Video-Based Heart Rate Estimation Using Facial Features

ABSTRACT:

The project "Video-Based Heart Rate Estimation Using Facial Features" presents a non-contact method for monitoring heart rate through computer vision and signal processing techniques. The system captures a real-time facial video using a webcam and identifies the facial region of interest (ROI) where subtle color changes occur due to blood circulation. These temporal variations in skin tone are analyzed using the principle of remote photoplethysmography (rPPG). The extracted signal is then processed using filtering and Fast Fourier Transform (FFT) to estimate the dominant frequency corresponding to the heart rate in beats per minute (BPM). This approach removes the need for physical sensors such as ECG or pulse oximeters, offering a more comfortable and accessible alternative for health monitoring. The results demonstrate that facial video can be effectively used to estimate heart rate with reasonable accuracy, making this technique suitable for telemedicine, fitness tracking, and human–computer interaction applications.

Keywords: Heart rate estimation, facial video, photoplethysmography (PPG), signal processing, computer vision, FFT, remote monitoring, non-contact measurement.

1.INTRODUCTION:

In recent years, new technologies used for providing clinical health care re motely have appeared and new elds like telemedicine have experienced huge advancements. New ways for monitoring patients automatically have been developed, as well as techniques for measuring physiological parameters out of the hospital.

One of these parameters is the heart rate, and it is usually used by medi cal professionals to assist in diagnosis. However, the information provided by the pulse is not only useful in telemedicine, but also in other elds like auto matic emotion recognition, interactive videogames or sport-people monitoring.

Heart rate is one of the most important physiological indicators used to assess a person's health and physical condition. Conventional methods for heart rate measurement, such as electrocardiograms (ECG) and pulse oximeters, require direct skin contact through sensors or electrodes. Although these methods are accurate, they can be inconvenient for continuous or remote monitoring. With the advancement of computer vision and image processing technologies, non-contact techniques have emerged as promising alternatives. In this project, heart rate is estimated using a standard webcam that captures real-time facial videos. Subtle color fluctuations in specific facial regions, caused by blood volume changes during cardiac cycles, are extracted and analyzed using the principle of remote photoplethysmography (rPPG). The recorded signal is processed using filtering and Fast Fourier Transform (FFT) algorithms to obtain the heart rate in beats per minute (BPM). This video-based approach offers an effective, low-cost, and contactless solution for heart rate monitoring, which can be applied in telemedicine, fitness assessment, and emotion recognition systems.

For this reason, this project addresses the design, evaluation and imple mentation of a system able to estimate the heart rate of a person using only facial video information coming from a standard webcam. By the use of photo plethysmography techniques and data processing tools, the proposed method ology captures the small illumination changes produced in the user's face be cause of the variation in the amount of blood present in the surface of the skin. This technique allows an unobtrusive way to measure people's heart rate at any place, without any e ort more than being in front of a video camera.

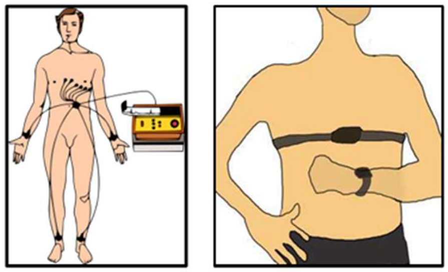


Figure 1.1: Examples of ECG devices.

This project consists in designing, evaluating and implementing a non contact heart rate estimation system using PPG techniques. This work is focused in the development of a real-time application capable of detecting the HR of a person using a standard webcam. This involves capturing the small light fluctuations that are produced in the user face because of the pulse. These small light fluctuations can be sensed by a standard digital camera.

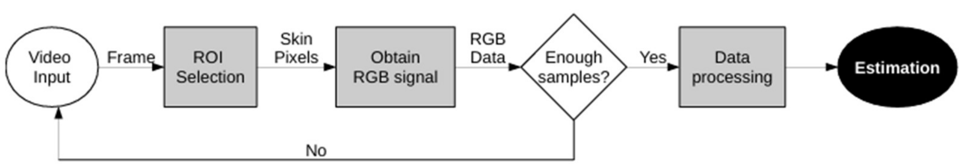


FIG 1.2. Scheme of the real time heart rate estimator system.

Specifically, this work is focused on achieving the following objectives:

- Designing a system that estimates the heart rate of a user using a standard webcam in normal ambient light conditions. The system must work in people with different kind of skins, as well as different ages and genders. Moreover, the system has to be robust to small variations in the position of the user's face.

