**Jurnal Penelitian Politeknik Penerbangan Surabaya**
Vol 11, No 2, Agustus, 2026
p-ISSN: 1978-6832
e-ISSN: 2622-5948

# Autonomous Optical Tracking System (AOTS): Development of A Computer Vision-Based Face-Tracking Pan-Tilt Prototype Using Arduino

**Bagus Rangge Plaewe Nusantara[1], Uvi Desi Fatmawati[2], Sunarta[3], Erlangga Aryo Prastya[4], Juanta Naveen Gabriel Ginting[5], Refandy Permana Gultom[6], Sri Wulandari Margaretha[7]**

[1,2,3,4,5,6,7] *Department of Electrical Engineering, Faculty of Engineering and Defense Technology, Republic of Indonesia Defense University*

Corresponding author*:* *[1]bagusrangge46@gmail.com*

*DOI : https://doi.org/10.46491/jp.v11i2.2629*

**Abstract**

Recent advances in computer vision and embedded control have enabled low-cost systems to detect and follow visual targets in real time. This study developed and preliminarily evaluated an Autonomous Optical Tracking System (AOTS) that combines MediaPipe/CVZone and OpenCV face detection on a laptop with Arduino Uno control of two SG90 pan-tilt servos and a 5 mW KY-008 visual alignment indicator. Face-centroid coordinates were converted into pan and tilt commands and transmitted through serial communication. Eight functional scenarios were each repeated 20 times, producing 160 trials. A trial was successful when the face state and the expected actuator or indicator response were correct. The system achieved 159 successful trials (99.4%): seven scenarios reached 20/20, while the centered-target scenario reached 19/20 (95%). Representative response times across six movement conditions ranged from 0.32 to 0.42 s for the pan servo and from 0.28 to 0.50 s for the tilt servo. These results demonstrate prototype feasibility under controlled conditions but should not be interpreted as benchmark face-detection accuracy because pixel-level tracking error, environmental variation, and repeated timing distributions were not recorded. The system is intended solely for benign laboratory and educational use. Further work should evaluate repeated latency, tracking error, illumination, distance, and user safety under standardized conditions.

**Keywords:** Arduino; computer vision; face tracking; MediaPipe; pan-tilt control.

**Article History:**
Received: July 27, 2026 ; Revised: August 3, 2026 ; Accepted: August 6, 2026



## INTRODUCTION

Computer vision and embedded control enable systems to convert visual observations into actuator commands in real time. OpenCV remains a widely used foundation for image acquisition and processing, while contemporary perception frameworks such as MediaPipe provide modular pipelines for efficient detection and tracking (Bradski, 2000; Lugaresi et al., 2019; Szeliski, 2022). MediaPipe face detection is associated with lightweight detector designs intended for low-latency operation, making it suitable for interactive prototypes with limited computational resources (Bazarevsky et al., 2019).

Active pan-tilt tracking combines perception with visual feedback control: the target position in the image plane is estimated, converted into angular commands, and used to reposition a camera or indicator toward the image center. Recent research has improved this process through model-based control, high-speed vision, predictive tracking, and nonlinear coordinate mapping (Guo et al., 2021; Jiang et al., 2021; Nebeluk et al., 2023; Qu et al., 2020; Zhou et al., 2025). These studies show that tracking performance depends not only on detection but also on coordinate mapping, mechanical response, communication delay, tolerance-zone design, and recovery after target loss.

However, many advanced systems rely on high-speed cameras, specialized mechanisms, or complex controllers, which can limit their accessibility for laboratory instruction and low-cost prototyping. A practical research gap therefore remains in documenting how a simple webcam-laptop-Arduino architecture performs when face detection, serial communication, two-axis servo actuation, target-loss behavior, and a visible alignment indicator are integrated and evaluated through repeated functional scenarios. Prototype studies must also distinguish functional success rate from algorithmic detection accuracy and explicitly acknowledge limitations related to tracking error, environmental variability, and latency measurement.

To address this gap, this study develops an Autonomous Optical Tracking System (AOTS) for benign laboratory and educational use. The objectives are to: (1) design and build a low-cost Arduino-based face-tracking prototype; (2) implement face localization using MediaPipe/CVZone and OpenCV; (3) control a two-axis pan-tilt mechanism through serial communication; and (4) evaluate functional success and representative actuator response. The main contribution is the integrated implementation and transparent prototype-level evaluation of these subsystems without claiming biometric identification or weapon-related functionality.

