PupilHeart: Heart Rate Variability Monitoring via Pupillary Fluctuations

Pallanti Lokesh
Reg. No. 24Q71F0040
lokeshpallanti2001@gmail.com
Department of Master of Computer Applications
Avanthi Institute of Engineering and Technology (Autonomous)
Vizianagaram, Andhra Pradesh, India
Under the guidance of Ms.N.Komali, MCA, Assistant Professor
naravakomali066@gmail.com

Abstract—*Heart Rate Variability (HRV) is an important indicator of autonomic nervous system activity and overall cardiovascular health. This project, "PupilHeart: Heart Rate Variability Monitoring via Pupillary Fluctuations," presents a novel, non-invasive approach to HRV estimation using smartphone cameras. The system captures subtle changes in pupil size caused by physiological responses linked to cardiac activity. Advanced image-processing and signal-analysis techniques are applied to extract meaningful patterns from pupillary fluctuations and estimate HRV metrics in real time. Unlike traditional methods that require wearable sensors or electrodes, this approach leverages widely available mobile devices, making HRV monitoring more accessible and cost-effective. The application provides users with insights into stress levels, mental well-being, and cardiovascular condition. By integrating computer vision and mobile-health technologies, PupilHeart offers a convenient and scalable solution for continuous health monitoring, promoting early detection of potential health issues and supporting preventive healthcare practices. The prototype is implemented in Python using OpenCV for video processing, MediaPipe for face and eye-landmark detection, NumPy and SciPy for signal processing, and a Streamlit interface with Plotly visualisation, and was validated through functional test cases covering valid and short videos, no-face and low-quality inputs, and stress-level classification, all of which passed. The system demonstrates how computer vision on commodity smartphones can democratise physiological monitoring.*

Keywords—*Heart Rate Variability; Pupillary Fluctuations; Non-Invasive Monitoring; Computer Vision; MediaPipe; Signal Processing; Mobile Health; Stress Detection.*

I. INTRODUCTION

Heart Rate Variability (HRV) has emerged as a crucial physiological marker for assessing the functioning of the autonomic nervous system and overall cardiovascular health. It reflects the variation in time intervals between consecutive heartbeats and is widely used to evaluate stress levels, emotional well-being, and cardiac conditions. Traditional HRV-monitoring techniques rely on electrocardiograms (ECG) or wearable sensors, which may be expensive, intrusive, and not always accessible to the general population.

With the rapid advancement of mobile technology and computer vision, there is growing interest in developing non-invasive and cost-effective health-monitoring solutions. The proposed system, PupilHeart, introduces an innovative approach to estimating HRV by analysing pupillary fluctuations captured through standard mobile-device cameras. The human pupil exhibits subtle changes in size due to physiological responses regulated by the autonomic nervous system, which are closely linked to cardiac activity. By leveraging image-processing techniques and signal-analysis methods, the system extracts meaningful patterns from video recordings of the eye, which are then processed to estimate HRV metrics in real time, eliminating the need for specialized medical equipment while maintaining reliability and convenience.

The integration of mobile-health (mHealth) technology with computer vision not only enhances accessibility but also enables continuous monitoring in everyday environments, empowering individuals to track their stress levels and cardiovascular health conveniently and promoting preventive healthcare and early detection of potential health issues. Overall, the PupilHeart system represents a significant step toward democratising health monitoring by making advanced physiological analysis available through widely accessible devices such as smartphones.

II. LITERATURE SURVEY

Heart Rate Variability has long been studied as a marker of autonomic nervous system activity and cardiovascular health. The foundational standards by Malik et al. (Task Force, 1996) defined the measurement, physiological interpretation, and clinical use of HRV, establishing the time-domain and frequency-domain metrics that subsequent work builds upon, and Shaffer and Ginsberg (2017) provided an overview of HRV metrics and norms that clarifies how these measures relate to health and stress. These works underpin the HRV metrics that any estimation system, including a camera-based one, must ultimately approximate.

