IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Practical Cognitive Radar System for UAV Tracking Using ESP32, ESP32-CAM, 24-GHz mmWave Radar, OpenCV-YOLO and Web-Based Visualization

[1]Ms. Suvarna Sahebrao Patil, [2]Dr. Shivleela Mudda
[1]PG Student, [2]Associate Professor
Department of Electronics and Telecommunication Engineering
M. S. Bidve Engineering College, Latur, Maharashtra, India

***Abstract*—**
This paper presents a practical low-cost UAV surveillance and tracking prototype that combines a 24-GHz mmWave radar sensor, an ESP32 radar controller, an ESP32-CAM video node, OpenCV with a YOLO object-detection model, a 28BYJ-48 motorized scanning mechanism, and a custom web application. The radar contributes target range/motion reports, while the camera provides visual confirmation. On the host computer, radar telemetry and video observations are processed at the application level to maintain a lightweight target state, generate a recent trajectory, and support re-acquisition when the target is temporarily lost. The web interface provides separate radar, camera, and trajectory views. Unlike high-end cognitive radar research that adapts waveforms or radar resources, the present system implements a simplified perception-action loop suitable for an M-Tech hardware prototype. Functional integration is demonstrated without fabricating outdoor range, accuracy, or probability-of-detection values; a quantitative evaluation framework is therefore included for controlled final testing.

Keywords— ESP32, ESP32-CAM, 24-GHz mmWave radar, UAV tracking, OpenCV, YOLO, motorized scanning, target re-acquisition, web visualization.

I. INTRODUCTION

Small unmanned aerial vehicles (UAVs) are increasingly used in photography, inspection, agriculture, infrastructure monitoring, and security applications. Their small radar cross section, low-altitude motion, changing aspect angle, and visually small image footprint make continuous observation difficult for a single low-cost sensor. Radar can provide useful range and radial-motion information under poor illumination, whereas a camera can provide object-level visual confirmation. Research on micro-Doppler and FMCW radar has shown that rotor motion and range-Doppler structure are informative for small-UAV detection and discrimination [13]-[25].

The original project concept was motivated by cognitive radar, where sensing and action form a closed perception-action cycle [3], [4]. In this practical implementation, 'cognitive' is used in a deliberately limited engineering sense: the system observes radar and camera data, maintains a target state, continues tracking when a valid target is available, and commands a motorized scan to support re-acquisition when the target is lost. It does not claim adaptive waveform design or advanced radar resource management of the type studied in multifunction cognitive radar systems [6]-[12].

The proposed platform uses a 24-GHz mmWave radar sensor connected to an ESP32, a separate ESP32-CAM for network video, a 28BYJ-48 stepper motor with ULN2003 driver for scan motion, Python/OpenCV with a YOLO model for visual detection, and a custom web application. Recent work has demonstrated the feasibility of compact 24-GHz radar architectures for small-drone detection [41], while deep-learning surveys identify one-stage detectors such as YOLO as attractive for real-time UAV applications, with small-object scale and complex backgrounds remaining important challenges [42].

II. RELATED WORK AND RESEARCH GAP

A. Cognitive Radar and Radar Resource Management

Haykin introduced cognitive radar as a radar system that learns from the environment and adapts its sensing actions through a perception-action cycle [3], with later work further connecting cognition, feedback and radar system design [4]. Subsequent research has extended this concept to adaptive and cognitive resource management [6], including deep-reinforcement-learning approaches for detection, tracking, scheduling and constrained scan-time allocation [7]-[12]. These methods are important for advanced multifunction radars, but they normally assume control over waveform, dwell time, beam scheduling or other radar resources that are not exposed by a simple low-cost mmWave module.

B. Small-UAV Radar Detection and Classification

Small-UAV radar research has focused strongly on micro-Doppler signatures, range-Doppler representations and machine-learning classification. Foundational studies established the micro-Doppler phenomenon and its usefulness for distinguishing rotorcraft-like motion [13], followed by UAV/bird classification using radar signatures, CNNs, random forests and transformer-based models [14]-[25]. Recent work continues to improve radar-only detection and classification, including high-resolution FMCW systems [26], dual-band micro-Doppler learning [27], trajectory-feature learning [28], and a compact 24-GHz hybrid-beamforming radar specifically designed for small-drone detection [41].

C. Tracking, Prediction and Vision-Based Detection

Advanced radar tracking methods include probabilistic data association, interacting multiple models and random-finite-set filtering [29]-[34]. For micro-UAVs, STIF and JTEA demonstrate trajectory tracking, prediction and movement estimation using richer 4-D radar measurements containing range, Doppler, azimuth and elevation [35], [36]. On the vision side, deep-learning-based UAV object detection offers fast one-stage approaches, but small object size, scale variation, background complexity and image quality remain central difficulties [42].

D. Identified Research Gap

The literature shows strong results with specialized radar signal processing, 4-D MIMO radar, advanced tracking filters and deeply trained visual detectors. However, these approaches can be difficult to reproduce in a low-cost academic prototype. The gap addressed here is integration-oriented: combine an inexpensive 24-GHz radar report, ESP32 control, ESP32-CAM vision, YOLO detection, motorized scanning and a single web interface in a closed-loop workflow that can be assembled, demonstrated and quantitatively tested without claiming the performance of high-end surveillance radar.

Table 1. Representative literature and the practical gap addressed by the proposed prototype.

