiter=',')
                x = list(reader)
                matrix = np.array(x).astype("float")
```

```

# Use Christov Segmentation method on 6th lead of ECG data (Lead I)
# Since this is the most stable lead (by inspection) for the R peak
r_peaks = bioecg.christov_segmenter(matrix[5,:])

# Using the detected R-peaks of the signal, segment the ECG signal into
# individual beats
# by taking the RR interval from previous beat and dividing by 2
big_matrix = []
for i in range(len(r_peaks[0])-1):
    start = 0
    end = 0
    if (i != 0):
        start = (r_peaks[0][i] - ((r_peaks[0][i] - r_peaks[0][i-1])/2)).\
            astype("int")
    if (i == (len(r_peaks[0]) - 1)):
        big_matrix.append(matrix[:,start:])
    else:
        end = (r_peaks[0][i] + ((r_peaks[0][i+1] - r_peaks[0][i])/2)).astype(
            "int")
        big_matrix.append(matrix[:,start:end])
big_matrix.append(matrix[:,end:])

# frameon=False stops the figure frame from being drawn
fig = plt.figure(frameon=False)

# Set to an appropriate size for displaying the ECG data
fig.set_figheight(20)
fig.set_figwidth(3)

plt.subplots_adjust(0,0,1,1,0,0)

# Create a directory for the patient if it doesn't already exist
dir_string = dirpath.split("/")[1]
filename = file.split(".")[0]
if not os.path.exists(image_dir + dir_string + "/" + filename):
    os.mkdir(image_dir + dir_string + "/" + filename)
# Plot all beats and process/save as their own files within patient
# directory
for i in range(len(r_peaks[0])):
    print("printing beat: ", i)

```

```

for j in range(num_leads):
    plot(big_matrix[i][0], big_matrix[i][j+1], j+1)

# Use matplotlib to save the figure that was plotted to a png file
fig.savefig(image_dir + dir_string + "/" + filename + "/" + filename +
            "_beat" + str(i) + ".png", dpi=500, bbox_inches='tight')
# Read the file into OpenCV
image = cv2.imread(image_dir + dir_string + "/" + filename + "/" +
                    filename + "_beat" + str(i) + ".png")

# Crop image to remove white space on the left and right edges and
# convert to grayscale
if (image.shape[1] >= 1600):
    cropped_image = image[:, 115:-115]
    gray_image = cv2.cvtColor(cropped_image, cv2.COLOR_BGR2GRAY)
else:
    gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Down-sample image - 25% of width and maintaining aspect ratio for the
# height so the image doesn't get squished
(height, width) = gray_image.shape
resized = cv2.resize(gray_image, (math.floor(width/4), math.floor(math.
    floor(width/4)/(width/height)))))

# Write the processed file back
cv2.imwrite(image_dir + dir_string + "/" + filename + "/" + filename +
            "_beat" + str(i) + ".png", resized)
plt.clf()
plt.close()

```

E Python Image Filtering Code

```
import numpy as np
import cv2
from os import walk
import os

data_dir = "data2/"
stable_dir = "data2/stable"
unstable_dir = "data2/unstable"

print("Getting data...")
for (dirpath, dirnames, filenames) in walk(data_dir):
    # Are there files that aren't directories inside dirpath?
    if filenames:
        # For each file in this current dirpath
        for file in filenames:
            # Just in case
            if file == ".DS_Store":
                continue
            # Want to make sure the file we're reading is a .png file
            if file.split('.')[1] != "png":
                continue
            # Dont want to recursively filter
            if (dirpath.split('\\')[-1] == "filtered"):
                continue
            if (os.path.exists(dirpath + "/filtered/" + file)):
                continue
            print("reading file: ", dirpath + "/" + file)
            img = cv2.imread(dirpath + "/" + file, cv2.IMREAD_GRAYSCALE)
            # Discovered that the size can be reduced further using INTER_AREA
            # interpolation
            img_resized = cv2.resize(img, (180, 900), interpolation=cv2.INTER_AREA)
            # Padded image 184x904
            img_padded = cv2.copyMakeBorder(img_resized, 2, 2, 2, 2, cv2.BORDER_CONSTANT,
                                           value=[255, 255, 255])
            # The size reduction blurs the image quite substantially so a sharpening kernel
            # is applied to the image to improve the quality
            kernel = np.array([[-1, -1, -1], [-1, 9, -1], [-1, -1, -1]])
            img_filtered = cv2.filter2D(img_padded, -1, kernel)
```