- Analyzing the accuracy of the system, which must present similar results to the ones obtained with other conventional HR monitor. Besides, the heart rate estimator system must achieve practical results, with special interest in elds like telemedicine or automatically emotion recognition.
- Implementing a final application capable to work in real time, with a reduced exposure period in front of the camera to obtain the first HR estimation.

2.OBJECTIVES:

2.1 FACE DETECTION:

Face detection is the technique that determines locations and sizes of faces in input images. In literature, several algorithms have been developed to solve this problem, however, the method proposed by P. Viola and M. Jones is the first algorithm to provide competitive face detection rates with real-time performance. Viola-Jones algorithm involves obtaining a series of simple features from an input image and applying a modified version of the AdaBoost classification scheme in order to detect human faces. In fact, Viola-Jones is an iterative algorithm that searches faces along the complete image using different scales. To be able to perform real-time detection, a group of weak classifiers are combined in a cascade to form a final complex classifier. Both the image descriptors used for detection and the Ada-Boost-based scheme for selecting weak classifiers are explained below, as well as the cascade construction of the final face detector.

2.2 SKIN PIXEL DETECTION:

As mentioned before, each heart beat produces a variation in the amount of blood present in the face skin, which causes small illumination changes in it. To detect and process these light variations, the first thing we have to do is to locate those regions in the input video image containing facial skin.

The inclusion of non-skin pixels in the selected region, such as hair or background, is a strong drawback as well. By adding a skin pixel detector the system can distinguish skin from non-skin pixels. This reduces the amount of noise coming from non facial skin regions in the image and improves the precision of the system.

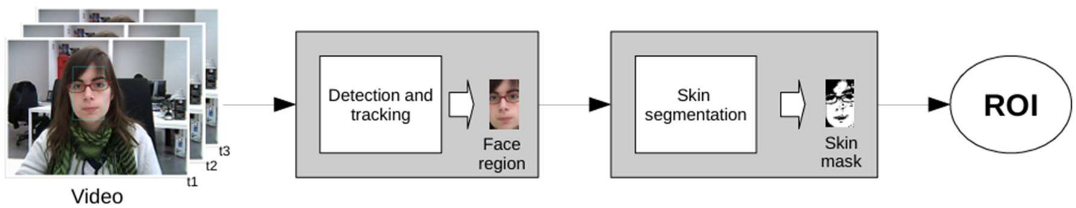


Figure 2.1: Skin pixel detection methodology.

2.3 HAAR CASCADE CLASSIFIER:

The Haar Cascade Classifier is a machine learning–based object detection method used to identify faces or other objects in images and video streams. It was introduced by Viola and Jones and is one of the most widely used algorithms for real-time face detection due to its accuracy and speed.

The classifier is trained using a large number of positive images (containing faces) and negative images (without faces). It uses Haar-like features, which are simple rectangular patterns that measure the contrast between adjacent regions of an image, such as between the eyes and the cheeks or between the nose bridge and the surrounding area. These features help distinguish facial characteristics from the background.

To make the detection fast, the algorithm uses an integral image technique that allows rapid computation of these Haar features. It also employs a cascade of classifiers, where simple classifiers are arranged in multiple stages — each stage filters out non-face regions quickly, and only the most promising regions pass through to the next stage. This step-by-step rejection process allows the algorithm to detect faces efficiently, even in real-time applications.

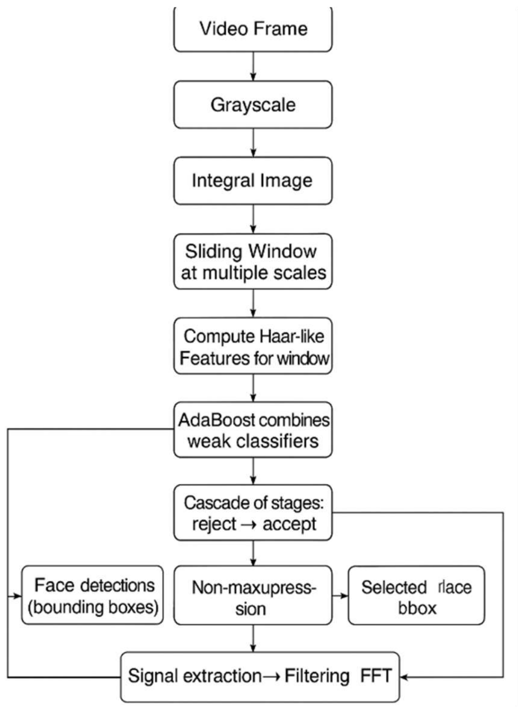


FIG 2.2:Flow diagram of Haar cascade

2.3.1 Purpose of AdaBoost

AdaBoost is used to select the most important Haar features and combine them into a strong classifier. Since an image can contain thousands of possible Haar features, using all of them would be computationally expensive and unnecessary. AdaBoost helps by:

- Picking only the most discriminative features that distinguish a face from a non-face.
- Combining many weak classifiers (each based on a single Haar feature) into a strong classifier that gives high accuracy.

2.4 ROI selection:

For each frame of the video input, which is recorded at the average speed of 20 fps, a ROI is selected with the process described. For this purpose, the Viola-Jones algorithm included in the version of the OpenCV library is used, as well as other computer vision algorithms also included in this library. The tracking and skin pixel selection methods have also been implanted using this library.

In this project, the forehead region is chosen as the ROI because it has smooth skin, minimal facial movement, and fewer occlusions such as facial hair or shadows. After detecting the face, the bounding box coordinates provided by the face detector are used to define a rectangular ROI corresponding to the upper 15% of the face, centered horizontally. This ensures that the selected region consistently covers the forehead area across different subjects.

The ROI is extracted in every frame of the recorded video sequence. The green color channel from the ROI is averaged to obtain an intensity value for each frame, forming a time-series signal. This signal represents subtle changes in skin reflectance caused by variations in blood volume with each heartbeat.

A correctly chosen ROI improves the signal-to-noise ratio (SNR) and ensures that the extracted photoplethysmographic (PPG) signal truly represents physiological activity rather than motion artifacts or illumination changes. Thus, effective ROI selection is fundamental for achieving reliable and stable heart rate estimation.

In the recovery procedure, the raw traces are processed with the methodology shown in Figure 2.3. As it can be observed, first the raw RGB signals are pre processed and treated to become ready for their analysis

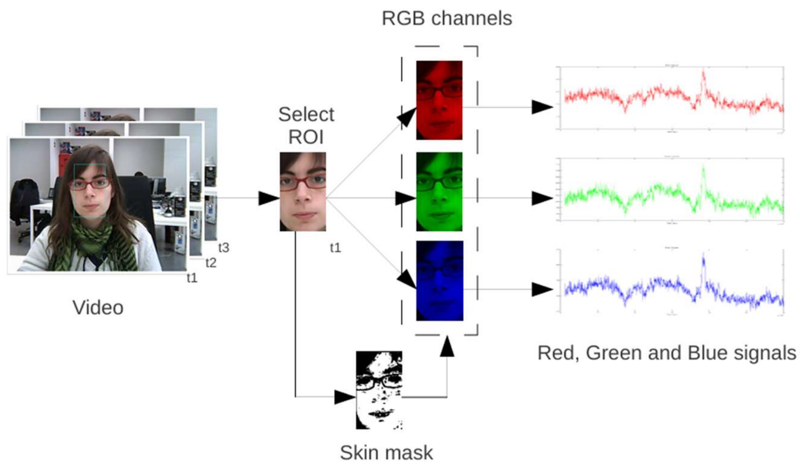


FIG 2.3: RGB traces recovery.

3.METHODOLOGY:

After identifying the Region of Interest (ROI) on the detected face, the system extracts the heart rate signal using remote photoplethysmography (rPPG). The key steps and formulas used in this process are as follows:

3.1: Signal Extraction

From each captured frame, the green channel intensity of the ROI is averaged to obtain a single intensity value representing that frame:

$$G(t_i) = \frac{1}{N} \sum_{x,y \in ROI} I_G\,(x, y, t_i)$$

where:

- $G(t_i)$= average green intensity at frame i

- $I_G(x, y, t_i)$= pixel intensity of the green channel at position $(x, y)$and time t_i
- N= total number of pixels in the ROI

This sequence of $G(t_i)$values over time forms the raw photoplethysmographic (PPG) signal.

3.2: Detrending

To remove slow-varying illumination effects and center the signal around zero, the mean is subtracted:

$$G'(t_i) = G(t_i) - \bar{G}$$

where $\bar{G}$is the mean of all green intensity values. This step helps eliminate low-frequency drift from lighting or minor motion.

3.3: Butterworth Bandpass Filtering

A 3rd-order Butterworth bandpass filter is used to isolate the heart rate frequency band between 1.25 Hz and 2.0 Hz, corresponding to 75–120 BPM.

The normalized cutoff frequencies are calculated as:

$$W_L = \frac{f_L}{f_s/2}, W_H = \frac{f_H}{f_s/2}$$

where:

- $f_L = 1.25\ Hz$(low cutoff)
- $f_H = 2.0\ Hz$(high cutoff)
- f_s= sampling frequency (frames per second)

The filtered signal $S(t)$is then:

$$S(t) = \text{filtfilt}(b, a, G'(t))$$

3.4: Frequency Analysis (FFT)

To determine the dominant heart rate frequency, the Fast Fourier Transform (FFT) is applied to the filtered signal:

$$F(f) = \mid \text{FFT}(S(t)) \mid$$

The frequency axis is computed as:

$$f_k = \frac{k \cdot f_s}{N}, k = 0,1,2, \dots, \frac{N}{2}$$

where $N$is the number of frames (samples). The peak frequency $f_{peak}$is determined as the frequency where $F(f)$is maximum within the defined heart rate range (1.25–2.0 Hz):

$$f_{peak} = \arg \max_{f \in [f_L, f_H]} F(f)$$

3.5: Heart Rate Estimation

Finally, the heart rate in beats per minute (BPM) is calculated by:

$$\text{BPM} = f_{peak} \times 60$$

This gives the average heart rate over the recorded duration.