Research direction	Representative work	Main strength	Limitation for this prototype
Cognitive radar	[3], [4], [6]-[12]	Adaptive perception-action and resource management	Requires programmable radar resources and greater algorithmic complexity
Micro-Doppler / FMCW UAV classification	[13]-[28]	Rich UAV/bird discrimination from radar signatures	Often needs raw radar spectra, range-Doppler maps, or custom training data
4-D radar tracking and prediction	[35], [36]	3-D trajectory tracking and movement estimation	Uses high-dimensional range-Doppler-azimuth-elevation radar
Compact 24-GHz small-drone radar	[41]	Shows practical feasibility of 24-GHz small-drone sensing	Specialized beamforming radar hardware
Vision-based UAV detection	[42]	Fast one-stage deep-learning detection is suitable for real-time tasks	Small, distant UAVs and complex backgrounds remain difficult
Proposed system	This work	Low-cost integration, web visualization and motorized re-acquisition	Current stage requires controlled quantitative performance testing

III. PROBLEM STATEMENT AND OBJECTIVES

The practical problem is to build a low-cost system that can observe a moving UAV using both radar and camera information and present the result clearly to the user. The system should be simple enough to implement with commonly available modules and should avoid unnecessary high-complexity algorithms.

The main objectives of the project are:

- Read range and motion information from a 24-GHz mmWave radar sensor using ESP32.
- Obtain a live video stream from ESP32-CAM.
- Use OpenCV and a YOLO model to detect the UAV or moving target in the camera frame.
- Send radar and camera-related information to a host computer or server.
- Display radar data, the OpenCV camera feed, and target trajectory in a custom-built web application.
- Use recent target positions to show the movement path and support basic target re-acquisition after short loss of detection.

IV. PRACTICAL SYSTEM ARCHITECTURE

The system is divided into four practical blocks: sensing hardware, embedded control, host processing, and web visualization. The 24-GHz radar sends target range/motion reports to the ESP32; the ESP32 also drives the ULN2003/28BYJ-48 scanning mechanism. In parallel, ESP32-CAM supplies the video stream. The host computer processes radar telemetry and camera frames, applies OpenCV/YOLO detection, maintains the target state and recent trajectory, and provides the three monitoring views in the web application.

A. Design Rationale and Data Paths

The radar controller and camera node are kept as separate embedded subsystems. This separation prevents continuous camera streaming from competing directly with radar reading and motor-control duties on the same microcontroller. The ESP32 therefore concentrates on deterministic radar telemetry and scan actuation, while ESP32-CAM concentrates on image acquisition and network video. At the host level, both data paths are combined without requiring the low-cost radar board to perform image processing or the camera board to perform radar processing.

Radar reports reach the ESP32 through the interface exposed by the selected 24-GHz module, typically UART or a module-specific digital output. The camera stream travels independently over the local Wi-Fi network. The host application reads these sources asynchronously and stores only the latest valid radar report and recent camera detections. Application-level timestamps can be attached when each item is received so that observations occurring within a short time window are treated as approximately concurrent. Because the prototype does not process raw RF samples, synchronization is intentionally coarse and suitable for visualization rather than precision multisensor estimation.

The motorized scan path forms the action side of the simplified cognitive loop. When the state changes from TRACKING to LOST or RE-ACQUIRE, the host sends a bounded scan request to the ESP32. The ESP32 converts that request into stepper commands through the ULN2003 driver. In the initial implementation, the motor sweep is used to search the nearby field of view; it should not be described as an accurately steered radar beam until motor-step angle, mechanical backlash and camera/radar alignment have been calibrated. After calibration, the last known visual direction can be used to bias the first scan direction and reduce re-acquisition time.

The custom web application is deliberately separated from acquisition logic. The radar screen presents the latest telemetry and controller state, the camera screen presents annotated video and YOLO output, and the trajectory screen presents recent accepted positions. This modular design makes faults visible: a lost camera stream, missing radar update or disconnected controller can be reported independently instead of silently appearing as a tracking failure. It also allows later replacement of the radar sensor or vision model without redesigning the complete operator interface.

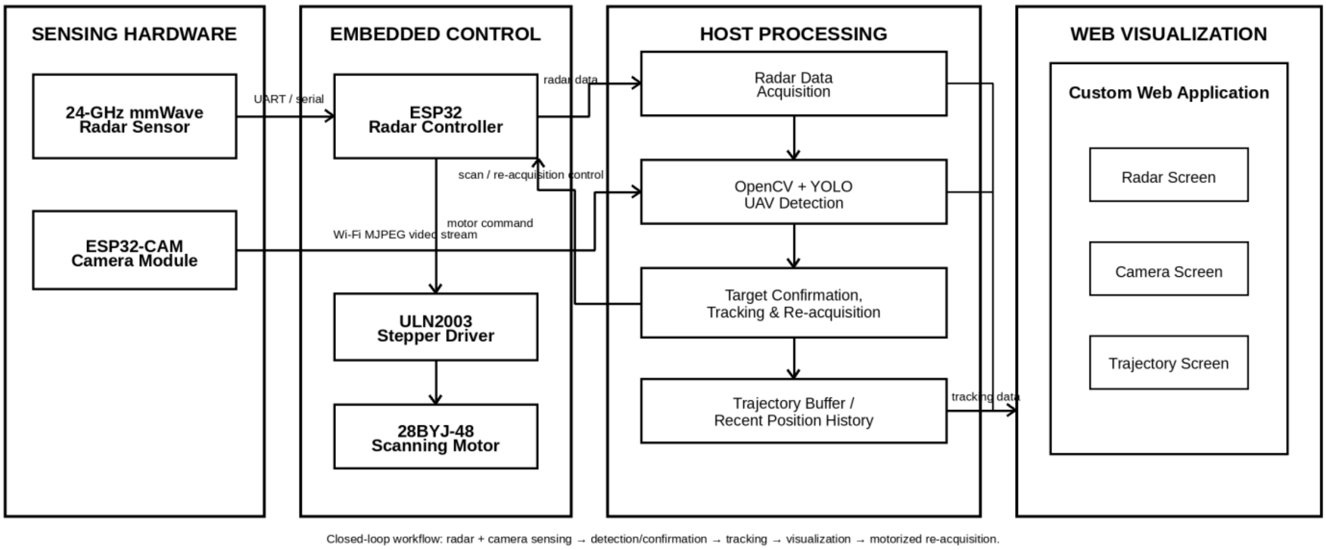


Fig. 1. Block diagram of the proposed practical cognitive radar system for UAV tracking.

