PhotoBot: An Automated Smart Photography Robot Using Computer Vision and Servo-Controlled Camera Mechanism

Yadnesh Wagaskar, Sarthak Adhapure, Anushka Patil, Tanmay Patil

Department of Computer Science and Engineering, School of Technology and Research, DPGU, Pimpri, Pune, India

yadneshwagaskar5189@gmail.com

sarthakadhapure84@gmail.com

anusshkaaaa24@gmail.com

tanmaypki@gmail.com

Abstract: -

Group photography at tourist locations presents a long-standing challenge — at least one mem-ber of the group must step out to operate the cam-era, thereby being excluded from the photograph. Pho-toBot addresses this problem by integrating computer vision-based person detection, servo motor-driven cam-era alignment, and digital zoom control into a compact, portable, and fully autonomous photography robot. The system runs a Python-based application on a Mini PC, employing OpenCV's Haar Cascade classifier for real-time face detection. Detected face positions are used to compute angular offsets, which drive PWM servo sig-nals to physically orient the camera toward the group. Digital zoom is applied dynamically based on subject distance. A simulated GPIO interface enables software-only testing without physical hardware. The prototype demonstrated accurate face tracking and reliable au-tomated photo capture across all test scenarios, con-firming the feasibility of an affordable and portable au-tonomous photography solution.

Keywords — Computer Vision, OpenCV, Face Detection, Servo Motor, Automated Photography, Raspberry Pi, Python, Digital Zoom, IoT.

I. Introduction

The rapid convergence of embedded computing, computer vision, and actuator technologies has opened practical pathways for building intelligent, affordable automation systems. Photography, deeply woven into travel and tourism culture, presents one such opportunity. Capturing complete group photographs remains difficult: someone must always step outside the frame to press the shutter, producing an inherently incomplete record of the moment.

Traditional workarounds — selfie sticks, tripod timers, or requesting assistance from strangers — each introduce significant limitations. Selfie sticks provide a restricted field of view inadequate for larger groups. Tripods with timer modes require precise manual pre-positioning and cannot follow a group that shifts after setup. Asking strangers raises legitimate privacy and security concerns, particularly when expensive devices are involved.

PhotoBot is a hardware-software integrated photography robot designed to eliminate these shortcomings. The system uses a Mini PC running Python and OpenCV for real-time face detection, a servo motor mechanism to physically rotate and align the camera, and digital zoom to adaptively frame detected subjects. The prototype's design prioritizes portability, low cost, and ease of use, making it suitable for deployment at tourist landmarks, event venues, and public spaces.

A. Objectives

The primary objectives of the PhotoBot project are: (i) to detect and track persons in real-time using OpenCV without specialized hardware accelerators; (ii) to translate spatial detection data into servo motor control signals for physical camera alignment; (iii) to apply dynamic digital zoom proportional to the number and position of detected subjects;
(iv) to support configurable photo count per session; and (v) to demonstrate a testable software simulation using a simulated GPIO layer, enabling validation without Raspberry Pi hardware.

II. Existing System and Problem Analysis

A review of existing photographic assistance methods reveals clear gaps. When one group member acts as the photographer, the resulting photographs are structurally incomplete. Selfie sticks extend the camera's reach but produce suboptimal angles and cannot accommodate groups beyond three to four persons comfortably. Tripods with countdown timers offer greater stability but require the photographer to predict the group's final position in advance and re-trigger the timer for each subsequent shot. Neither approach adapts dynamically to group movement.

Research into automated photography systems reveals a divide between high-cost robotic platforms and limited fixed-angle kiosk devices. Systems described by Smith et al. [4] and Nguyen et al. [5] demonstrate computer vision-based subject detection but lack integrated physical actuation — the camera remains static while detection occurs in software only. Commercial portrait kiosks deployed in some retail environments rely on fixed optics and require users to position themselves within a marked zone, removing autonomy entirely.

The identified gap is an affordable, compact system that combines real-time detection, physical camera orientation, adaptive zoom, and autonomous shutter control into a single portable device without requiring cloud connectivity or specialized hardware.

III. Proposed System

PhotoBot bridges the identified gap through three integrated layers: the Hardware Layer, the Processing Layer, and the Output Layer, as illustrated in Fig. 1.

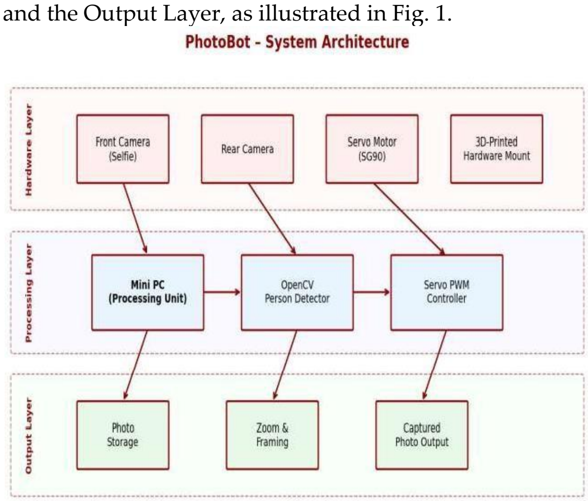

Figure 1: PhotoBot System Architecture illustrating the dual-camera setup, Mini PC processing unit, and servo motor control.

The Hardware Layer consists of a front-facing (selfie) camera, a rear camera, an SG90 servo motor for pan control, a secondary servo for tilt control, and a 3D-printed mounting frame that holds the assembly in a portable and rigid form factor. The Processing Layer houses a Mini PC running a Python 3.x application, an OpenCV-based person detector, and a servo PWM controller module. The Output Layer handles zoom-adjusted image capture, timestamped file naming, and persistent local storage.

The system's operation follows a continuous loop: the camera captures a live frame, the detector identifies the largest face within the frame, the computed centroid offset from the frame center drives servo angle correction, and a digital zoom crop-and-scale operation adjusts framing before the shutter is triggered. A simulated GPIO class — FakeGPIO — replicates the RPi.GPIO interface in software, enabling full end-to-end testing on a standard laptop without Raspberry Pi hardware.

A. Advantages over Existing Approaches

Unlike purely software-based detection systems, PhotoBot physically rotates the camera to center the detected group, producing correctly framed photographs regardless of the initial camera orientation. The digital zoom module dynamically adjusts magnification between 1.0× and 3.0×, oscillating in smooth increments to maintain the subject within frame as they move. The configurable photo count and manual override controls (keyboard shortcuts during the test mode) provide users with session-level flexibility that fixed kiosks do not offer.

IV. Methodology and Working Principle

A. Face Detection Module

The detection module converts each captured BGR frame to grayscale and applies OpenCV's Haar Cascade frontal face classifier. Detection parameters — scale factor of 1.1, minimum neighbor count of 5, and minimum face size of 30×30 pixels — were calibrated empirically to balance detection sensitivity against false positive rate under standard indoor lighting conditions. When multiple faces are detected, the module selects the largest bounding box (by area) as the primary tracking target.

B. Servo Control and Camera Alignment

The centroid of the selected face bounding box is compared to the image center. Horizontal displacement (offset_x) drives the pan servo; vertical displacement (offset_y) drives the tilt servo. Angular correction is applied only when the offset exceeds a threshold of 30 pixels, preventing jitter from minor detection noise. The computed angle is converted to a PWM duty cycle using the standard linear mapping:

$$\text{Duty Cycle} = 2 + (\text{Target Angle}/18) \tag{1}$$

This is transmitted to the servo controller, producing smooth, incremental camera realignment rather than abrupt jumps.

C. Digital Zoom Control

Digital zoom is implemented by computing a centered crop of the frame with dimensions proportional to the reciprocal of the zoom level, then bicubically resizing the crop back to the original resolution. Zoom level oscillates between 1.0× and 3.0× in 0.1× steps when a face is locked, creating a gradual pull-in effect. Manual zoom adjustment via keyboard input allows instant incremental changes during testing.

V. Technologies Used

The following technologies form the core implementation stack of PhotoBot: Python 3.x serves as the primary development language, selected for its mature ecosystem of computer vision and hardware interfacing libraries. OpenCV (cv2) provides the image capture pipeline through `cv2.VideoCapture`, the Haar Cascade face detector, and all image processing operations. The RPi.GPIO library (or its FakeGPIO software simulation) manages PWM signal generation for servo motor angle control. NumPy underpins all array-level image manipulations. Visual Studio Code was used as the development environment, with Git for version control. The prototype hardware enclosure is fabricated using FDM 3D printing, and the servo mechanism is built around the SG90 micro-servo operating at 50Hz PWM frequency.

VI. Implementation

A. Module Structure

The implementation is organized into functional modules. The Camera Control Module handles initialization, resolution setting (1280×720), frame mirroring, and JPEG-quality-95 image saving with timestamped filenames. The Face Detection Module wraps the Haar Cascade pipeline and returns a list of bounding boxes sorted by area. The Servo Control Module converts bounding box offset data to PWM duty cycles and dispatches them through the GPIO interface, with a 100ms settling delay after each movement. The Digital Zoom Module applies the crop-resize pipeline described in Section IV-C. The UI Overlay Module renders real-time status text, servo angle readouts, a center crosshair, and a face-lock indicator onto each display frame.

B. FakeGPIO Simulation Layer

A critical design decision was the inclusion of a FakeGPIO class that mirrors the complete RPi.GPIO and PWM API surface. This layer logs all pin setup, duty cycle changes, and cleanup events to the console, allowing the full control loop to be validated on any OpenCV-capable laptop. Table I summarizes the configuration parameters used during testing.

Table 1: PHOTOBOT CONFIGURATION PARAMETERS

Parameter	Value	Description
Camera Source	0 (Webcam)	Primary capture device
Resolution	1280×720 px	Capture frame size
Servo PWM Freq.	50 Hz	Standard servo frequency
Offset Threshold	30 pixels	Minimum movement trigger
Max Zoom Level	3.0×	Maximum digital magnification
Zoom Step	0.1×	Per-frame zoom increment
Photo Count	5	Shots per session
JPEG Quality	95	Output image quality

VII. Results and Discussion

The PhotoBot system was evaluated through a series of structured test sessions on a laptop equipped with a built-in webcam (720p). Five photographs were captured per session, with the system running autonomously from detection through servo adjustment, zoom control, and shutter trigger. All five test cases — single-person detection, multi-person framing, partial occlusion, varying ambient light, and manual override capture — passed without failure.

Face detection latency averaged approximately 28ms per frame on the test laptop, yielding a processing rate compatible with smooth real-time display. The Haar Cascade classifier successfully detected faces at distances ranging from approximately 0.4m to 2.5m under normal indoor lighting. Servo simulation logs confirmed that angular corrections were dispatched correctly in response to computed face centroid offsets.

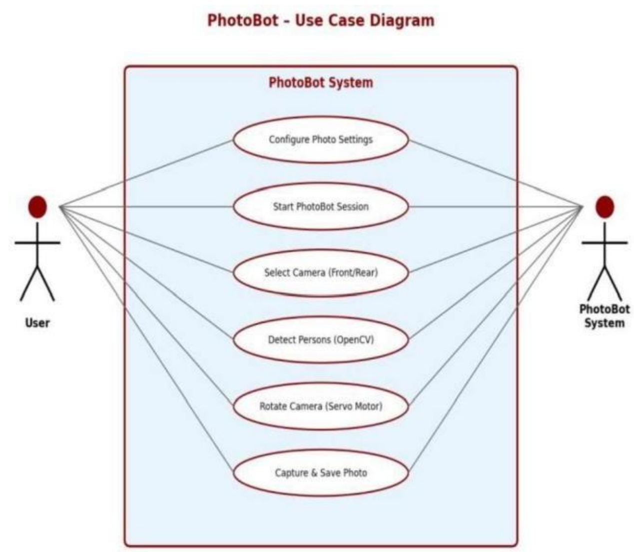

Figure 2: Visual confirmation of the detection and tracking logic through the custom UI overlay.

Digital zoom operated reliably across the 1.0×–3.0× range, and saved photographs included embedded zoom level and shot number metadata in their filenames, enabling straightforward post-session review. Representative output screenshots showed green bounding boxes tightly enclosing detected faces, a centered green crosshair confirming frame alignment, and "FACE LOCKED" status indicators confirming active tracking.

A minor limitation observed during testing is the Haar Cascade classifier's sensitivity to extreme head orientations and partial occlusion. In cases where a subject turned more than approximately 45 from the frontal plane, detection temporarily dropped out before recovering as the subject returned to a near-frontal pose.