4.MATLAB CODE

```
clc; clear; close all;


%% --- Settings ---

duration = 20;   % seconds to record

faceDetector = vision.CascadeObjectDetector('FrontalFaceCART');

cam = webcam;

cam.Resolution = '640x480';  % lower resolution for better FPS


disp(" Detecting face... Please stay still with good lighting.");

pause(2);


%% --- Detect Face Once ---

frame = snapshot(cam);

frame = flip(frame, 2);

gray = rgb2gray(frame);

bboxes = step(faceDetector, gray);


if isempty(bboxes)

    clear cam;

    error(" No face detected. Try again with better lighting.");

end


[~, idx] = max(bboxes(:,3).*bboxes(:,4));
```

```
bbox = bboxes(idx,:);

x = bbox(1); y = bbox(2); w = bbox(3); h = bbox(4);


% Define smaller FOREHEAD ROI (top 15% of face box, centered)

roi_y1 = y;

roi_y2 = y + round(0.15*h);

roi_x1 = x + round(0.35*w);

roi_x2 = x + round(0.65*w);


disp(" Face detected. Recording... Stay still!");


%% --- Initialize Arrays ---

green_values = [];

timestamps = [];

start_time = tic;

frame_count = 0;

fps_window = 15;

live_bpm = NaN;


figure('Name','Heart Rate Detection','NumberTitle','off');



%% --- Recording Loop ---

frame_skip = 10;   % detect face every 10 frames
```

```
while toc(start_time) < duration
    frame = snapshot(cam);
    frame = flip(frame, 2);
    frame_count = frame_count + 1;

    % --- Detect face occasionally ---
    if mod(frame_count, frame_skip) == 1
        gray = rgb2gray(frame);
        bboxes = step(faceDetector, gray);

        if isempty(bboxes)
            frame = insertText(frame, [10 40], 'NO FACE DETECTED', ...
                               'FontSize', 16, 'BoxColor', 'red');
            imshow(frame); drawnow limitrate;
            continue;
        else
            [~, idx] = max(bboxes(:,3).*bboxes(:,4));
            bbox = bboxes(idx,:);
            x = bbox(1); y = bbox(2); w = bbox(3); h = bbox(4);
            roi_y1 = y;
            roi_y2 = y + round(0.15*h);
            roi_x1 = x + round(0.35*w);
            roi_x2 = x + round(0.65*w);
        end
```

```
% --- Extract ROI and intensity ---
roi = frame(roi_y1:roi_y2, roi_x1:roi_x2, :);
green = mean(roi(:,:,2), 'all');
green_values(end+1) = green;
timestamps(end+1) = toc(start_time);

% --- FPS Estimate ---
if numel(timestamps) > 5
    fps_live = 1 / mean(diff(timestamps(max(1,end-fps_window):end)));
else
    fps_live = NaN;
end

% --- Live Heart Rate (update every 15 frames) ---
if numel(green_values) > 30 && mod(frame_count, 15) == 0
    fs = max(fps_live, 10); % prevent NaN
    low_cut = 1.0;  % 60 BPM
    high_cut = 2.2; % 132 BPM
    [b,a] = butter(3, [low_cut high_cut]/(fs/2), 'bandpass');
    filtered = filtfilt(b,a, green_values - mean(green_values));

    n = length(filtered);
    f = (0:n-1)*(fs/n);
    fft_val = abs(fft(filtered(1:end)));
```

```
        f = f(1:floor(n/2));
        fft_val = fft_val(1:floor(n/2));

        idx = (f >= low_cut & f <= high_cut);
        [~, pk] = max(fft_val(idx));
        freq_est = f(idx);
        live_bpm = freq_est(pk)*60;
    end

    % --- Display ---
    if mod(frame_count,5)==0
        frame = insertShape(frame,'Rectangle',[roi_x1 roi_y1 (roi_x2-roi_x1) (roi_y2-roi_y1)], ...
                    'Color','green','LineWidth',3);
        frame = insertText(frame,[10 10],sprintf("FPS: %.1f",fps_live),'FontSize',16,'BoxColor','yellow');
        if ~isnan(live_bpm)
            frame = insertText(frame,[10 40],sprintf("HR: %.0f BPM",live_bpm), ...
                        'FontSize',16,'BoxColor','red');
        end
        imshow(frame); drawnow limitrate;
    end
end



    if isempty(bboxes)
```

```
    % No face detected — show message and skip processing
    frame = insertText(frame,[10 40],'NO FACE DETECTED','FontSize',16,'BoxColor','red');
    imshow(frame); drawnow limitrate;
    continue;  % skip this iteration
end

% Proceed only if face detected
[~, idx] = max(bboxes(:,3).*bboxes(:,4));  % largest face
bbox = bboxes(idx,:);
x = bbox(1); y = bbox(2); w = bbox(3); h = bbox(4);

% Define ROI (forehead region)
roi_y1 = y;
roi_y2 = y + round(0.15*h);
roi_x1 = x + round(0.35*w);
roi_x2 = x + round(0.65*w);

% Ensure ROI inside frame boundaries
if roi_y2 > size(frame,1) || roi_x2 > size(frame,2)
    frame = insertText(frame,[10 40],'INVALID ROI','FontSize',16,'BoxColor','red');
    imshow(frame); drawnow limitrate;
    continue;
end
% Extract ROI and compute green intensity
```

```
roi = frame(roi_y1:roi_y2, roi_x1:roi_x2, :);

green = mean(roi(:,:,2), 'all');


% Store values for HR calculation

green_values(end+1) = green;

timestamps(end+1) = toc(start_time);


% Show ROI on frame

frame = insertShape(frame,'Rectangle',[roi_x1 roi_y1 (roi_x2-roi_x1) (roi_y2-roi_y1)],'Color','green');

imshow(frame);

drawnow limitrate;



%% --- FPS Calculation ---

if frame_count > 5

    fps_live = 1 / mean(diff(timestamps(max(1,end-fps_window):end)));

else

    fps_live = NaN;

end


%% --- Live Heart Rate Estimation (update every second) ---

if numel(green_values) > 30 && mod(frame_count, 10) == 0

    fs = fps_live;
```

```
    if isnan(fs) || fs < 5
        continue;
    end
    % Restrict band to 1.25–2.0 Hz (≈ 75–120 BPM)
    low_cut = 1.25;
    high_cut = 2.0;
    if high_cut > fs/2
        high_cut = fs/2 - 0.1;
    end
    [b,a] = butter(3, [low_cut high_cut]/(fs/2), 'bandpass');
    filtered = filtfilt(b,a, green_values - mean(green_values));

    n = length(filtered);
    f = (0:n-1)*(fs/n);
    fft_val = abs(fft(filtered));
    f = f(1:floor(n/2));
    fft_val = fft_val(1:floor(n/2));

    idx = (f >= low_cut & f <= high_cut);
    [~, pk] = max(fft_val(idx));
    freq_est = f(idx);
    live_bpm = freq_est(pk)*60;
end
%% --- Visualization (every 5th frame) ---
```