B. Hardware Components

Table 2. Main hardware components of the practical prototype.

Component	Connection / Role	Practical Purpose
ESP32	Connected to mmWave radar sensor	Reads radar target data and communicates it to the host system.
24-GHz mmWave Radar Sensor	UART / digital interface depending on module	Provides target distance and motion information.
ESP32-CAM	Wi-Fi camera stream	Provides live visual view of the surveillance area.
Wi-Fi / Local Network	ESP32 and ESP32-CAM to host	Carries sensor data and camera stream.
Laptop / PC	Runs Python and web application	Performs OpenCV/YOLO processing and displays the tracking interface.
28BYJ-48 + ULN2003	ESP32 digital outputs to stepper driver	Provides low-cost motorized scan/re-acquisition movement for the radar-camera mount.

Fig. 4. 28BYJ-48 stepper motor with ULN2003 driver used for scanning movement.

Fig. 5. 24 GHz mmWave radar sensor used for target detection and motion sensing.

Representative images of the major hardware modules used in the prototype are shown separately in Fig. 2 to Fig. 5.

Fig. 2. ESP32 development board used as the main controller of the practical prototype.

Fig. 3. ESP32-CAM module used for image acquisition and live video streaming.

C. Software Components

- Embedded firmware on ESP32 for receiving radar measurements, transmitting telemetry, and commanding the scanning motor.
- ESP32-CAM firmware for providing a live JPEG/MJPEG stream over Wi-Fi.
- Python with OpenCV for opening the camera stream, frame handling, coordinate extraction and interface-side processing.
- YOLO object-detection model for locating the UAV or selected visible moving target and returning class, confidence and bounding-box coordinates.
- Custom web application for displaying radar status, annotated camera output, recent trajectory and re-acquisition state.

V. WORKING METHODOLOGY

The practical workflow starts when the mmWave radar reports a moving target inside its field of view. The ESP32 reads the available range/motion data, adds controller status, and forwards the information to the host. In parallel, ESP32-CAM continuously provides the video stream. When the host detects a valid visual target, the software associates the radar report and the camera observation within a short application-level time window.

The host computer opens the camera stream using OpenCV and passes frames to the YOLO detector. For each valid detection, the class label, confidence and bounding box are obtained. The center of the bounding box is used as the visual target position. The initial prototype uses rule-based confirmation rather than a mathematically heavy sensor-fusion filter: a target is considered stronger when radar motion/range and the visual detection are simultaneously valid, while either sensor alone can temporarily maintain a provisional target state.

For trajectory display, the application stores the most recent valid positions in sequence. Each point is appended to a bounded trajectory buffer and joined to earlier points. If the camera target is briefly lost but radar motion remains available, the system keeps the last valid visual position and enters a re-acquisition state. The ESP32 can then command a limited motorized scan of the radar-camera assembly. When a valid detection returns, the state changes back to TRACKING and the trajectory continues from the newest accepted point.

A. Radar and Visual Measurement Basis

For an FMCW radar, the physical basis of range and radial velocity can be described by the beat-frequency and Doppler relationships below. The low-cost radar module used in this prototype may calculate these quantities internally and expose only processed target reports; therefore, the equations are included to explain the measurement principle rather than to claim that raw chirp processing is performed on the ESP32.

$$R = (c\, f_b) / (2\, S) \qquad (1)$$

where c is propagation speed, f_b is beat frequency and S is FMCW chirp slope.

$$v_r = (\lambda\, f_D) / 2 \qquad (2)$$

where λ is radar wavelength and f_D is the Doppler frequency shift.

B. Camera Coordinate and Target State

For a YOLO bounding box with image coordinates (x1, y1, x2, y2), the visual center used by the trajectory display is:

$$u_c = (x1 + x2) / 2, \quad v_c = (y1 + y2) / 2 \qquad (3)$$

The application maintains four practical states: SEARCH, TRACKING, LOST and RE-ACQUIRE. SEARCH waits for a valid radar or visual target; TRACKING stores accepted positions; LOST preserves the latest valid state for a short interval; and RE-ACQUIRE requests motorized scanning until a new valid target is observed or a timeout returns the system to SEARCH. This state-machine approach is intentionally simpler than probabilistic multi-target tracking [29]-[34], but it is suitable for a first hardware prototype.

C. Software and Simulation Environment

The project is primarily a real-hardware prototype rather than a physics-level radar simulation. Embedded firmware runs on ESP32 and ESP32-CAM, while the host uses Python, OpenCV and the selected YOLO implementation together with the custom web application. For software debugging, recorded camera video and previously captured radar packets can be replayed through the same host-processing path. This replay mode is useful for interface testing, trajectory plotting and state-machine verification, but it must not be reported as a substitute for measured radar detection range or real-time hardware latency.

A practical test sequence is therefore hardware-in-the-loop: initialize the radar controller and camera, verify network connectivity, start the camera stream, confirm radar telemetry, run YOLO inference, move a visible target through the monitored area, intentionally hide or move the target outside the camera field of view, and observe the LOST/RE-ACQUIRE transition and scan command. The same sequence can be repeated at different distances and lighting conditions to populate the quantitative metrics defined later in Table 5.

VI. OPENCV AND YOLO CAMERA PROCESSING

OpenCV is used as the main camera-processing library because it can open the ESP32-CAM network stream, resize or annotate frames, and provide image coordinates for display. The YOLO model supplies detected class, confidence and bounding-box coordinates. One-stage object detectors are attractive for real-time UAV applications because of their speed, although surveys emphasize that small target size, scale variation, image quality and complex backgrounds can reduce performance [42].

