ENHANCING UAV TARGET DETECTION AND TRACKING ALGORITHMS FOR IMPROVED EFFICIENCY

Abstract - **Unmanned Aerial Vehicles (UAVs) play an important role in many applications such as surveillance, reconnaissance, and disaster management. However, effective target detection and tracking remains a major challenge to maximize efficiency. This research paper aims to study and develop UAV target detection and tracking systems to improve the performance of target identification and tracking tasks. These studies will investigate computer vision techniques, machine learning algorithms and sensor fusion methods to improve the accuracy, robustness and real-time performance of target detection and tracking systems of drone platforms. By analysing and comparing performance, this study aims to gain a deeper understanding of the advantages and limitations of existing algorithms and propose new strategies to overcome these limitations. The results of this research are expected to help improve drone-based monitoring and surveillance capabilities, thereby increasing their effectiveness in important projects and applications.**

Keywords - Unmanned Aerial Vehicles (UAVs), Surveillance, Reconnaissance, Target Detection and systems, Computer vision techniques, Machine learning algorithms, Monitoring, Accuracy

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have become important tools in many fields, including military, civil and commercial, due to their high performance, effectiveness, and ability to operate in remote or dangerous areas. Among the many applications, one of the main roles of drones is target detection and tracking, which is important for tasks such as surveillance, surveillance, surveillance borders, disaster response and search and rescue. Good target detection and tracking algorithms are crucial to optimizing drones' ability to perform these tasks. However, achieving accurate and robust target detection and tracking faces significant challenges due to factors such as environmental changes, occlusion, target structures and sensor limit methods.

UAV target detection and tracking algorithms often face limitations in terms of accuracy, real-time performance, and flexibility. Therefore, there is a growing need to explore technologies and methods to improve the performance of algorithms.

This research aims to solve these problems by examining and developing UAV target detection and tracking algorithms. This research aims to improve the accuracy, robustness, and real-time performance of target detection to acquire and track systems of UAV platforms by leveraging cutting-edge computer vision technology, machine learning algorithms and sensor fusion methods.

Through a comprehensive review of the existing literature, followed by clinical evaluation and comparative study, this study aims to evaluate the quality and drawbacks of existing algorithms and propose new ideas to solve their limitations. The ultimate goal is to create new solutions that improve the capabilities of UAVs in mission detection and tracking operations. The results of this research are expected to have a significant impact in many areas such as prevention, surveillance, law enforcement, damage control, damage, and property maintenance. This research focuses on the advancement of UAV technology and its applications in the field, alarm, and operation issues by increasing the effectiveness of UAV target detection and tracking systems.

Unmanned Aerial Vehicles (UAVs) have become increasingly integral in various mapping applications, including target detection and tracking. Olsen et al. (Year) provide a comprehensive survey detailing UAV usage in photogrammetric surveying within the mapping industry, shedding light on the diverse applications and challenges in this domain. Moreover, Al-Falluji et al. (Year) propose a real-time UAV detection and tracking system leveraging machine learning techniques. Their work emphasizes algorithmic efficiency, crucial for seamless operation in dynamic environments. Smith et al. (Year) introduce a novel approach combining Kalman filters and optical flow techniques to enhance UAV target tracking accuracy. This fusion-based strategy demonstrates promising results in improving tracking performance, especially in scenarios with occlusions or rapid motion changes. Additionally, Gupta et al. (Year) offer a comprehensive survey of deep learning-based object detection and tracking methods tailored for UAV applications. Their analysis highlights the state-of-the-art techniques, paving the way for future advancements in algorithmic sophistication and performance. Furthermore, Chen et al. investigate the benefits of sensor fusion techniques for enhancing UAV target detection accuracy and efficiency. By integrating data from multiple sensors, their approach demonstrates robustness against environmental uncertainties, contributing to improved target tracking capabilities. Zhang et al. (Year) explore the utilization of swarm intelligence algorithms to optimize target detection and tracking in UAV networks. Their research showcases the potential of decentralized decision-making processes to enhance overall system efficiency and adaptability to dynamic environments. Moreover, Li et al. (Year) delve into the development of adaptive algorithms capable of dynamically adjusting UAV target detection and tracking strategies based on environmental cues. This adaptive framework enables UAVs to operate effectively in challenging conditions, ensuring reliable performance across diverse scenarios.

Lastly, Wang et al. present a multi-sensor fusion framework designed to improve UAV target tracking performance, particularly in adverse environmental conditions. By integrating data from diverse sensor modalities, such as visual, infrared, and radar, their approach enhances tracking robustness and accuracy, essential for mission-critical applications. Collectively, these studies contribute to the advancement of UAV target detection and tracking algorithms, offering valuable insights and methodologies for enhancing efficiency, accuracy, and adaptability in various operational scenarios.

II. RELATED WORKS

[1] A vision-based approach for detecting and tracking cooperative flying vehicles using UAVs equipped with monocular cameras is presented. The method integrates template matching and morphological filtering algorithms, leveraging navigation hints from cooperative formations to predict target positions efficiently. However, limitations in complex backgrounds and rapid changes in illumination conditions suggest the exploration of alternative tracking methods for improved performance.

[2] An enhanced YOLOv4 algorithm, termed YOLOv4Drone, is proposed to detect small targets in UAV images against complex backgrounds. The algorithm integrates hollow convolution, ultra-lightweight subspace attention mechanism, and soft non-maximum suppression to mitigate missed targets due to occlusion. Although effective, further exploration of advanced image enhancement techniques and robust post-processing algorithms could enhance detection accuracy.

[3] A saliency-enhanced MDnet algorithm is proposed for remote sensing target tracking in UAV aerial videos. The algorithm incorporates a saliency module and sample screening mechanisms to improve tracking performance significantly. Further validation on diverse datasets and exploration of computational efficiency in real-time applications is warranted. Additionally, generalizability to various tracking scenarios requires exploration.

[4] A multitarget real-time tracking algorithm for UAV IoT applications is introduced. The algorithm integrates detection and tracking processes into a single framework, addressing the limitations of traditional tracking algorithms in dynamic environments. Further exploration of real-world UAV IoT applications and discussion on specific challenges of the proposed algorithm is needed.

[5] A camera-based approach for multi-target detection and tracking in UAVs to enhance collision avoidance systems is proposed. While effective, reliance on optical sensors may limit performance in adverse conditions. Exploration of alternative sensor modalities such as radar could provide complementary capabilities. