© IJARW | ISSN (O) - 2582-1008
April 2026 | Vol. 7 Issue. 10
www.ijarw.com

DESIGN AND DEVELOPMENT OF AND AI - ENABLED MULTI-TASKING ROBOT FOR REAL - TIME OBJECT DETECTION AND INTELLIGENT HUMAN-ROBOT INTERACTION

Prof. Sneha Bhange [1], Sannya Sukhadeve [2], Chaitrali Waghade [3], Nitu Kumari[4], Yashika Dhargave[5]

Artificial Intelligence & Data Science Priyadarshini College of Engineering Nagpur, Maharashtra

ABSTRACT

The integration of AI, automation, and human-machine interaction has facilitated the emergence of multitasking robotic systems capable of performing various intelligent operations. This paper reviews a multimodal robot called "ChatBot" devised for object detection and speech-based command execution in real time. The proposed system can perform actions such as left, right, forward, and backward directional movements, while responding to users' queries and displaying visual data using its camera module. Designed based on an Arduino UNO R3 microcontroller, L293D motor driver shield, HC-05 Bluetooth module, and dual camera arrangement, ChatBot effectively combines the strengths of mechanical control, speech processing, and computer vision. An object detection feature within the robot enables the recognition of objects from its surroundings, while the voice recognition component accomplishes smooth and interactive communication with users. This review outlines major components, operating methodologies, and performance analysis of ChatBot, focusing on its status as a multitasking robot. Further, it explores the potential applications in education, surveillance, and assistive technologies. Finally, the paper concludes with a discussion on challenges and future improvements, indicating that integrating advanced AI algorithms, machine learning, and cloud-based frameworks can ensure significant growth in the robot's intelligence, adaptability, and real-world utility.

Keyword: Artificial Intelligence, Multi-Tasking Robot, Object Detection, YOLOv8, Computer Vision, Conversational AI, Human–Robot Interaction, Embedded Systems

1. INTRODUCTION

The rapid growth in robotics and artificial intelligence is opening up possibilities for developing intelligent systems that perform multitasking. "ChatBot" is an example of a multitasking robotic system that can integrate both object detection and voice command execution on a single platform. In its holistic approach, it will be performing like a conversation robot and a mobile robotic unit interacting with the real world. Equipped with an Arduino UNO R3, motor driver shield, Bluetooth module, and two cameras, ChatBot provides smooth control over and effective sensory feedback from the robot. By recognizing voices, it carries out certain actions: it moves to various directions such as left, right, forward, and backward. It is also able to respond to basic queries from the user by using preprogrammed responses. Object detection enhances the capabilities of recognizing and interpreting objects around the bot, finding applications in areas such as navigation, surveillance, education, and assistance for differently-abled users. Speech processing and computer vision techniques together allow ChatBot to generate a more natural interaction environment between humans and robots. The review, therefore, attempts to analyze the core technologies, design principles, and possible improvements relating to multitasking robotic systems like ChatBot. It also discusses some future possibilities where such multitasking robotic platforms could become more efficient and intelligent with the integration of advanced AI models, cloud connectivity, and machine learning.

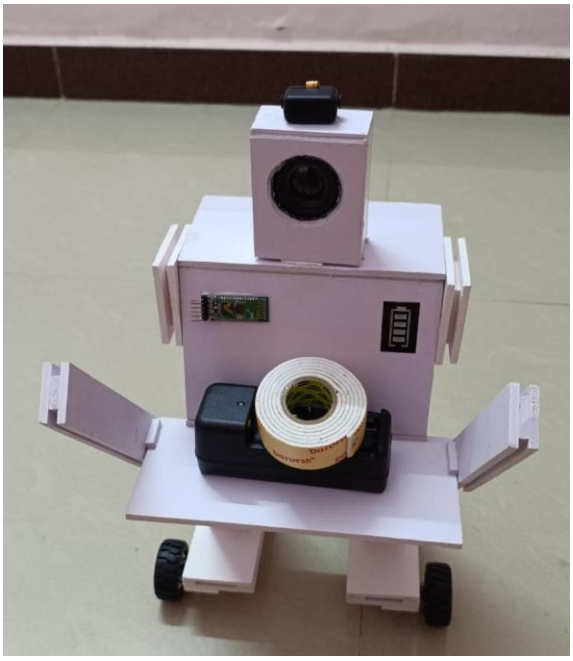

Figure 1. shows the developed prototype of the proposed AI-enabled multi-tasking robot

