International Journal on Engineering, Science and Technology
2022, Volume 4, Issue 4 ISSN: 2642-4088 (Online)

Remote Heart Diagnostic System

Emma Miller
Trinity University, United States, https://orcid.org/0000-0003-3087-1993

Edward Armitage
Trinity University, United States, https://orcid.org/0000-0003-4341-2614

Farzan Aminian
Trinity University, United States, https://orcid.org/0000-0002-1050-4611

Darin George
Trinity University, United States, https://orcid.org/0000-0002-5339-8986

Ashkan Aminian
University of Texas at San Antonio (UTSA), United States, https://orcid.org/0000-0001-6940-5645

Komron Aminian
University of Texas Health Science Center at San Antonio, United States, https://orcid.org/0000-0002-9901-6912

Mehran Aminian
St. Mary's University, United States, https://orcid.org/0000-0002-5146-1697

Abstract: For people in remote regions, limited access to specialists like cardiologists can turn treatable conditions into a serious threat. In this work, a heart diagnostic system capable of collecting and analyzing an electrocardiogram (ECG) signal in a remote location and sending the results to be reviewed by a cardiologist is designed and implemented. This system is controlled using a Raspberry Pi 4 microcomputer, with a physical subsystem designed to collect the ECG signal. This signal is utilized to diagnose certain heart conditions using an Artificial Intelligence (AI) based system. A touchscreen mini-terminal guides the user through collecting and compiling patient data, sending this data to the AI agent for classification, and then storing and transferring the results to another location for further examination. A user-friendly website is implemented to give the health professionals remote access to the collected data. The entire system is housed in a 3-D printed enclosure for protection. It is important to note that the AI system is trained using multiple data sets obtained from Physionet, as the collection of training data using this system is a multiyear project requiring a staff of health professionals, which is beyond the scope of this project.

Keywords: Engineering, Heart diagnosis, Neural networks

Introduction

Despite extensive modern advancements in healthcare, heart conditions continue to plague the world population, with heart disease being the primary cause of death in the United States for most demographic groups (Centers for Disease Control and Prevention, 2022). Furthermore, the Centers for Disease Control and Prevention estimate that 12.1 million people will have atrial fibrillation in the United States by 2030 (Centers for Disease Control and Prevention, 2021). Many heart conditions or risk factors can be alleviated if identified in the early stages, preventing life-threatening consequences.

However, due to high cost, limited availability, and long distances between patients and specialist facilities, these conditions can go unnoticed until they become fatal. For example, between 2016 and 2017, the United States lost approximately $363 billion dollars per year towards heart disease in the form of medical services, medication, and lost labor hours (Centers for Disease Control and Prevention, 2022). While heart disease is a significant strain on the US, impoverished nations and those without healthcare or stable employment cannot afford these costs, leading to a lack of proper medical care for a life-threatening condition. By reducing the travel and time expenses for cardiologists and patients through semi-automated remote care, this device seeks to increase the accessibility of specialized cardiac healthcare for hard-to-access and impoverished populations.

The objective of this project was to develop a proof-of-concept prototype capable of providing remote care for diagnosing heart disease using artificial intelligence and networking technologies. The prototype is capable of remotely collecting and analyzing an electrocardiogram (ECG) signal and transferring the recorded data to a cardiologist in a different location for final examination. The AI system is trained to diagnose normal sinus rhythm, atrial flutter, atrial fibrillation, myocardial infarction, and congestive heart failure with at least a 90% accuracy rate in validation testing. This device can also record the ECG signal with minimal noise and at least 5 full cardiac propagations to avoid undermining the accuracy of the AI algorithms.

Furthermore, the networking component operates between the device, database, and website with loading times under 5 minutes and a 90% accuracy rate. For cardiologists to be able to accurately confirm the diagnoses made by the AI system, the system must be able to replicate commercial ECG signals with high accuracy. Finally, the device is portable and durable enough to be operated in remote regions. Specifically, it is designed to fit within a carry-on suitcase matching the dimensions specified by most airlines, and to survive a drop of 4 feet.

To record the ECG signal and reduce noise, a circuit consisting of a signal monitor and power filter, along with an analog to digital converter to read the signal, is employed. The AI component of the system was designed as a convolutional neural network (Saha, 2018) using an image rendering of each ECG signal to perform classification. A relational cloud database accessed over Wi-Fi serves as the information transfer subsystem to ensure short loading times, data preservation, and constantly accessible connections. The full design is described in the next section.

Method

System Overview

The full system process begins with the collection of the patient's data and concludes with the display of the relevant information to a doctor via the website. When started, the device directs the nurse to enter the patient's information such as birth date and phone number using the touchscreen, attach the three leads to the patient, and then trigger an ECG reading using a virtual button. The ECG reading is then displayed on the device to allow the nurse to either save the reading or retake it if necessary. If saved, the ECG is analyzed using a deep learning algorithm to predict possible heart conditions, and then the ECG and predictions are stored in a remote database using Wi-Fi. A doctor, once registered by the website administrator, can then search for patients by name and view their ECG signal and either confirm or dispute the diagnosis. In this manner, not only will ECG analysis take less time for the doctors, as the diagnostic results could be used to identify patients with higher risk for more immediate analysis, but patients will be able to get an accurate ECG taken without visiting a specialist's office in person.

