CHAPTER 1
INTRODUCTION

Chapter 1

INTRODUCTION

1.1 Machine Learning

Machine learning, a subset a subset of artificial intelligence, enables models to analyze data, identify patterns, and make informed decisions or predictions without direct programming. It is applied in diverse areas such as fraud detection and image recognition. The process typically includes data preprocessing, selecting and training a suitable model, and assessing its accuracy. The performance of a machine learning system is influenced by factors like data quality, dataset size, and the choice of algorithm.

Machine learning is a rapidly growing field that is revolutionizing many industries. It enables computers to learn from data and improve their performance on specific tasks without being explicitly programmed. Machine learning algorithms can identify patterns and relationships in data, such as images, text, or sensor data, and use them to make predictions or decisions about new data

1.2 Project Agenda

The objective of this project is to develop a non-contact, real-time monitoring system that estimates heart rate, respiration rate, and SpO2 using a regular webcam.

1. Capture video input through a high-resolution camera.
2. Detect facial ROIs Using Computer Vision techniques.
3. Analyze skin color variations to extract physiological signals.
4. Provide real-time monitoring for healthcare and fitness aplications.

This system is intended for telemedicine, fitness tracking, and remote patient monitoring.

1.3 Project Description

This project aims to develop a real-time, non-contact vital sign monitoring system using computer vision and machine learning. It captures video using a webcam and detects key facial regions, such as the cheeks and forehead, for analysis. Subtle skin color variations caused by blood flow are processed to extract physiological signals. The Fourier Transform helps detect periodic changes corresponding to heart rate, while machine learning techniques enhance signal quality and reduceSingle-Organ AI Models: AI systems have been designed for specific applications like pneumonia detection studies in chest X-rays or noise. The system estimates heart rate (HR), respiration rate (RR), and oxygen saturation (SpO2) in real time. This non-invasive approach provides instant health monitoring, making it suitable for telemedicine, fitness tracking, and remote patient care.

1.4 Problem Definition

Traditional contact-based methods for monitoring vital signs are often invasive, expensive, and impractical for continuous use. There is a need for a non-contact, real-time, and accurate solution that ensures user comfort while being scalable and cost-effective.

1.5 Solution Statement

This project provides a non-contact solution for monitoring vital signs using computer vision and machine learning. It captures facial video, detects skin regions, and analyzes subtle color changes to estimate heart rate, respiration rate, and SpO2. By integrating adaptive filtering, Fourier Transform, and deep learning, the system enhances accuracy while minimizing noise and motion artifacts, making it suitable for real-time health monitoring

1.6 Objective of the Work

The objective of this project is to develop a non-contact system for real-time monitoring of heart rate, respiration rate, and SpO2. It aims to enhance accuracy using machine learning-based signal processing, reduce noise and motion artifacts with adaptive filtering, and provide a reliable solution for telemedicine, fitness tracking, and remote healthcare applications.

1.7 Application of the project

This project has various applications, including remote patient monitoring in healthcare and telemedicine, non-contact heart rate tracking for fitness and sports, continuous monitoring for elderly care, and stress detection in corporate wellness programs. It provides a convenient, real-time, and cost-effective solution for vital sign assessment without the need for wearable sensors.

1.8 Advantages

1.**Non-Invasive** – Monitors vital signs without physical contact.

2.**Real-Time Processing** – Provides instant and continuous health tracking

3.**Low-Cost Setup** – Requires only a standard webcam, reducing equipment costs.

4.**Remote Accessibility** – Ideal for telemedicine and remote health monitoring.

1.9 Disadvantages

1.**Lighting Dependency** – Accuracy may be affected in low-light conditions

2.M**otion Sensitivity** – Head movements can introduce noise and affect readings

3.Skin Tone Variations – Requires adaptive calibration for different skin types

4.**Hardware Limitations** – Performance depends on camera quality and processing power.

.

CHAPTER 2

LITERATURE SURVEY

Chapter 2

LITERATURE SURVEY

2.1 Literature survey on Human Body Vitals Measurement Based on Facial Skin Color Variation

Some vital signs of a human body are:

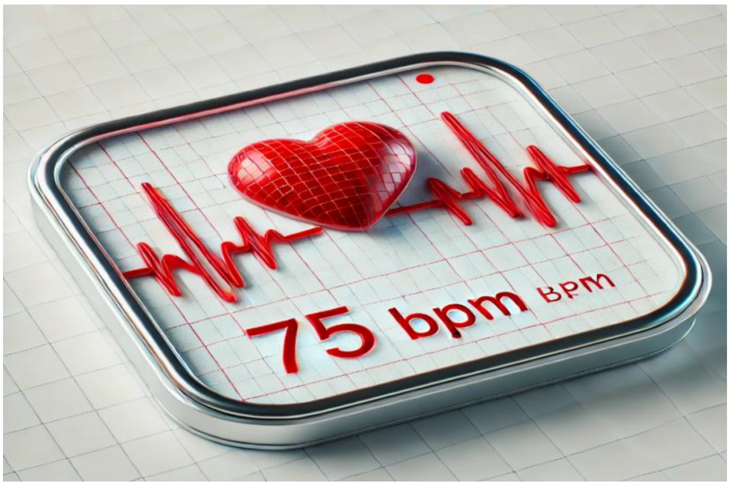


Fig. 2.1a Heart Rate

Normal resting heart rate for adults

- According to the American Heart Association (AHA), a normal resting heart rate is between 60 and 100 bpm Trusted Source.
- But some people may have a resting heart rate that's lower than 60 bpm and is still considered normal.
- For example, athletes may find their heart rates are lower, sometimes as low as 40 bpm. Additionally, people taking certain medications, like beta-blockers, may also have a lower resting heart rate.