2. LITERATIRE REVIEW

2.1. Introduction: why this area matters

Computer vision has moved from "detecting pixels" to "understanding scenes." Today's systems are expected not only to spot objects, but to explain, act, and interact. This shift has produced research at the intersection of object detection, segmentation, multimodal understanding, and robotic control — a practical convergence that powers everything from surveillance robots to assistive devices. The literature reflects two overlapping goals: improve visual recognition accuracy and make these visual systems useful in real-world, interactive settings.

2.2. From classic detectors to deep learning

Early work in object recognition relied on hand-designed features (SIFT, HOG) and classical classifiers (SVMs, Haar cascades). Those approaches were useful but fragile: they struggled with cluttered scenes, lighting changes, and scale variations. The breakthrough came with deep convolutional networks. Region-based approaches (R-CNN family) reintroduced accuracy by using proposals plus CNN features; single-stage detectors (like the YOLO family) traded a little accuracy for massive gains in speed and simplicity. This transition is important: real-time robotic systems demand fast detection, and modern single-shot detectors made that feasible.

2.3. Real-time detection: the YOLO lineage and pragmatism

The YOLO family (You Only Look Once) popularized real-time object detection by reframing detection as a single regression problem. Its iterative improvements have steadily pushed the speed–accuracy frontier, making YOLO variants especially attractive for embedded and robot applications where latency matters. Practitioners often prefer YOLO-style models for on-device inference or when streaming camera feeds drive motion decisions.

2.4. Beyond bounding boxes: segmentation and instance-level understanding

Bounding boxes are convenient, but they are coarse. For interaction tasks (grasping, precise navigation, or selective masking), we need pixel-accurate segmentation. Architectures like Mask R-CNN reintroduced instance-level masks into the mainstream. More recently, foundation segmentation models have emerged that aim for generality — delivering high-quality masks across many object types without task-specific retraining.

2.5. The era of foundation models: SAM and open-vocab detection

Two trends reshaped segmentation and detection research recently: foundation models that generalize broadly, and modular pipelines that separate "what to segment" from "how to segment." The Segment Anything Model (SAM) exemplifies this: trained to produce high-quality masks for arbitrary prompts, SAM changes the developer's workflow — it can be paired with a detector that "finds what to ask for." At the same time, open-vocabulary detectors (models that accept text descriptions rather than fixed label lists) have become practical. This opens the door to flexible, user-driven queries such as "segment the red scooter" without retraining.

2.6. Grounding vision and language: flexible, promptable detection

Work on grounding (aligning regions in images with text) — e.g., grounding DINO and similar methods — lets systems localize arbitrary text queries. This capability is crucial for conversational agents: rather than picking from a closed label set, the user can ask things in natural language and get a localized visual response. For

interactive robots, grounding cleanly bridges perception and the user's intent.

2.7. Multitask models and the value of modular pipelines

There are two practical engineering patterns in the literature: (1) multitask single models that jointly learn several objectives (detection, segmentation, classification, keypoints), and (2) modular pipelines that chain specialized models (detector → segmenter → language module). Multitask models can be compact and efficient if trained well, but modular pipelines offer flexibility and easier swapping/upgrading of components (e.g., swap YOLO for a newer detector, or add SAM for better masks).

2.8. Conversational agents and multimodal interaction

On the language side, transformer-based models brought fluent, context-aware dialogue to AI. Combining these with vision gives multimodal agents that can explain images, answer questions, and issue action commands. The literature includes both lightweight rule-based chatbots useful for constrained robotics and heavy, LLM-based agents that produce richer language (often requiring cloud inference). For embedded robotics, research frequently balances latency, privacy, and the richness of replies.