To describe the development process, the next subsections split the design into five categories, including data collection, signal analysis, information transfer, display, and casing.

Data Collection

To ensure that a proper diagnosis is made for the correct individual, the prototype must collect not only the patient's ECG signal, but also their biographical data. Consequently, a touchscreen interface system has been designed to guide the user through each step of the data collection procedure. The user interface was developed in Python using the Tkinter graphical user interface library on a Raspberry Pi 4 with a Raspberry Pi 7-inch touchscreen attached (Python, n.d.). A Raspberry Pi was selected due to its large quantities of open-source code and its versatility. It serves as the brain of the system, collecting and analyzing the ECG signal and storing it in the database. A simplified flow chart of the system is shown in Fig. 1.

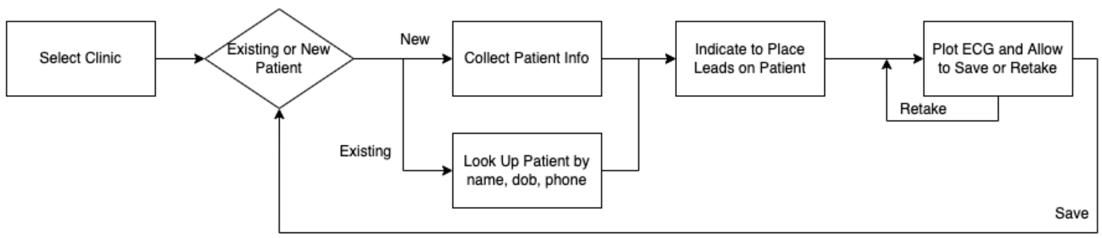

Figure 1. User Interface Software Flow Diagram

Through the graphical user interface, the unit collects the clinic's assigned identification number, and then allows the nurse to either look up an existing patient by name, date of birth, and phone number, or to record data on a new patient with name, gender, date of birth, phone number, and address. This data allows the doctor to access previous ECG signals for comparison and reach out to the patient with any findings. Once the patient information has been collected, the interface triggers the collection of an ECG using the circuit designed for this purpose and plots the recorded ECG for the nurse's approval. Finally, the program returns to the start, so that the device can be immediately used on another patient.

The ECG signal is collected via three leads connected to the patient in a standard triangular configuration, specifically right chest, left chest, and right abdomen. The three leads feed into a heart rate monitor integrated circuit (AD8232), and the resulting signal is converted to a digital signal using an analog to digital converter (MCP3008). The digital signal is then passed to the Raspberry Pi, as the Raspberry Pi cannot accept analog inputs. A filter is included in the form of two grounded capacitors between the 3.3 V output of the Raspberry Pi and the power inputs for the rest of the circuit, as it was discovered in testing that noise in the wall power outlet could transfer through the circuit and lead to significant noise in the recorded ECG signal. For the final prototype, the circuit was transitioned from a portable breadboard to a printed circuit board (PCB). The PCB was etched using a Bantam Tools Desktop PCB Milling Machine in the Trinity University Makerspace based on a schematic created in Eagle, and all the relevant components were either soldered directly into the PCB or attached via wires. The schematic for this circuit is included as Fig. 2.