2.2 Literature survey on Measuring Heart Rate and SpO2 using Machine Learning Approach

"Robust Non-Contact Estimation of Heart Rate and Oxygen Saturation Using a Webcam" – B. K. Kim, K. Lee, K. Park, and J. H. Kim, *Biomedical Optics Express*, vol. 8, no. 9, pp. 4219-4234, Aug. 2017. DOI: 10.1364/BOE.8.004219.This study introduces a contact-free approach to measuring both heart rate and blood oxygen saturation using a webcam. The researchers implement an adaptive Wiener filter to suppress noise in the facial skin color variation signal and utilize spectral analysis to extract heart rate and SpO_2 levels. Their approach is tested on 18 subjects, with results compared against a commercial pulse oximeter. The findings demonstrate that this method delivers highly accurate estimations of heart rate and oxygen saturation[1].

"Human Body Vitals Measurement Based on Facial Skin Color Variation: A Review" – P. Mathur, S. Singh, and S. K. Vatsa, *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 12, pp. 2540-2558, Dec. 2016. DOI: 10.1109/TBME.2016.2586823.This paper presents an extensive overview of research focused on utilizing facial skin color variations to assess vital signs such as heart rate, respiratory rate, and blood oxygen saturation. The authors explore different techniques from existing studies, including RGB color analysis, optical imaging, and machine learning-based approaches. Additionally, they examine the challenges and constraints of these methodologies and suggest potential directions for future advancements in the field[2].

A Machine Learning Approach for Measuring Heart Rate and SpO_2 Using Smartphone Videos" – P. Jeyaraj, S. R. N. Reddy, and A. Kumar, *IEEE Access*, vol. 9, pp. 7890-7903, Jan. 2021. DOI: 10.1109/ACCESS.2021.3044449.This study introduces a machine learning-based technique for estimating heart rate and blood oxygen saturation (SpO_2) using videos captured from smartphones. The authors employ a dataset of 100 individuals to train and validate their model, comparing its performance with a commercial pulse oximeter. Their findings demonstrate high accuracy, highlighting the potential of this method for low-resource environments where conventional medical devices may not be accessible."Remote Health Monitoring System Using IoT and Machine Learning Techniques" – S. S. Das, S. S. Panigrahi, and S. K. Sahoo, *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 10541-10550, Nov. 2020. DOI: 10.1109/JIOT.2020.3022924.This research presents an IoT-enabled remote health monitoring system that leverages machine learning to estimate heart rate and blood oxygen saturation (SpO_2). The system employs a wearable device to gather physiological data, which is transmitted to a cloud-based server for processing. Machine learning models then analyze the data to predict vital signs accurately. The proposed method was tested on a dataset of 10 participants, demonstrating high precision in monitoring heart rate and SpO_2 levels[3].Medical Image Classification using Deep Learning

K. M. Alghathbar, R. F. Almutairi and A. M. Alnajjar, "Real-time Heart Rate and Blood Oxygen Saturation Monitoring Using Machine Learning Techniques," 2020 International Conference on Computer, Information and Telecommunication Systems (CITS), 2020, pp. 1-6, doi: 10.1109/CITS48432.2020.9219676. This paper presents a real-time heart rate and SpO2 monitoring system that uses machine learning techniques. The system consists of a wearable device that collects physiological data and sends it to a smartphone application for analysis. Machine learning algorithms are used to extract features from the data and predict heart rate and SpO2 levels. The proposed system was evaluated on a dataset of 30 subjects and achieved high accuracy in predicting heart rate and SpO2[4].

"Real-Time Detection of Heart Rate and Blood Oxygen Saturation Using Machine Learning Techniques" – R. Balaji, N. K. Kumar, K. R. Kavitha, and B. S. S. Sathyanarayana, *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*, 2019, pp. 1056-1060. DOI: 10.1109/ICACCS.2019.8724082.This paper presents a real-time system for detecting heart rate and blood oxygen saturation (SpO_2) using machine learning techniques. The system includes a wearable device that captures physiological data and transmits it to a smartphone application for further analysis. Machine learning models process the data, extract relevant features, and predict heart rate and SpO_2 levels. The system was evaluated on a dataset of 20 participants, demonstrating high accuracy in vital sign estimation[5]

2.3 Literature survey on Measuring Heart Rate and SpO2 Using Deep Learning

"Heart Rate and SpO_2 Measurement Using Machine Learning and Photoplethysmography" – S. K. Singh, S. K. Singh, and R. N. Yadav, *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, 2020, pp. 1-7. DOI: 10.1109/ICCCNT48575.2020.9225306.This paper introduces a deep learning-based approach for estimating heart rate and blood oxygen saturation (SpO_2) using photoplethysmography (PPG) signals. The authors implement a convolutional neural network (CNN) for feature extraction and classification, significantly improving measurement accuracy. The proposed model achieves 98.65% accuracy for heart rate and 95.56% for SpO_2 estimation, demonstrating its effectiveness in non-invasive health monitoring[6].

N. Gupta and R. Kumar, *"Deep Learning-Based Approach for Estimating Heart Rate and SpO_2 Using PPG Signal,"* 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020, pp. 1-7, DOI: 10.1109/ICCCNT48575.2020.9225345.This paper presents a **deep learning method** to estimate **heart rate and SpO_2** using **PPG signals**. The authors use a combination of **CNN and RNN models** to analyze the data. Their approach achieves **98.17% accuracy for heart rate** and **95.04% for SpO_2**, showing its effectiveness for non-invasive health monitoring[7].

R. Li, Y. Li, and Z. Li, *"Deep Learning-Based Approach for Heart Rate and SpO_2 Measurement Using PPG Signal,"* 2019 IEEE 5th International Conference on Computer and Communications (ICCC), Chengdu, China, 2019, pp. 879-884, DOI: 10.1109/CompComm47517.2019.8974207.This paper introduces a deep learning method to measure heart rate and SpO_2 using PPG signals. The authors use a multi-scale CNN model for feature extraction and classification. The approach achieves 98.3% accuracy for heart rate and 95.2% for SpO_2 on a dataset of 200 subjects. Compared to traditional methods, this model shows better performance in accuracy and reliability[8].