```
if not os.path.exists(dirpath + "/filtered/"):
    os.makedirs(dirpath + "/filtered/")
print("writing file to: ", dirpath + "/filtered/" + file)
cv2.imwrite(dirpath + "/filtered/" + file, img_filtered)
```

F 2D CNN Python Code

```
#!/usr/bin/env python
import keras
from keras.models import Sequential
from keras.layers import *
from keras import regularizers
from keras.callbacks import *
from keras import backend as K
from keras.utils import multi_gpu_model
import numpy as np
from os import walk
from sklearn.model_selection import train_test_split
import matplotlib.image as mpimg

data_dir = "data/"
stable_dir = "data/stable"
unstable_dir = "data/unstable"
img_width = 184
img_height = 904

train_size_percent = 0.9 # % of data to split into training + validation data
val_split = 0.2 # % of training data to split into validation data

KERNEL = (3, 3)

# Higher batch size = less training time = get results faster
BATCH_SIZE = 32
EPOCHS = 1000

# Stable = 0
# Unstable = 1

# Callback for calculating sensitivity of prediction
def sensitivity(y_true, y_pred):
    true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
    possible_positives = K.sum(K.round(K.clip(y_true, 0, 1)))
    return true_positives / (possible_positives + K.epsilon())

# Callback for calculating specificity of prediction
```

```

def specificity(y_true, y_pred):
    true_negatives = K.sum(K.round(K.clip((1-y_true) * (1-y_pred), 0, 1)))
    possible_negatives = K.sum(K.round(K.clip(1-y_true, 0, 1)))
    return true_negatives / (possible_negatives + K.epsilon())

# Callback for monitoring test accuracy over the epochs
class testing_callback(keras.callbacks.Callback):
    def __init__(self, test_data):
        self.test_data = test_data
        self.test_history = np.empty((0, 5))

    def on_epoch_end(self, epoch, logs={}):
        x, y = self.test_data
        loss, acc, sens, spec = self.model.evaluate(x, y, verbose=0)
        self.test_history = np.append(self.test_history, np.array([[epoch, loss, acc,
            sens, spec]]), axis=0)
        filename = 'test_history.txt'

        if os.path.exists(filename):
            append_write = 'a' # append if already exists
        else:
            append_write = 'w' # make a new file if not

        hist = open(filename, append_write)
        hist.write('{}, {}, {}, {}, {}]\n'.format(epoch, loss, acc, sens, spec))
        hist.close()
        print('\nTesting loss: {}, acc: {}, sensitivity: {}, specificity: {}{}\n'.format(
            loss, acc, sens, spec))

# Callback for monitoring training accuracy over the batches
class per_batch_accuracy(keras.callbacks.Callback):
    def __init__(self):
        self.acc = []

    def on_batch_end(self, batch, logs={}):
        filename = 'accuracy_hist.txt'

        if os.path.exists(filename):
            append_write = 'a' # append if already exists
        else:

```

```

append_write = 'w' # make a new file if not

hist = open(filename, append_write)
hist.write('{}, {}, {}, {}, {}]\n'.format(batch, logs.get('loss'), logs.get('acc'),
    ), logs.get('sensitivity'), logs.get('specificity')))
hist.close()
print('Training loss: {}, acc: {}, sens: {}, spec: {}]\n'.format(logs.get('loss'),
    logs.get('acc'), logs.get('sensitivity'), logs.get('specificity')))

# This function will get training and testing data
def get_data():
    X = []
    Y = np.array([])
    i = 0
    print("Getting data...")
    # Search directories to find stable/unstable data and put them into an array
    for (dirpath, dirnames, filenames) in walk(data_dir):
        # Are there files that aren't directories inside dirpath?
        if filenames:
            # For each file in this current dirpath
            for file in filenames:
                # Just in case
                if file == ".DS_Store":
                    continue
                # Want to make sure the file we're reading is a .png file
                if file.split('.')[1] != "png":
                    continue
                # Values are [0,1] by default
                img = mpimg.imread(dirpath + "/" + file)
                print("Got file: ", file)
                # X is a list of np.array objects
                if (dirpath == stable_dir):
                    # All filenames in here are stable images
                    Y = np.append(Y, 0)
                    X.append(np.array(img))
                elif (dirpath == unstable_dir):
                    # All filenames in here are unstable images
                    Y = np.append(Y, 1)
                    X.append(np.array(img))

```