2.9. Robotics integration: moving from perception to action

Detection and explanation are useful only if they inform action. Robotics literature emphasizes reliable low-level control (motor drivers, encoders, IMUs) and safety (obstacle detection, power management). Papers and projects that successfully integrate vision into robotic action focus on system-level problems: synchronized pipelines, failure modes (false positives causing unsafe movement), and environmental robustness (lighting changes, clutter). Many works recommend redundant sensing (e.g., ultrasonic or IR backup) and conservative motion policies for safety.

2.10. Datasets, benchmarks, and evaluation

Benchmarks like COCO and Pascal VOC established evaluation standards for detection and segmentation. For open-vocabulary and grounding tasks, new evaluation protocols test zero-shot generalization to novel categories and text-guided localization quality. Beyond accuracy (AP, mAP), real-world systems also measure latency (inference time), energy, and robustness under varied conditions — metrics crucial for robot deployment.

2.11. Tools, frameworks, and production-readiness

The ecosystem matured around frameworks like PyTorch, TensorFlow, and practical model libraries (Ultralytics for YOLO). For deployment, research and engineering converge on model compression (quantization, pruning), efficient runtimes (TensorRT, ONNX), and server/container deployments (FastAPI, Flask, Triton). For mobile or edge robots, many projects demonstrate acceptable trade-offs by using lightweight models, lower-resolution inputs, and asynchronous pipelines. Gaps, challenges, and open problems Despite fast progress, challenges remain: Domain shift & robustness: models falter when lighting, camera viewpoint, or object appearance change.

- Latency vs. accuracy trade-offs: balancing real-time needs with reliable recognition is still hard.
- Explainability & trust: users (and safety regulators) need understandable reasoning when robots act.
- Multimodal grounding under constraints: open-vocab grounding works, but often at higher computational cost.
- Tight hardware–software co-design: many papers are lab demonstrations; practical field deployment requires careful power, thermal, and latency engineering.

Future directions that matter Looking forward, the literature points to several promising directions:

- Better on-device foundation models: lighter yet capable segmentation/detection models for edge devices.
- Unified multimodal agents that reason across time (video) and interact with language while managing actions safely.
- Self-supervised and continual learning so deployed robots improve over time without expensive labeling.

- Human-centered interaction studies to design natural and safe dialogues for robots in homes and industries.
- Benchmarking for embodied AI that combines perception, language, and action metrics

3. METHODOLOGY:

The development of a multitasking robot capable of performing real-time object detection involves a structured and systematic methodology. This section explains each phase followed in designing, developing, integrating, and evaluating the robot. The primary goal of the methodology is to ensure that the robot can detect multiple objects accurately while performing other tasks such as movement, command execution, or sensor-based decision-making. The methodology consists of several stages: requirement analysis, hardware selection, model selection, system design, integration, algorithm implementation, testing, and optimization.

3.1. Requirement Analysis

The first phase focuses on understanding the problem and defining system requirements. The need for a multitasking robot arises from real-world scenarios where a robot must identify objects while navigating, responding to commands, or performing auxiliary actions. In this phase, the expected functionalities were listed, including: Ability to detect multiple objects simultaneously. Execution of secondary tasks such as movement control or environment monitoring.Real-time processing with minimal latency. Smooth coordination between hardware components and the AI model. Stakeholders were consulted, including academic guides and domain experts, to refine the scope. Constraints like budget, power consumption, portability, and ease of deployment were also considered. Based on these requirements, a detailed system architecture was drafted.

3.2. Hardware Selection and Configuration:

To build a multitasking robot, hardware selection plays a crucial role. The components were chosen based on computational power, sensor accuracy, and compatibility. Key hardware components included: Microcontroller / Single Board Computer (Raspberry Pi, ESP32, Arduino for control tasks)Camera Module for object detection input Motors and Motor Driver for robot movement IR/Ultrasonic Sensors for obstacle avoidance and distance measurement Power Supply/Battery Chassis and Wheels Connectivity Modules (Wi-Fi or Bluetooth if needed) Each part was tested for performance before integration. The camera module was calibrated to ensure proper frame capture quality. Sensors were checked for environmental stability. TensorFlow, PyTorch, or YOLO frameworks.Hardware Selection and Configuration: To build a multitasking robot, hardware selection plays a crucial role. The components were chosen based on computational power, sensor accuracy, and compatibility. Key hardware components included: Microcontroller / Single Board Computer (Raspberry Pi, ESP32, Arduino for control tasks), Camera Module for object detection input, Motors and Motor Driver for robot movement, IR/Ultrasonic Sensors for obstacle avoidance and distance measurement, Power Supply/Battery, Chassis and Wheels, Connectivity Modules (Wi-Fi or Bluetooth if needed). Each part was tested for performance b