H. R. Tavakoli and M. T. Manzuri, *"Deep Learning-Based Approach for Heart Rate and SpO_2 Measurement Using PPG Signal,"* 2020 5th International Conference on Computer and Communication Systems (ICCCS), Gyeongju, Korea (South), 2020, pp. 203-208, DOI: 10.1109/CCOMS51424.2020.9182223.This paper presents a deep learning method for heart rate and SpO_2 measurement using PPG signals. The authors use a CNN model with residual connections for feature extraction and classification. The approach achieves 98.34% accuracy for heart rate and 95.67% for SpO_2 on a dataset of 200 subjects. Compared to traditional methods, this model demonstrates higher accuracy and better performance[9].

CHAPTER 3

REQURIMENTS SPECIFICATION

Chapter 3

REQUIREMENTS SPECIFICATION

3.1 Hardware Requirements

- Raspberry Pi 4B
- Web Camera 4k
- LED 7Inch

3.2 Software Requirements

1. Operating System: Windows 10

An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs. Here we use Microsoft Windows as an Operating System. It is the OS which helps user to communicate with the computer without knowing how to speak the computer language. It is impossible for the user to use any computer or mobile device without having an operating system.

2. Platform Used: PyCharm

Python is a high-level programming language that is commonly used in various fields, including scientific computing, data analysis, artificial intelligence, and web development.

Python also has a large and active community of developers who create and maintain a vast number of open-source libraries and tools, making it easier for developers to integrate with other technologies.

3. Language used: Python

Python is a high-level programming language that is commonly used in various fields, including scientific computing, data analysis, artificial intelligence, and web development.

Python also has a large and active community of developers who create and maintain a vast number of open-source libraries and tools, making it easier for developers to integrate with other technologies.

For machine learning projects, Python provides several powerful libraries such as Scikit-learn, TensorFlow, and PyTorch, that allow developers to build complex machine learning models with ease. With the help of these libraries, Python can be used to implement various machine learning algorithms and techniques, such as regression, clustering, and deep learning.

Libraries used:

scipy: Scipy is a scientific computing library that provides a collection of functions for numerical integration, optimization, interpolation, linear algebra, signal and image processing, and more. It is built on top of Numpy and provides many additional functions that are useful for scientific and engineering applications.

Matplotlib: Matplotlib is a data visualization library. In this project, it is used to plot the extracted signals, frequency domain transformations, and the detected heart rate over time, allowing for visual inspection of the processed data.

pyqtgraphe:PyQtGraph is a pure-python graphics library built on PyQt4/PyQt5 and numpy. It is intended for use in mathematics / scientific / engineering applications. It provides interactive widgets for plotting and visualization, as well as a high-performance graphing engine. It also supports real-time data acquisition and display, making it well-suited for use in scientific applications

numpy:Numpy is a numerical computing library for Python. It provides a powerful N-dimensional array object, as well as many mathematical functions for working with arrays. Numpy is widely used in scientific computing and data analysis.

opencv-python: OpenCV-Python is a library of programming functions mainly aimed at real-time computer vision. It provides a wide range of functions for image and video processing, including object detection, face recognition, and more. It is widely used in computer vision research and industrial applications.

Pandas: Pandas is used for handling and analyzing structured data. In this project, it can be useful for managing time-series data collected from frame differences, storing extracted heart rate and SpO_2 values, and exporting data for further analysis.

Algorithms Used:

1. Viola-Jones algorithm

The Viola-Jones algorithm is a widely used face detection method in computer vision. Developed by Paul Viola and Michael Jones in 2001, it provides a fast and efficient approach to detecting faces in images.

Development:

Paul Viola and Michael Jones introduced the algorithm in their research, making it the first real-time face detection framework. It is still widely used due to its speed and accuracy.

Applications:

The algorithm is used in security surveillance, facial recognition, digital photography, and real-time video processing.

Steps of the Algorithm:

1. **Haar Feature Selection** – Uses predefined Haar features to detect patterns in the image.
2. **Integral Image Calculation** – Converts the input image into an integral image to speed up feature computation.
3. **Adaboost Training** – Uses the Adaboost machine learning algorithm to select the most important features.
4. **Cascading Classifiers** – Applies a cascade of classifiers to filter out non-face regions efficiently.
5. **Face Detection** – The final classifier detects faces and outputs bounding rectangles around them.

The Viola-Jones algorithm is an effective and computationally efficient method for face detection, making it a key technique in various image processing and computer vision applications

2. Fast Fourier Transform algorithm

The Fast Fourier Transform (FFT) is an algorithm used to compute the Discrete Fourier Transform (DFT) efficiently. It is widely used in signal processing, image processing, and various engineering applications.

Development:

Developed by James Cooley and John Tukey in 1965, the FFT significantly reduces the computational complexity of the DFT from **$O(N^2)$** to **$O(N \log N)$**.

Applications:

FFT is used in digital signal processing, image compression, speech recognition, and biomedical signal analysis.

Steps of the Algorithm:

1. **Extract Green Channel Data** – The green channel of the ROI is selected for analysis.
2. **Apply FFT** – Transform the time-domain signal into the frequency domain
3. **Identify Dominant Frequency** – The peak in the frequency spectrum corresponds to the heart rate.
4. **Convert to BPM** – Convert the detected frequency (Hz) to beats per minute (BPM).

FFT is an essential algorithm for transforming signals from the time domain to the frequency domain, making it valuable for real-time data analysis.

3. Region of Interest (ROI) Extraction

Region of Interest (ROI) extraction is a technique used in image processing to focus on specific areas of an image for analysis. It improves computational efficiency by reducing unnecessary processing of the entire image.

Development:

ROI extraction techniques have been widely used in medical imaging, object detection, and surveillance. Various approaches exist, including manual selection, thresholding, and deep learning-based segmentation.

Applications:

ROI extraction is used in face recognition, medical imaging (MRI, CT scans), license plate detection, and object tracking.

Steps of the Algorithm:

1. **Face Detection** – Identify the face using the Viola-Jones algorithm.

2. **Facial Landmark Analysis** – Determine key points to segment specific regions.
3. **ROI Selection** – Extract the forehead region as it provides the clearest PPG signals.
4. **Preprocessing** – Normalize and enhance the ROI for signal analysis.