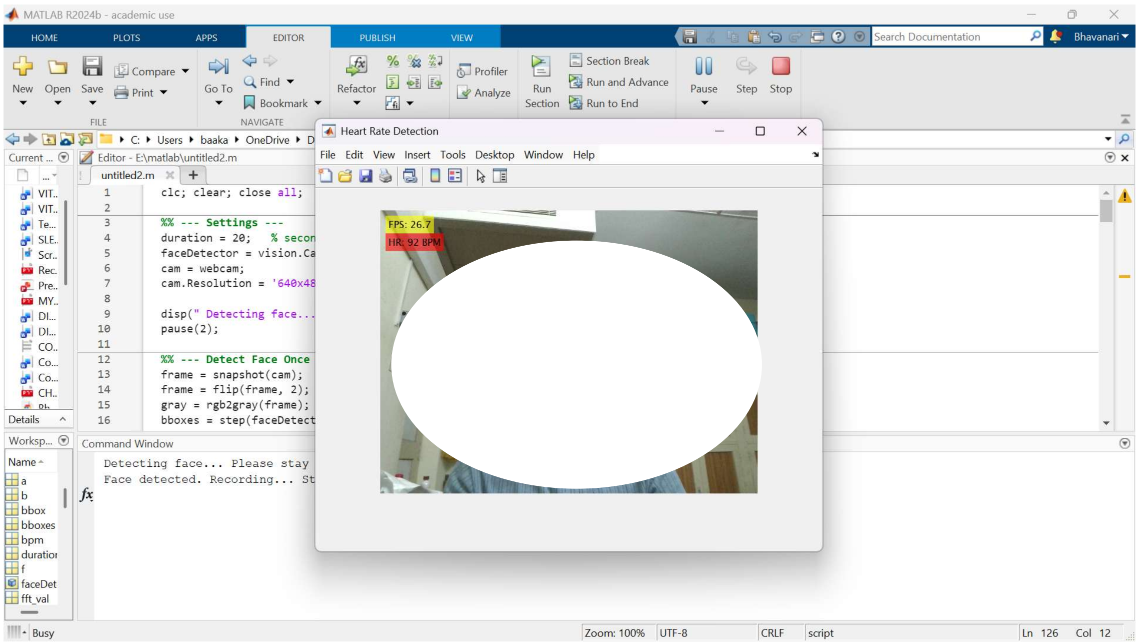
```
    if mod(frame_count,5)==0
        frame = insertShape(frame,'Rectangle',[roi_x1 roi_y1 (roi_x2-roi_x1) (roi_y2-roi_y1)], ...
                        'Color','green','LineWidth',3);
        frame = insertText(frame,[10 10],sprintf("FPS: %.1f",fps_live),'FontSize',16,'BoxColor','yellow');
        if ~isnan(live_bpm)
            frame = insertText(frame,[10 40],sprintf("HR: %.0f BPM",live_bpm),'FontSize',16,'BoxColor','red');
        end
        imshow(frame); drawnow limitrate;
    end
end

clear cam;
disp("Recording complete.");

%% --- Post-processing for Final Heart Rate ---
fps = 1 / mean(diff(timestamps));
green_values = green_values - mean(green_values);

low_cut = 1.25;
high_cut = min(2.0, 0.45*fps);
[b,a] = butter(3, [low_cut high_cut]/(fps/2), 'bandpass');
filtered_signal = filtfilt(b,a, green_values);
n = length(filtered_signal);
```

```
frequencies = (0:n-1)*(fps/n);

fft_values = abs(fft(filtered_signal));

frequencies = frequencies(1:floor(n/2));

fft_values = fft_values(1:floor(n/2));


idx = (frequencies >= low_cut & frequencies <= high_cut);

filtered_freqs = frequencies(idx);

filtered_fft = fft_values(idx);

[~, peak_idx] = max(filtered_fft);

peak_freq = filtered_freqs(peak_idx);

bpm = peak_freq * 60;


fprintf('\n Measured FPS: %.2f\n', fps);

fprintf(' Final Estimated Heart Rate: %.2f BPM\n', bpm);


%% --- Plot ---

figure;

subplot(2,1,1);

plot(timestamps, filtered_signal, 'g');

title('Filtered Green Signal (Forehead ROI)');

xlabel('Time (s)'); ylabel('Intensity');

subplot(2,1,2);

plot(filtered_freqs*60, filtered_fft, 'r');

title('FFT (Heart Rate Detection)');
```