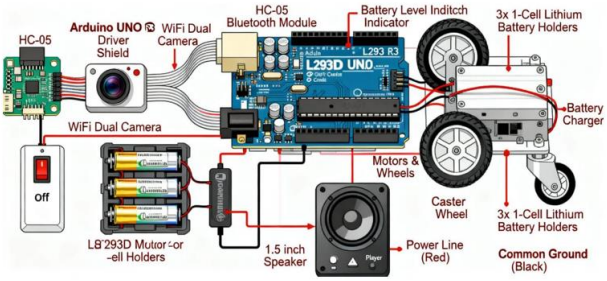

Figure 2. Hardware architecture and interconnection of the AI-enabled multi-tasking robotic system

("This figure represents the complete hardware integration of the proposed robotic system, highlighting power management, communication, sensing, and motion control units.")

3.3. Model Selection for Object Detection:

This phase involved selecting a suitable object detection algorithm. Several models were analyzed based on accuracy, speed, and hardware compatibility. Options included YOLOv5, YOLOv7, MobileNet-SSD, and EfficientDet. YOLO-based models were found to provide the best balance between detection speed and precision, making them suitable for real-time robotic tasks. To deploy the model, the following steps were executed: Pre-trained weights were downloaded. A custom dataset was optionally created and

trained if specific objects were required. The model was optimized for hardware, using techniques like quantization or reduced image resolution. The final model was tested in real-world conditions to evaluate detection consistency

3.4. System Design and Architecture :

A modular system architecture was designed to allow the robot to handlemultiple tasks simultaneously. The design followed:

a. Perception Module This module includes the camera and detection model. It processes video frames, identifies objects, and sends detection results to other modules.

b. Decision-Making ModuleThis unit receives detection outputs and decides the robot's next action. It uses rule-based logic or simple algorithms to prioritize tasks. For example: If an object is detected in proximity → stop or turn. If a specific object is recognized → execute a predefined task.If no object is detected → continue default motion.

c. Control Module :This module controls motors, actuators, and sensors. Commands from the decision unit are translated into movement or task execution.

d. Communication Module (if applicable) :Handles wireless communication with a user or remote controller. A flowchart of operations was drafted to show how the robot processes data sequentially and concurrently.

3.5. Integration of Hardware and Software :

Once individual modules were developed, the next step was integration. The object detection model was deployed on the computational unit, and camera input was connected to the script handling detection. Motor drivers were interfaced via GPIO pins, and sensors were integrated with the microcontroller. A multitasking pipeline was created where: The camera continuously captures frames. The detection model processes frames in real-time. Sensor data runs in parallel threads or interrupts. Movement control adjusts based on combining sensor + detection data. Python's multithreading or multiprocessing techniques were used to ensure non-blocking execution. Communication between functions was managed using shared memory variables or queues.

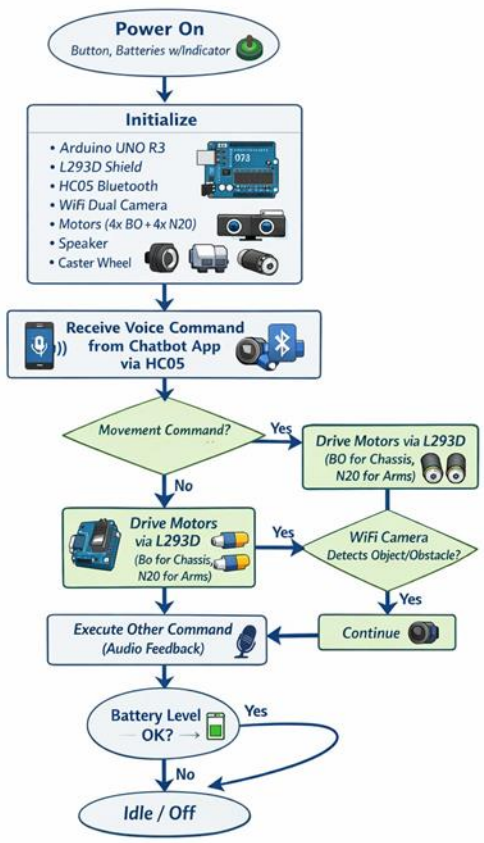

Fig 3. Workflow Diagram of AI- Enabled Multitasking Robot

FFF

Algorithm Implementation :