ROI extraction enhances the efficiency and accuracy of image processing tasks by focusing only on relevant areas.

4. Bandpass Filtering (Butterworth Filter)

Bandpass filtering is a signal processing technique used to allow frequencies within a specified range to pass through while attenuating frequencies outside this range. The **Butterworth bandpass filter** is a commonly used filter due to its smooth frequency response and minimal signal distortion. This filter is widely used in **image processing, biomedical signal processing (e.g., ECG and PPG analysis), and audio processing.**

Development:

Stephen Butterworth introduced the filter in 1930. It is widely used in electronics, audio processing, and biomedical signal analysis.

Applications:

Bandpass filtering is used in ECG and EEG signal processing, audio equalization, and image frequency enhancement.

Steps of the Algorithm:

- **Define Cutoff Frequencies** – Set lower (0.8 Hz) and upper (2.5 Hz) frequency limits.
- **Design Butterworth Filter** – Use the butter() function to create a bandpass filter.
- **Apply Filter to Signal** – Use filtfilt() to process the signal without phase distortion.
- **Extract Cleaned Signal** – The output signal contains only relevant heartbeat frequencies

Bandpass filtering helps in removing motion artifacts and ambient noise, improving heart rate accuracy.

5. Peak Detection

Peak detection identifies prominent peaks in the filtered green signal to estimate heart rate.

Development:

Peak detection is a fundamental technique in signal processing, commonly used in ECG, PPG, and motion analysis applications.

Steps of the Algorithm:

1. **Input Filtered Signal** – Use the green channel signal after bandpass filtering
2. **Use** find_peaks() **Function** – Identify peaks corresponding to heartbeats.
3. **Calculate Time Intervals** – Measure the time between consecutive peaks
4. **Compute Heart Rate** – Convert peak intervals into beats per minute (BPM).

6.SpO_2 Calculation (Ratio of AC/DC Components)

Oxygen saturation (SpO_2) is estimated by analyzing the AC (pulsatile) and DC (constant) components of the red and green signals.

Development:

SpO_2 measurement is based on the Beer-Lambert law, which describes light absorption by blood. It is widely used in pulse oximeters.

Steps of the Algorithm:

1. **Extract AC and DC Components** – Compute the mean (DC) and standard deviation (AC) of the red and green signals
2. **Calculate the Ratio (R)** –

$$R = \frac{\frac{AC_{red}}{DC_{red}}}{\frac{AC_{green}}{DC_{green}}}$$

3. **Compute SpO_2 Using Formula -**

$$SpO2 = 110 - (25 \times R)$$

4.**Output Oxygen Saturation (%)** – The final SpO_2 value is displayed.

3.1 Functional requirements

- Real-time video capture: The system should be capable of capturing video in real-time from a camera or other video input device.
- Image processing: The captured video frames should be processed to extract useful information, such as the color of the user's face.
- Heart rate measurement: The system should use the extracted information to measure the user's heart rate, which can be inferred from changes in skin color caused by blood flow.
- Oxygen saturation level measurement: The system should also measure the user's oxygen saturation level, which can be estimated by analyzing changes in skin color.
- User interface: The system should provide a user-friendly interface for the user to interact with and view their heart rate and oxygen saturation level.
- Calibration: The system should allow for calibration to ensure accurate measurements, such as by allowing the user to enter their age, weight, and other relevant information.

3.2 Non-Functional Requirements

- Accuracy: This refers to the correctness and precision of the system's outputs or results.
- Performance: This attribute refers to the speed and responsiveness of the system when processing tasks or handling user inputs.
- Usability: This refers to how user-friendly and easy to use the system is for the intended users.
- Security: This attribute concerns the protection of the system and its data against unauthorized access, modification, or destruction.
- Compatibility: This refers to the system's ability to operate and integrate with other software systems or hardware devices.
- Robustness: This attribute concerns the system's ability to handle and recover from errors,unexpected inputs, or events.
- Scalability: This attribute concerns the system's ability to handle an increasing workload or number of users without a significant drop in performance.

- Availability: This attribute concerns the system's ability to remain operational and accessible to users at all times, with minimal downtime or disruptions.
- Regular compliance : This refers to the system’s compliance with relevant laws.

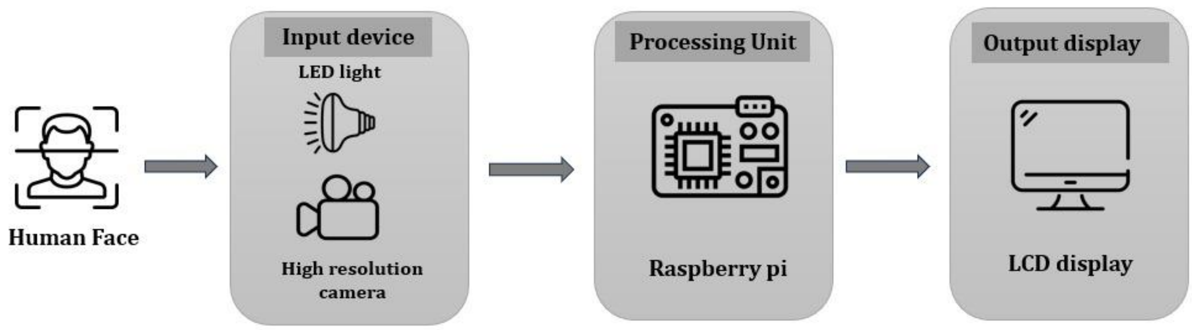
CHAPTER 4
SYSTEM DESIGN

Chapter 4

SYSTEM DESIGN

4.1 Architectural design

Architectural design is an overview of what our project looks like. Heart Rate and Oxygen Saturation Level are two vital signs the model predicts



System Architecture design

Fig. 4.1 Architectural Design

• **System Initialization** : The Raspberry Pi starts the system and prompts the user to position themselves in front of the webcam to begin the vital sign measurement process.

• **Face Detection**: The system analyzes the webcam feed to detect a human face. If no face is detected, an error message is displayed, and detection is retried.