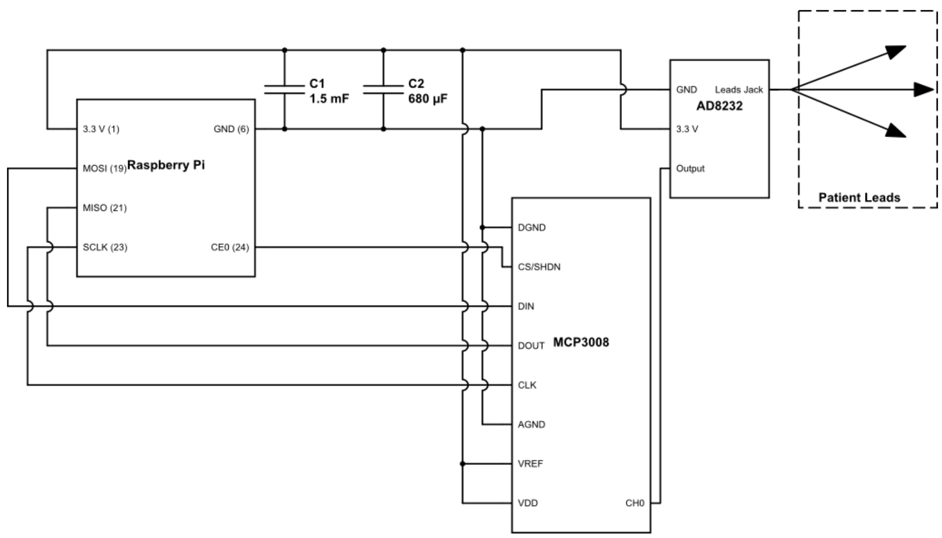

Figure 2. Simplified Schematic of the Electrocardiogram Circuit

A key requirement governing this subsystem is that the ECG circuit must record a signal with at least 5 full cardiac propagations and enough detail to be analyzed by the AI algorithms. To meet this requirement, the software collects a full three-minute ECG signal, and a filter was added to ensure that excessive noise would not prevent an accurate diagnosis. Other restrictions on the circuit design included size constraints on the components, and the safety requirements imposed by the Trinity University Institutional Review Board for testing on human subjects. Additionally, Transportation Security Administration and airline requirements for carry-on dimensions and battery types were considered in the data collection design.

Signal Analysis

As mentioned before, a convolutional neural network (CNN) was used to classify electrocardiograms for the selected heart conditions. A CNN accepts a structured data array, such as an image, as input and performs feature extraction to identify key components of the image. The image is then classified based on the identified features using multilayer neural networks (IBM Cloud Education, 2020). The weights and biases are initially randomized and then developed through training prior to the deployment of the network for classifications. In training, a series of input images are provided to the network, and for each image, the output is predicted and then compared to the expected output from human labeling, and the differences between the results and the intended prediction are then back propagated through the algorithm to update the weights and biases using a user-selected learning rate and a gradient descent function to minimize the loss, which in turn increases the accuracy.

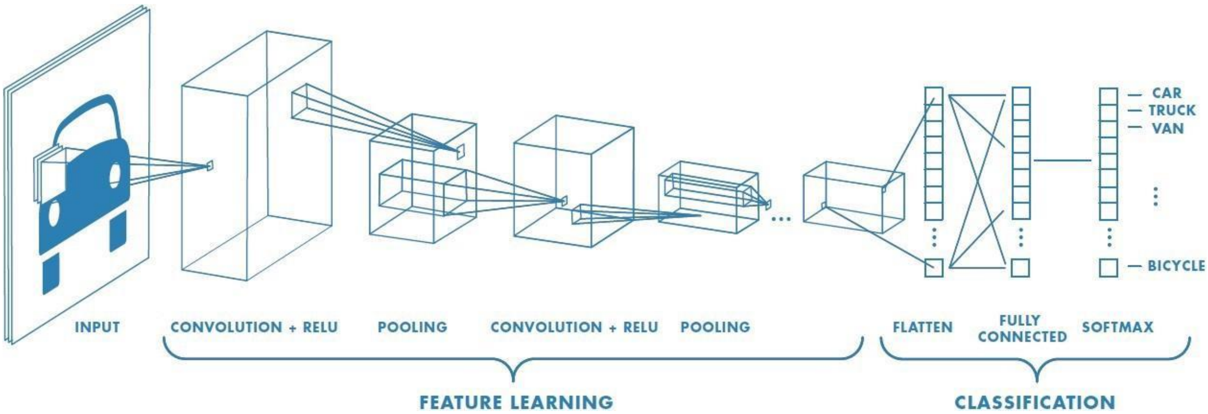

Figure 3. Diagram of a Convolutional Neural Network (Saha, 2018)

As displayed in Fig. 3, a convolutional neural network consists mainly of two parts, feature learning and classification. The first type of layer is the convolutional layer, in which a 2-dimensional matrix, known as a kernel, with values chosen to extract important features such as edges, is stepped across the input and a dot product is performed between the kernel and the overlapped subsection of the input, until this procedure has been performed for the entirety of the input matrix. This process produces a feature map, which is often reduced in size compared to the input matrix. The final step in this layer is the application of a Rectified Linear Unit (ReLU) transformation to the feature map to provide non-linearity to the layers. The ReLU function returns 0 if the input is below zero, otherwise it returns the input.

The next key layer in a convolutional neural network is the pooling layer, which reduces the dimensions of the matrix being passed through the network to improve efficiency and reduce computational complexity (Saha, 2018). To do so, a kernel is traversed through the matrix, but instead of performing a dot product with weights in the kernel matrix, an aggregation function is applied to the section of the input covered by the dimensions of the kernel. Typically, the pooling layer either computes the maximum or the average of the values in each region as it traverses, producing a reduced-size output array. The convolutional and pooling layers can then both be repeated as many times as desired to refine the feature extraction in the network.

Once the convolutional and pooling processes are fully completed, the output of these layers is passed to the fully connected layer to perform the classification. First, the input is flattened to produce a column vector (Saha, 2018). This vector is then passed to a neural network, which consists of a series of hidden layers of nodes that activate, return 1, or do not activate, return 0, based on the inputs from the previous layer, the trained weights and biases, and an activation function, such as the ReLU function. The neural network concludes with an output layer that functions similarly to the hidden layers, but often employs a different activation function. For convolutional neural networks, the activation function typically used in the output layer is the softmax function, *Eq. (1)*, which converts the numerical inputs into probabilities to represent the likelihood of the image matching each classification category (Saha, 2018). In *Eq. (1)*, σ represents the softmax function, z represents the vector input, K is the number of classification options, the i subscript denotes input, and the j subscript indicates the output. The output of the layer with the softmax activation function is the result of the convolutional neural network and includes the probabilities that the image is a match for each potential class.