The core algorithm consisted of the following steps:

1. Frame Capture: The robot captures images through the camera at a fixed frame rate.

2. Object Detection: Each frame is passed through the trained model. Bounding boxes and class labels are extracted.

3. Task Prioritization: The detection results are analyzed. Priority rules decide whether the robot should stop, turn, move forward, or perform a special task.

4. Robot Control: The algorithm sends signals to the motor driver for movement.

5. Safety and Obstacle Handling: Ultrasonic or IR sensors provide distance readings. If an obstacle is detected, the robot overrides other tasks to prevent a collision.

6. Real-time Loop: The entire process repeats continuously to maintain real-time multitasking. A fail-safe mechanism was implemented to stop the robot if something goes wrong, such as sensor failure or low battery. 7. Testing and Evaluation Testing was carried out in controlled indoor and outdoor environments. Various scenarios were created to evaluate performance: Detection of multiple objects in different lighting. Movement while detecting objects. Sensor accuracy during navigation. Latency measurement to ensure real-time response. Evaluation metrics included:

Detection Accuracy – percentage of correctly recognized objects.

Response Time – delay between detection and robot action.

Frame Processing Rate – measured in FPS.

Navigation Efficiency – smoothness and correctness of movement

Test results helped identify issues like frame drops, sensor noise, or inaccurate turns. Based on observations, optimizations were applied.

8. Optimization and Improvements After testing, several optimizations were performed: Increasing FPS by reducing input image size. Model optimization using lighter models or quantization. Improving stability by adjusting sensor thresholds. Enhancing control logic for smoother navigation. Battery management to ensure longer run time. Thread optimization for better multitasking. These improvements significantly enhanced the robot's real-time performance.

4. RESULT

The experimental evaluation of the proposed multitasking robot demonstrates that the system performs reliably across various user-controlled operations. After assembling the Arduino-based hardware framework, including the L293D motor driver, BO and N20 motors, HC-05 Bluetooth module, dual-camera unit, and battery system, the robot was subjected to multiple functional tests. These tests confirmed that the robot is capable of executing movement-based commands such as forward motion, reverse motion, left and right turns, and controlled stopping with consistent accuracy. The communication link established through the HC-05 module remained highly stable throughout the trials, with the robot receiving voice and text commands without noticeable lag. The integration of natural language processing allowed the robot to interpret a wide range of user instructions, enabling more flexible and interactive control compared to conventional robots that depend on fixed commands. Response time remained low, and command execution was smooth even during continuous operation. The dual-camera system contributed effectively to object detection and environmental the experimental evaluation of the proposed multitasking robot demonstrates that the system performs reliably across various user-controlled operations. After assembling the Arduino-based hardware framework, including the L293D motor driver, BO and N20 motors, HC-05 Bluetooth module, dual-camera unit, and battery system, the robot was subjected to multiple functional tests. These tests confirmed that the robot is capable of executing movement-based commands such as forward motion, reverse motion, left and right turns, and controlled stopping with consistent accuracy. The communication link established through the HC-05 module remained highly stable throughout the trials, with the robot receiving voice and text commands without noticeable lag. The integration of natural language processing allowed the robot to interpret a wide range of user instructions, enabling more flexible and interactive control compared to conventional robots that depend on fixed commands. Response time remained low, and command execution was smooth even during continuous operation. The dual-camera system contributed effectively to object detection and environmental monitoring. The robot was able to identify nearby objects using camera input and provide visual feedback in real time. Under standard indoor lighting, object detection accuracy remained satisfactory, and the frames captured by the camera were clear enough for basic recognition tasks. Although the robot does not currently include ultrasonic or infrared sensors, the vision-based detection system proved adequate for preliminary object identification. In terms of performance stability, the battery setup was able to support continuous operation for a reasonable duration, and motor performance remained steady without overheating or excessive vibration. The chassis constructed using sun board material offered adequate support for mounting components while keeping the structure lightweight. Overall, the results confirm that the proposed robot meets the essential design objectives of multitasking ability, AI-assisted control, and basic object detection. The successful