## METHOD

1. Research Approach and Design

This study used an experimental engineering-design approach involving requirement definition, subsystem integration, software implementation, calibration, and functional evaluation. The system operates as a visual-feedback loop rather than a fully instrumented servo closed loop: each camera frame is processed, the detected face centroid is converted into pan and tilt commands, servo movement changes the subsequent camera view, and the new frame is processed again. Testing was conducted in a controlled indoor setting with one unobstructed face presented to the camera. Eight functional scenarios were evaluated, and each scenario was repeated 20 times, producing 160 trials. The response-time dataset contained one representative value for each of six target-movement scenarios; therefore, these values were analyzed descriptively rather than inferentially.

2. Hardware Components

The physical implementation of the prototype used the following components:

a. Processing Unit: a laptop/computer running the Python-based image-processing program.
b. Visual Sensor: a JETE W7 Pro Full HD 1080p webcam used to acquire real-time video frames.
c. Microcontroller: an Arduino Uno R3 (ATmega328P) that received serial commands and generated control signals for the servos and visual indicator.
d. Actuators: two SG90 servo motors, with the pan servo connected to digital pin D9 and the tilt servo connected to digital pin D10.
e. Visual Alignment Indicator: a 5 mW KY-008 laser module connected to digital pin D8. The module was used only to indicate the direction of the tracking mechanism and was not intended for identification, aiming, or weapon-related use.
f. Power Supply: a 9 V HYW battery supplied the Arduino; the servos used 5 V and the KY-008 module used 3.3 V according to the prototype wiring configuration.

**Figure 1.** Hardware Assembly of the AOTS Prototype

3. ***System Architecture and Workflow***

The AOTS architecture separates image processing from actuator control. As shown in Figure 2, the webcam supplies frames to the laptop, where MediaPipe through the CVZone library detects a face and OpenCV supports frame acquisition and display. The detected face centroid is translated into pan and tilt commands, which are sent to the Arduino Uno through PyFirmata serial communication. The Arduino then updates the two SG90 servos and the KY-008 indicator. This distributed design allows computationally intensive vision processing to remain on the laptop while the microcontroller handles low-level actuation.

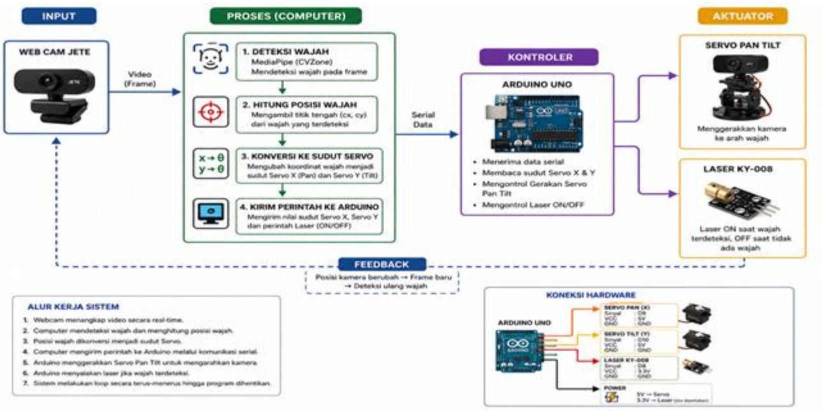


**Figure 2.** Architecture and Data Flow of the AOTS Face-Tracking System

The operational sequence is as follows. First, the webcam captures a video frame. Second, the face detector identifies the face region and estimates its centroid coordinates (cx, cy). Third, the program compares the centroid with the image center and linearly maps the horizontal and vertical coordinate offsets to pan and tilt commands within the calibrated mechanical range. A predefined central tolerance zone reduces unnecessary movement when the face is near the frame center. Fourth, the commands and indicator state are transmitted to the Arduino. Finally, servo motion changes the camera view and the next frame is processed, producing a continuous visual-feedback cycle.

The hardware connections were configured as follows: pan-servo signal on D9, tilt-servo signal on D10, and KY-008 signal on D8. The program did not perform facial identity recognition or store facial images; it used the detected face location only for immediate control. The laser module requires controlled operation: direct eye exposure and highly reflective surfaces must be avoided, and use should follow the safety principles in IEC 60825-1 (International Electrotechnical Commission, 2014).