To keep the camera side practical, the initial build can start from a pretrained YOLO detector and then move to a UAV-specific dataset if the general model fails to detect small or distant drones reliably. The final paper should report precision, recall and mAP only from a labeled validation set collected or selected for the completed system; these values are not estimated in advance.

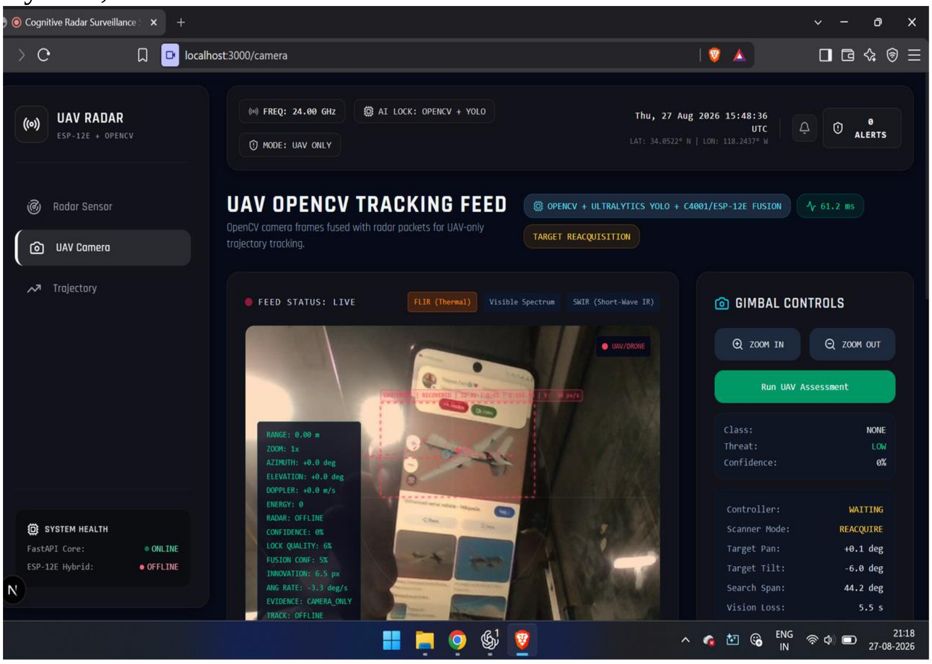


Fig. 6. OpenCV and YOLO camera screen in the custom web application.

VII. RADAR DATA DISPLAY

The radar screen converts telemetry received through ESP32 into an operator-friendly view containing controller state, scan state, target availability and the latest range/motion report. Micro-Doppler and range-Doppler literature demonstrates that radar can provide richer UAV classification features when raw signal data are available [13]-[28]. The present low-cost prototype uses only the processed measurements exposed by its radar module and therefore does not claim micro-Doppler classification.

For the final hardware build, ESP32 acts as the radar/motor controller and the web interface represents the latest telemetry and system state. The use of a 24-GHz band is consistent with recent compact small-drone radar research [41], but detection range and angular performance depend strongly on antenna, radar architecture, target radar cross section and environment; they must therefore be measured for the actual module rather than copied from unrelated radar systems.

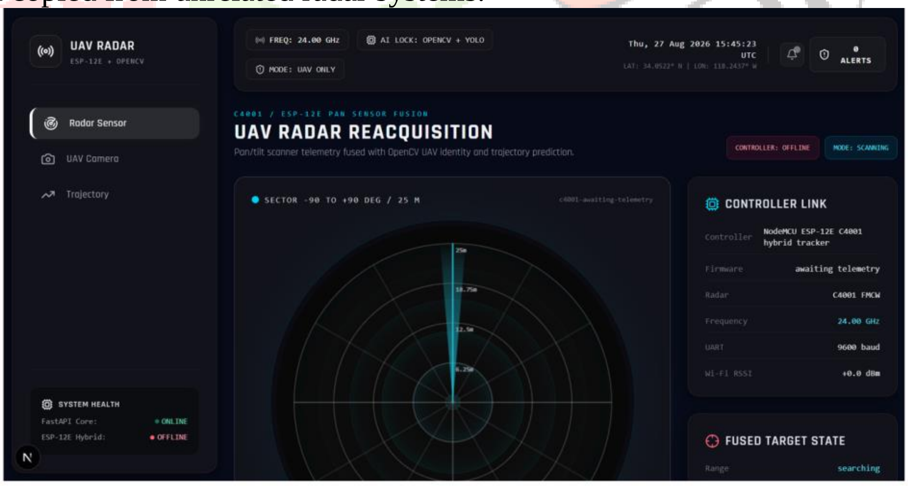


Fig. 7. Radar-screen view used to display mmWave target information.

VIII. TRAJECTORY DISPLAY IN THE WEB APPLICATION

The trajectory screen displays successive accepted target positions as a path so that target movement is visible instead of appearing only as isolated sensor values. The latest point represents the current application-level target location and earlier points form the recent movement history. Advanced research such as STIF and JTEA performs much richer trajectory tracking and prediction with 4-D radar measurements [35], [36]; the present trajectory should be interpreted only as a lightweight visualization.

A short moving-average window may optionally be used to suppress display jitter. For N recent two-dimensional visual points p_i, a smoothed display point can be written as p̄_k = (1/N) Σ p_i over the selected recent window. More advanced Kalman, interacting-multiple-model or multi-target filters can be added only after stable sensor timing and calibration are available.

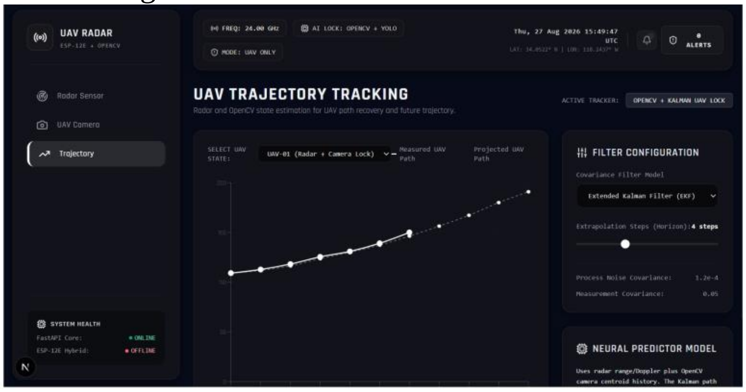


Fig. 8. Separate trajectory screen of the custom-built UAV monitoring web application.