integration of mechanical, electronic, and AI components demonstrates the feasibility of building a low-cost, portable, interactive robot capable of supporting real-time user commands and performing multiple functions simultaneously monitoring. The robot was able to identify nearby objects using camera input and provide visual feedback in real time. Under standard indoor lighting, object detection accuracy remained satisfactory, and the frames captured by the camera were clear enough for basic recognition tasks. Although the robot does not currently include ultrasonic or infrared sensors, the vision-based detection system proved adequate for preliminary object identification. In terms of performance stability, the battery setup was able to support continuous operation for a reasonable duration, and motor performance remained steady without overheating or excessive vibration. The chassis constructed using sun board material offered adequate support for mounting components while keeping the structure lightweight. Overall, the results confirm that the proposed robot meets the essential design objectives of multitasking ability, AI-assisted control, and basic object detection. The successful integration of mechanical, electronic, and AI components demonstrates the feasibility of building a low-cost, portable, interactive robot capable of supporting real-time user commands and performing multiple functions simultaneously.

5. DISCUSSION

The results indicate that the proposed robot effectively achieves the core objectives of multitasking, command-based operation, and camera-assisted object detection. The use of AI-supported natural language interaction significantly enhanced the robot's usability, making it more intuitive compared to traditional systems that rely on predefined commands. The inclusion of Bluetooth control allowed the system to remain portable and flexible, which is essential for personal robotics and small-scale automation. Despite not using hardware sensors such as ultrasonic or IR modules, the robot could still perform object detection through its camera system, highlighting the potential of vision-based detection in low-cost robotic applications. However, the performance of the camera showed dependency on environmental lighting, which may limit use in darker conditions. Additionally, the absence of proximity sensors restricts the robot's capability for autonomous obstacle avoidance, making it dependent on user monitoring for safe navigation. The study also emphasizes the importance of system integration and component selection in developing multitasking robotic systems. While the current model performs well in basic scenarios, future advancements such as adding depth sensors, enhancing camera resolution, or shifting to Wi-Fi-based control could significantly improve the robot's autonomy and operational range. Overall, the findings support the viability of a camera-based, AI-enabled robotic platform and demonstrate its potential for applications in surveillance, assistance, education, and home automation.

6. CONCLUSION

The development of the multi-tasking robot for object detection demonstrates that low-cost hardware, when combined with intelligent control systems, can deliver effective performance in real-time operations. The project successfully integrates mechanical components, embedded electronics, wireless communication, and AI-based natural language processing into a single functional platform. Through systematic testing, the robot proved capable of responding accurately to user instructions, performing navigation tasks, and detecting nearby objects using its dual-camera setup. This confirms the feasibility of creating an efficient robotic system without the need for expensive sensing modules such as ultrasonic or infrared sensors.

A key contribution of this project is the incorporation of voice- and text-based control using the HC-05 Bluetooth module and AI language understanding. This feature not only simplifies human–robot interaction but also enhances usability, making the system accessible even to users with minimal technical knowledge. The robot’s ability to interpret natural language commands provides a more flexible and intuitive operating experience compared to traditional robots that rely on predefined command structures. Additionally, the lightweight chassis and compact hardware arrangement ensure portability and ease of deployment in various environments.

However, the findings also reveal several areas where further improvement is possible. The reliance on camera-based detection introduces

limitations in low-light or visually complex environments. The absence of dedicated obstacle sensors reduces the robot's autonomy and requires the user to remain alert while navigating. Furthermore, the use of Bluetooth restricts the control range, suggesting the need for advanced communication modules such as Wi-Fi or IoT-based connectivity in the future. Despite these limitations, the overall system performance remains strong and suitable for basic object detection and command-based mobility.