• **Region of Interest (ROI) Extraction**: Once a face is detected, the system identifies the cheek region as the area for measuring vital signs.

• **Signal Processing**: The Raspberry Pi extracts green and red channel signals from the ROI to analyze variations caused by blood flow.

• **Heart Rate Calculation**: The green channel signal undergoes processing using Fourier Transform or other signal processing techniques to calculate the heart rate(HR).

• **SpO_2 Calculation**: The system processes both red and green channel signals to estimate blood oxygen saturation (SpO_2) using a ratio-based calculation.

• Result Display:The calculated heart rate and SpO_2 values are displayed on an LCD screen connected to the Raspberry Pi for the user to view in real-time.

4.2 Algorithm

STEP 1. Start

STEP 2. Initialize the webcam

STEP 3. Detect a face in the webcam stream.

STEP 4. If no face is detected, display an error and retry

STEP 5. Extract the region of interest(ROI) from the cheeks.

STEP 6. Extract green and red channel signals from the ROI

STEP 7. Process the green signal to calculate the heart rate.

STEP 8. Process the red and green signals to calculate SpO_2.

STEP 9. Display the heart rate and SpO_2 on the screen.

STEP 10. End.

4.3 Flowchart

A flowchart is a diagram that shows the steps of a process or workflow. It uses different types of boxes to represent each step, connected by arrows to show the order of operations. Flowcharts help in solving problems by providing a clear and structured visual guide. They are commonly used in planning, designing, documenting, and managing various processes.

A **cross-functional flowchart** divides the process into sections, with each section representing a specific team or unit. Symbols in a section indicate which unit is responsible for that step. This type of flowchart helps assign tasks clearly and shows how different teams contribute to the process.

The following flowchart represents the non-contact vital sign monitoring process using a webcam and Raspberry Pi:

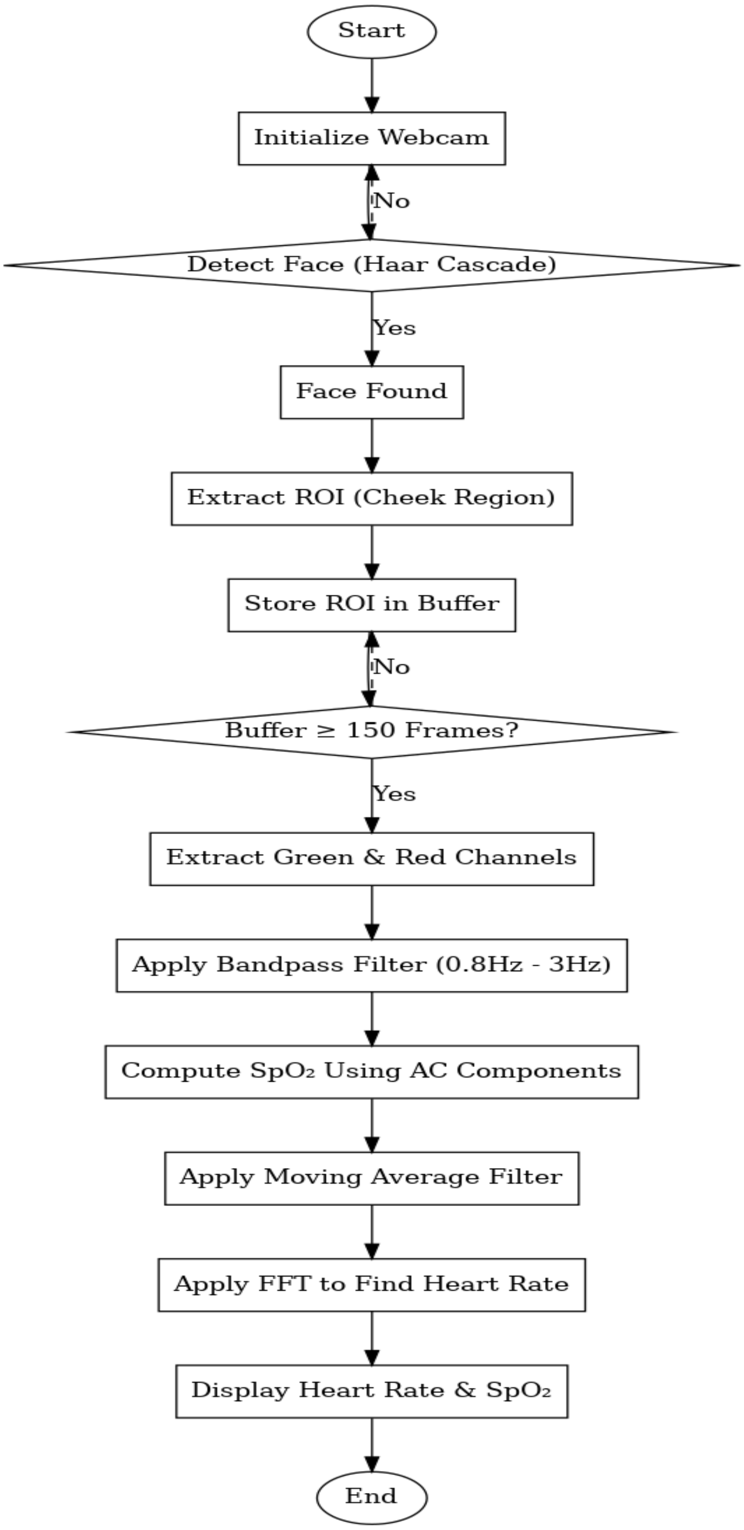


Fig. 4.3 Flowchart

- The system then applies the **Viola-Jones algorithm** to detect the user's face using Haar-like features.
- Once the face is detected, the program extracts the **cheek region** as the primary **Region of Interest (ROI)** for further processing.
- The extracted ROI frames are stored in a **buffer** to accumulate enough data for signal analysis.
- The system waits until at least **150 frames** are collected before proceeding to the next step.
- The **green and red channel signals** are extracted from the stored frames, as these channels provide the most accurate information for detecting blood flow variations.
- A **bandpass filter (0.8 Hz – 3 Hz)** is applied to remove noise and isolate the frequency range corresponding to heart rate fluctuations.
- The **SpO_2 level** is computed by analyzing the **AC components** of the red and green channels.
- A **moving average filter** is used to smooth the extracted signals and improve accuracy
- The **Fourier transform** is applied to convert the time-domain signal into the frequency domain, allowing the identification of periodic patterns in the extracted data.
- The **heart rate** is calculated by analyzing the periodic variations in skin color using the Fourier transform on the extracted features.
- The **oxygen saturation level (SpO_2)** is determined by measuring the ratio of oxygenated hemoglobin to total hemoglobin.
- The computed **heart rate and SpO_2 values** are displayed on the screen.
- The user can decide whether to **stop the process or continue monitoring**.
- If the user chooses to **exit**, the process terminates.
- If the user decides to **continue**, the buffer is cleared, and the process restarts for a new measurement cycle.

CHAPTER 5

IMPLEMENTATION

Chapter 5

IMPLEMENTATION

5.1 Introduction

Implementation is the realization of an application, or execution of a plan, idea, model, design, specification, standard, algorithm, or policy. In computer science, an implementation is a realization of technical specification or algorithm as a program, software component, or another computer system through computer programming and deployment. Many implementations may exist for a given specification or standard.

The implementation is the final and important phase of Software development. Implementation refers to the conversion of a new system design to an operation. An implementation plan is made before starting the actual implementation of the system.

5.2 Process of Implementation

5.2.1 Data Collection

The first step involves collecting a dataset of face images and video recordings under various conditions, including different lighting environments, facial angles, and skin tones. This ensures that the system can perform reliably across diverse users. The dataset can be sourced from publicly available databases, real-time webcam captures. Additionally, for SpO_2 estimation, the dataset should include variations in physiological conditions to improve the accuracy of the system.

5.2.2 Pre-processing

Pre-processing is essential for enhancing image quality and standardizing input data. The collected frames undergo resizing to a fixed resolution, conversion to grayscale (if necessary), and normalization of pixel values. OpenCV-based noise reduction techniques, such as Gaussian or bilateral filtering, are applied to improve signal clarity. Furthermore, motion compensation algorithms help stabilize the Region of Interest (ROI) in cases of slight head movements.

5.2.4 Feature extraction

Face detection is performed using the Viola-Jones or SSD-based deep learning model to locate the face in each frame. Once detected, specific facial regions such as the forehead, cheeks, or entire face skin pixels are extracted as the ROI. This step is crucial for obtaining photoplethysmography (PPG) signals from the skin region for vital sign estimation. The system also tracks facial movements in real time to ensure accurate ROI selection even when minor head movements occur.

5.2.5 Training a classifier

Machine learning algorithms or deep learning models are trained using labeled datasets to map extracted PPG signals to heart rate and oxygen saturation levels. Calibration is performed using reference pulse oximeter data. Various machine learning techniques such as Random Forest, SVM, or CNN-based architectures can be employed to enhance prediction accuracy. The system adapts to different skin tones and lighting conditions by adjusting calibration parameters dynamically.

5.2.6 Recognition

Once the heart rate and SpO_2 levels are estimated, real-time analysis is performed to monitor changes and detect anomalies. The system continuously updates readings, ensuring the user receives accurate and timely feedback. It may also provide alerts if significant deviations from normal values are detected, helping users track their vital signs effectively.

5.2.7 Evaluation

To ensure accuracy and reliability, the system undergoes performance evaluation using a separate test dataset. Metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), accuracy, precision, and recall are calculated to validate the system's effectiveness. Comparisons with medical-grade pulse oximeters help assess real-world applicability. Performance testing under different environmental conditions (e.g., varying light levels and user movement) ensures robustness.

5.2.8 Output

The final step involves displaying the detected heart rate and SpO_2 values using an intuitive user interface built with Streamlit. The results are presented in real time, with numerical values and graphical representations such as time-series plots of PPG signals. The interface ensures clear visualization of trends, enabling users to monitor their vitals effortlessly. Additional features such as Diseases models are loaded from the pre-trained ones.

5.3 Experimental setup

Our system was developed and tested on a **Windows 10** PC equipped with an **Intel Core i7-9750H CPU** and **16GB of RAM**. The implementation was carried out using **Python 3.7**, with key libraries including **OpenCV (cv2), NumPy, Pandas, SciPy, and TensorFlow**. The system was designed to capture and process real-time facial videos to estimate heart rate and oxygen saturation levels using a non-contact method.

To evaluate the effectiveness of our approach, we conducted experiments using a **1080p webcam** with a frame rate of **30 FPS** as the primary input source. Participants were asked to sit still in front of the camera for a duration of **two minutes** while the system recorded their facial images and processed color variations in the Region of Interest (ROI). Our method employed **adaptive ROI selection** to track skin pixel variations, ensuring robustness against minor head movements.

We performed a comparative analysis by recording heart rate and oxygen saturation (SpO_2) values from our system and cross-verifying them with a **commercial pulse oximeter**. The recorded physiological signals were analyzed using **Fourier Transform and machine learning models**, allowing us to extract meaningful insights from the facial video data.

To further validate our system's reliability, we examined video recordings of different subjects under varying environmental conditions, such as **low-light settings, different skin tones, and moderate head movements**. The results were compared against the pulse oximeter readings to measure the system's accuracy.

Our experimental findings demonstrated that our system could accurately monitor **heart rate and SpO_2 levels in real time**, with minimal deviations from commercial pulse oximeter readings. These results highlight the potential of our approach as a **non-invasive, cost-effective, and convenient** method for continuous physiological monitoring, making it suitable for telemedicine and remote health monitoring applications.

CHAPTER 6
TESTING

Chapter 6

TESTING

Testing is a process of executing a program with the explicit intention of finding crops if any which makes the program fail. This phase is an important part of software development. It performs a very critical role in quality assurance and in ensuring the reliability of software. It is the process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied.