On the computer-vision side, Poh, McDuff, and Picard (2010) demonstrated non-contact, automated cardiac-pulse measurement using video imaging and blind source separation, establishing that physiological signals can be recovered from ordinary camera video and motivating camera-based cardiovascular monitoring. Research on the pupillary system, such as Beatty and Lucero-Wagoner (2000), documents how the pupil responds to autonomic activity, supporting the premise that pupillary fluctuations carry information related to cardiac and stress states. Together, the clinical HRV standards, video-based physiological sensing, and pupillary-response literature motivate PupilHeart's approach of estimating HRV from pupillary fluctuations captured on a smartphone camera, combining computer vision with established signal-processing methods.

TABLE I. SUMMARY OF REPRESENTATIVE PRIOR WORK

S.No	Author(s) / Year	Contribution	Relevance
1	Malik et al. (Task Force), 1996	HRV measurement standards	Defines HRV metrics & clinical use
2	Shaffer & Ginsberg, 2017	Overview of HRV metrics and norms	Reference for HRV interpretation
3	Poh, McDuff & Picard, 2010	Non-contact pulse from video	Camera-based physiological sensing

S.No	Author(s) / Year	Contribution	Relevance
4	Beatty & Lucero-Wagoner, 2000	The pupillary system	Pupil response to autonomic activity

III. EXISTING SYSTEM AND PROPOSED SYSTEM

A. Existing System

Traditional HRV-monitoring systems primarily rely on electrocardiograms (ECG), chest-strap heart-rate monitors, and wearable sensors such as smartwatches and fitness bands. These methods are clinically accurate but require physical contact with sensors or electrodes, can be expensive, and may be intrusive or inconvenient for continuous everyday use. They also depend on specialized hardware that is not always accessible to the general population, limiting their use for large-scale, preventive, or casual self-monitoring.

Limitations of the existing system:

- Requires physical contact with sensors or electrodes.
- Specialized hardware can be expensive.
- Intrusive and inconvenient for continuous daily use.
- Not always accessible to the general population.
- Limited suitability for casual, large-scale self-monitoring.

B. Proposed System

The proposed system introduces a non-invasive HRV-monitoring approach using mobile camera-based video capture. The system records a short video of the user's face/eye, detects the face and eye landmarks, tracks subtle pupillary fluctuations, and applies signal-processing techniques to estimate HRV metrics and classify stress levels in real time. By using a standard smartphone camera rather than dedicated sensors, the approach is accessible, cost-effective, and convenient for continuous monitoring in everyday environments.

Advantages of the proposed system:

- Non-invasive: no physical sensors or electrodes required.
- Uses widely available smartphone cameras; cost-effective.
- Real-time HRV estimation and stress-level classification.
- Convenient for continuous, everyday monitoring.
- Accessible to the general population; scalable.
- Supports preventive healthcare and early detection.