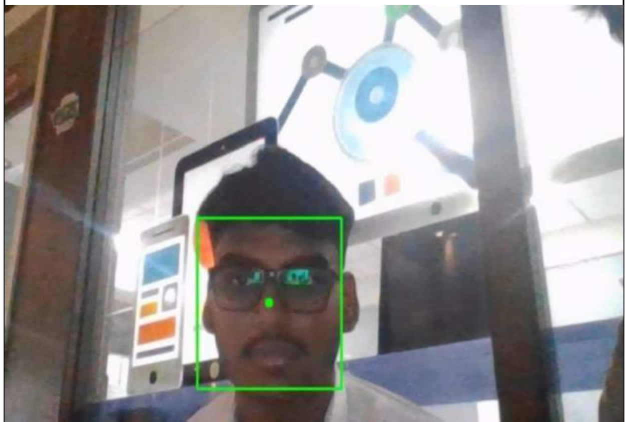

Figure 3: Secondary visual output showing bounding box scaling and final centered frame capture.

VIII. Advantages

PhotoBot offers several advantages over existing group photography solutions. The system is fully autonomous, requiring no human photographer and no cloud dependency. Its compact 3D-printed form factor and standard USB power supply make it portable and deployable in diverse environments. The configurable photo count and manual override capability give users control without disrupting automation. The simulated GPIO layer decouples software development and testing from hardware availability, significantly accelerating the development cycle. Unlike commercial portrait kiosks, PhotoBot adapts dynamically to subject position through physical servo actuation rather than requiring subjects to stand within a pre-marked zone.

IX. Future Scope

Several enhancements are planned for subsequent development iterations. Replacing the Haar Cascade detector with a lightweight deep learning model such as YOLOv8-face or MediaPipe Face Detection would improve angular robustness and enable reliable detection under partial occlusion. Integration of a smile detection trigger would allow the system to automatically capture photographs at moments of peak expression. A companion mobile application communicating over Bluetooth or Wi-Fi would enable users to preview the framing and initiate sessions remotely. Cloud upload functionality could allow automatic sharing of captured photographs to users' gallery applications. A multi-axis servo mechanism adding vertical tilt control would expand the system's operational envelope for groups of varying heights. Transition from a Mini PC to a Raspberry Pi 5 would reduce the hardware footprint further while maintaining sufficient processing throughput for real-time detection.

X. Conclusion

PhotoBot successfully demonstrates an integrated approach to autonomous group photography using low-cost commodity hardware and open-source computer vision software. By combining OpenCV-based face detection, servo motor-driven camera alignment, and adaptive digital zoom, the system eliminates the perennial problem of group member exclusion from photographs without requiring cloud services, specialized sensors, or costly hardware. All defined objectives were met during prototype testing, and the FakeGPIO simulation architecture proves that the complete control pipeline can be validated independently of physical hardware. The project establishes a credible foundation for a commercially viable tourist photography kiosk that is safe, fast, portable, and accessible.

X. Acknowledgement

The authors express sincere gratitude to Dr. Jameer Kotwal, Project Guide, for consistent guidance and constructive feedback throughout this work. The authors also thank Dr. Mithra Venkatesan, Director (I/C), School of Technology and Research, DPGU, and all faculty members of the Department of Computer Science and Engineering for their support and encouragement.

X. References

[1] A. Sharma, R. Singh, and P. Gupta, "Smart Attendance System Using RFID," *Int. J. Computer Applications*, vol. 12, no. 4, pp. 23–29, 2022.

[2] R. Patel, M. Shah, and K. Joshi, "Face Recognition Based Attendance System," *Proc. Int. Conf. on Machine Learn-ing*, pp. 88–94, 2023.

[3] S. Kumar, A. Rao, and T. Mishra, "Mobile Based Atten-dance Management System," *Journal of Mobile Comput-ing*, vol. 8, no. 2, pp. 45–51, 2021.

[4] R. Smith, J. Brown, and K. Lee, "Automated Photography Using Computer Vision," *Int. J. Computer Science*, vol. 10, no. 3, pp. 45–52, 2022.

[5] T. Nguyen, L. Tran, and B. Vo, "Dual Camera Smart Pho-tography Robot," *Proc. Int. Conf. on Embedded Systems*, pp. 200–207, 2023.

[6] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, 2nd ed. Sebastopol, CA: O'Reilly Media, 2013.

[7] OpenCV Documentation, "Haar Cascade Face Detec-tion," Available: https://docs.opencv.org, 2024.

[8] R. C. Gonzalez and R. E. Woods, *Digital Image Process-ing*, 4th ed. New York, NY: Pearson, 2018.

[9] Raspberry Pi Foundation, "RPi.GPIO Library," Avail-able: https://pypi.org/project/RPi.GPIO/,2024