4. **Software Logic Flow (Flowchart)**

The software logic shown in Figure 3 begins by initializing the webcam, PyFirmata connection, servos, and visual indicator. During operation, the program repeatedly captures a frame and runs MediaPipe/CVZone face detection. When a face is present, the centroid is extracted, converted into servo commands, and transmitted to the Arduino. When no face is detected, the interface displays "NO TARGET," the indicator is switched off, and the servos retain their last commanded position.

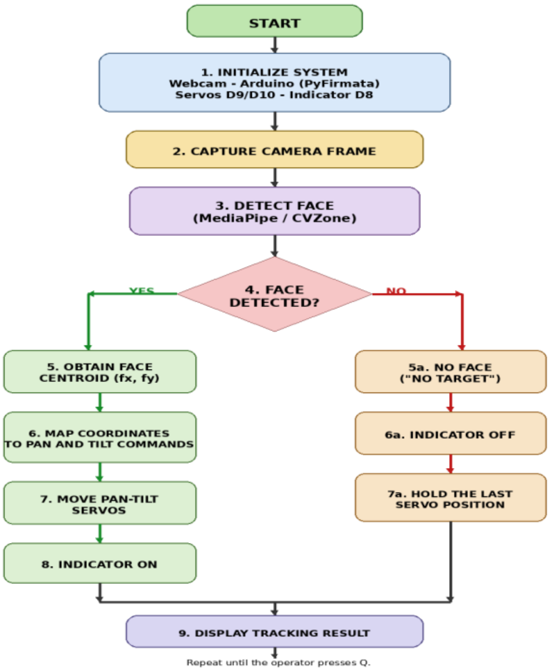


**Figure 3.** Software Logic Flow of the AOTS Prototype

The tracking display was updated continuously until the operator pressed the Q key. Functional success was defined as a correct face-presence state together with the expected pan, tilt, or indicator response for the specified scenario. The functional success rate was calculated as the number of successful trials divided by the total number of trials and multiplied by 100%. This measure evaluates integrated prototype behavior and is not equivalent to standard machine-learning measures such as precision, recall, or intersection-over-union.

Tracking error was operationally defined as the Euclidean pixel distance between the detected face centroid and the center of the image frame. However, pixel-level error was not logged in the original test, so tracking-error statistics are not reported. Response time was defined as the elapsed interval from detection of a new target position to completion of the corresponding servo response. Because repeated timing observations and measurement uncertainty were not retained, Figure 6 presents representative scenario values only. The present work is a prototype-level system test rather than a formal human-participant study; any future usability study should include informed consent, privacy safeguards, and institutional ethics review.

## RESULTS AND DISCUSSION

1. Functional Testing Results

Functional testing used the eight scenarios summarized in Figure 4. Each scenario was repeated 20 times. A trial was classified as successful when the program correctly represented face presence or absence and the expected servo or indicator state occurred without an unintended command. The test matrix covered face detection, horizontal and vertical movement, centered-target behavior, target loss, and visual-indicator activation.

| No | Skenario Pengujian | Hasil yang Diharapkan | Status |
|---|---|---|---|
| 1 | Deteksi wajah | Wajah terdeteksi | Berhasil |
| 2 | Wajah ke kanan | Servo Pan ke kanan | Berhasil |
| 3 | Wajah ke kiri | Servo Pan ke kiri | Berhasil |
| 4 | Wajah ke atas | Servo Tilt ke atas | Berhasil |
| 5 | Wajah ke bawah | Servo Tilt ke bawah | Berhasil |
| 6 | Wajah di tengah | Kamera mengarah ke target | Berhasil |
| 7 | Tidak ada wajah | Laser OFF, NO TARGET | Berhasil |
| 8 | Wajah terdeteksi | Laser ON | Berhasil |

**Figure 4.** Functional Testing Matrix for the AOTS Face-Tracking System

The repeated testing produced 159 successful responses from 160 trials. Face detection, right, left, upward, and downward movement, the no-target condition, and indicator activation each achieved 20 successful trials. The centered-target condition achieved 19 successful trials, indicating one unnecessary corrective movement near the tolerance-zone boundary.

These results confirm that the integrated subsystems operated consistently in the controlled prototype environment. When a face moved horizontally or vertically, the appropriate servo followed the expected direction. When the face remained near the frame center, the mechanism generally maintained alignment. When no face was visible, the program displayed "NO TARGET" and switched the visual indicator off.

2. Functional Detection and Tracking Success Rate

Figure 5 presents the functional success rate for each scenario. The rate was calculated from 20 repeated trials per condition rather than from a single binary test. The overall rate was 159/160, equivalent to 99.4% after rounding.