```

print("Running dstack . . .")
# Make X a 3D array (depth is each image)
X = np.dstack(X)
print("After d stack")
X_new = X.transpose(2,0,1)

x_train, x_test, y_train, y_test = train_test_split(X_new, Y, test_size = 1 -
    train_size_percent, train_size = train_size_percent)

# CNN needs the layer to be (number of images, height, width, 1)
x_train = np.expand_dims(x_train, axis=3)
x_test = np.expand_dims(x_test, axis=3)
print("Got all data!")
# X = array of images, depth is how you select the image
# Y = classification of image [0,1]
return x_train, x_test, y_train, y_test

def build_model():
    cnn = Sequential()
    cnn.add(Conv2D(32, KERNEL, input_shape = (img_height, img_width, 1),
        kernel_initializer='glorot_uniform'))
    cnn.add(keras.layers.LeakyReLU(alpha=0.1))
    cnn.add(BatchNormalization())

    cnn.add(Conv2D(32, KERNEL, strides=(1, 1), kernel_initializer='glorot_uniform'))
    cnn.add(keras.layers.LeakyReLU(alpha=0.1))
    cnn.add(BatchNormalization())

    cnn.add(MaxPool2D(pool_size=(2, 2), strides=(2, 2)))

    cnn.add(Conv2D(64, KERNEL, strides=(1, 1), kernel_initializer='glorot_uniform'))
    cnn.add(keras.layers.LeakyReLU(alpha=0.1))
    cnn.add(BatchNormalization())

    cnn.add(Conv2D(64, KERNEL, strides=(1, 1), kernel_initializer='glorot_uniform'))
    cnn.add(keras.layers.LeakyReLU(alpha=0.1))
    cnn.add(BatchNormalization())

    cnn.add(MaxPool2D(pool_size=(2, 2), strides=(2, 2)))

```

```

cnn.add(Conv2D(128, KERNEL, strides=(1, 1), kernel_initializer='glorot_uniform'))
cnn.add(keras.layers.LeakyReLU(alpha=0.1))
cnn.add(BatchNormalization())

cnn.add(Conv2D(128, KERNEL, strides=(1, 1), kernel_initializer='glorot_uniform'))
cnn.add(keras.layers.LeakyReLU(alpha=0.1))
cnn.add(BatchNormalization())

cnn.add(MaxPool2D(pool_size=(2, 2), strides=(2, 2)))

cnn.add(Flatten())

cnn.add(Dense(256, kernel_regularizer=regularizers.l1(0.001)))
cnn.add(keras.layers.LeakyReLU(alpha=0.1))
cnn.add(BatchNormalization())
cnn.add(Dropout(0.4))

cnn.add(Dense(256, kernel_regularizer=regularizers.l1(0.001)))
cnn.add(keras.layers.LeakyReLU(alpha=0.1))
cnn.add(BatchNormalization())
cnn.add(Dropout(0.4))

cnn.add(Dense(1, activation='sigmoid'))

return cnn

print("Building model...")
model = build_model()

parallel_model = multi_gpu_model(model, gpus=4)

print("Compiling optimizer...")
# optimizer can change to rmsprop
parallel_model.compile(
    loss='binary_crossentropy',
    optimizer='adam',
    metrics=['accuracy', sensitivity, specificity])
parallel_model.summary()

```

```

print("Getting data...")
X_train, X_test, Y_train, Y_test = get_data()

csv_log_cb = CSVLogger("training.csv", append=True)
training_acc_cb = per_batch_accuracy()

early_stop_cb = EarlyStopping(monitor='val_loss', min_delta=0, patience=200, verbose=1,
    mode='min', baseline=None, restore_best_weights=False)

model_save_filepath = "model-best.hdf5"
model_save_cb = ModelCheckpoint(model_save_filepath, monitor='val_loss', verbose=1,
    save_best_only=True, mode='min')
test_hist = testing_callback((X_test, Y_test))

model_hist = parallel_model.fit(X_train, Y_train, batch_size=BATCH_SIZE, epochs=EPOCHS,
    validation_split=val_split, verbose=1, shuffle=True, callbacks=[training_acc_cb,
    model_save_cb, csv_log_cb, test_hist, early_stop_cb])
print(model_hist.history)

print("Testing history: ")
print(test_hist.test_history)

```

G 2D CNN Prediction Testing Python Code