IX. WEB APPLICATION MODULES

Table 3. Main modules of the custom-built web application.

Web Module	Input	Displayed Output
Radar Screen	ESP32 + mmWave radar	Target range/motion, radar plot and controller status
Camera Screen	ESP32-CAM + OpenCV + YOLO	Live video, detected target box and detection information
Trajectory Screen	Recent target positions	Movement path and latest target position

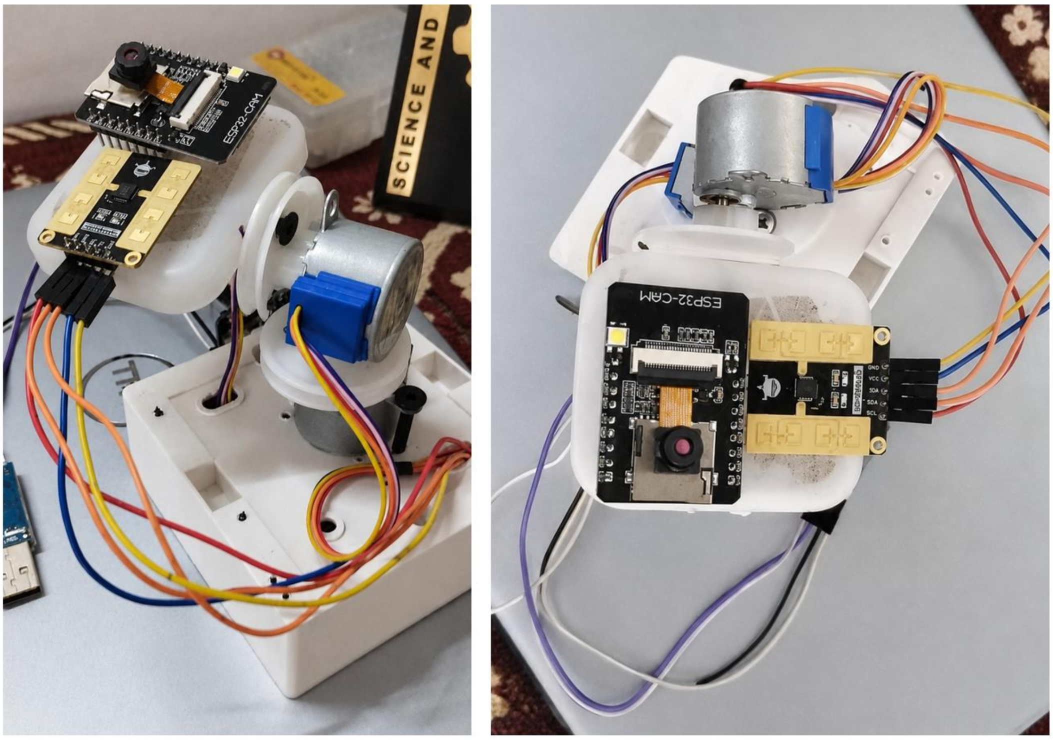
X. PROTOTYPE TESTING AND OBSERVATIONS

Prototype testing is separated into functional validation and quantitative performance measurement. Functional validation checks whether radar telemetry, camera streaming, YOLO detection, trajectory plotting, web visualization and motorized re-acquisition operate together. Quantitative values such as radar range, camera frame rate, detection precision, end-to-end latency and re-acquisition time must be collected with the completed hardware under controlled test conditions.

Table 4. Functional validation of the integrated project prototype.

Test	Expected Operation	Prototype Observation
Radar interface	Show radar/controller state and target report	Radar monitoring screen implemented in the web application.
Camera stream	Open live camera feed	Camera feed can be displayed in the OpenCV/YOLO screen.
YOLO detection	Draw a detection box on visible target	Detection overlay and status are supported by the camera module.
Trajectory plot	Store and join recent target points	Separate trajectory screen displays the movement path.
Combined monitoring	Use radar, camera, trajectory and scan state as one system	Radar, camera and trajectory views are integrated; motor scan command path is included.
Re-acquisition command	Enter LOST/RE-ACQUIRE and command a scan	ESP32-to-ULN2003/28BYJ-48 scan path is included for practical re-acquisition.

The assembled hardware configuration used for practical integration is shown in Fig. 9. The two views contain the ESP32-CAM, 24-GHz mmWave radar sensor and the motorized scanning mount. This assembly is consistent with the architecture in Fig. 1 and provides the physical basis for final outdoor and indoor controlled testing.



(a) (b)

Fig. 9. Practical hardware prototype: (a) front/side view and (b) top view of the radar-camera scanning assembly.

The current observations demonstrate the software and integration workflow of the project. They should not be interpreted as measured outdoor detection-range or accuracy claims. Final performance values must be produced from the completed ESP32, ESP32-CAM, radar and motorized mount using the metrics in Table 5, with test distance, lighting, target type, radar mounting, network conditions and model version reported.

A. Quantitative Evaluation Framework

To convert the functional prototype into a defensible final experimental paper, the following metrics should be recorded from repeated controlled tests. The current manuscript intentionally marks these values as pending rather than inserting simulated or estimated results.

Table 5. Quantitative metrics recommended for final controlled testing.