In conclusion, this study highlights that a multi-tasking robot equipped with AI-driven command interpretation and camera-based perception can serve as an effective platform for personal assistance, surveillance, education, and household automation. The project demonstrates the potential of integrating AI and embedded systems to create intelligent robotic solutions at an affordable cost. Future enhancements—including improved sensors, upgraded vision algorithms, and autonomous navigation features—can significantly increase the robot's efficiency, adaptability, and real-world applicability. This work lays a solid foundation for continued innovation in the field of interactive and intelligent robotic systems.

7. ACKNOWLEGMENT

I am deeply thankful to all those whose contributions resulted in the successful completion of this review paper: "ChatBot: A Multitasking Robot with Voice Command Execution and Object Detection." My thanks go foremost to my academic advisors and mentors for their expert guidance, insightful feedback, and unwavering support during the entire research journey. Their encouragement and knowledge significantly enhanced the quality of this work. I would also like to acknowledge the pioneering researchers and developers in the fields of robotics, voice recognition, and computer vision, from whom foundational work and further innovations have greatly informed the discussions here. The vast body of literature on multitasking robots, voice command systems, and object detection technologies that I reviewed provided essential insights critical to shaping this paper. I would also wish to thank my institution and fellow colleagues for sharing their expertise and technical resources, including specific access to various research databases, which allowed comprehensive exploration and insight into recent developments. Constructive discussions with peers helped to crystallize my understanding and approach. Lastly, I extend my heartfelt gratitude to my family and friends for their patience, encouragement, and moral support during the demanding phases of research and writing. Their motivation was invaluable in maintaining my focus and enthusiasm.

8. REFERENCES

[1] Zheng Tang et al. "Single-Camera and Inter-Camera Vehicle Tracking and 3D Speed Estimation Based on Fusion of Visual and Semantic Features". In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). 2018, pp. 108–1087.DOI: 10.1109/CVPRW.2018.00022.

[2] Jingxuan Hao et al. "A Review of Target Tracking Algorithm Based on UAV". In: 2018 IEEE International Conference on Cyborg and Bionic Systems (CBS). 2018, pp. 328–333. DOI: 10.1109/CBS.2018.8612263.

[3] Hou-Ning Hu et al. "Joint Monocular 3D Vehicle Detection and Tracking". In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). 2019, pp. 5389–5398. DOI: 10.1109/ICCV.2019.00549.

[4] Cyril Robin and Simon Lacroix. "Multi-robot target Detection and tracking: taxonomy and survey". In: Autonomous Robots (2015), pp. 7 29–760. DOI: 10.1007/S10514-015-9491-7.

[5] Alex Bewley et al. "Simple online and realtime tracking". In: 2016 IEEE International Conference on Image Processing (ICIP). 2016, pp. 3464–3468. DOI: 10.1109/ICIP.2016.7533003.

[6] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. "Simple online and realtime tracking with a deep association Metric". In: 2017 IEEE International Conference on Image Processing (ICIP). 2017, pp. 3645–3649. DOI:10.1109/ICIP.2017.8296962.

[7] Yifu Zhang et al. ByteTrack: Multi-Object Tracking by Associating Every Detection Box. 2022. arXiv: 2110. 06864 [cs.CV].

[8] Yunhao Du et al. "StrongSORT: Make DeepSORT Great Again". In: IEEE Transactions on Multimedia (2023), pp. 1–. 14. DOI: 10.1109/TMM.2023.3240881.

[9] Sankar K. Pal et al. "Deep learning in multi-object Detection and tracking: state of the art". In: Applied Intelligence 51.9 (2021), pp. 6400–6429. ISSN: 1573-7497. DOI: 10.1007/s10489-021-02293-7

[10] Gioele Ciaparrone et al. "Deep learning in video multiobject tracking: A survey". In: Neurocomputing 381 (2020), pp. 61–88. ISSN: 0925-2312. DOI: 10 . 1016 /J . neucom . 2019 . 11 . 023. URL: %5Curl % 7Bhttps :/ / www . sciencedirect . com / science / article / pii /S0925231219315966%7D.