```
#!/usr/bin/env python
import keras
from keras.models import Sequential
from keras.layers import *
from keras import regularizers
from keras.callbacks import *
import time
import numpy as np
from os import walk
from sklearn.model_selection import train_test_split
import matplotlib.image as mpimg
from keras.models import model_from_json

data_dir = "predict-data/"
stable_dir = "predict-data/stable"
unstable_dir = "predict-data/unstable"

img_width = 184
img_height = 904
KERNEL = (3, 3)
# This function will get training and testing data
def predict(predict_model):
    X = []
    total_correct = 0
    N = 0
    fneg = 0
    fpos = 0
    tpos = 0
    tneg = 0
    total_time = 0
    print("Getting data...")
    for (dirpath, dirnames, filenames) in walk(data_dir):
        # Are there files that aren't directories inside dirpath?
        if filenames:
            # For each file in this current dirpath
            for file in filenames:
                # Just in case
                if file == ".DS_Store":
```

```

continue
# Want to make sure the file we're reading is a .png file
if file.split('.')[1] != "png":
    continue
# print("reading file: ", dirpath + "/" + file)
# Values are [0,1] by default
img = mpimg.imread(dirpath + "/" + file)
print("Got file: ", file)
print("From: ", dirpath)
# X is a list of np.array objects
X = np.array(img)
X = np.dstack(X)
X_new = X.transpose(2,1,0)
X_new = np.expand_dims(X_new, axis=0)
start = time.time()
prediction = predict_model.predict(X_new, verbose=0)
end = time.time()

time_taken = end - start
total_time += time_taken

print("Prediction time: ", time_taken, "s")
if (prediction > 0.5):
    pred_class = 1
else:
    pred_class = 0
out_class = 0
if (dirpath == stable_dir):
    out_class = 0
elif (dirpath == unstable_dir):
    out_class = 1

if pred_class == 0:
    if out_class == 0:
        tneg += 1
        total_correct += 1
    elif out_class == 1:
        fneg += 1
elif pred_class == 1:
    if out_class == 0:

```

```

        fpos += 1
    elif out_class == 1:
        tpos += 1
        total_correct += 1
    print('Actual class: {}, predicted class: {}{}'.format(out_class,
        pred_class))
    print('True prediction: {}'.format(prediction))
    N += 1
    # X.append(np.array(img))
# Make X a 3D array (depth is each image)
print('Total time taken: {}s'.format(total_time))
print('Total correct predictions: {} out of {} images\n'.format(total_correct, N))
print('False positives: {}; False negatives: {}{}'.format(fpos, fneg))
print('True positives: {}; True negatives: {}{}\n'.format(tpos, tneg))
print('Sensitivity = {}; Specificity = {}; Accuracy = {}{}\n'.format(
    (tpos/(tpos+fneg)) * 100,/
    (tneg/(tneg + fpos)) * 100, ((tpos+tneg)/(tpos+tneg+fpos+fneg)) *
    100))

def build_model():
    cnn = Sequential()
    cnn.add(Conv2D(32, KERNEL, input_shape = (img_height, img_width, 1),
        kernel_initializer='glorot_uniform'))
    cnn.add(keras.layers.LeakyReLU(alpha=0.1))
    cnn.add(BatchNormalization())

    cnn.add(Conv2D(32, KERNEL, strides=(1, 1), kernel_initializer='glorot_uniform'))
    cnn.add(keras.layers.LeakyReLU(alpha=0.1))
    cnn.add(BatchNormalization())

    cnn.add(MaxPool2D(pool_size=(2, 2), strides=(2, 2)))

    cnn.add(Conv2D(64, KERNEL, strides=(1, 1), kernel_initializer='glorot_uniform'))
    cnn.add(keras.layers.LeakyReLU(alpha=0.1))
    cnn.add(BatchNormalization())

    cnn.add(Conv2D(64, KERNEL, strides=(1, 1), kernel_initializer='glorot_uniform'))
    cnn.add(keras.layers.LeakyReLU(alpha=0.1))
    cnn.add(BatchNormalization())

```