Metric	Measurement method	Recommended report	Current status
Radar usable range	Known UAV/target distances under stated environment	Maximum repeatable detection distance and detection continuity	Pending controlled test
Camera frame rate	Count decoded frames over measured interval	Mean FPS and stream resolution	Pending controlled test
YOLO detection quality	Labeled validation frames	Precision, recall and mAP50 (or selected mAP metric)	Pending labeled test
End-to-end latency	Timestamp sensor/video input and web update	Mean and 95th-percentile latency in ms	Pending controlled test
Re-acquisition time	Force temporary target loss and restore visibility	Mean time from LOST to TRACKING	Pending controlled test
Trajectory stability	Compare successive displayed positions / reference path	Mean position error or jitter statistic	Pending controlled test

XI. RESULTS AND DISCUSSION

A. Functional Integration Result

The prototype demonstrates the complete information path required by the proposed architecture. Radar-side information can be received by the ESP32 and represented in the radar interface, the ESP32-CAM stream can be opened by the host, YOLO detections can be drawn on the camera view, and recent target positions can be displayed as a trajectory. The assembled radar-camera mount also provides a physical path for scan commands through the ULN2003 driver and 28BYJ-48 motor. These observations confirm system-level interoperability among embedded sensing, computer vision, motor control and web visualization. They do not yet establish a maximum UAV detection range or a statistically validated detection probability.

B. Interpretation Relative to Recent Research

Compared with research-grade radar approaches, the present system operates at a different level of complexity. Micro-Doppler and range-Doppler studies [13]-[28] extract discriminative information directly from radar signals, and 4-D radar approaches such as STIF and JTEA [35], [36] estimate richer trajectories using range, Doppler, azimuth and elevation. The low-cost module used here does not expose an equivalent measurement space, so attempting to reproduce those algorithms would create unsupported claims. The contribution of this prototype is instead the practical fusion of available radar reports with camera-based confirmation. The recent 24-GHz small-drone radar work in [41] supports the suitability of the band for compact sensing, while also illustrating how antenna and beamforming design strongly influence real detection performance.

On the vision side, the camera result is consistent with the broader observation that one-stage detectors provide useful real-time performance but become more difficult when the UAV occupies only a small number of pixels or appears against a complex background [42]. This is why the proposed workflow does not rely on YOLO alone: radar motion can maintain a provisional target state when the visual detector is temporarily weak, and the scan mechanism provides an explicit re-acquisition action. A UAV-specific training set is expected to improve the camera branch, but its benefit should be demonstrated using labeled validation data rather than assumed from model choice.

C. Practical Trade-offs

The architecture favors reproducibility and low cost over sensing precision. Separate ESP32 and ESP32-CAM boards simplify development and isolate motor/radar timing from video traffic, but Wi-Fi introduces variable delay. The 28BYJ-48 motor is inexpensive and suitable for a demonstrator, but backlash and finite step resolution limit directional accuracy unless the mount is calibrated. Likewise, a processed mmWave target report is easy to integrate but provides less information than raw FMCW data. These trade-offs are acceptable for an academic prototype because each limitation is visible and measurable, and each subsystem can later be replaced without changing the overall sensing-processing-visualization architecture.

D. Experimental Readiness

The next experimental milestone is not additional interface development but controlled measurement. Repeated tests should vary target distance, target direction, lighting and network conditions while recording radar reports, YOLO outputs, motor state and timestamps. Table 5 defines the minimum metrics required to convert the present functional demonstration into a quantitative study. Reporting mean values together with the number of trials and test conditions will make the results interpretable. If the hardware is later upgraded to a radar with angle or raw range-Doppler output, the same web and host-processing framework can be retained while more advanced tracking and classification methods are added.

XII. ADVANTAGES, LIMITATIONS AND APPLICATIONS

A. Advantages

- Low-cost and easy-to-understand hardware architecture.
- Radar and camera provide complementary information.
- ESP32 and ESP32-CAM support simple Wi-Fi based integration.
- OpenCV and YOLO provide practical real-time visual detection.
- Custom web application for displaying radar status, annotated camera output, recent trajectory and re-acquisition state.
- The system can be expanded step by step without changing the complete architecture.

B. Limitations

- The low-cost mmWave module may expose only processed range/motion reports and has limited angular resolution and target-classification capability compared with research-grade 4-D radar.
- ESP32-CAM video quality and frame rate depend on Wi-Fi signal and lighting.

- A general YOLO model may not detect small distant UAVs reliably without suitable training data.
- The current trajectory is a prototype visualization and not a high-precision navigation measurement.
- Outdoor testing is required to measure actual range, latency and tracking accuracy.

C. Applications

- Academic UAV surveillance demonstration.
- Low-cost perimeter monitoring experiments.
- Radar-camera sensor-fusion teaching laboratory.
- Drone detection and tracking research prototype.
- Base platform for future motorized pan/tilt or long-range radar development.

XIII. FUTURE SCOPE

Future development should first complete controlled experiments and populate Table 5 with measured values. The next practical step is to calibrate the motor scan angle, relate motor steps to camera/radar direction, and log synchronized radar, camera and state-machine data. After stable timing is achieved, the system can add a Kalman filter or interacting-multiple-model tracker, a UAV-specific YOLO model, and more formal radar-camera association. A later research version could replace the simple radar module with a higher-resolution FMCW/4-D radar so that micro-Doppler classification, angle estimation and predictive tracking techniques from [13]-[28], [35], [36] can be evaluated.

XIV. CONCLUSION

A practical UAV tracking system has been developed around ESP32, ESP32-CAM, a 24-GHz mmWave radar sensor, OpenCV/YOLO, a 28BYJ-48 motorized scan mechanism and a custom web application. The design intentionally translates the broad cognitive-radar idea into a lightweight perception-action loop: sense with radar and camera, confirm and track at the host, visualize the latest state, and command a scan when re-acquisition is required.

The main contribution is system integration rather than a claim of high-end radar accuracy. The prototype combines complementary sensing, computer vision, motor control and web visualization in an architecture that can be assembled with commonly available components. The paper also defines a transparent quantitative test framework so that range, accuracy, frame rate, latency and re-acquisition performance can be reported from real measurements instead of assumed values. This makes the work suitable as a reproducible academic base for later upgrades to UAV-specific vision training, filtering, higher-resolution radar and predictive tracking.

