Software Engineering Department

ORT Braude College

Capstone Project Phase B – 61999

Camera measurement of physiological vital signs monitoring system

23-2-D-7

Shimon Rubin | shimonr111@gmail.com

Lior Guzovsky | liorguzovsky@gmail.com

Supervisors:

Dr. Dan Lemberg | lembergdan@braude.ac.il

Mrs. Elena Kramer | elenak@braude.ac.il

Table Of Contents

Abstract

Measurement of heart rate is necessary to assess the status of the patient's health. Performing non-contact heart rate measurement can provide many benefits in various fields such as medicine, hospitals, and sports. In addition, it can be mainly significant in the field of remote medicine. For it to be a worthy replacement for the existing technologies that include contact equipment for heart rate measurement, there needs to be great accuracy for the heart rate measurement results, which is something that is not easy to achieve given the fact that it is a measure that changes in a very short period. In our project, we suggest a camera-based heart rate monitoring system that detects objects using image processing methods such as the Haar cascade classifier. We use the Fast Fourier Transform (FFT) for analyzing frequency domain heart rate signals.

Keywords:

Image processing, Real-time, Heart rate, Video, Fast Fourier Transform (FFT), Haar Cascade classifier.

1 Introduction

In today's fast-paced world, the demand for quality healthcare remains paramount, especially in nursing homes where the well-being of our beloved family members becomes a top priority.

From a personal story of a family member who was in a nursing home, we have noticed that the therapist-patient ratio during the night shifts is relatively small and therefore dangerous for the patients. This experience highlighted the need for improved patient-therapist ratios, particularly during the night shifts, when resources are limited, and monitoring becomes a challenging task.
Inspired by this personal story, the idea for our system to measure heart rate from video emerged and was approved as our final project.

Traditional methods rely on the use of pulse oximeters or other contact-based devices to obtain vital signs. However, such devices often require direct physical contact with the patient, which can be intrusive and uncomfortable, especially for elderly individuals in nursing homes. In addition, the traditional methods demand more therapists which is a problem we won’t be able to solve.

In recent years, the field of non-contact heart rate monitoring through video has gained significant attention, and particularly in the post-COVID era. The use of such systems (contactless monitoring systems) was introduced in airports where automated thermal screening systems have been deployed to monitor body temperature without physical contact.

As a solution, we developed a system that monitors the heart rate of the patients and present them to the medical team, to enable life-saving treatment if necessary. The system process video coming from live streaming. Then, it locates the region of interest (ROI), more especially the face and forehead, which offer indicators of the underlying cardiovascular activity, by using a Haar cascade classifier which is an efficient image processing algorithm. From the ROI we have found, the green channel information is then extracted from the video data, and the Fast Fourier Transform (FFT) is used to examine the frequency spectrum of those frames. This enables us to precisely extract the crucial information and identify conspicuous peaks that correspond to the heart rate.

In the subsequent sections of this article, we will explore deeper into the complexities of contactless heart rate monitoring from video in nursing homes.
We will explore the existing related work in the field. Then present the background information necessary for a thorough understanding of the challenges and opportunities associated with contactless monitoring. Building upon the knowledge gained from related work and background analysis, we will describe the challenges we encountered and the way we handled them during the development of the application. The subsequent section will detail the research and engineering process we have made to develop the system including the architectural design and data flow to provide an overview of the system's structure and functionality. We will discuss the evaluation and verification plan. Finally, we will describe the solution, the process and the results of our system.

2 Background and Related work

The field of monitoring heart rate from video data has gained considerable attention in recent years due to its potential for non-contact vital sign measurement. In the medical domain, accurate and timely monitoring of heart rate is crucial for diagnosing and managing various cardiovascular conditions.

This section is dedicated for the description of the medical background, PPG and RPPG theory, mathematical background, RGB model, ICA algorithm, Bandpass filter, GrabCut algorithm and related work we found relevant during our research to solve the problem we have described.

2.1 Medical background – cardiovascular system

The cardiovascular system is responsible for transporting blood throughout the body and is composed of the heart, blood vessels, and blood.

Hemoglobin is a vital component of the cardiovascular system. It is an iron-containing protein found in red blood cells that responsible for the transport of oxygen from the lungs to the body's tissues. When hemoglobin binds with oxygen in the lungs, it forms oxyhemoglobin, which provides a bright red color to oxygenated blood. This oxygen-rich blood is then carried by arteries to various organs and tissues, where oxygen is released for cellular respiration.

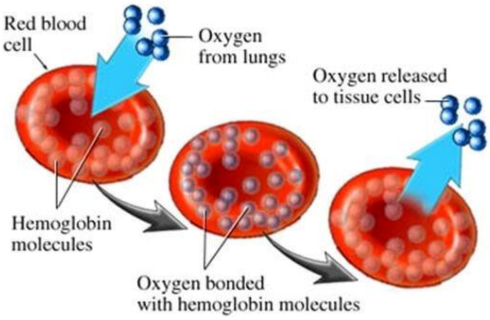


Figure 1

Blood circulation involves the heart pumping oxygenated blood from the lungs into arteries, which carry it throughout the body. Deoxygenated blood returns to the heart through veins and is pumped back to the lungs for re-oxygenation.

When The heart beats by contracting and relaxing its muscles in a rhythmic pattern, creating pressure that forces blood out of the heart and into arteries during systole and allows blood to flow back into the heart during diastole.

Blood flows through arteries, arterioles, capillaries, venules, and veins in a continuous loop throughout the body.

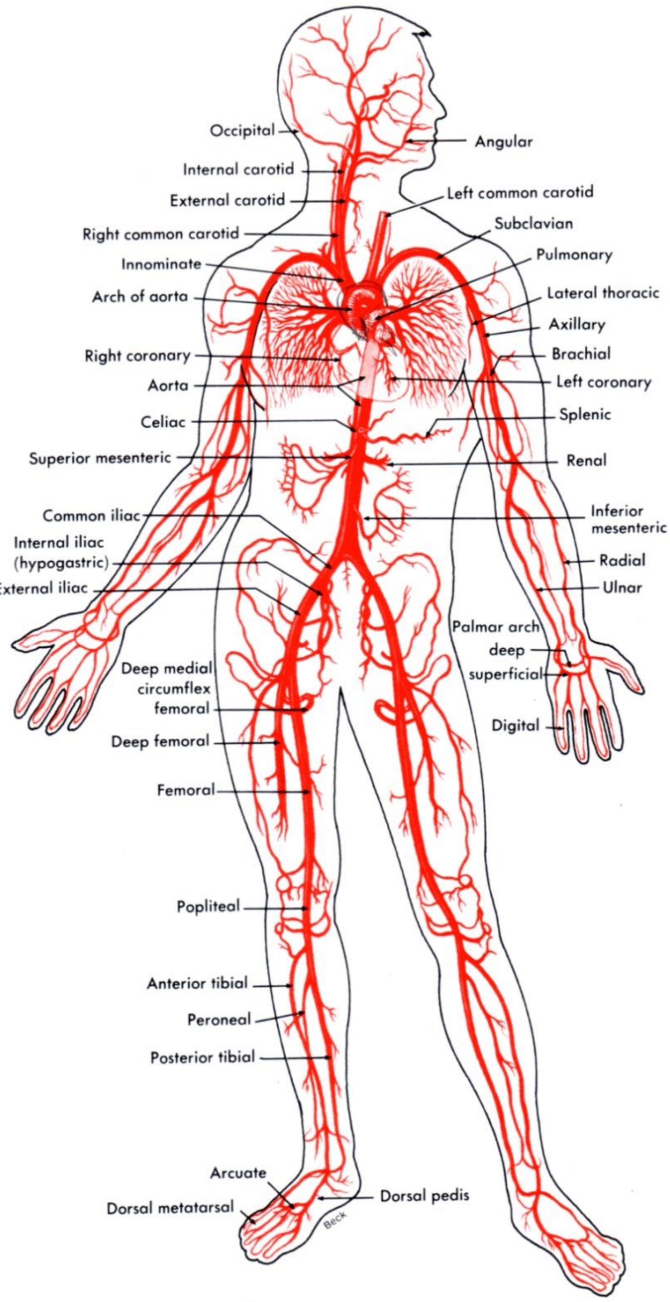
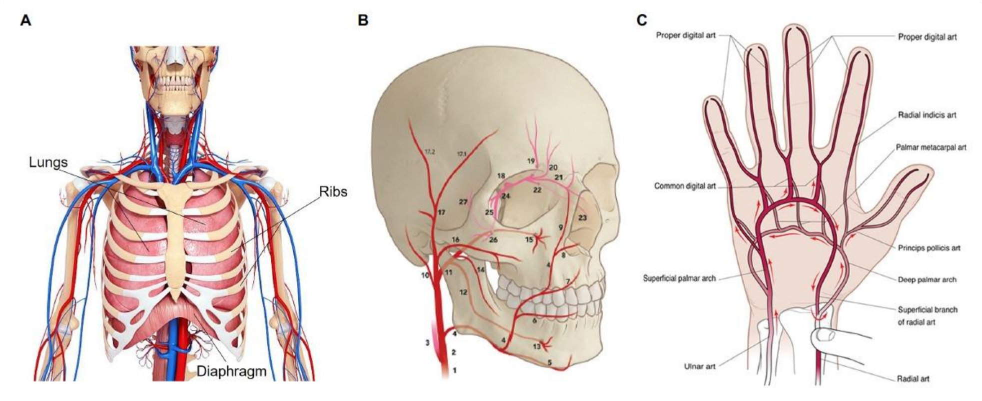


Figure 2

The major arteries that supply blood to the face and head are located on either side of the neck: carotid arteries (internal and external) and vertebral arteries. In particular, the forehead receives blood primarily from branches of both internal carotid arteries, including the supraorbital artery [Fig 1 – B].