```

cnn.add(MaxPool2D(pool_size=(2, 2), strides=(2, 2)))

cnn.add(Conv2D(128, KERNEL, strides=(1, 1), kernel_initializer='glorot_uniform'))
cnn.add(keras.layers.LeakyReLU(alpha=0.1))
cnn.add(BatchNormalization())

cnn.add(Conv2D(128, KERNEL, strides=(1, 1), kernel_initializer='glorot_uniform'))
cnn.add(keras.layers.LeakyReLU(alpha=0.1))
cnn.add(BatchNormalization())

cnn.add(MaxPool2D(pool_size=(2, 2), strides=(2, 2)))
cnn.add(MaxPool2D(pool_size=(2, 2), strides=(2, 2)))
cnn.add(MaxPool2D(pool_size=(2, 2), strides=(2, 2)))

cnn.add(Flatten())

cnn.add(Dense(256, kernel_regularizer=regularizers.l1(0.001)))
cnn.add(keras.layers.LeakyReLU(alpha=0.1))
cnn.add(BatchNormalization())
cnn.add(Dropout(0.4))

cnn.add(Dense(256, kernel_regularizer=regularizers.l1(0.001)))
cnn.add(keras.layers.LeakyReLU(alpha=0.1))
cnn.add(BatchNormalization())
cnn.add(Dropout(0.4))

cnn.add(Dense(1, activation='sigmoid'))

return cnn
model = build_model()
# my-weights.h5 is a file containing the learned weights generated by the training process
model.load_weights('my-weights.h5')
predict(model)

```

H Raspberry Pi 3B+ Testing Output

```
Got file: ptbdb_s0531_re_beat55.png
From: predict-data/stable
Prediction time: 10.737799882888794 s
Actual class: 0, predicted class: 0
Actual prediction: [[0.00052029]]
```

```
Got file: staffiii_105a_beat9.png
From: predict-data/stable
Prediction time: 5.126259088516235 s
Actual class: 0, predicted class: 0
Actual prediction: [[0.00668119]]
```

```
Got file: ptbdb_s0470_re_beat7.png
From: predict-data/stable
Prediction time: 5.1545798778533936 s
Actual class: 0, predicted class: 0
Actual prediction: [[6.5975655e-05]]
```

```
Got file: ptbdb_s0476_re_beat9.png
From: predict-data/stable
Prediction time: 5.1439433097839355 s
Actual class: 0, predicted class: 0
Actual prediction: [[0.00010813]]
```

```
Got file: ptbdb_s0466_re_beat59.png
From: predict-data/stable
Prediction time: 5.147115707397461 s
Actual class: 0, predicted class: 0
Actual prediction: [[2.0544196e-05]]
```

```
Got file: ptbdb_s0552_re_beat19.png
From: predict-data/stable
Prediction time: 5.151220321655273 s
Actual class: 0, predicted class: 0
Actual prediction: [[0.00254849]]
```