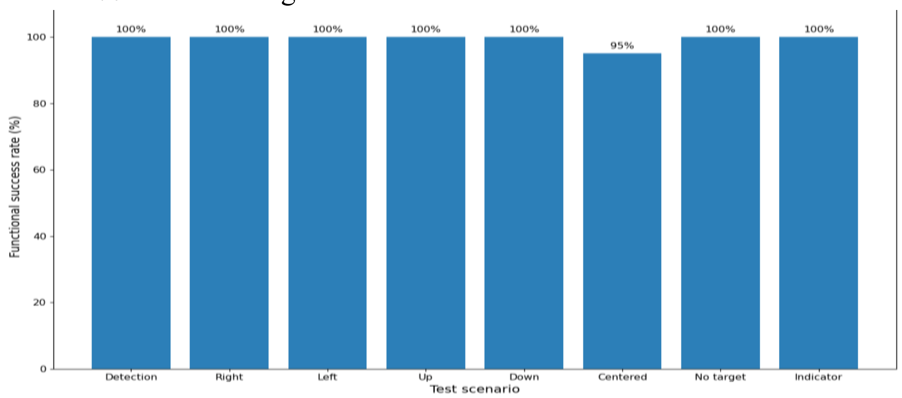


Figure 5. Functional Detection and Tracking Success Rate by Test Scenario

Seven scenarios achieved a functional success rate of 100%, while the centered-target scenario achieved 95%. The isolated centered-target failure suggests sensitivity near the tolerance-zone boundary, where small fluctuations in the detected centroid can trigger an unnecessary servo correction. Similar visual-servoing research emphasizes that centering performance depends on image-plane error, controller design, and command stability (Nebeluk et al., 2023; Qu et al., 2020).

The reported percentages should not be interpreted as benchmark face-detection accuracy. The test evaluated whether the complete prototype produced the expected response under predefined conditions; it did not use an annotated image dataset and did not quantify false positives, false negatives, precision, recall, or pixel-level tracking error. Accordingly, the term functional success rate is used throughout the revised manuscript.

Compared with advanced pan-tilt systems employing high-speed cameras, predictive control, or specialized stabilized mechanisms (Guo et al., 2021; Jiang et al., 2021), the present prototype prioritizes low cost, accessibility, and instructional value. Its contribution is therefore the practical integration of widely available components rather than state-of-the-art tracking speed or control precision.

The findings nevertheless provide a useful baseline for further development. Logging centroid error, servo command history, and frame timestamps would allow future studies to evaluate settling time, overshoot, jitter, target-loss recovery, and the effect of illumination, viewing distance, occlusion, and multiple faces.

3. Servo Response Time Analysis

Representative response times were recorded for six movement scenarios: right, left, up, down, upper-right diagonal, and lower-left diagonal. Values were recorded separately for the pan servo on D9 and the tilt servo on D10. Figure 6 displays the available scenario-level observations; because repeated timing measurements were not retained, the values are presented as descriptive prototype observations rather than population estimates.


**Figure 6.** Representative Pan and Tilt Servo Response Times by Target-Movement Scenario

Across the six scenarios, the pan-servo response ranged from 0.32 to 0.42 s, with a mean of 0.365 s and a standard deviation across scenarios of 0.041 s. The tilt-servo response ranged from 0.28 to 0.50 s, with a mean of 0.397 s and a standard deviation across scenarios of 0.088 s. These standard deviations describe variation among movement scenarios and should not be interpreted as repeated-trial uncertainty.

The shortest pan responses occurred during right and left movement (0.32 and 0.34 s), while the largest pan value occurred during upper-right diagonal movement (0.42 s). Tilt values increased from 0.28-0.30 s for horizontal target changes to 0.46-0.50 s for diagonal movement. The pattern is consistent with the need to update both axes during diagonal tracking, but the available data do not establish whether the difference was caused by computational delay, serial communication, mechanical loading, or measurement variability.

The response times are substantially slower than those reported for specialized high-speed pan-tilt platforms (Jiang et al., 2021), which is expected given the use of low-cost SG90 servos, laptop-based processing, and serial communication. Related studies also show that improved mapping and controller design can reduce centering error and latency (Guo et al., 2021; Nebeluk et al., 2023; Zhou et al., 2025). Future testing should repeat each timing condition, report the mean, standard deviation, confidence interval, and measurement method, and compare the prototype with alternative dead-zone, proportional, or predictive controllers.