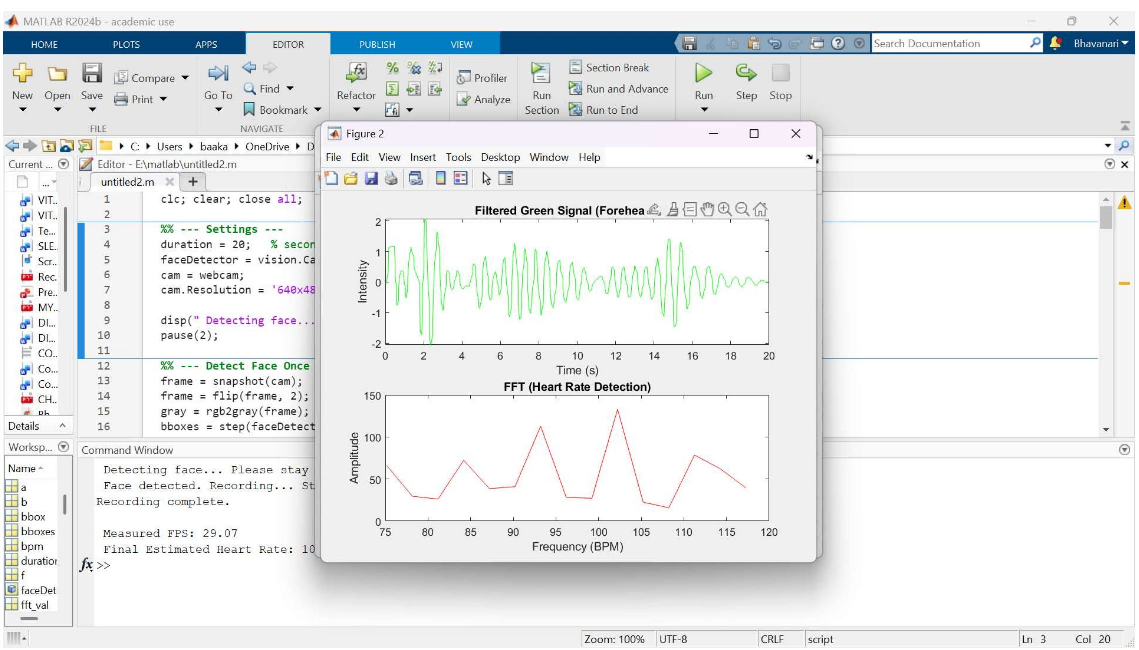
xlabel('Frequency (BPM)'); ylabel('Amplitude');

5.RESULTS:

5.1 Detecting face:

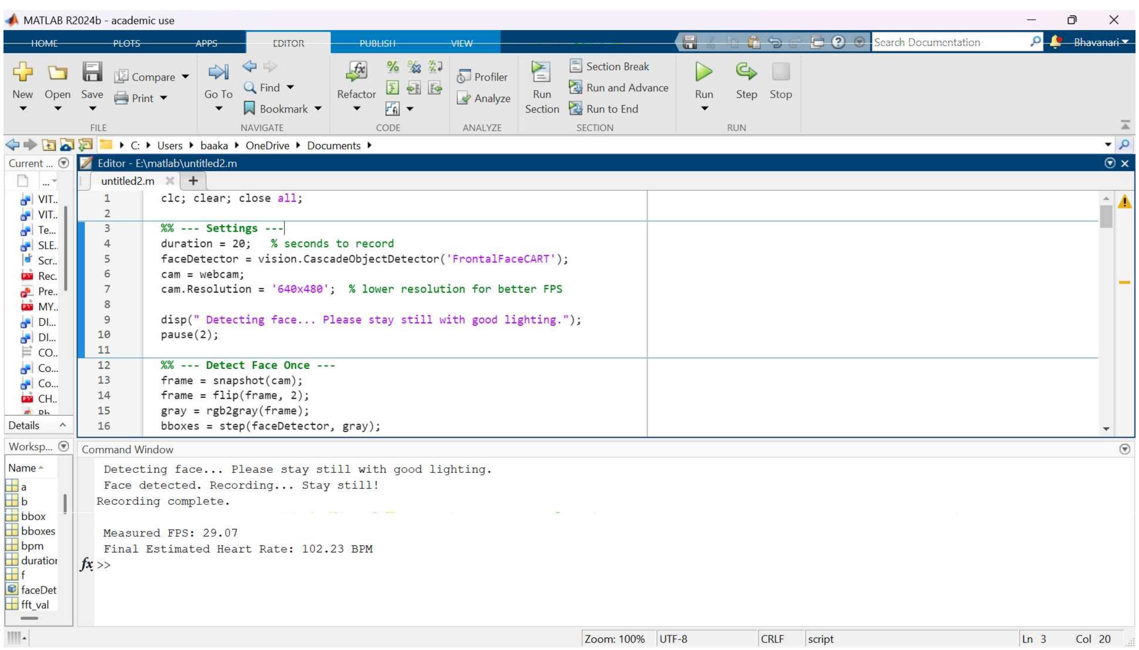


5.2 Obtaining Graphs:



The output graph shows how the heart rate is calculated from the video. The top graph is the filtered green signal, which comes from the forehead area of the face. It shows how the green color value changes with time because of blood flow under the skin. Each wave in this signal represents one heartbeat. The bottom graph shows the FFT (Fast Fourier Transform) result of this signal. It converts the signal from time to frequency and helps find the main heartbeat frequency. The highest peak in this graph represents the actual heart rate. In this result, the peak appears around 100 BPM, meaning the person's heart rate is about 100 beats per minute. This confirms that the system detected the face correctly and calculated the heart rate accurately.