REFERENCES

[1] M. J. Page et al., “The PRISMA 2020 statement: An updated guideline for reporting systematic reviews,” BMJ, vol. 372, Art. no. n71, 2021, doi: 10.1136/bmj.n71.

[2] B. Kitchenham and S. Charters, “Guidelines for performing systematic literature reviews in software engineering,” Keele Univ. and Durham Univ., Tech. Rep. EBSE-2007-01, 2007.

[3] S. Haykin, “Cognitive radar: A way of the future,” IEEE Signal Process. Mag., vol. 23, no. 1, pp. 30–40, Jan. 2006, doi: 10.1109/MSP.2006.1593335.

[4] S. Haykin, Y. Xue, and P. Setoodeh, “Cognitive radar: Step toward bridging the gap between neuroscience and engineering,” Proc. IEEE, vol. 100, no. 11, pp. 3102–3130, Nov. 2012.

[5] M. R. Bell, “Information theory and radar waveform design,” IEEE Trans. Inf. Theory, vol. 39, no. 5, pp. 1578–1597, Sep. 1993, doi: 10.1109/18.259642.

[6] A. Charlish, F. Hoffmann, C. Degen, and I. Schlangen, “The development from adaptive to cognitive radar resource management,” IEEE Aerosp. Electron. Syst. Mag., vol. 35, no. 6, pp. 8–19, Jun. 2020, doi: 10.1109/MAES.2019.2957847.

[7] C. E. Thornton, M. A. Kozy, R. M. Buehrer, A. F. Martone, and K. D. Sherbondy, “Deep reinforcement learning control for radar detection and tracking in congested spectral environments,” IEEE Trans. Cogn. Commun. Netw., vol. 6, no. 4, pp. 1335–1349, Dec. 2020, doi: 10.1109/TCCN.2020.3019605.

[8] S. Durst and S. Brüggenwirth, “Quality of service based radar resource management using deep reinforcement learning,” in Proc. IEEE Radar Conf. (RadarConf21), 2021, doi: 10.1109/RadarConf2147009.2021.9455234.

[9] S. Akbar, R. S. Adve, Z. Ding, and P. W. Moo, “Transfer-based DRL for task scheduling in dynamic environments for cognitive radar,” IEEE Trans. Aerosp. Electron. Syst., vol. 60, no. 1, pp. 37–50, 2024.

[10] Z. Lu, M. C. Gursoy, C. K. Mohan, and P. K. Varshney, “Learning-based cognitive radar resource management for scanning and multi-target tracking,” in Proc. IEEE Int. Conf. Commun. (ICC), 2024, pp. 2785–2790, doi: 10.1109/ICC51166.2024.10622225.

[11] Z. Lu, S. Kalia, M. C. Gursoy, C. K. Mohan, and P. K. Varshney, “Multi-objective reinforcement learning for cognitive radar resource management,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), 2025, pp. 1–5.

[12] Z. Lu, M. C. Gursoy, C. K. Mohan, and P. K. Varshney, “Constrained deep reinforcement learning for cognitive radar resource management,” IEEE Trans. Radar Syst., vol. 4, pp. 627–644, 2026, doi: 10.1109/TRS.2026.3669518.

[13] V. C. Chen, F. Li, S.-S. Ho, and H. Wechsler, “Micro-Doppler effect in radar: Phenomenon, model, and simulation study,” IEEE Trans. Aerosp. Electron. Syst., vol. 42, no. 1, pp. 2–21, Jan. 2006.

[14] P. Molchanov, R. I. A. Harmanny, J. J. M. de Wit, K. Egiazarian, and J. Astola, “Classification of small UAVs and birds by micro-Doppler signatures,” Int. J. Microw. Wireless Technol., vol. 6, no. 3-4, pp. 435–444, 2014.

[15] S. Björklund, “Target detection and classification of small drones by boosting on radar micro-Doppler,” in Proc. 15th Eur. Radar Conf. (EuRAD), 2018, pp. 182–185, doi: 10.23919/EuRAD.2018.8546569.

[16] A. G. Huizing, M. Heiligers, B. Dekker, J. de Wit, L. Cifola, and R. Harmanny, “Deep learning for classification of mini-UAVs using micro-Doppler spectrograms in cognitive radar,” IEEE Aerosp. Electron. Syst. Mag., vol. 34, no. 11, pp. 46–56, Nov. 2019, doi: 10.1109/MAES.2019.2933972.

[17] B.-S. Oh, X. Guo, and Z. Lin, “A UAV classification system based on FMCW radar micro-Doppler signature analysis,” Expert Syst. Appl., vol. 132, pp. 239–255, 2019, doi: 10.1016/j.eswa.2019.05.007.

[18] J. Park, D. H. Jung, K. B. Bae, and S.-O. Park, “Range-Doppler map improvement in FMCW radar for small moving drone detection using the stationary point concentration technique,” IEEE Trans. Microw. Theory Techn., vol. 68, no. 5, pp. 1858–1871, May 2020, doi: 10.1109/TMTT.2019.2961911.

[19] S. Rahman and D. A. Robertson, “Classification of drones and birds using convolutional neural networks applied to radar micro-Doppler spectrogram images,” IET Radar Sonar Navig., vol. 14, no. 5, pp. 653–661, 2020, doi: 10.1049/iet-rsn.2019.0493.

[20] M. Passafiume, S. Maddio, G. Collodi, and A. Cidronali, “Modeling small UAV micro-Doppler signature using millimeter-wave FMCW radar,” Electronics, vol. 10, no. 6, Art. no. 747, 2021, doi: 10.3390/electronics10060747.