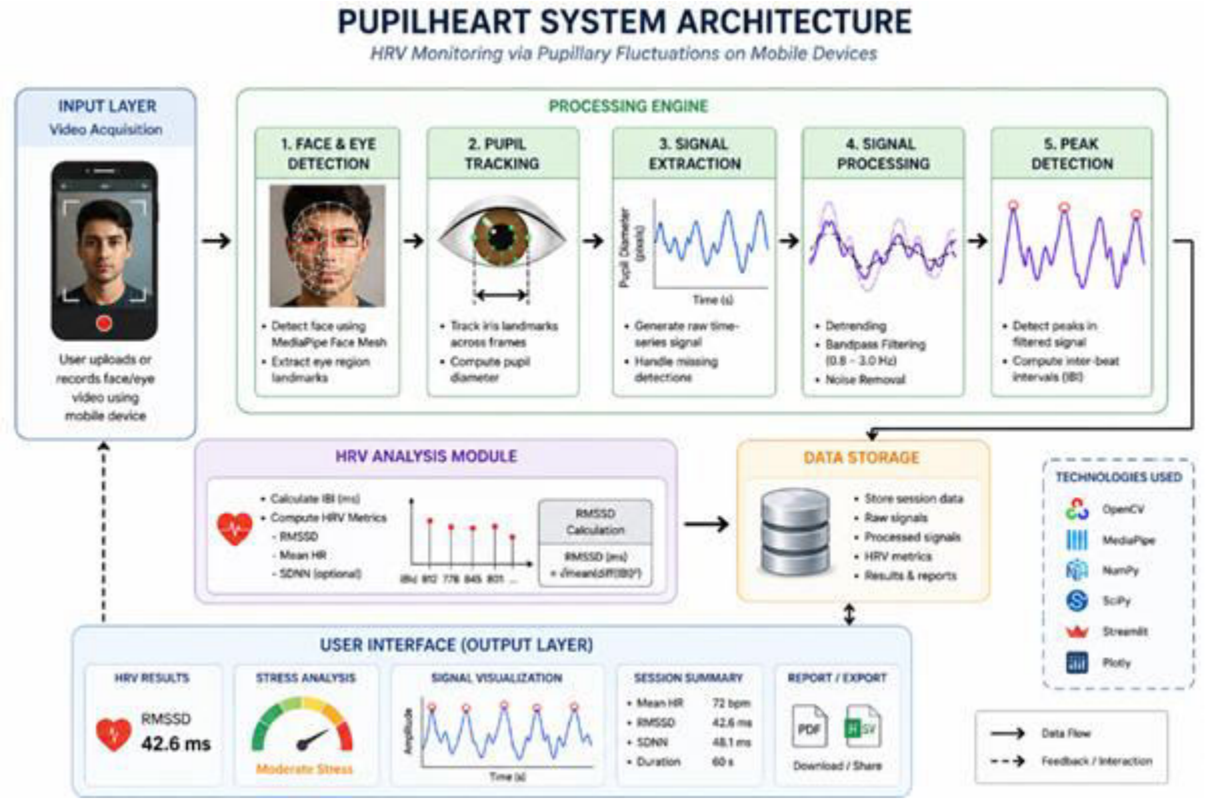
IV. SYSTEM DESIGN AND METHODOLOGY

A. Requirements

Functionally, the system must capture or accept a face/eye video, detect the face and eye landmarks, track pupil-size fluctuations over time, apply signal processing to derive HRV metrics, classify stress level, and present results including a graph. It must validate input quality and handle error cases such as short videos, no face detected, and low-quality video. Non-functional considerations include accessibility on commodity devices, real-time responsiveness, usability through a simple interface, and reasonable robustness to lighting and video-quality variation.

B. System Architecture

The architecture comprises a video-capture layer, a detection layer, a signal-processing layer, and a presentation layer. The video-capture layer obtains eye/face video from a camera or uploaded file. The detection layer uses MediaPipe for face and eye-landmark detection and OpenCV for frame processing, isolating the pupil region and measuring its fluctuations across frames. The signal-processing layer uses NumPy and SciPy to convert the pupillary-fluctuation time series into HRV metrics and to classify stress level. The presentation layer, built with Streamlit and Plotly, displays the HRV value, a graph, and the stress classification.



C. Workflow

A user provides a short face/eye video; the system validates its length and quality; MediaPipe detects the face and eye landmarks while OpenCV processes the frames; the pupil region is tracked and its size fluctuations are recorded over time; SciPy/NumPy signal processing converts these fluctuations into HRV metrics; a stress level is classified; and the result—HRV value, graph, and stress level—is displayed. Error or warning messages are produced for short videos, undetected faces, or low-quality input.

V. SYSTEM IMPLEMENTATION

A. Technology Stack

TABLE II. TECHNOLOGY STACK

Component	Technology / Tool
Programming Language	Python
Video Processing	OpenCV
Face / Eye Landmark Detection	MediaPipe
Signal Processing	NumPy, SciPy
User Interface	Streamlit
Visualisation	Plotly
Input	Smartphone-camera face/eye video
Output	HRV value, graph, stress-level classification

B. Implementation Details

The system is implemented in Python. OpenCV handles video input and frame-level processing, while MediaPipe provides robust face and eye-landmark detection that localises the eye region in each frame. The pupil region is tracked across frames and its fluctuations are recorded as a time series. NumPy and SciPy perform the signal processing that transforms the pupillary-fluctuation signal into HRV metrics and supports stress-level classification. The Streamlit application provides a simple interface for capturing or uploading video, running the analysis, and displaying results, and Plotly renders the HRV graph. The modular design keeps capture, detection, signal processing, and presentation cleanly separated.