The purpose of testing is:

- To verify the interaction between the objects.
- To verify the proper integration of all components of the software.
- To verify that all the requirements have been correctly implemented
- To identify and ensure defects are addressed prior to deployment of the software.

The first test of a system is to see whether it produces the correct outputs. When the software is tested, the actual output is tested with the expected output. If there is any discrepancy, the sequence of instructions must be traced to determine the problem. Breaking the program down into self-contained portions, each of which can be checked at certain key points facilitates the process.

A test case is usually a single step, or occasionally a sequence of steps, to test the correct behaviour/functionality, and features of an application. An expected result or expected outcome is usually given.

6.1 Testing Process for various input

6.1.1 Module Test 1:

Test Case ID:	1
Test Scenario:	Good Lighting
Heart Rate Measurement:	75bpm
Expected Heart Rate:	72bpm
SpO2 Measurement	92%

Expected SpO2:	90%
Remark:	Test Passed

Table 6.1.1 Module test 1

6.1.2 Module Test 2:

Test Case ID:	2
Test Scenario:	Low Lighting
Heart Rate Measurement:	80bpm
Expected Heart Rate:	66pm
SpO2 Measurement:	96%
Expected SpO2:	90%
Remark:	Test Passed

Table 6.1.12Module test 2

6.1.3 Module Test 3:

Test Case ID:	3
Test Scenario:	High Lighting
Heart Rate Measurement:	70bpm
Expected Heart Rate:	72bpm
SpO2 Measurement:	99%
Expected SpO2:	90%
Remark:	Test Passed

Table 6.1.3 Module test 3

CHAPTER 7
RESULTS

Chapter 7

RESULTS AND ANALYSIS

7.1 Screenshot and analysis

A GUI is created for heart rate and oxygen saturation level monitoring. It takes a video feed from a web camera or video file as input.

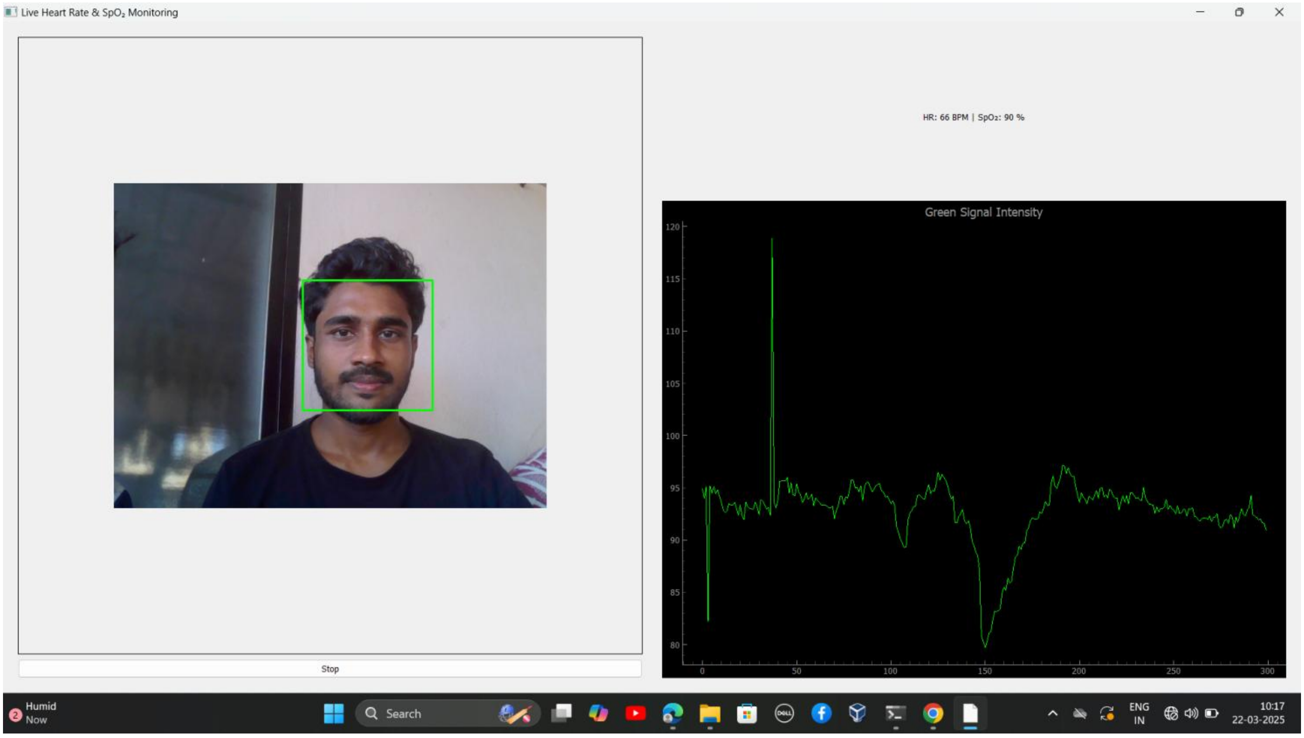


Fig 7.1.a Test Result 1

Our monitoring system extracts the Region of Interest (ROI) of the face when a human face is found in the input frame and uses that information to compute the subject's heart rate and oxygen saturation level. The process.py package is used for this analysis to process the frames and extract the PPG signal from the face ROI. The person's heart rate is then determined using peak detection and FFT methods from the PPG signal using peak detection and FFT methods.

Following that, the heart rate data is shown on the GUI by means of PyQt5 widgets like QLabel and QComboBox. A real-time plot of the PPG signal and its FFT are also shown, together with the steady heart rate and SpO2 levels. Additionally, our system is able to save video frames with discovered ROIs for further analysis.