5.3 Final Result



6.FUTURE SCOPE

The future scope of heart rate calculators using facial video-based heart rate estimation is highly promising, with ongoing advances in algorithms, hardware, and application areas. The main conclusions underline the technique's accuracy, potential for broad deployment in telemedicine, and its contributions to unobtrusive health monitoring. However, future research is essential to address challenges related to diverse populations, motion artifacts, lighting conditions, and integration with other vital sign measurements.

Key Future Scope Areas

- **Robustness Across Conditions**: Current systems work best in stable lighting and with minimal subject movement. Improvements in algorithmic robustness for varying skin tones, lighting environments, and motion will broaden acceptance and practical use.
- **Extension to Multiple Vital Signs**: Facial video for remote photoplethysmography (rPPG) can be further developed to estimate respiratory rate, blood oxygen levels (SpO2), and even blood pressure, providing comprehensive contactless health monitoring.
- **Broader Applicability**: Applications can extend to emotion detection, athletic performance monitoring, mental health, and interactive entertainment systems, in addition to telemedicine.
- **Algorithmic Advancements**: Ongoing development of signal processing (e.g., ICA, PCA, adaptive filtering, and deep learning) will help overcome noise and improve reliability, even in real-world noisy conditions.
- **Integration and Trust**: Integrating facial video heart rate estimation into consumer devices (phones, webcams, wearables) and building trust

through clinical validation and privacy safeguards will be critical for wide adoption.

- Information proceeding from the heart signal is usually very useful for knowing some aspects of a person, like feelings or physical condition. Another application in which this project can be very useful is in emotions recognition, as physiological parameters are often used in that kind of applications

- As the heart rate varies with the age and the gender of the user [48], an interesting future work to measure the effectiveness of the system would be to carry on an exhaustive study of the working of the system along different age stages.

- Another interesting future line of investigation would be to find methodologies to measure more complex physiological parameters such as heart rate variability, blood oxygen level or blood pressure.

7. **CONCLUSIONS**:

This section summarizes the conclusions that can be extracted from the development of this project. Achievements in both the image processing and the data processing phase are described, as well as the general conclusions about the whole project

- Non-contact, video-based heart rate measurement provides accurate, unobtrusive results in controlled environments and has significant promise for scalable remote health monitoring.
- Developed systems show strong accuracy and low error rates in ideal conditions, with errors often less than 1 bpm, but further testing and development for everyday settings are needed.

- During the realization of this project, an exhaustive study of the involved algorithms and techniques has been carried out. In particular, independent component analysis, which is the base of the heart rate recovery procedure, A deep background in computer vision algorithms (as face detection, tracking systems and skin segmentation) as well as knowledge in image processing tools (OpenCV) have been acquired.
- With motion, the extracted RGB traces are affected by a greater number of signals than the available mixture signals, This could be solved by improving the ROI selection system) that take into account some a-prior information about the heart signal.

8.**REFERENCES**

[1]. M.Z. Poh, D.J. McDu and R. Picard. Non-contact, automated cardiac pulse measurements using video imaging and blind source separation. Op tics Express, vol. 18, pp. 10762-10774, 2010.

[2]. K.H. Kim, S.W. Bang and S.R. Kim. Emotion recognition system us ing short-term monitoring of physiological signals. Medical and Biological Engineering Computing, vol. 42, pp. 419-427, 2004.

[3]. P. Sahindrakar. Improving motion robustness of contact-less monitoring of heart rate using video analysis. Technische Universiteit Eindhoven, Department of Mathematics and Computer Science, 2011.

[4]. C.K. Chen, C.L. Lin, C.T. Chiang and S.L. Lin. Personalized information encryption using ECG signals with chaotic functions. Information Sciences, vol. 193, pp. 125-140, 2012.

[5]. C. Takano and Y. Ohta. Heart rate measurement based on a time-lapse image. Medical Engineering and Physics, vol. 29, pp. 853-857, 2007.

[6]. M.Z. Poh, D.J. McDu and R. Picard. Advancements in noncontact, multiparameter physiological measurements using a webcam. IEEE Transactions on Biomedical Engineering, vol. 58, no. 1, pp. 7-11, 2011.

[7]. K. Umetani, D. H. Singer, R. McCraty and M. Atkinson. Twenty-four hour time domain heart rate variability and heart rate: relations to age and gender over nine decades. Journal of the American College of Cardiology, vol. 31, pp. 593-601, 1998.