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad (1)$$

A drawback of convolutional neural networks is that they require significant quantities of labeled data to develop the weights and biases for the neural network layers in training. To overcome this issue, a strategy known as transfer learning can be applied, in which a pre-trained neural network for a similar task is retrained by changing only a few layers, such as the output layer, to adjust for the new potential classification options. The model can then be supplied with a smaller set of labeled training data to readjust the weights, as they have been partially pre-trained. However, too small of a training dataset can lead to the AI system memorizing the data without learning its underlying features, leading to poor classification of the testing and validation data.

For this project, transfer learning was utilized to retrain GoogLeNet to classify ECG signals for the selected five conditions. GoogLeNet is a convolutional neural network designed for image classification, with a base architecture like that described above, but with some additional features to improve accuracy and reduce the likelihood of overfitting (Alake, 2021). The GoogLeNet convolutional neural network was retrained to classify true color images of plotted electrocardiograms by changing the output layer along with key training parameters. The inspiration and base code for the training program was derived from the MATLAB example, “Classify Time Series Using Wavelet Analysis and Deep Learning” (MathWorks, 2018). Initially, another image classification convolutional neural network, SqueezeNet, was also retrained, but GoogLeNet was selected as it produced a higher validation accuracy in all testing.

The training was developed using the MATLAB platform, and the final code was translated into a C++ script and executable, for speed and functionality on a Raspberry Pi. The datasets used are as follows: MIT-BIH Normal Sinus Rhythm Database (Goldberger et al., 2000), BIDMC Congestive Heart Failure Database (Goldberger et al., 2000; Baim et al, 1986), MIT-BIH Atrial Fibrillation Database (Goldberger et al., 2000; Moody & Mark, 1983), and PTB-XL (Goldberger et al., 2000; Wagner et al., 2020; Wagner et al., 2020).

The deployed network is called from the main program during typical device use. When the user chooses to save the ECG on the touchscreen interface, the prototype performs the signal analysis, and then saves the results in the database along with the full ECG signal. The signal analysis program takes the collected array of ECG data as input, plots the data over time, converts the plot into a true color image, and passes this image into the convolutional neural network for classification. The network then returns an array containing the probability that the patient has any of the five possible conditions included in the AI training.

Transmission of Information

All patient data and the AI results are transmitted from the device via Wi-Fi to a PostreSQL database hosted on Heroku. Heroku is a company providing cloud services that allows for limited free deployment of databases and websites (Heroku, 2022). The use of a cloud database provides easy connectivity and allows users to seamlessly access the data. Both the device and the website access the database through a wireless connection and Structured Query Language (SQL) commands. The database accesses on the device are performed using the "psycopg2" library for PostgreSQL in Python (Psycopg, 2021), and the database is accessed on the website using GET and POST requests. The database stores a variety of relevant fields related to the ECG and patient data, as well as key verification and account information. The structure of the deployed database can be viewed in Fig. 4.