C. Signal Processing and Stress Classification

From the tracked pupil-size time series, the signal-processing module derives HRV-related measures and maps them to a stress-level classification, presenting the user with both the numeric HRV output and an interpretable stress indicator. Because the approach is camera-based, the quality of detection and the derived signal depend on video clarity, eye visibility, and lighting, which are handled through input validation and warnings rather than assumed away.

VI. SYSTEM TESTING AND RESULTS

The system was validated through functional test cases covering a valid face video, a short video, a no-face-detected case, a low-quality video, and a normal video for stress classification. All test cases passed and behaved as expected, producing HRV values and graphs for valid input and appropriate error or warning messages for invalid or low-quality input.

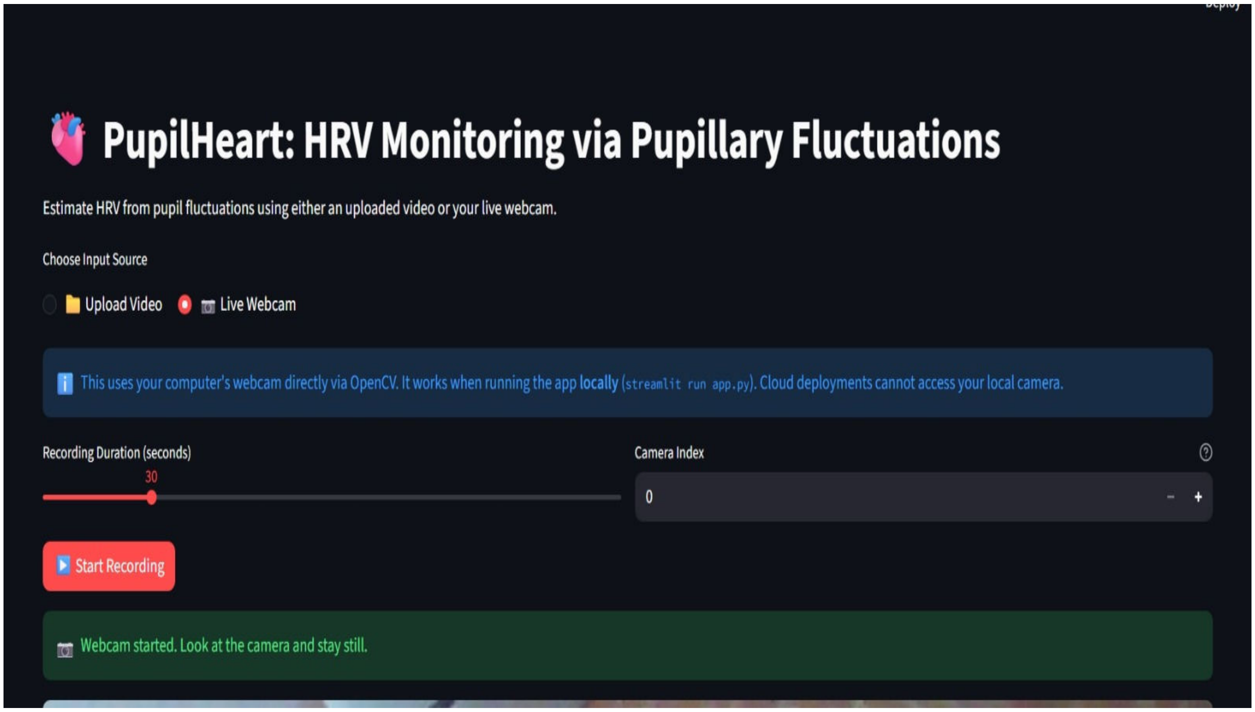
TABLE III. FUNCTIONAL TEST CASES

ID	Input	Expected Output	Result
TC01	Valid face video	HRV value + graph	**Pass**
TC02	Short video	Error message	**Pass**
TC03	No face detected	Warning message	**Pass**
TC04	Low-quality video	Reduced-accuracy warning	**Pass**
TC05	Normal video	Stress-level classification	**Pass**

A. *Observed Results*

Testing confirmed that HRV values were generated correctly, stress levels were classified, graph visualisation worked properly, and the system handled edge cases effectively. Some limitations were identified during testing: performance may reduce under poor lighting conditions, accuracy depends on video clarity and eye visibility, and face detection may fail if the face is partially hidden. The source reports these outcomes qualitatively; no specific numeric accuracy values are claimed here, and real-world performance depends on camera quality and capture conditions.