Anatomy of the respiratory system **(A)**, face (**B**; adapted from von Arx et al., 2018), and hand **(C)** vascularization (from Habib et al., 2012). In **(B)**, numbers indicate the main arteries that carry the blood to the forehead and cheeks, specifically: common carotid artery (1), external carotid artery (2), internal carotid artery (3), facial artery (4), transverse facial artery (16), superficial temporal artery (17) and frontal branch (17.1), ophthalmic artery (18).

Figure 3

Normal heart rate parameters vary depending on age, sex, and physical activity level but typically range between 60-100 beats per minute at rest which are corresponding to a frequency range of 1-1.67 Hz.

2.2 PPG and RPPG

2.2.1 PPG

Photoplethysmography, sometimes known as PPG, is a non-invasive optical technique that assesses changes in blood volume in peripheral tissues.
It involves shining a light into the skin and detecting changes in light absorption created by the continuous nature of blood flow.

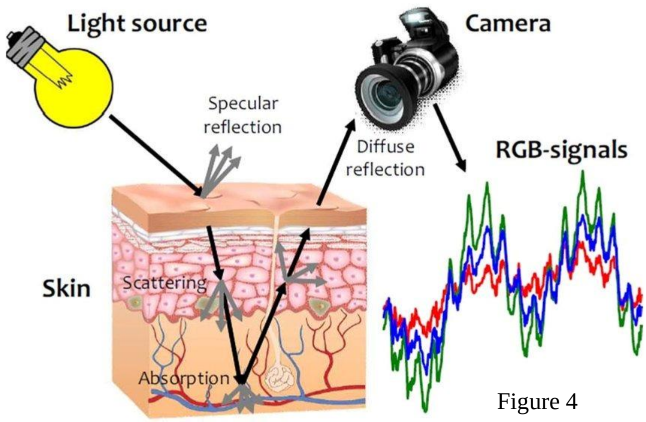


Figure 4

In PPG, light passes onto the skin, generally at the fingertip or earlobe from a light source. The blood arteries below the tissues partially absorb some of the light. The amount of blood in the vessels affects how much light is absorbed. Blood flow during heartbeats changes the blood vessels volume on a regular basis, changing how much light is absorbed in the process.

The PPG waveform, an electrical signal, is created from these intensity variations. The PPG waveform symbolizes a pulsatile blood flow element and can reveal information regarding the heart rate.

PPG is used in many different types of medical research and monitoring. To determine the heart rate and evaluate its variability, the waveform is typically studied for heart rate monitoring. Pulse oximeters[1] also use PPG to measure peripheral perfusion and measure blood oxygen saturation levels.

Improvements in technology have ended up in the development of wearable devices with PPG sensors, enabling non-invasive continuous monitoring of vital signs.

2.2.2 RPPG

RPPG, or Remote Photoplethysmography, is a new method that builds on the fundamentals of photoplethysmography (PPG) to monitor physiological parameters without having to encounter the subject directly. RPPG uses video-based photography to capture small color changes on the subject's face or other exposed areas, unlike traditional PPG, which necessitates direct physical contact between the light source and the skin.

In RPPG, a camera collects a video of the subject while focusing on specific areas like the forehead or cheeks. The recorded video offers details about

[1] The pulse oximeter, or Pulse Ox, is an electronic device that measures the saturation of oxygen carried in your red blood cells.

changes in blood volume beneath the skin's surface that affect skin color. The continuous blood flow caused by the cardiac cycle is connected to these variations in color.

The continuous signals can be extracted from the video frames using RPPG algorithms, which can then produce a remote photoplethysmogram. The remote photoplethysmogram shows distinctions in blood volume over time similarly to a traditional PPG. The subject's heart rate, heart rate variability, blood flow characteristics, possibly other cardiovascular parameters, can all be learned from this information.

Due to several variables that may affect the precision and dependability of the remote measurements, RPPG does, however, also present difficulties. These elements include of subject mobility, alterations in skin tone and changed ambient lighting conditions. As a result, current research concentrates on creating reliable algorithms and strategies to deal with these constraints and raise the precision of RPPG-based measurements.

2.3 FFT - Fast Fourier Transform

A fast Fourier transform (FFT) is a mathematical algorithm used to transform a time-domain signal into its frequency-domain representation. A fast Fourier transform can be used in various types of signal processing. It may be useful in reading things like sound waves, or for any image-processing technologies. A fast Fourier transform can be used to solve various types of equations or show various types of frequency activity in useful ways.

The Fast Fourier Transform is an optimized implementation of the Fourier transform algorithm. It significantly reduces the computational complexity of the transform, making it feasible to compute the frequency-domain representation of signals in real-time or near real-time applications. For heart rate detection, the frequency domain of a PPG signal can be inspected for peaks in the region corresponding to a reasonable heart rate.

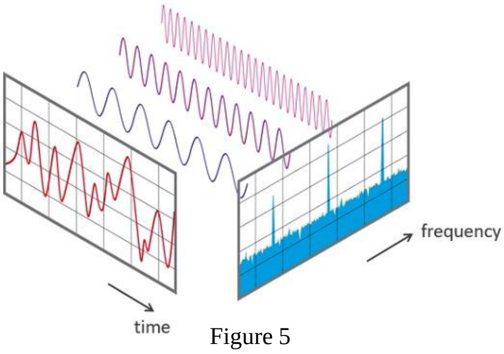


Figure 5

2.4 RGB Model

A typical color model used in digital imaging and displays is RGB (Red-Green-Blue). Red, green, and blue light are combined in various intensities to represent colors in the RGB model. Each color component can have values between 0 and 255, which represent brightness or intensity. A wide spectrum of colors can be produced by adjusting the red, green, and blue component intensities.

Due to the optical properties of human tissues and the absorption spectrum of hemoglobin, photoplethysmography (PPG) chooses to detect heart rate using the green channel. PPG makes use of the green channel to find a balance between reducing interference from other tissue elements and raising the sensitivity to changes in blood volume. Both oxygenated and deoxygenated hemoglobin partially absorbs green light, making it possible to record a distinct rapid signal. When using the green channel, we can notice that more hemoglobin-rich blood flows into the veins close to the skin's surface when the heart beats and the blood volume rises. As a result, there is an increase in the absorption of green light and a corresponding decrease in pixel intensity. Heart rate can be determined by tracking these changes over time.

Although PPG can also be performed using red and near-infrared channels, they are more sensitive to interference from things like ambient light and tissue variance. Green light is commonly used for PPG measurements of heart rate and other cardiovascular parameters because it provides a far better compromise between signal quality and interference.

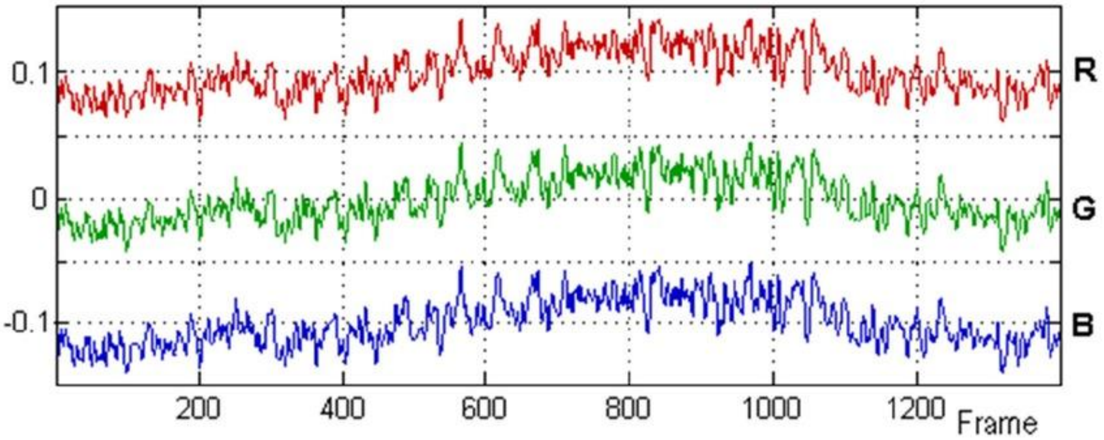


Figure 6

2.5 ICA Algorithm

Independent Component Analysis is referred to as ICA. It is a method for computing the independent parts of a multidimensional signal. ICA is used to separate the independent source signals from the observed mixed color signals. The observed color changes in the three channels (red, green, and blue) are assumed to be the result of three source signals. According to ICA, these source signals are combined linearly to produce the observed mixed signals. These distinct source signals can be extracted using ICA to produce more precise signals that can be used to determine heart rate from video data.

2.6 Bandpass filter

A bandpass filter is a sort of electronic filter that reduces (decreases) signals outside of a specific frequency range while allowing signals within that range to pass through. A bandpass filter can be used to isolate frequency peaks in the power spectrum in the range of 0.75 to 4 Hz, which correspond to physiological heart rate ranges of 45 to 240 bpm. The accuracy of heart rate measurements from video data can be increased by applying a bandpass filter to remove noise and other undesired frequencies from the signal.

2.7 GrabCut algorithm

GrabCut is an image segmentation method created by Rother et al. that is used to distinguish between an image's foreground and background pixels. By creating a graph model to describe the image and figuring out the smallest cut for the graph to generate two sets of nodes, which represent the foreground and background pixels of the image, it is possible to achieve the algorithm's goal of iteratively decreasing an energy cost function. GrabCut is used during the stage of ROI selection in the context of monitoring heart rate from video of a person's face. GrabCut is used to change the image after a region of interest (ROI) on the person's forehead has been chosen where blood vessels are visible and there is little motion artifact.

Within a few iterations, GrabCut is used to change the bounding box around the face of the subject and remove background pixels. This helps remove the background pixel variations that was present when using a simple, rigid bounding box ROI, which may improve the accuracy of heart rate measurements from video data.

2.8 Haar Cascade classifier

A machine learning-based object detection system called the Haar cascade classifier is used to identify objects in images or video streams. Viola and Jones first mentioned it in their study from 2001.

The research article "Rapid Object Detection using a Boosted Cascade of Simple Features" by Viola and Jones makes use of edge or line detection characteristics. The algorithm is trained on both many positive photos with faces in them and many negative images without any faces in them.

When using Haar cascade classifier the characteristics of the image make it simple to identify the image's borders or lines as well as regions where there is a sharp change in the pixel brightness. Below you can see the filters used by the Haar cascade classifier:

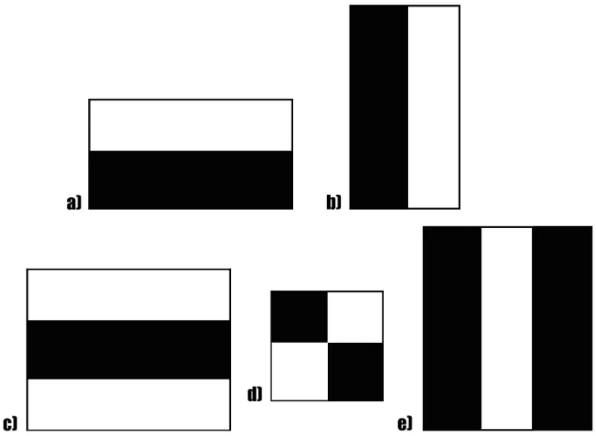


Figure 7

In the above filters, the goal is to calculate the total of all the image pixels located in the Haar feature's darker and lighter areas, respectively. then identify what makes them different. The Haar value will be closer to 1 if there is an edge in the image separating light pixels on the left from dark pixels on the right. In other words, if the Haar value is closer to 1, we declare that an edge has been found. Since the Haar number is distant from 1, there is no edge in the previous scenario.

For example, the left picture consists of arbitrary pixel values and the right one is the Haar filter. After calculating the result, we get the outcome on the right.

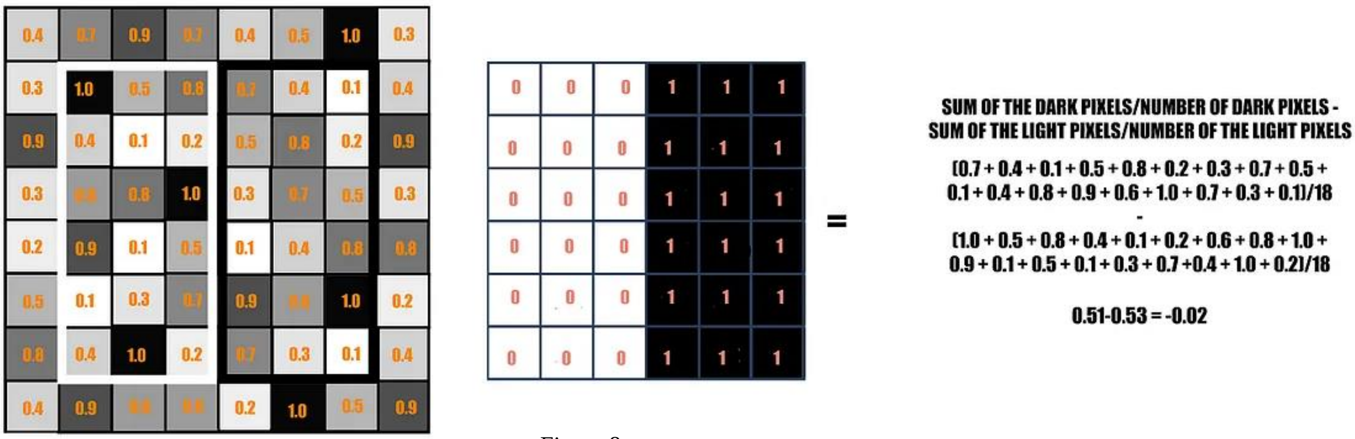


Figure 8

This is just an instance of a specific Haar characteristic that divides a vertical edge. There are currently more Haar characteristics as well, which will recognize any additional image structures as well as edges in other directions. The Haar feature must go across the entire image to find an edge anywhere in it.

Traversing the Haar features on an image would need numerous mathematical calculations. Even with a high-performance system, this would be an intensive operation.

The authors developed a different solution known as **The Integral Image** to address this problem and carry out the same function. Each pixel in an integral image is calculated from the original image so that it equals the total of all the pixels to its left and to its upper right.

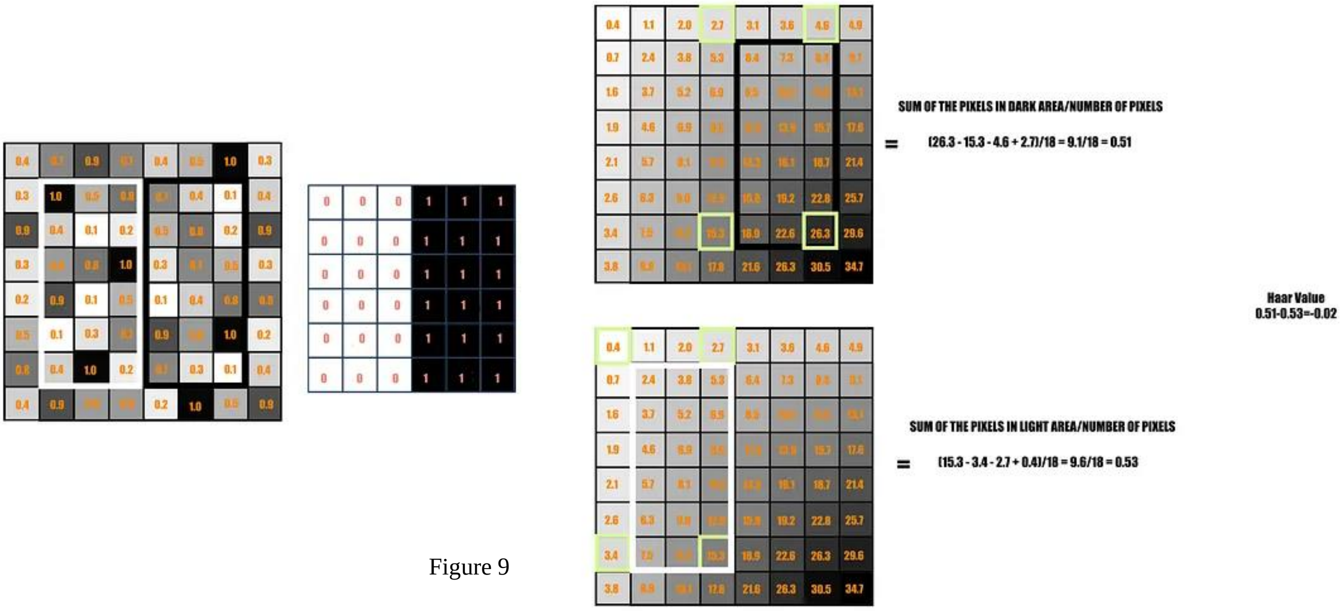


Figure 9

In contrast to the 18 additions previously, the Integral Image simply requires 4 constant value adds each time for any feature size. As a result, the number of adds does not depend on the number of pixels enclosed anymore, continually decreasing the time complexity of each addition.

2.9 AdaBoost

The focus of the previous chapter was on the features and representation of the image utilized in the Haar Cascade research. Let's look at some of the implementation details right now.

There is a collection of features that would accurately depict specific facial features, such as the lips, the bridge connecting the two eyes, or the brows. Most of these features won't fit the facial features well or won't be relevant to them because they are too random to be of any use. In this situation, they need a feature selection technique to pick a small subset of features from a large set that would not only pick features that performed better than the others but would also remove the irrelevant ones. They used a technique known as AdaBoost. Their final collection of features was reduced using this method from 180,000 to a total of 6000 features.

We created *weak learners* by applying each of these 180,000 features to the photos separately. Some of them, however, produced low error rates because they distinguished between Positive and Negative images more effectively than the others. These weak learners are created so that they would incorrectly classify a minimum number of photos.

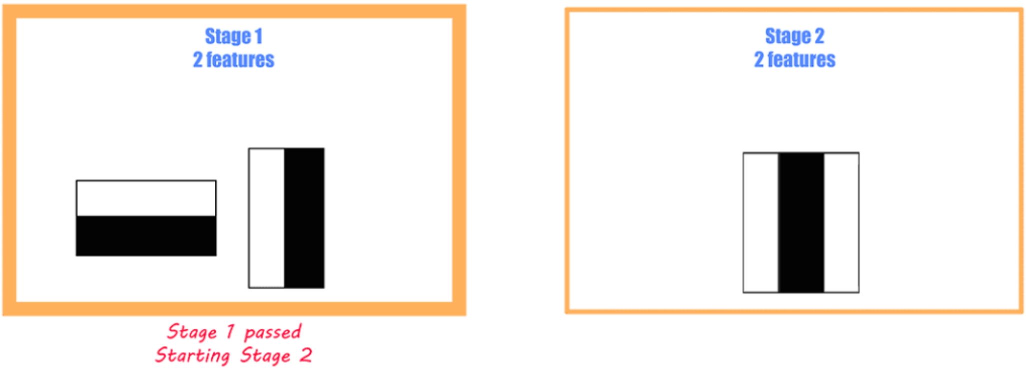
2.10 Attentional Cascade

The next step is the cascading section is to determine if a facial feature is present or absent. The subset of all 6000 features will again be run on the training photos. A standard window size of 24x24 was chosen for the feature detection to be running in at this point.

Another method called The Attentional Cascade was suggested to make this easier to understand. Not all features must be used on every window, according to the rationale behind this. We can infer that the face features are absent if a feature fails on a specific window. Therefore, we can continue to the following windows where facial traits might be present.

- The images have features in stages added to them. Beginning stages include simpler features compared to later stages, which have features that are sophisticated enough to reveal the finer details on the face. The window itself will be removed from the remaining process and replaced with the next window if the initial step is unable to detect anything on it. As the irrelevant windows won't be processed in the bulk of the steps, a significant amount of processing time will be saved.

- Only when the first stage's features are found in the image would the second stage of processing begin. This is how the process proceeds: if one stage succeeds, the window is moved on to the next; if it fails, the window is discarded.



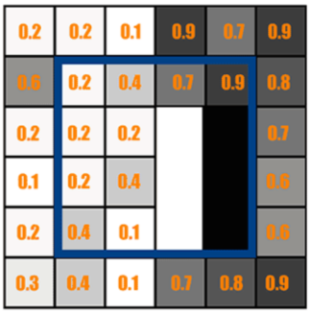


Figure 10

2.11 Related work

There is a growing amount of related work and research because of the significant interest in the subject of heart rate measurement using video data. Numerous studies have investigated different methods and strategies for precisely obtaining heart rate data from video recordings, allowing for non-contact. These studies addressed a wide range of topics, including machine learning, signal processing, computer vision, and healthcare.

For estimating heart rate from video data, researchers developed new technique, proposed innovative algorithms, and analyzed the efficacy of different approaches. In this section, we provide an overview of the body of research on heart rate measurement from video, highlighting significant contributions and methodology.

In the paper “Measuring Heart Rate from Video” [1] by Isabel Bush, Stanford Computer Science faculty, we have encountered the first method for extracting heart rate from video.

There are multiple steps involved in the process of extracting heart rate from video data using non-contact techniques based on photoplethysmography (PPG). The person's face is first videotaped using a standard color camera to give a clean view of the forehead and cheeks. The input for the associated analysis is this video.

The person's forehead is the next area to be chosen as a region of interest (ROI), as blood vessels are visible and motion artifacts are at a minimum there. This ROI serves as the basis for further research. The ROI is then tracked throughout time, considering any movements or position changes, to ensure accurate measurement. For this, tracking algorithms like template matching or optical flow are used. GrabCut algorithm is used to separate the foreground (in this case, the person's face) from the background in an image or video frame. It is used in the context of measuring heart rate from video of a person's face because it can help to exclude background pixels and hair that may interfere with accurate heart rate measurements.

Signals are taken from each video frame within the chosen ROI when the ROI has been successfully monitored using ICA method (Independent Component Analysis) to extract independent source signals from observed mixed color signals. For heart rate estimation, these signals show changes in pixel brightness over time and offer useful data. These signals do, however, frequently include noise and undesirable deviations. Therefore, to improve the quality of the recovered signals and eliminate undesired noise, signal processing techniques including bandpass filtering and Fourier analysis are used.

The heart rate is determined after signal processing by examining the frequency components of the processed signals. Usually, techniques like peak detection or spectrum evaluation are used for this. The heart rate can be precisely calculated in the frequency domain by locating peaks or dominating frequencies.

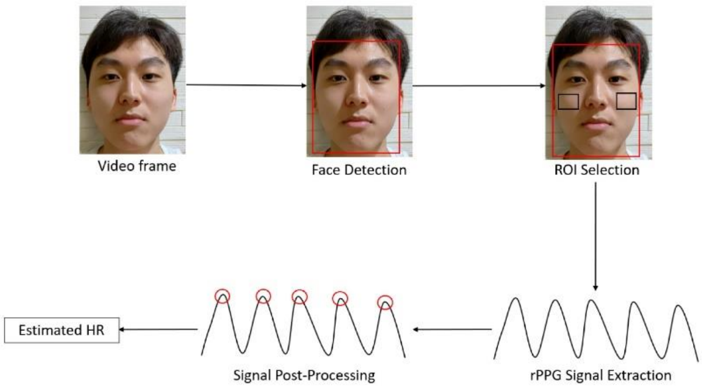
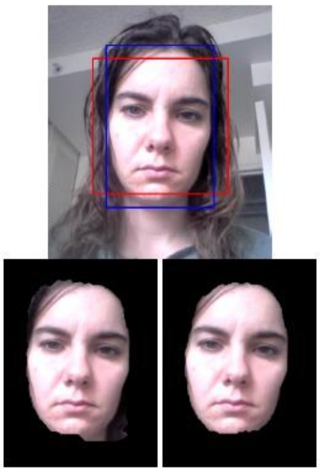


Figure 11

In the paper “Beat-to-Beat Cardiac Pulse Rate Measurement from Video” [2] by Brian L. Hill, University of California, Los Angeles, Xin Liu, University of Washington and Daniel McDuff, Microsoft, a new method for extracting heart rate from video was introduced. The authors proposed a neural architecture and model for camera-based vital sign measurement that effectively captures both spatial and temporal information, enabling the recovery of cardio-pulmonary signals from video recordings.

Several steps are involved in the procedure, all of which help to accurately extract the heart rate. The process begins with a camera taking a clear video of the subject's face in ideal lighting. The subject's face is then precisely located inside the video frames using face identification algorithms, which combine both conventional computer vision approaches and cutting-edge deep learning techniques (basically they used landmark points on the face such as corners of the eyes, nose, and mouth to find the ROI). A region of interest (ROI) is carefully chosen around the forehead area once the face has been detected because this is where small variations in skin color caused by changes in blood flow are most noticeable.

The remote photoplethysmography (RPPG) signal, which shows variations in skin color over time, is extracted using the ROI as the basis. Various signal processing techniques, including peak identification, normalization, and filtering, are used to improve the RPPG signal's quality. These techniques extract relevant cardiac pulse rate-related characteristics. The subject's heart rate is finally calculated over time using these derived features.



The original bounding box in red and adjusted bounding box that was input into the GrabCut segmentation algorithm in blue (top) as well as the first two iterations of the GrabCut implementation (bottom).

Figure 12

3 Medical research

A thorough research procedure including several fields of study is necessary to comprehend the medical basis for heart rate and the circulatory system. To begin with, a thorough understanding of cardiovascular physiology and anatomy is essential to understanding the complex operations of the heart and blood vessels. This includes researching subjects including the cardiac cycle, the anatomy of the heart, and the mechanics of blood flow.

An in-depth knowledge of the medical field is necessary to develop a remote heart rate monitoring system. Understanding hemoglobin and its function in determining heart rate in the face is crucial. Photoplethysmography, a non-invasive technology for heart rate monitoring, relies heavily on hemoglobin, the protein in red blood cells that carries oxygen. Accurate measurements may be taken from areas with strong blood flow, like the forehead or cheeks, by understanding the concepts underlying this technique related to its medical background.

There are various difficulties while researching medical information as a non-medical professional. It can be difficult to comprehend and interpret the complicated vocabulary and technical language used in scientific articles when conducting research. It is tricky to separate reliable sources from personal experiences since it is difficult to evaluate the quality and reliability of medical information without experience.

To sum up, the medical basis for heart rate and the circulatory system must be understood, and this requires extensive research in many different academic areas. This include studying cardiovascular anatomy, physiology, and the function of hemoglobin in influencing heart rate. Understanding ideas like photoplethysmography and its dependency on hemoglobin is necessary for creating a remote heart rate monitoring system. However, there are issues when conducting medical research as a non-medical professional, such as understanding scientific papers, understanding technical language, and assessing the reliability of sources.

4 Camera measurement of physiological vital signs monitoring system application

4.1 General Description

Our project's goal was to develop a desktop application that is easy to use and measures heart rate using cutting-edge video analysis algorithms. Heart rate is a crucial important indication that indicates a person's overall cardiovascular health. This cutting-edge technology is intended to increase the effectiveness and accessibility of health monitoring, particularly for communities that would find it challenging to use more conventional heart rate monitoring equipment.

The principal objective of our system is to process video data and analyze and report on everyone's heart rate. The necessity for an easy-to-use, non-invasive technique for monitoring heart rate propelled this goal. Our software uses video footage to measure heart rate in an easy-to-use manner without requiring any specialist medical equipment or physical touch, which makes it a vital tool for ongoing health monitoring and early identification of any cardiovascular problems.

The first stage in our four-step approach for extracting heart rate is to receive video input from any transmitting device. To precisely record physiological signals, the program finds the region of interest (ROI) on the subject after the video has been entered. The heart rate data is analyzed and evaluated at the end of the procedure to provide users with instant insights into their cardiovascular health. The application's goal of offering a straightforward yet powerful tool for heart rate monitoring is embodied in this straightforward approach.

4.2 Algorithm

In the second part of the project, we noticed that the data was noisy and therefore we had to change our pipeline. The main steps in the process of extracting heart rate remained the same, e.g. applying FFT, reading the input data and extracting the green channel. But some changes were applied to the pipeline. The next bullets will describe the final process:

4.2.1 Receiving video input

The system receives both recorded and live streaming video with fps = 30 Hz and 1920 X 1080 resolution.

4.2.2 Locating the eyes using the Haar cascade classifier

An efficient approach for identifying objects, such as face traits like eyes, is the Haar cascade classifier. The goal is to precisely identify the eyes in the video frames.

4.2.3 Locating the Region of Interest (ROI)

Once the eyes have been located, the algorithm should move on to choose the forehead as the ROI because it frequently offers a good location for detecting the pulse (related to the hemoglobin we explained earlier).

4.2.4 Address the green color channel

Convert the ROI image into a suitable color space, such as RGB (Red, Green, Blue). Split the ROI image into its component red, green, and blue color channels. Then, while working with the picture channels, access the green channel by choosing the proper index. At this point we decided to calculate the median along the green channel to remove some of the noise we encountered, this approach helped us in getting better results and made the analyzation process less complex.

4.2.5 Perform FFT

Fast Fourier Transform, known as FFT, is a mathematical procedure that converts time-domain data into the frequency domain. The system expects to examine the signal's frequency components by applying the FFT to the extracted data. In addition, it contributes to the noise reduction in our input data. The input signal size that the FFT worked with was 600 samples. Initially, we started with 300 samples, but the results were less than expected and therefore we changed the signal size.

4.2.6 Absolute FFT values

After performing the Fast Fourier Transform, we calculated the absolute values of the outcome since the output of this approach is a vector of complex numbers and we wanted to calculate the frequency of the signal.

4.2.7 Calculate heart rate low and high threshold

To search for the right candidates for the heart rate frequency we calculated the threshold for the low and high heart rate corresponding to the bin in the $abs\big(FFT(input\ signal\ vector)\big)$ results.

we used a formula to identify the start and end bin for our search after the highest frequency (main frequency):

$$frequency = 1\ Hz$$

$$start_{bin} = frequency * \frac{size(input\ sample\ vector)}{sampling\ rate}$$

The same process was applied for the end_{bin} with slight change to calculated frequency.

$$frequency = 2.3\ Hz$$

$$end_{bin} = frequency * \frac{size(input\ sample\ vector)}{sampling\ rate}$$

4.2.8 Search for the highest value in threshold range

The highest value in the $abs\big(FFT(input\ signal\ vector)\big)$ in described threshold range is corresponding to the bin with the desired heart rate we were looking for.

4.2.9 Calculate the Heart rate frequency

To calculate the heart rate frequency, we use the same formula described above with a slight change in the desired outcome:

$$HR_{frequency} = \frac{sampling\ rate * Max_{bin}}{size(input\ sample\ vector)}$$

4.2.10 Calculate the heart rate

The calculated heart rate is formed by applying $60 \cdot HR_{frequency}$

4.3 Solution Description

We developed a client-server desktop application in Python that safely stores patient data in a db.json file. Our application makes it simple to access and monitor patient data with its intuitive GUI. Its ability to directly analyze heart rates from video inputs is one of its most notable features. We will display diagrams that clarify the architecture of the app and how heart rate data is extracted from videos. With the goal of providing an easy-to-understand overview of the application's internal components, these diagrams seek to clarify its functions.

4.3.1 Package Diagram

The following package diagram shows the architecture of our client-server application, developed for extracting heart rate from user-

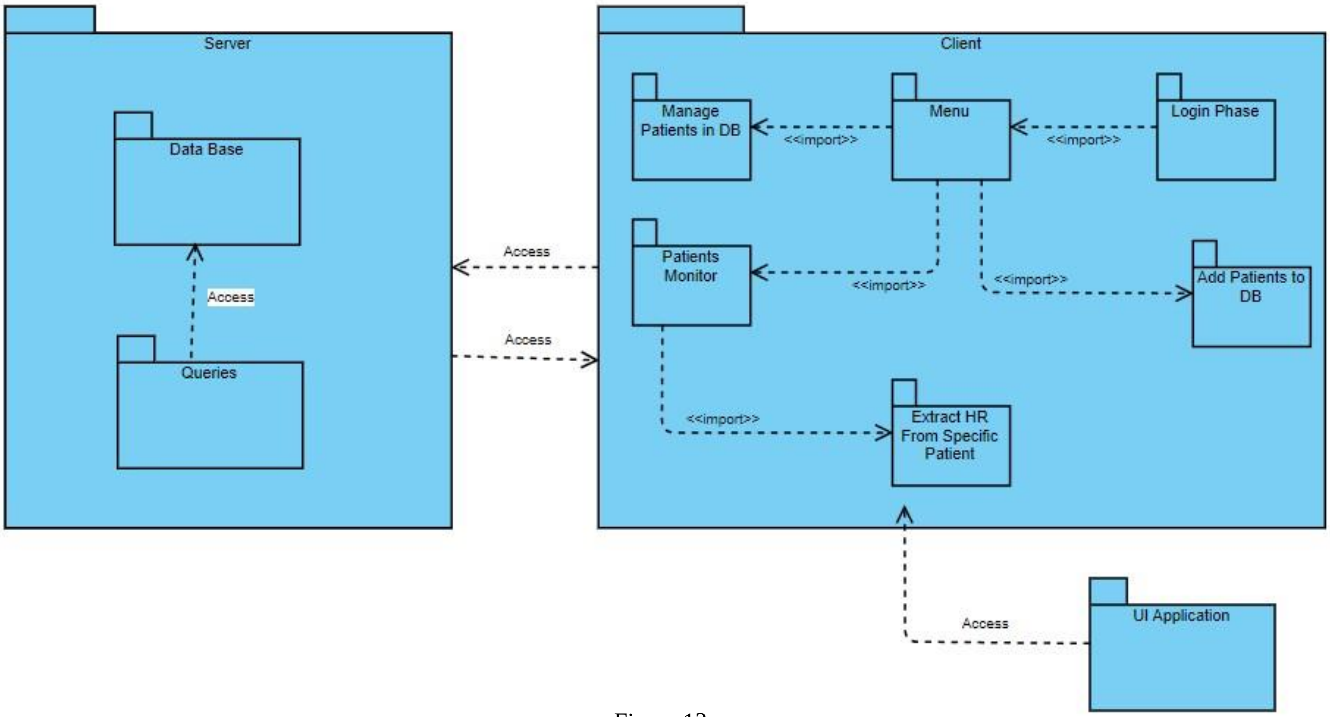


Figure 13

provided video inputs. It illustrates the system's modular design and the interaction between its components.

4.3.2 Activity Diagram

The following activity diagram shows the process of extracting heart rate from a chosen video input by the user, to the point it is shown to the user.

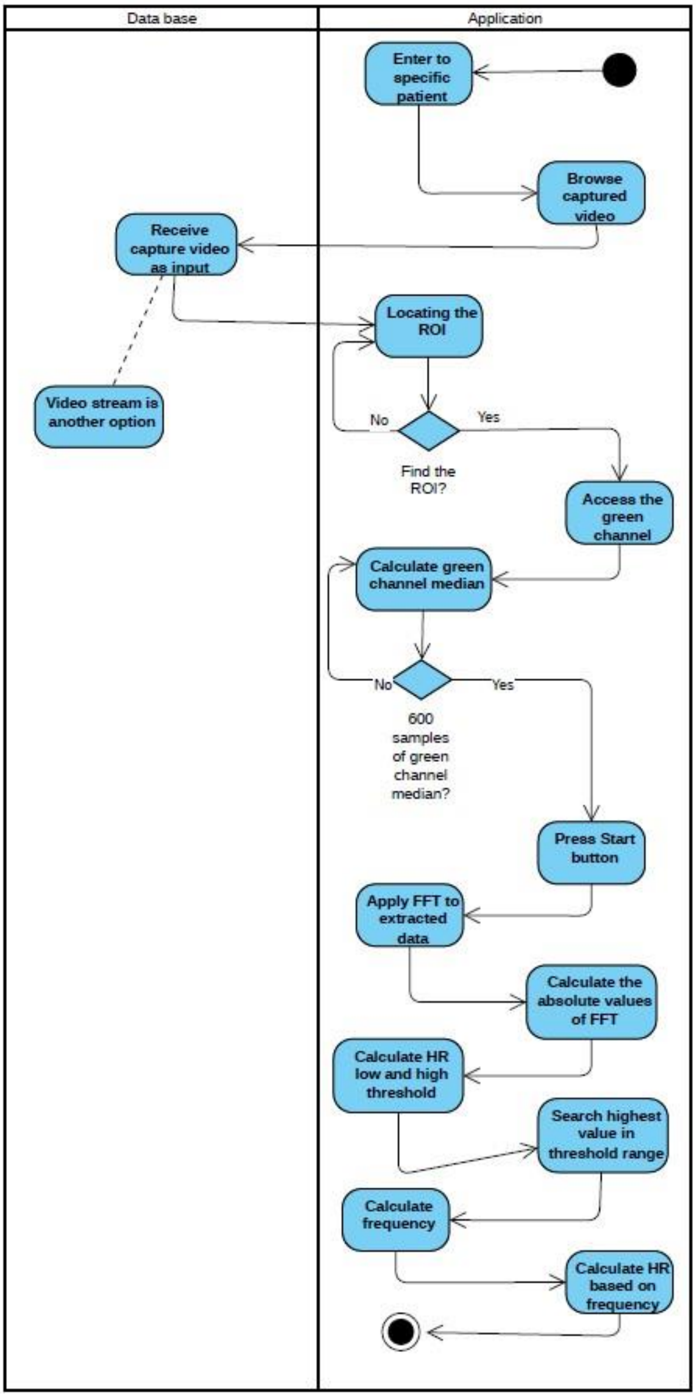


Figure 14

4.4 Development Process Description

4.4.1 Process

Our heart rate monitoring software was developed using a methodical procedure that began with the construction of a sizable dataset that would act as the basis for our algorithms. To mimic real-world situations found in clinical settings, we first concentrated on filming indoor footage in a variety of lighting settings, particularly fluorescent and LED illumination. We then carefully developed a backlog that included each task needed to construct the application, guaranteeing a methodical and structured approach to project management.

After creating a backlog, we divided the tasks among the team members and used GIT Hab to facilitate collaborative development, which allowed version control and code modifications to be seamlessly integrated. Weekly code merges and integration meetings made it easier for various parts to synchronize, which helped us move forward cooperatively toward our objectives. We moved to the main task of heart rate extraction as the program developed, working closely together to design and improve the algorithm.

However, when we ran into problems with the lighting conditions in our source videos, we were unable to continue further. We realized we needed a solution, so we got back together and came up with a different plan. We used FFT analysis to determine which frequency component corresponded to the heart rate. This novel approach produced encouraging outcomes.

Despite overcoming this hurdle, we encountered further setbacks when attempting to process indoor videos with noise from lighting. Despite numerous attempts with various filters and parameters, progress remained elusive. This iterative approach to problem-solving and adaption highlighted the dedication and collaboration that characterized the course of our project.

4.4.2 Decisions

During Phase B of our project, a pivotal decision was made to temporarily halt development to address the persistently poor outcomes we were encountering. This decision proved to be instrumental as it allowed us to delve deeper into the root causes of the issues we were facing, particularly regarding the impact of noise from LED and fluorescent lighting sources within our home environment. Through thorough investigation and experimentation, we gained valuable insights into the significant impact that such noise sources were having on our results—a crucial realization that had not been fully anticipated during the research phase of Phase A. This pause in development provided us with an opportunity to reassess our approach and make necessary adjustments to mitigate the effects of noise on our outcomes. This strategic decision not only led to a deeper understanding of the challenges posed by noise sources but also paved the way for significant advancements in our project, ultimately strengthening the foundation for future development and implementation efforts.

4.4.3 Tools

In the development of our remote heart rate monitoring application, we employed several tools and technologies to facilitate accurate data extraction and analysis. A pivotal component of our toolkit was the pulse oximeter, a vital device that played a central role in our project. The pulse oximeter served as the primary source for establishing ground truth heart rate measurements for the source videos. By directly measuring the oxygen saturation levels in the blood and detecting changes in blood volume, the pulse oximeter provided reliable data on heart rate variations. This ground truth data served as a reference point for validating the accuracy of our application's heart rate extraction algorithms.

Furthermore, we opted to develop our application in Python, leveraging its versatility and extensive ecosystem of libraries and packages. Python's flexibility allowed us to seamlessly integrate various functionalities and conduct complex mathematical operations essential for heart rate analysis. Specifically, we utilized packages such as NumPy and SciPy for conducting Fast Fourier Transform (FFT) and other mathematical operations critical to our heart rate extraction algorithms. This choice of programming language provided us with a robust framework for implementing and refining our algorithms with ease.

Additionally, we employed the mock library for Python to facilitate unit testing tailored to the specific requirements of our application. Unit testing played a crucial role in ensuring the reliability and robustness of our codebase by systematically verifying the functionality of individual components.
Overall, the combination of these tools and technologies enabled us to develop a sophisticated and reliable remote heart rate monitoring application capable of delivering accurate and actionable insights for healthcare professionals.

4.4.4 Client Interaction with the UI

Our collaboration with the end users, particularly those who will be utilizing our application in real-world healthcare settings, was instrumental in shaping our project's direction. To gain valuable insights into the needs and expectations of the nursing staff, we engaged with a family member who works as a doctor. Her firsthand experience provided invaluable inputs regarding the functionalities and features that she deemed essential for the nursing staff's efficient use of the application. She emphasized that the user interface (UI) should be as simple as possible, allowing for quick and intuitive navigation. Furthermore, she highlighted the importance of prominently displaying the heart rate result, ensuring that nursing staff could easily monitor patients' vital signs and respond promptly to any changes or emergencies. By leveraging her expertise and perspective, we gained a deeper understanding of the practical considerations and requirements specific to healthcare environments. This collaborative approach enabled us to tailor our application to better meet the needs of the end users, ensuring its usability, effectiveness, and relevance in clinical settings.

4.5 Testing Process Description

The testing process was carefully designed to cover all the essential features of our desktop application, which is meant to extract heart rate from video. To make sure the application was reliable and resilient.

4.5.1 Tools and Frameworks

- The unittest framework in Python: It allowed us to create a set of test cases that automatically check if various application components function as expected under different scenarios
- We were able to mimic interactions with the graphical user interface (GUI) elements and other services without the need for real implementations thanks to unittest.mock and MagicMock, which made it easier to create mock objects and functions

4.5.2 Testing GUI Components

To create the application's GUI, we used PyQt5 library. It was necessary to simulate user activities like clicks and text entry to test these components. To simulate user activities, this was accomplished by directly calling slot methods and programmatically setting widget states, For example:

- Verifying successful and unsuccessful login attempts
- Testing the ability to navigate between several screens, including the manage users, login, menu, and signup displays
- Simulating the procedure of managing current users and adding new ones

4.5.3 Testing Heart Rate Extraction Logic

- Tests for video processing included simulating video streams and confirming that facial recognition and green channel extraction functions performed as anticipated
- Tests for Heart Rate Calculation: Focusing on the algorithm used to extract heart rate from video. In order to prove that the output was accurate, the input data for the heart rate calculation routines had to be mocked up

4.5.4 Acceptance Testing

Acceptance Testing was done after the development and testing stages were finished to make sure the application complied with all criteria and was prepared for deployment.

Test	Feature to Test	Test Steps	Expected Result
1	Login Functionality	1. Launch the application 2. Enter valid username and password 3. Click on the login button	The application transitions to the main menu screen indicating successful login.
2	Logout Functionality	From the main menu, click on the logout button	The application returns to the login screen
3	Signup Functionality	1. From the login screen, click on the signup link 2. Fill in the required fields 3. Submit the form	A new user account is created, and the application navigates back to the login screen
4	Navigate to Heart Rate Extraction Screen	1. Log in to the application 2. Select the option to start heart rate extraction	The heart rate extraction screen is displayed
5	Heart Rate Extraction from Live Video	1. Select the live video source 2. Start the heart rate extraction process	The application displays the heart rate extracted from the live video feed
6	Heart Rate Extraction from Recorded Video	1. Select a recorded video file 2. Start the heart rate extraction process	The application displays the heart rate extracted from the recorded video

8	Data Accuracy of Heart Rate Extraction	Compare the application's heart rate reading with a known standard or external device for the same video feed	The heart rate readings should have a minimal acceptable deviation from the standard
9	User Interface and Usability	1. Navigate through all screens and features of the application 2. Evaluate the ease of use and intuitiveness	The application should be easy to use, with intuitive navigation and clear instructions

Table 1

4.6 User Manual

4.6.1 Open Application

The first step after configuring and downloading the application is to open an IDE such as PyCharm. Afterwards, run the server application and the client application. This procedure opens the client application.

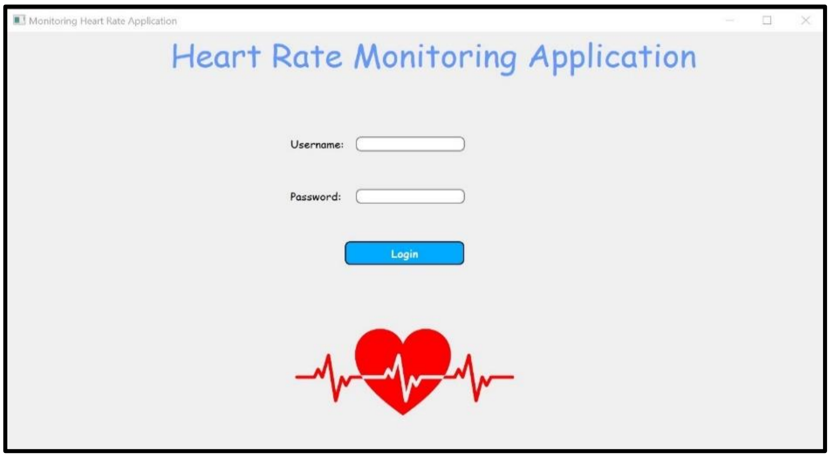


Figure 15

This is the login phase, please enter the username and password according to the data base and click "Login".

4.6.2 Menu Screen

In the menu screen, you have several options, including:

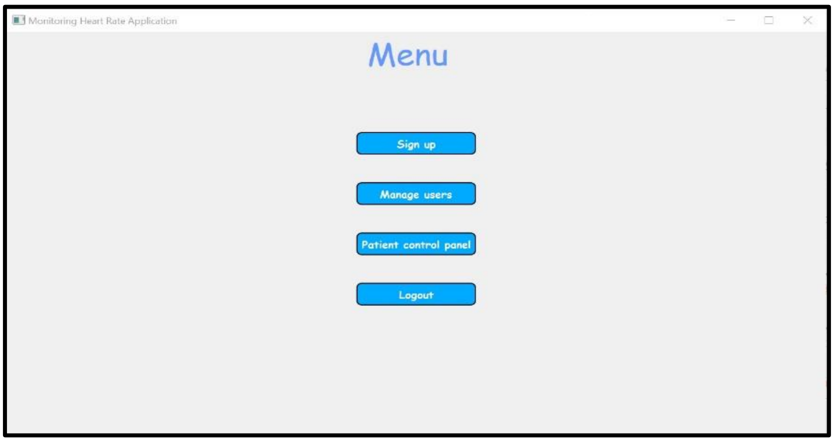


Figure 16

4.6.2.1 Sign up

This screen gives the user ability to add a new patient to the system data base.

Figure 17

4.6.2.2 Manage users

This screen gives the user ability to edit the system data base, e.g. add/remove patients.

Figure 18

4.6.2.3 Patient control panel

This screen displays the patient's heart rate in real time to the health care team who monitor the patients. When pressing the "Enter" button, the user will be able to see the heart rate raw data.

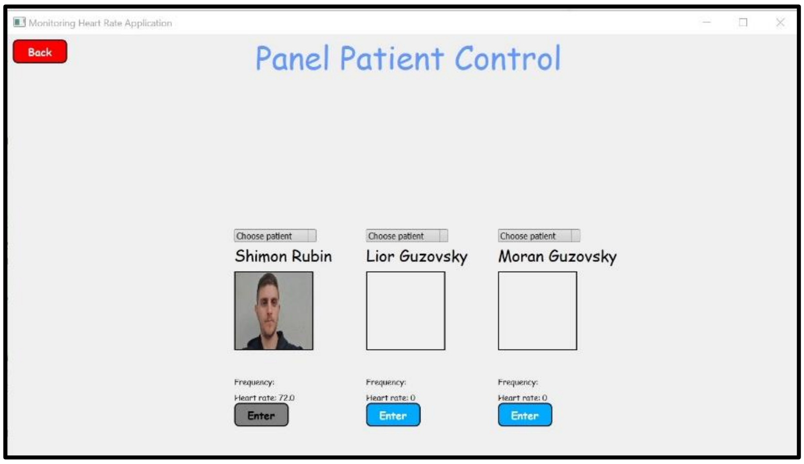


Figure 19

4.6.2.4 Logout

This button logouts the user from the system.

4.6.3 Specific Patient Screen

This screen is responsible for the actual calculation of the heart rate. The screen displays the patients recorded \ live streaming video, heart rate graph with the calculated frequency and heart rate over time and a graph that shows the FFT used to calculate the heart rate.

This screen enables two options for the raw data needed for our algorithm:

4.6.3.1 Recorded Video

To use the recorded video, the user should choose the "Video" option in the combo box located on the left side of the "Open" button. Afterwards, press the "Open" button, and specify the recorded video file he wants to use, notice that the recorded video is of the patient's face. After choosing the file, press the "Start" button and wait for the results.

4.6.3.2 Live Streaming Video

To use the live streaming video, the user should choose the "Webcam" option in the combo box located on the left side of the "Open" button. Afterwards, press the "Start" button and wait for the results, notice that the patient camera is connected properly (in our case, we used the laptop camera).

Figure 20

4.7 Developer Manual

4.7.1 Setting Up the Development Environment

- Set up the Python IDE on your workstation. PyCharm, which offers excellent support for Python projects, including PyQt5 apps, is the program we suggested
- Clone the project repository from the terminal, using the repository URL
- Install required libraries from the terminal, by running the following command to install all the required libraries: "pip install -r requirements.txt"

4.7.2 Running the Project

- Run the Server:
  Locate the server.py file within the server folder, right-click on the server.py file, and select 'Run'. Ensure the server is running before initiating any clients instances.
- Run the Client:
  Locate the client.py file within the client folder, right-click on the client.py file, and select 'Run'. The GUI should now be displayed, allowing you to interact with the application.
- Connecting Multiple Clients:
  Simply launch a second instance of the client script and follow the same instructions as before to connect multiple clients.
  Multi-user interaction with the program is possible since each client will establish an individual connection with the server.

4.7.3 Configuring IDE for Running Tests

- Go to File -> Settings
- In the Settings dialog, go to Project -> Project Structure
- Highlight client, server and test folders as Sources
- Apply the changes and click 'OK'
- Running the tests

4.8 Results

4.8.1 Challenges

4.8.1.1 Data Set

In the initial phases of our project, we encountered a significant challenge: the absence of a suitable dataset that matched our project requirements. As a result, we were compelled to generate our own video dataset from scratch. This process involved meticulously filming individuals in various environments and lighting conditions to capture a diverse range of color tones. Additionally, we needed participants with varying heart rates, both high and low, to comprehensively test our pipeline's performance. To accurately measure the heart rates during filming, we acquired a pulse oximeter, which was generously provided by our instructors. This device enabled us to capture ground truth heart rate data, essential for validating our pipeline's accuracy. Despite the complexity introduced by the need for diverse scenarios and participant characteristics, this approach was crucial for ensuring the robustness and effectiveness of our project. Through careful planning and execution, we successfully created a comprehensive dataset that enabled thorough testing and evaluation of our pipeline under diverse real-world conditions.

4.8.1.2 Lightning Conditions

Heart rate detection accuracy can be greatly impacted by variations in lighting. Accurately detecting heart rate signals in videos can be challenging due to changes in skin color caused by ambient light changes, shadows, or artificial lighting.
We first used recorded videos when we first started to work with our algorithm. The videos were recorded in-house. We have seen that the algorithm produced extremely inaccurate, noisy, and unstable results. We discovered that ambient light

from LED and fluorescent lights is the source of noise after evaluating the raw data. Since demonstrating the effectiveness of our pipeline was our main goal, we made the decision to record under natural illumination. This method brought us one step closer to the intended result by almost completely removing the noise. After successfully proving our algorithm works, we tested several filtering techniques to remove the cyclic noise we saw coming from the ambient light source we have mentioned earlier.

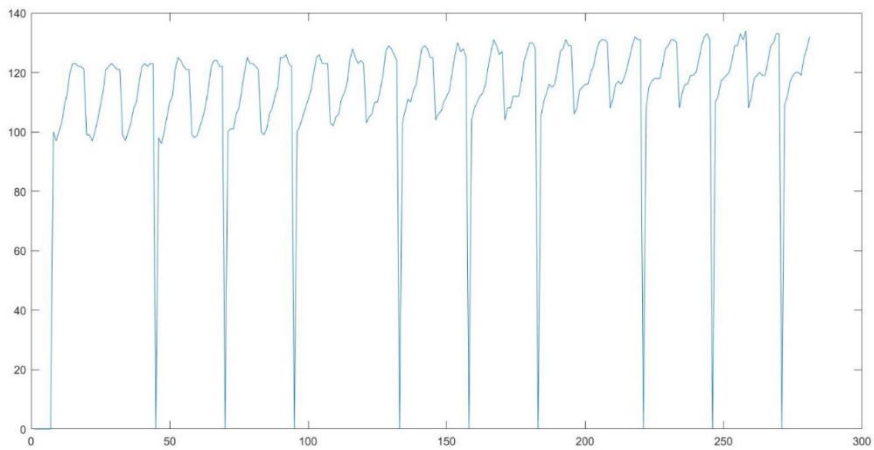


Figure 21

4.8.1.3 Subject Movement

The subject's movement might add a lot of noise to the data, making it challenging to identify the heart rate signal. This covers both minor motions like speaking or breathing as well as more significant motions.

Same as in the lightning condition approach, we decided to minimize the noise by recording videos with minimal movement.

4.8.1.4 Unfamiliar Field – Image Processing

Image processing was extremely challenging when we first started learning about it, and its complexity overwhelmed us. We used a range of articles and tutorials as well as the concept of trial and error to navigate this uncharted territory. No matter how the experiments worked out, they all contributed to our eventual understanding of the topic, together with the knowledge we gained from reading a lot. This approach increased our technical competency while teaching us the value of perseverance and adaptability in learning new skills.

4.8.1.5 Raw data size

In our data processing pipeline, we initially utilized raw video data with a resolution of 1920 by 1080 pixels. Recognizing the computational intensity of processing such high-resolution data, we decided to apply a median operation to every matrix of pixels. Specifically, we extracted the green channel data and performed a median operation over each pixel matrix. This process effectively condensed the information while preserving essential features, ultimately resulting in a reduction of data complexity. By retaining only one value per matrix from the green channel we made subsequent analysis more computationally tractable without sacrificing significant information content. This approach allowed us to manage the data more efficiently while still capturing the essential characteristics needed for our analysis.

4.8.1.6 Noise Filtering

To minimize undesired noise, we have experimented with several filtering techniques. We addressed the problem of noise coming from fluorescent and LED lights. Finding methods to distinguish the desired signals from the surrounding noise was one of our objectives. We wanted to enhance the quality of our signal processing algorithm by experimenting with various methods. Our goal was to determine which filtering techniques would reduce interference and improve the clarity of our signals through careful testing and analysis.

Low pass and high pass filters are used in signal processing to control signals by attenuating certain frequency components and allowing others to pass through. To efficiently smooth out high-frequency noise, a low pass filter allows frequencies below a specific cutoff point to pass, but a high pass filter allows frequencies above a given cutoff point to pass, hence removing low-frequency noise. We tested with several low pass and high pass filter parameters in our study to reduce noise from LED and fluorescent illumination. We specifically evaluated three and four-sized low pass filters and ten and fifteen-sized high pass filters. Nevertheless, the outcomes were insufficient despite these efforts. Numerous things could have contributed to this result. A possible cause could be that the window sizes chosen were not the most effective for efficiently absorbing and removing the frequencies linked to noise from LED and fluorescent lights.

As part of our investigation into noise reduction strategies, we experimented with different window sizes for median filtering. As a type of non-linear filtering, median filtering provides a straightforward but efficient way to reduce noise while maintaining the integrity of the original signal. We experimented with various window widths to determine how the window's size affected the filtering efficiency. While smaller window widths may enable greater sharp feature preservation but perhaps less effective noise reduction, larger window sizes tend to produce smoother output by considering a wider range of surrounding samples. In the end, we tried to find the ideal window size for our application requirements by balancing noise reduction and signal quality in our experiments.

Unfortunately, we discovered that despite our best efforts and experimentation with a range of noise reduction methods - including median filtering with varying window sizes and low \ high pass filtering - these approaches were unable to achieve the necessary degree of noise reduction. We tried optimizing procedures and parameters as much as we could, but the noise remained at levels that interfered with our desired signal. This result emphasizes how difficult it is to adequately mitigate noise interference and how complicated the noise environment is. To accomplish the intended outcomes in following projects, more investigation and improvement of noise reduction tactics might be required.

4.8.2 Evaluation

During the evaluation phase, we begin a thorough examination of the outcomes produced by our system. Using a pulse oximeter was essential to our assessment procedure since it allowed us to accurately calculate the ground truth heart rates of the candidates we examined. Three men and one woman, all between the ages of twenty and thirty, made up our study group, which ensured a varied representation for our assessment. To conduct an extensive evaluation of our application's performance over a wide range of heart rates, we specifically chose a range that extends from 60 to 120 beats per minute.

We were able to efficiently test the system's accuracy and robustness under various physiological situations thanks to its broad range of tested heart rates. Additionally, to enable a comprehensive analysis, we separated the raw input data into two classes: one for algorithm verification and another for performance testing. We concentrated on high heart rate scenarios during the testing phase, while a smaller selection of data was used for the verification process. Our careful assessment of the system's effectiveness and identification of areas for improvement and modification were made possible by this methodical methodology.

In the subsequent analysis, we will present the outcomes of the six recorded videos, with one sourced from the internet and the remaining five meticulously filmed by our team. This evaluation will be encapsulated in a comprehensive table, featuring key data items critical to our assessment process. These include the video number, Ground Truth Heart Rate (HR), Evaluated Heart Rate (HR), HR variation in Beats Per Minute (BPM), Ground Truth HR frequency, Evaluated HR frequency and HR accuracy in percentage. This structured presentation will provide invaluable insights into the accuracy and performance of our system across a range of real-world scenarios and conditions, guiding our understanding of its strengths and areas for enhancement.

Video	Ground truth HR [BPM]	Evaluated HR [BPM]	Ground truth HR [Hz]	Evaluated HR [Hz]	HR [BPM] Variation [BPM]	HR Accuracy [Percentage]
1	64	63	1.067	1.05	1	98.44%
2	68	69	1.13	1.15	1	98.55%
3	70	72	1.16	1.2	2	97.2%
4	76	72	1.26	1.2	4	94.74%
5	80	75	1.3	1.25	5	93.75%
6	108	111	1.8	1.85	3	97.3%

Table 2

Our application demonstrates remarkable accuracy, achieving over 93% precision when operating under natural lighting conditions. This high level of accuracy underscores the effectiveness of our approach in extracting heart rate data from video sources in optimal lighting environments.

4.8.3 Conclusions

To sum up, it appears we've achieved a significant milestone in our project by successfully extracting heart rate data from videos. However, our journey was not without its challenges. The absence of a suitable dataset posed an initial hurdle, necessitating the creation of our own dataset, which proved to be a time-consuming endeavor. Recording the videos required careful planning and execution, as we sought to capture a diverse range of individuals spanning different age groups and genders, ensuring a representative sample for our evaluation. Furthermore, our efforts were compounded by unforeseen complications arising from ambient lighting conditions. We discovered that variations in lighting, both natural and artificial, had a pronounced impact on the accuracy of our outcomes.

Despite multiple attempts to record videos under varied conditions, including adjustments to camera settings and the use of external lighting sources, we encountered difficulties in managing indoor ambient lighting, leading to limitations in achieving all desired outcomes. Nevertheless, a pivotal decision to film our recordings in natural light proved to be advantageous, as it provided consistent and uniform lighting conditions, thereby facilitating more reliable results. This strategic choice served as a proof of concept for our idea and application, demonstrating its feasibility and potential for real-world implementation. Looking ahead, we propose that future iterations of this project be continued by subsequent cohorts of students, building upon our groundwork, and addressing the challenges we encountered to further refine and enhance the effectiveness of our approach.

4.8.4 Insights

As we consider the lessons learned from our heart rate monitoring app, careful dataset preparation is essential to the project's success. One important lesson learned is how crucial it is to start the project's preparation of the dataset as early as possible—ideally, during phase A of the previous semester. A better way would be to split the dataset into two groups: videos taken in ambient light, and videos with noise from LED and fluorescent lights. This realization became very clear when we were developing the product because we had a lot of difficulties reducing lighting-related noise. The challenges presented by the noise from LED and fluorescent lights were different, highlighting the need for a more thoughtful strategy to dataset preparation that takes these factors into consideration. During the development we spent a significant amount of time in collecting the videos and creating our data set which could have been easier if we have done it in phase A.

Our trip also made clear how important it is to expand on our knowledge of noise and noise reduction strategies as soon as the project is launched. An important insight about the relevance of real-world applications became clear, especially in situations like nursing homes where heart rate monitoring is essential. The lighting conditions seen in clinical settings were similar to the simulated circumstances used in our experiment and were common in indoor spaces. This realization emphasizes how crucial it is to match project goals with actual situations, which motivates us to increase our knowledge and proficiency in noise reduction strategies adapted to environmental circumstances. We were able to obtain significant insights into the practical issues and concerns necessary for the effective adoption of heart rate monitoring devices in clinical settings by drawing comparisons between our project environment and real-world applications.

4.8.5 Project Metrics

In conclusion, we have fulfilled our project metrics in several ways. First, we were able to show in a proof of concept that our method enables the extraction of heart rate information from films made using basic laptop and phone cameras. This accomplishment represents a critical turning point in our project and demonstrates the viability and potential of our methodology in practical settings. Nevertheless, even with this achievement, we faced difficulties in reducing subject-caused motion distortions and eliminating noise from surrounding illumination. We tried our best, but we were unable to completely resolve these problems, which highlights areas that still need work.

However, we think that the main goal of our project—proving that heart rate can be extracted from video sources—has been achieved. Our realizations during the project highlight how crucial it is to carry out more research and development to identify practical noise reduction methods. Notwithstanding the difficulties we have faced, we are confident in the potential of our strategy and strongly support continued attempts to improve and perfect our methods. We advise concentrating on creating strong noise reduction strategies going forward to enhance the precision and dependability of our heart rate monitoring software in different lightning conditions and other factors that generate noise.

5 References

1. Bush, I. (2016). Measuring Heart Rate from Video. Stanford Computer Science.
2. Hill, B. L., Liu, X., McDuff, D. (2021). Beat-to-Beat Cardiac Pulse Rate Measurement From Video. University of California, Los Angeles.
3. Lee, Y. C., Syakura, A., Khalil, M. A., Wu, C. H., Ding, Y. F., Wang, C. W. (September 2020). A real-time camera-based adaptive breathing monitoring system.
4. Molinaro, N., Schena, E., Silvestri, S., Bonotti, F., Aguzzi, D., Viola, E., Buccolini, F., Massaroni, C. (February 2022). Contactless Vital Signs Monitoring From Videos Recorded With Digital Cameras: An Overview.
5. Viola, P., Jones, M. (2001). Rapid Object Detection using a Boosted Cascade of Simple Features. Mitsubishi Electric Research Labs, Compaq CRL.
6. Chen, X., Cheng, J., Song, R., Liu, Y., Ward, R., Wang, Z. J. (October 2019). Video-Based Heart Rate Measurement: Recent Advances and Future Prospects.