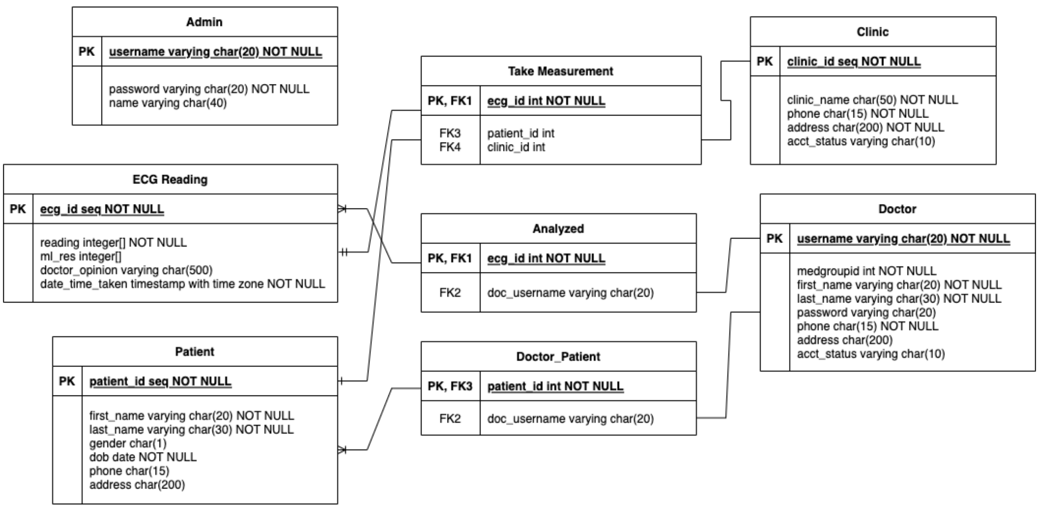

Figure 4. Diagram of the database structure

Display

The doctor-facing website was developed in JavaScript using the ReactJS markup language. Using the basic React libraries, a framework for a single web application was created. To transfer data to and from the database, an Application Programming Interface (API) was implemented using Microsoft's .NET framework, written in C#. The API makes HTTP requests to the deployed database. To pull data to the front-end of the application, a variety of web development libraries are employed. The most important of these libraries is the "react-chartjs-2" library, which allows the application to plot the data retrieved from the database as an array (react-chartjs-2, n.d.). The website is styled using Cascading Style Sheets (CSS). The website was locally built and tested in the Windows operating system using two separate folders and the Visual Studio Code Integrated Development Environment (IDE). All of the code was written and run in Visual Studio until the website was deployed on Heroku for widespread access using a link. The overall website architecture is shown in Fig. 5.

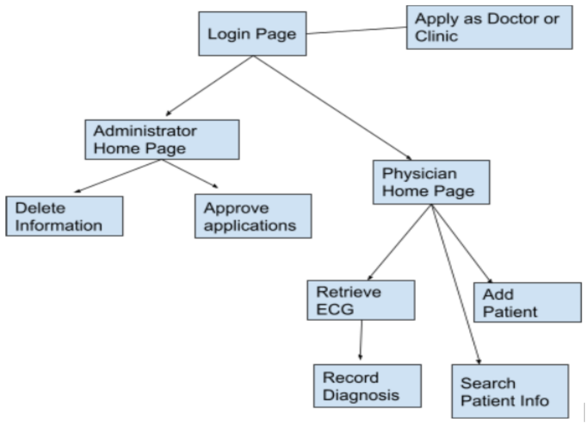

Figure 5. Website Architecture Diagram

Casing

The casing for the device was designed in Autodesk Fusion360 and printed on a 3D printer, a method preferred for its adaptability, low cost, and ease of use. In addition, 3D printed materials are relatively durable, and the high degree of measurement control in 3D printed design ensures that the circuit components are securely housed. The case was required to be large enough to firmly house the screen and the electronic components without exceeding the accepted carry-on dimension range, 22” by 14” by 9”. The case is modeled and printed as two different pieces: the case and the front panel. The CAD model for the casing is displayed in Fig. 6.

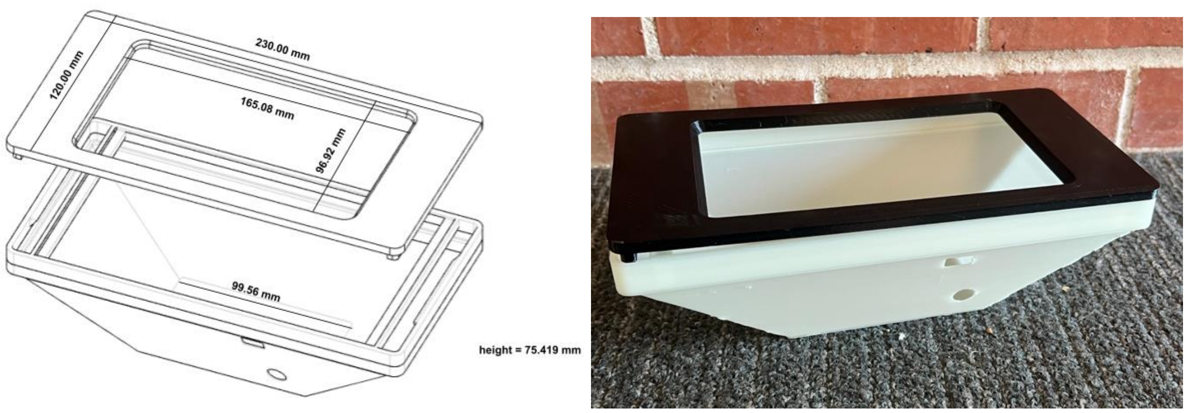

Figure 6. 3-D Computer-Aided Design Model and Photograph of Casing

The case houses the screen and the circuit. Two L-shaped mounts are precisely positioned to securely mount the touchscreen to the front panel, while providing for easy insertion and removal. The Raspberry Pi is attached with screws to the bottom of the touchscreen. The ECG collection circuit is placed at the bottom of the case, with room left between the Raspberry Pi and the circuit for proper connections. Two port-holes are included on one side of the case to allow both the power cord and ECG leads to be plugged into the circuit components.

The front panel and the large side of the case have the same dimensions and are attached to each other by a snap fit. The front panel is designed so that only the touch portion of the screen is visible, excluding the border from view to ensure a stronger hold on the screen. A slight downward curve along the inside borders of the front panel improves the accessibility of the corners of the touchscreen, as well as providing enhanced aesthetics.