[21] R. M. Narayanan, B. Tsang, and R. Bharadwaj, “Classification and discrimination of birds and small drones using radar micro-Doppler spectrogram images,” Signals, vol. 4, no. 2, pp. 337–358, 2023, doi: 10.3390/signals4020018.

[22] J. Liu, Q. Y. Xu, and W. S. Chen, “Classification of bird and drone targets based on motion characteristics and random forest model using surveillance radar data,” IEEE Access, vol. 9, pp. 160135–160144, 2021, doi: 10.1109/ACCESS.2021.3130231.

[23] Y. Yang, F. Yang, L. Sun, T. Xiang, and P. Lv, “Echoformer: Transformer architecture based on radar echo characteristics for UAV detection,” IEEE Sensors J., vol. 23, no. 8, pp. 8639–8653, Apr. 2023, doi: 10.1109/JSEN.2023.3254525.

[24] S. K. Han, J. H. Lee, and Y. H. Jung, “Convolutional neural network-based drone detection and classification using overlaid frequency-modulated continuous-wave range-Doppler images,” Sensors, vol. 24, no. 17, Art. no. 5805, 2024, doi: 10.3390/s24175805.

[25] M. Larrat and C. Sales, “Classification of flying drones using millimeter-wave radar: Comparative analysis of algorithms under noisy conditions,” Sensors, vol. 25, no. 3, Art. no. 721, 2025, doi: 10.3390/s25030721.

[26] P. Janpangngern et al., “High-resolution FMCW radar for small UAV detection using GNU software-defined radio,” IEEE Access, vol. 13, pp. 86396–86412, 2025, doi: 10.1109/ACCESS.2025.3570635.

[27] T. Zhang and X. Song, “A classification algorithm of UAV and bird target based on L/K dual-band micro-Doppler and Mamba,” Drones, vol. 10, no. 4, Art. no. 265, 2026, doi: 10.3390/drones10040265.

[28] F. Tong, K. Zhang, G. Liao, L. Li, J. Xu, and K. Jiang, “Radar UAV/bird trajectory feature classification based on TCN-Transformer and the PC-TimeGAN data augmentation framework,” Sensors, vol. 26, no. 8, Art. no. 2528, 2026, doi: 10.3390/s26082528.

[29] Y. Bar-Shalom and E. Tse, “Tracking in a cluttered environment with probabilistic data association,” Automatica, vol. 11, no. 5, pp. 451–460, 1975, doi: 10.1016/0005-1098(75)90021-7.

[30] D. B. Reid, “An algorithm for tracking multiple targets,” IEEE Trans. Autom. Control, vol. 24, no. 6, pp. 843–854, Dec. 1979, doi: 10.1109/TAC.1979.1102177.

[31] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," IEEE Trans. Autom. Control, vol. 33, no. 8, pp. 780–783, Aug. 1988.
[32] R. P. S. Mahler, "Multitarget Bayes filtering via first-order multitarget moments," IEEE Trans. Aerosp. Electron. Syst., vol. 39, no. 4, pp. 1152–1178, Oct. 2003, doi: 10.1109/TAES.2003.1261119.
[33] B.-T. Vo and B.-N. Vo, "Labeled random finite sets and multi-object conjugate priors," IEEE Trans. Signal Process., vol. 61, no. 13, pp. 3460–3475, Jul. 2013, doi: 10.1109/TSP.2013.2259822.
[34] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," IEEE Trans. Signal Process., vol. 56, no. 8, pp. 3447–3457, Aug. 2008, doi: 10.1109/TSP.2008.920469.
[35] D. Huang, Z. Zhang, X. Fang, M. He, H. Lai, and B. Mi, "STIF: A spatial-temporal integrated framework for end-to-end micro-UAV trajectory tracking and prediction with 4-D MIMO radar," IEEE Internet Things J., vol. 10, no. 21, pp. 18821–18836, 2023, doi: 10.1109/JIOT.2023.3244655.
[36] X. Fang, M. He, D. Huang, Z. Zhang, L. Ge, and G. Xiao, "JTEA: A joint trajectory tracking and estimation approach for low-observable micro-UAV monitoring with 4-D radar," IEEE Trans. Instrum. Meas., vol. 73, pp. 1–14, 2024, doi: 10.1109/TIM.2023.3338650.
[37] M. I. Schöpe, H. Driessen, and A. Yarovoy, "Optimal threat-based radar resource management for multitarget joint tracking and classification," J. Adv. Inf. Fusion, vol. 17, no. 1, pp. 46–62, 2022.
[38] F. Arapoglou, P. Zacharia, and M. Papoutsidakis, "Intelligent counter-UAV threat detection using hierarchical fuzzy decision-making and sensor fusion," Sensors, vol. 25, no. 19, Art. no. 6091, 2025, doi: 10.3390/s25196091.
[39] Z. Zhang, L. Zhang, K. Huang, Z. Jia, M. Wu, L. Cui, and M. Zhang, "Three-dimensional dynamic UAV threat assessment using approach directionality and historical-trend correction for multi-asset protection," Drones, vol. 10, no. 7, Art. no. 536, 2026, doi: 10.3390/drones10070536.
[40] R. Di, X. Gao, Z. Guo, and K. Wan, "A threat assessment method for unmanned aerial vehicle based on Bayesian networks under the condition of small data sets," Math. Probl. Eng., vol. 2018, Art. ID 8484358, 2018, doi: 10.1155/2018/8484358.
[41] K. Jin, S.-S. Han, D. Baek, and H. L. Lee, "Small drone detection using hybrid beamforming 24 GHz fully integrated CMOS radar," Drones, vol. 9, no. 7, Art. no. 453, 2025, doi: 10.3390/drones9070453.
[42] G. Tang, J. Ni, Y. Zhao, Y. Gu, and W. Cao, "A survey of object detection for UAVs based on deep learning," Remote Sensing, vol. 16, no. 1, Art. no. 149, 2024, doi: 10.3390/rs16010149.