## CONCLUSION

This study developed an Autonomous Optical Tracking System (AOTS) that integrates MediaPipe/CVZone and OpenCV face localization on a laptop with Arduino-based control of a two-axis SG90 pan-tilt mechanism and a KY-008 visual alignment indicator. The prototype demonstrated that commonly available hardware and open-source software can be combined into a functional visual-feedback tracking platform for benign laboratory and educational applications.

Across eight scenarios repeated 20 times, the system completed 159 of 160 trials successfully, corresponding to an overall functional success rate of 99.4%. Seven scenarios achieved 20/20 successful trials, while the centered-target scenario achieved 19/20. This result supports functional feasibility under controlled conditions but does not represent benchmark face-detection accuracy because standard detection metrics and pixel-level tracking error were not measured.

Representative response times ranged from 0.32 to 0.42 s for pan and from 0.28 to 0.50 s for tilt. The scenario-level means were 0.365 s and 0.397 s, respectively. These observations describe the available prototype data; repeated timing distributions are required before stronger claims regarding real-time performance, stability, or comparative superiority can be made.

The principal limitations are the controlled single-face setting, absence of standardized illumination and distance measurements, lack of logged tracking error, and absence of repeated latency data. Future research should evaluate varied environments, occlusion, multiple targets, target-loss recovery, battery endurance, and alternative controllers on a standalone embedded platform. The system should remain restricted to peaceful research and educational use, and the laser must function only as a low-power visual indicator operated with appropriate eye-safety controls.


## ACKNOWLEDGEMENTS

The authors express their appreciation to Uvi Desi Fatmawati, S.Si., M.Eng., and to the faculty and laboratory staff of the Department of Electrical Engineering, Faculty of Engineering and Defense Technology, Republic of Indonesia Defense University, for their guidance and technical support during prototype development and testing. The authors remain responsible for the design, data interpretation, and conclusions presented in this manuscript.


## REFERENCES


Bazarevsky, V., Kartynnik, Y., Vakunov, A., Raveendran, K., & Grundmann, M. (2019). *BlazeFace: Sub-millisecond neural face detection on mobile GPUs* [Preprint]. arXiv. https://arxiv.org/abs/1907.05047

Bradski, G. (2000). The OpenCV library. *Dr. Dobb's Journal of Software Tools*, *25*(11), 120, 122–125.

Guo, J., Yuan, C., Zhang, X., & Chen, F. (2021). Vision-based target detection and tracking for a miniature pan-tilt inertially stabilized platform. *Electronics*, *10*(18), Article 2243. https://doi.org/10.3390/electronics10182243

International Electrotechnical Commission. (2014). *Safety of laser products—Part 1: Equipment classification and requirements* (IEC 60825-1:2014, 3rd ed.). https://webstore.iec.ch/en/publication/3587

Jiang, M., Shimasaki, K., Hu, S., Senoo, T., & Ishii, I. (2021). A 500-fps pan-tilt tracking system with deep-learning-based object detection. *IEEE Robotics and Automation Letters*, *6*(2), 691–698. https://doi.org/10.1109/LRA.2020.3048653

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M. G., Lee, J., Chang, W.-T., Hua, W., Georg, M., & Grundmann, M. (2019). *MediaPipe: A framework for building perception pipelines* [Preprint]. arXiv. https://arxiv.org/abs/1906.08172

Nebeluk, R., Zarzycki, K., Seredyński, D., Chaber, P., Figat, M., Domański, P. D., & Zieliński, C. (2023). Predictive tracking of an object by a pan–tilt camera of a robot. *Nonlinear Dynamics*, *111*, 8383–8395. https://doi.org/10.1007/s11071-023-08295-z

Qu, Y., Yang, B., Yu, Z., Song, Y., & Sun, Y. (2020). Active visual servo pan/tilt control design based on improved augmented LQR. In *Proceedings of the 32nd Chinese Control and Decision Conference (CCDC 2020)* (pp. 3702–3707). IEEE. https://doi.org/10.1109/CCDC49329.2020.9164025

Szeliski, R. (2022). *Computer vision: Algorithms and applications* (2nd ed.). Springer. https://doi.org/10.1007/978-3-030-34372-9

Zhou, G., Zhang, B., Li, Q., Zhao, Q., & Zhang, S. (2025). Optimizing success rate with nonlinear mapping control in a high-performance Raspberry Pi-based light source target tracking system. *PLOS ONE*, *20*(2), Article e0319071. https://doi.org/10.1371/journal.pone.0319071