To enhance durability and portability, the bottom portion of the case is printed utilizing Polylactic Acid (PLA) material which is 1.75mm thick using the Original Prusa i3 MK3 3D printer (Prusa Research, n.d.). The front panel is printed using 1.75mm ABS Nylon Black on an Ultimaker S5 (Ultimaker, n.d.).

Results and Conclusions

As discussed, the remote heart diagnostic system is designed to collect an ECG signal and transfer the data to a Raspberry Pi, where the collected signals are analyzed and the results are transferred to a database for storage and remote access. Eleven subjects were used to test this full system; the tests involved collecting, analyzing, and transmitting the data for each subject. To verify system performance, tests used an oscilloscope connected via a probe between the signal monitor (AD8232) and the analog to digital converter (MCP3008), three electrodes attached to a human subject, a monitor attached to the Raspberry Pi unit to read any collected data, and an Apple Watch worn by the subject to provide a commercial ECG signal for comparison. Each test subject was asked to complete a fake data entry on the touchscreen and then remain still for 3 minutes while an ECG was recorded. The use of fake patient information allowed the team to validate the full data collection process without violating the Institutional Review Board expectations that the data not be traced back to the test subject. The main acceptance criterion was that the noise visible in the electrocardiograms measured under reasonable conditions on the device microcomputer should match within a 10% margin of error the electrocardiogram produced by the commercial device and include 5 full cardiac propagations. In addition, no data from the patient data entry testing should be lost, and the model should diagnose at a minimum 90% of the input signals with the correct heart condition.

The patient data entry using the touchscreen functioned as expected. All personal data was collected without issue, and proper error handling measures for missing fields or improper data formats occurred as designed. The electrocardiogram recording was started and displayed after completion on the screen as expected and displayed visual similarities to the signal recording on the oscilloscope. In addition, the recording was properly passed to the artificial intelligence program, and both the recording and diagnostic results were saved in the database.

All data transfer attempts were successful in both transmitting and receiving data of the expected types with no lost data or delays in service. The distance in time and space between the submission and retrieval of data proved to have no effect on the access capabilities. All data transfer operations proceeded within milliseconds, with the longest database access for the website completed in 733.65 milliseconds. Furthermore, data submitted via the Raspberry Pi was accessible in well under a minute on the website, and vice versa. All attempts to access the database and the wireless connection were successful on both ends, indicating an 100% accuracy rate in connection on a stable wireless signal. As the prototype exceeded the accuracy, time, and distance expectations set forth in the requirements for the project and all data stored matched the data entered on the device, the design was considered successful both in this test and in the fulfillment of the wireless transmission requirements overall. This indicates that the wireless transmission process is fully capable of handling the data transfer operations necessary to achieve the primary objective, a device capable of assisting with remote automated healthcare for cardiology. To assess the accuracy of the ECG collection circuit, the signals recorded by the system and those collected by the Apple Watch were compared. Four key peaks relevant to ECG signals, namely the P, Q, R, and S deflections, were examined, as labeled in Fig. 7.

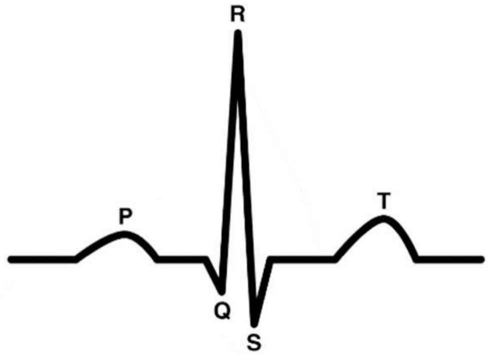

Figure 7. Labeled Diagram of Traditional ECG Waveform

An average height value for each peak from both recordings was estimated, and the two values were compared to compute a voltage correction coefficient and a percent difference, using *Eq. (2)*. The percent differences between the two values for each peak are shown in Table 1.

$$\%\ difference = \left|\frac{\text{Prototype peak height} - \text{Apple Watch peak height}}{\text{Apple Watch peak height}}\right| \times 100 \tag{2}$$

Table 1. Percent Differences Between Adjusted Peak Voltages from Apple Watch and Prototype

P (% diff.)	Q (% diff.)	R (% diff.)	S (% diff.)
10.6	6.7	29.3	5.9
13.6	14.6	43.9	1.2
46.3	23.3	39.3	18.6
5.3	16.4	12.6	12.2
15.5	14.8	35.7	14.7
22.9	52.8	12.0	7.2
0.3	23.7	40.5	11.1
10.7	5.9	11.3	1.1
18.6	24.7	1.5	8.3
22.1	31.0	19.7	3.9
10.2	21.3	9.0	18.5

On average, there is a 17.5% difference between the peak readings, with the S peaks displaying the most similarity with an average difference of 9.3%, followed by the P peaks with 16.0%, the Q peaks with 21.4%, and finally the R peaks with 23.2%. The differences in key metrics between the Apple Watch and prototype recordings were higher for many signals than the 10% metric set in the acceptance criteria. However, 5 out of 10 reference signals were analyzed by the Apple Watch as “Inconclusive” rather than “Sinus Rhythm”, indicating that the Apple Watch signals may not have been cleanly collected, and invalidating the assumption that the Apple Watch would provide accurate electrocardiograms. In addition, each recording is highly dependent on the motion of the test subject which can affect the signal received, and as they were taken at different times, the effects of motion may not be equally displayed on both recordings. Considering that visual analysis of each pair of recordings displayed distinct similarities, the team concluded that the prototype passed this test, but further comparisons should be performed comparing the results of the prototype to other commercial and medical-grade electrocardiography devices.