```
Got file: staffiii_105d_beat4.png
From: predict-data/stable
```

Prediction time: 5.180302619934082 s
Actual class: 0, predicted class: 0
Actual prediction: [[0.06581482]]

Got file: ptbdb_s0363lre_beat11.png
From: predict-data/stable
Prediction time: 5.108787536621094 s
Actual class: 0, predicted class: 0
Actual prediction: [[0.00079428]]

Got file: ptbdb_s0551_re_beat6.png
From: predict-data/stable
Prediction time: 5.217668294906616 s
Actual class: 0, predicted class: 0
Actual prediction: [[1.5471609e-05]]

Got file: ptbdb_s0481_re_beat7.png
From: predict-data/stable
Prediction time: 5.084589004516602 s
Actual class: 0, predicted class: 0
Actual prediction: [[0.00097541]]

Got file: ptbdb_s0273lre_beat12.png
From: predict-data/stable
Prediction time: 5.298357248306274 s
Actual class: 0, predicted class: 0
Actual prediction: [[0.00179965]]

Got file: ptbdb_s0532_re_beat9.png
From: predict-data/stable
Prediction time: 4.910542964935303 s
Actual class: 0, predicted class: 0
Actual prediction: [[0.00058063]]

Got file: staffiii_044b_beat1.png
From: predict-data/stable
Prediction time: 5.182345628738403 s
Actual class: 0, predicted class: 1
Actual prediction: [[0.67608213]]

Got file: staffiii_011c_beat2.png
From: predict-data/stable
Prediction time: 5.128925085067749 s
Actual class: 0, predicted class: 0
Actual prediction: [[0.2525279]]

Got file: staffiii_064d_beat6.png
From: predict-data/stable
Prediction time: 5.170924663543701 s
Actual class: 0, predicted class: 0
Actual prediction: [[0.01608829]]

Got file: staffiii_065d_beat4.png
From: predict-data/unstable
Prediction time: 5.119891881942749 s
Actual class: 1, predicted class: 1
Actual prediction: [[0.99999714]]

Got file: staffiii_019b_beat6.png
From: predict-data/unstable
Prediction time: 5.154499530792236 s
Actual class: 1, predicted class: 1
Actual prediction: [[0.9999994]]

Got file: ptbdb_s0126lre_beat7.png
From: predict-data/unstable
Prediction time: 5.123972654342651 s
Actual class: 1, predicted class: 1
Actual prediction: [[0.99999976]]

Got file: staffiii_044d_beat3.png
From: predict-data/unstable
Prediction time: 5.135965824127197 s
Actual class: 1, predicted class: 1
Actual prediction: [[0.882759]]

Got file: staffiii_051b_beat3.png
From: predict-data/unstable
Prediction time: 5.146641731262207 s
Actual class: 1, predicted class: 1

```
Actual prediction: [[0.99985707]]  
  
Got file: staffiii_098d_beat2.png  
From: predict-data/unstable  
Prediction time: 5.100493431091309 s  
Actual class: 1, predicted class: 1  
Actual prediction: [[0.9999261]]  
  
Got file: ptbdb_s0234lre_beat6.png  
From: predict-data/unstable  
Prediction time: 5.185266494750977 s  
Actual class: 1, predicted class: 1  
Actual prediction: [[0.99986875]]  
  
Got file: staffiii_026c_beat1.png  
From: predict-data/unstable  
Prediction time: 5.217010974884033 s  
Actual class: 1, predicted class: 1  
Actual prediction: [[0.9999945]]  
  
Got file: staffiii_037b_beat5.png  
From: predict-data/unstable  
Prediction time: 5.224578857421875 s  
Actual class: 1, predicted class: 1  
Actual prediction: [[0.9999218]]  
  
Got file: staffiii_084a_beat4.png  
From: predict-data/unstable  
Prediction time: 5.101965665817261 s  
Actual class: 1, predicted class: 1  
Actual prediction: [[0.99994326]]  
  
Got file: staffiii_008f_beat11.png  
From: predict-data/unstable  
Prediction time: 5.238185167312622 s  
Actual class: 1, predicted class: 1  
Actual prediction: [[0.99999785]]  
  
Got file: staffiii_107d_beat7.png  
From: predict-data/unstable
```

```
Prediction time: 5.1105732917785645 s
Actual class: 1, predicted class: 1
Actual prediction: [[0.9999995]]
```

```
Got file: ptbdb_s0395lre_beat5.png
From: predict-data/unstable
Prediction time: 5.1950154304504395 s
Actual class: 1, predicted class: 1
Actual prediction: [[0.99873155]]
```

```
Got file: ptbdb_s0030_re_beat4.png
From: predict-data/unstable
Prediction time: 5.19695520401001 s
Actual class: 1, predicted class: 1
Actual prediction: [[0.9999994]]
```

```
Got file: ptbdb_s0001_re_beat6.png
From: predict-data/unstable
Prediction time: 5.203875541687012 s
Actual class: 1, predicted class: 1
Actual prediction: [[0.99735993]]
```

```
Total time taken: 160.19825291633606 s
Total correct predictions: 29 out of 30 images
```

```
False positives: 1; False negatives: 0
```

```
True positives: 15; True negatives: 14
```

```
Sensitivity = 100.0
Specificity = 93.33333333333333
Accuracy = 96.66666666666667
```