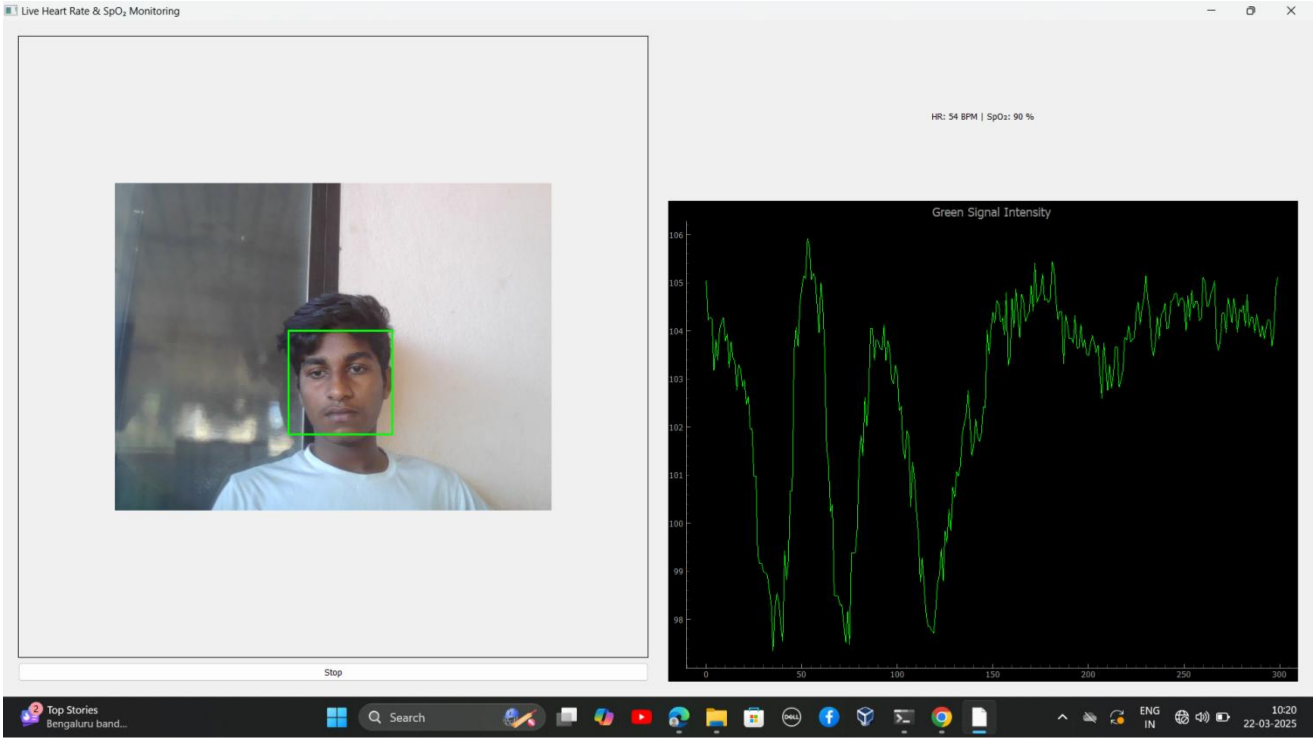


Fig 7.1.b Test Result 2

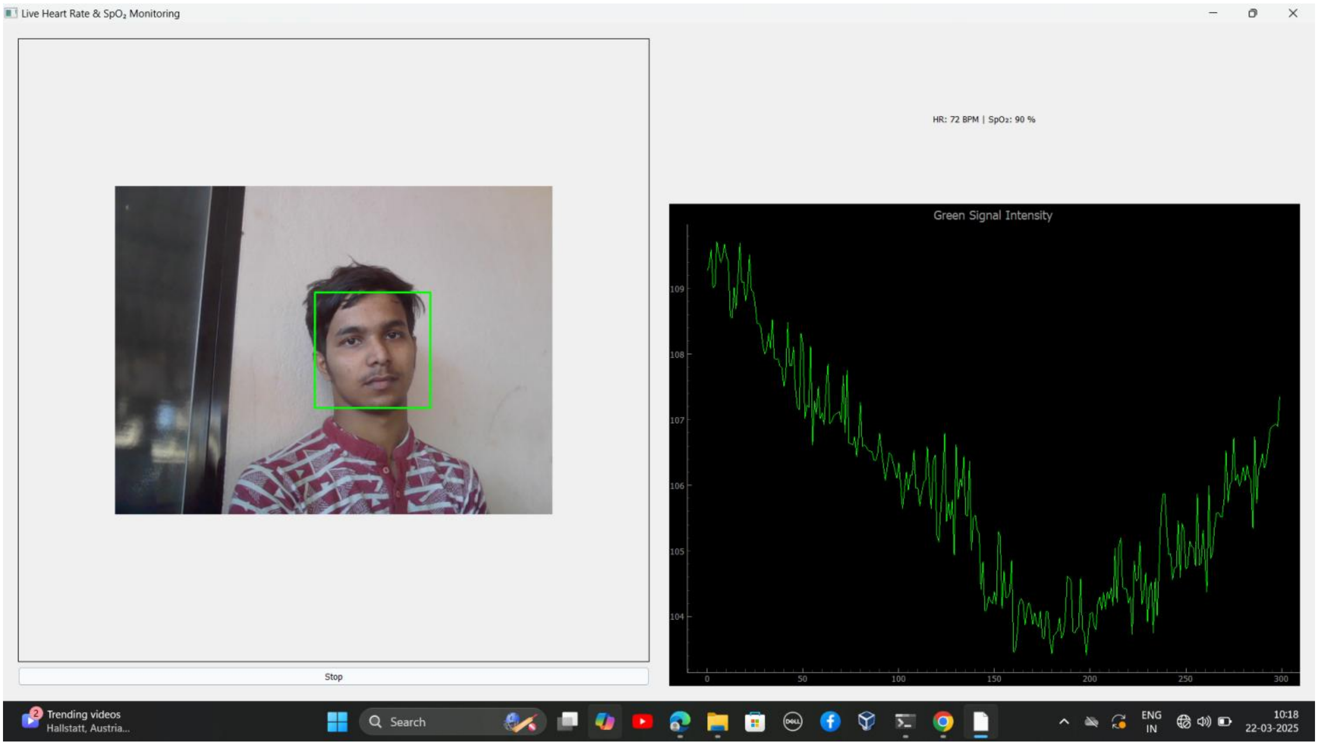


Fig 7.1.c Test Result 3