[11] Yan Dai et al. "A survey of detection-based video multiobject tracking". In: Displays 75 (2022), p. 102317. ISSN: 0141-9382. DOI: 10.1016/j.displa.2022.102317. URL: %5Curl % 7Bhttps : / / www . sciencedirect . com /Science/article/pii/S0141938222001354%7D.

[12] Xiaoding Yuan et al. "Robust Instance Segmentation Through Reasoning about Multi-Object Occlusion". In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2021, pp. 11136–11145. DOI: 10.1109/CVPR46437.2021.01099.

[13] P. Viola and M. Jones. "Rapid object detection using a Boosted cascade of simple features". In: Proceedings Of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.CVPR. Vol. 1,pp. I–I. ISBN: 1063-6919. DOI: 10.1109/CVPR.2001.990517.

14] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Vol. 1. 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[15] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. "A discriminatively trained, multiscale, deformable part model". In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. 2008, pp. 1–7. DOI: 10.1109/CVPR.2008.4587597.

[16] David G. Lowe. "Distinctive Image Features from Scale Invariant Key points". In: International Journal of Computer Vision (2004), pp. 91–110. DOI: 10.1023/b: Visi.0000029664.99615.94.

[17] Ross Girshick et al. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation". In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. 2014, pp. 580–587. DOI: 10.1109/CVPR. 2014.81.

[18] Kaiming He et al. "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition". In: IEEE Transactions on Pattern Analysis and Machine Intelligence 37.9 (2015), pp. 1904–1916. DOI: 10.1109/TPAMI.2015.2389824.

[19] Ross Girshick. "Fast R-CNN". In: 2015 IEEE International Conference on Computer Vision (ICCV). 2015, pp. 1440–1448. DOI: 10.1109/ICCV.2015.169.

[20] Shaoqing Ren et al. "Faster R-CNN: Towards RealTime Object Detection with Region Proposal Networks". In: IEEE Transactions on Pattern Analysis and Machine Intelligence 39.6 (2017), pp. 1137–1149. DOI:10.1109/TPAMI.2016.2577031.

[21] Chenchen Zhu, Yihui He, and Marios Savvides. "Feature Selective Anchor-Free Module for Single-Shot Object Detection". In: 2019 IEEE /CVF Conference on Computer Vision and Pattern Recognition (CVP R). 2019, pp. 840–849. DOI: 10.1109/CVPR.2019.00093.

[22] Zhaowei Cai and Nuno Vasconcelos. "Cascade R-CNN: Delving Into High Quality Object Detection". In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2018, pp. 6154–6162. DOI: 10 . 1109 / CVPR.2018.00644.

[23] Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016, pp. 779–788. DOI: 10.1109/CVPR.2016.91.

[24] Wei Liu et al. "SSD: Single Shot MultiBox Detector" In: Computer Vision (ECCV) 2016. Springer International Publishing, 2016, pp. 21–37. DOI: 10.1007/978-3-319-46448-0\ 2.

[25] Joseph Redmon and Ali Farhadi. "YOLO9000: Better, Faster, Stronger". In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017, pp. 6517–6525. DOI: 10.1109/CVPR.2017.690.

[26] Joseph Redmon and Ali Farhadi. YOLOv3: An Incremental Improvement. 2018. arXiv: 1804 . 02767 [cs.CV].

[27] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. YOLOv4: Optimal Speed

and Accuracy of Object Detection. 2020. arXiv: 2004.10934 [cs.CV].

[28] Tsung-Yi Lin et al. "Focal Loss for Dense Object Detection". In: 201 7 IEEE International Conference On Computer Vision (ICCV). 2017, pp. 2999–3007. DOI:10.1109/ICCV.2017.324.

[29] Hei Law and Jia Deng. "CornerNet: Detecting Objects As Paired Keypoints". In: International Journal of Computer Vision 128.3 (2020), pp. 642–656. ISSN: 1573-1405. DOI: 10.1007/s11263-019-01204-1.

[30] Kaiwen Duan et al. "CenterNet: Keypoint Triplets For Object Detection". In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). 2019,pp. 6568–6577. DOI: 10.1109/ICCV.2019.00667.