Representative screenshots from the prototype implementation:



PupilHeart: HRV Monitoring via Pupillary Fluctuations

Estimate HRV from pupil fluctuations using either an uploaded video or your live webcam.

Choose Input Source

○ Upload Video ● Live Webcam

This uses your computer's webcam directly via OpenCV. It works when running the app **locally** (`streamlit run app.py`). Cloud deployments cannot access your local camera.

Recording Duration (seconds): 30

Camera Index: 0

Webcam started. Look at the camera and stay still.

Fig. 1. Video capture / upload interface.

29.9s / 30s • Frames: 723 • Detected: 618

✓ Recording complete! Captured 723 frames at ~24.66 FPS.

Results

HRV (RMSSD)

268.38 ms

Frames Processed

723

Stress Analysis

Normal / Relaxed

Fig. 2. Stress-level classification output.

VII. CONCLUSION AND FUTURE SCOPE

The PupilHeart system successfully demonstrates a novel, non-invasive approach to Heart Rate Variability monitoring using pupillary fluctuations captured through a standard smartphone camera. By combining computer-vision techniques—OpenCV video processing and MediaPipe face/eye-landmark detection—with NumPy/SciPy signal processing and a Streamlit/Plotly interface, the system estimates HRV metrics and classifies stress levels in real time without the wearable sensors or electrodes required by traditional methods. This makes HRV monitoring more accessible, cost-effective, and convenient for continuous, everyday use, supporting preventive healthcare and early detection of potential health issues. Functional testing confirmed correct HRV generation, stress classification, graph visualisation, and effective handling of edge cases, while also identifying practical limitations related to lighting, video clarity, and eye visibility.

Future work can improve robustness and accuracy. Enhancing performance under poor lighting and low-quality video through better preprocessing and more advanced detection would increase reliability, and validation against clinical-grade HRV references (ECG or photoplethysmography) would quantify accuracy. Incorporating machine-learning models for more accurate signal extraction and stress classification, expanding to continuous and longitudinal monitoring, deploying as a full mobile application, and adding personalised baselines and privacy-preserving on-device processing are promising directions for real-world adoption.

REFERENCES

[1] Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology (M. Malik et al.), “Heart Rate Variability: Standards of Measurement, Physiological Interpretation, and Clinical Use,” European Heart Journal, vol. 17, pp. 354–381, 1996.

[2] F. Shaffer and J. P. Ginsberg, “An Overview of Heart Rate Variability Metrics and Norms,” Frontiers in Public Health, vol. 5, art. 258, 2017.

[3] M. Z. Poh, D. J. McDuff, and R. W. Picard, “Non-Contact, Automated Cardiac Pulse Measurements Using Video Imaging and Blind Source Separation,” Optics Express, vol. 18, no. 10, pp. 10762–10774, 2010.

[4] J. Beatty and B. Lucero-Wagoner, “The Pupillary System,” in Handbook of Psychophysiology, 2nd ed., Cambridge University Press, 2000.

[5] OpenCV, “Open Source Computer Vision Library Documentation.” [Online]. Available: https://opencv.org/

[6] Google, “MediaPipe Documentation.” [Online]. Available: https://developers.google.com/mediapipe

[7] SciPy, “SciPy Signal Processing Documentation.” [Online]. Available: https://scipy.org/

[8] Streamlit Inc., “Streamlit Documentation.” [Online]. Available: https://streamlit.io/

[9] Plotly, “Plotly Visualization Library Documentation.” [Online]. Available: https://plotly.com/

[10] NumPy Developers, “NumPy Documentation.” [Online]. Available: https://numpy.org/