As for the AI diagnosis capabilities, it is important to note that the access to training data was limited in the amount of available ECG sample signals labeled by a cardiologist to have normal sinus rhythm, arrhythmia, atrial flutter, congestive heart failure, myocardial infarction, or atrial fibrillation, with some conditions having as few as 30 samples. Another limitation of the training data was that the signals were taken and analyzed on a variety of different electrocardiographs across many years. Due to limited data availability, the accuracy of the AI system is measured primarily using validation during the training process. Using this process, the highest accuracy of 96.9% is achieved for the conditions of atrial fibrillation, atrial flutter, congestive heart failure, normal sinus rhythm, and myocardial infarction, with 128 diagnosed ECG signals being used as validation data. The confusion chart for this model is shown in Fig. 8.

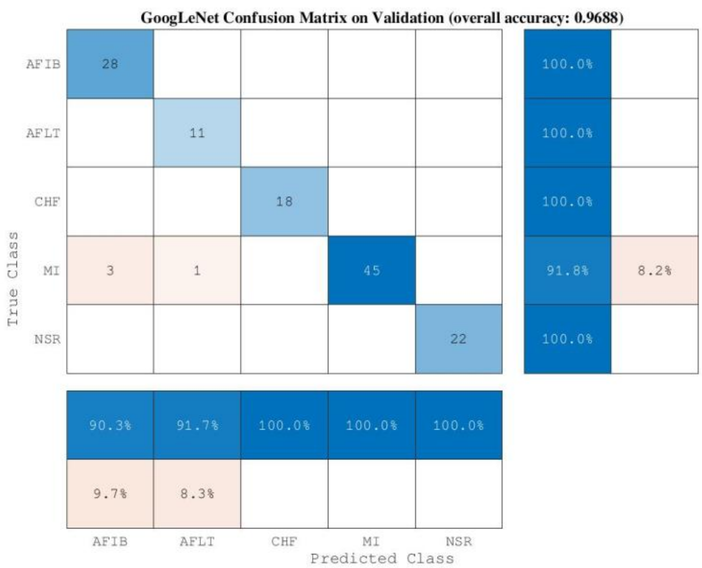

Figure 8. Confusion Matrix for Artificial Intelligence Heart Diagnostic Algorithm

Furthermore, the accuracy of the artificial intelligence is potentially limited by the decision to employ an image classification algorithm, and the use of only ten second signals for analysis due to the maximum length of available data. While image classification is used in many algorithms sourced for inspiration, such as “Classify Time Series

Using Wavelet Analysis and Deep Learning" (MathWorks, 2018), the conditions selected for analysis may not present in a manner that can be evaluated using feature extraction. This concern is highlighted in the comparison of the recorded signals to samples of the training data for the model. For example, a signal that should be normal sinus rhythm is classified as having a 99% chance of being positive for myocardial infarction; however, as shown in Fig. 9, this visually appears to be a logical result based on the matching T peaks, boxed in the diagrams, that do not appear in any of the other condition samples. While the plot only includes one sample of each training set, the small size of the training set enhances the effect of each sample, indicating that this could represent an issue in the functional accuracy of the model.

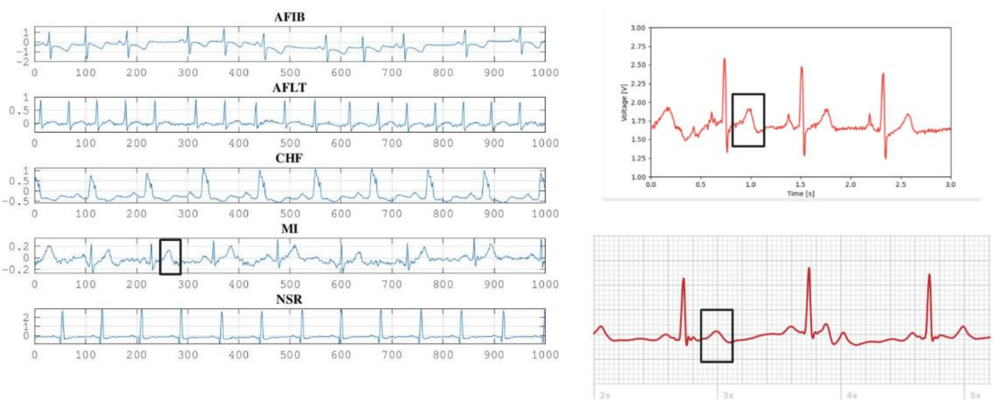

Figure 9. Comparison of Training Sample Electrocardiogram Data for Each Condition (left) to the Recorded Electrocardiogram from the Prototype (top right) and Apple Watch (bottom right), with Key Attributes Boxed in Black

The neural network meets the design requirement acceptance criteria, as it achieved over 90% accuracy in validation testing for all five desired conditions and was successfully deployed on the Raspberry Pi, achieving the intent of the proof-of-concept project. Prior to commercialization, the functional accuracy of the model needs to be improved through the retraining of the model with a significant amount of training data collected on the prototype and labeled by a professional cardiologist. In addition, the choice of network, potential conditions for diagnosis, duration of reading, and the feature extraction methods should be reviewed by a cardiologist.

Overall, this proof-of-concept prototype is functional, capable of collecting a clean electrocardiogram signal from a patient along with their biographical data, analyzing that electrocardiogram using a deployed artificial intelligence network, and storing that data for remote access via a website. However, many improvements must be made prior to implementation, including expanded training data to maximize the accuracy of the artificial intelligence, re-evaluation of the electrocardiogram reading circuit to ensure the reading meets medical accuracy standards, and increased security measures in all components, from the device to the website and database.

Acknowledgements

We would like to acknowledge Dr. Yu Zhang of the Trinity University Computer Science department for her insight on artificial intelligence algorithms, Neal Pape from the Trinity University IT department for providing access to the needed computing systems and MATLAB compatibility assistance, and Ryan Hodge and Marc Carpenter from the Trinity University Makerspace and Electronics Shop for their guidance and help during the circuit and casing construction. We would also like to acknowledge the key guidance and knowledge provided by the MathWorks Help Center examples and sample code in building the artificial neural network for this project.

References

Alake, R. (2021, November 3). *Deep learning: GoogLeNet explained*. Medium. https://towardsdatascience.com/deep-learning-googlenet-explained-de8861c82765

Baim, D. S., Colucci, W. S., Monrad, E. S., Smith, H. S., Wright, R. F., Lanoue, A., Gauthier, D. F., Ransil, B. J., Grossman, W., & Braunwald, E. (1986). Survival of patients with severe congestive heart failure treated with oral milrinone. *Journal of the American College of Cardiology, 7*(3), 661–670. https://doi.org/10.1016/s0735-1097(86)80478-8

Centers for Disease Control and Prevention. (2022, February 7). *Heart disease facts*. Centers for Disease Control and Prevention. https://www.cdc.gov/heartdisease/facts.htm

Centers for Disease Control and Prevention. (2021, September 27). *Atrial Fibrillation*. Centers for Disease Control and Prevention. https://www.cdc.gov/heartdisease/atrial_fibrillation.htm

Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., ... & Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation [Online]*, *101* (23), pp. e215–e220.

Heroku. (2022). *Heroku.* Heroku. https://www.heroku.com/

IBM Cloud Education. (2020, October 20). *What are convolutional neural networks?* IBM. Retrieved April 30, 2022, from https://www.ibm.com/cloud/learn/convolutional-neural-networks

MathWorks. (2018). *Classify Times Series Using Wavelet Analysis and Deep Learning*. MathWorks Help Center. https://www.mathworks.com/help/wavelet/ug/classify-time-series-using-wavelet-analysis-and-deep-learning.html

Moody, G. B. & Mark, R. G. (1983). A new method for detecting atrial fibrillation using R-R intervals. Computers in Cardiology, 10, 227-230.

Psycopg. (2021). *Psycopg – PostgreSQL database adapter for Python*. Psycopg 2.9.3 documentation. https://www.psycopg.org/docs/

Prusa Research. (n.d.). *Original Prusa i3 MK3S+ 3D printer*. Prusa Research. https://www.prusa3d.com/product/original-prusa-i3-mk3s-3d-printer-3

Python. (n.d.). *tkinter – Python interface to Tcl/Tk.* Python. https://docs.python.org/3/library/tkinter.html

react-chartjs-2. (n.d.). react-chartjs-2. https://react-chartjs-2.js.org/

Saha, S. (2018, December 15). *A comprehensive guide to Convolutional Neural Networks - the eli5 way*. Medium. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Ultimaker. (n.d.). *Ultimaker S5: Reliability at scale*. Ultimaker. https://ultimaker.com/3d-printers/ultimaker-s5

Wagner, P., Strodthoff, N., Bousseljot, R., Samek, W., & Schaeffter, T. (2020). PTB-XL, a large publicly available electrocardiography dataset (version 1.0.1). *PhysioNet*. https://doi.org/10.13026/x4td-x982.

Wagner, P., Strodthoff, N., Bousseljot, R., Kreiseler, D., Lunze, F.I., Samek, W., Schaeffter, T. (2020). PTB-XL: A Large Publicly Available ECG Dataset, Scientific Data. *PhysioNet*. https://doi.org/10.1038/s41597-020-0495-6.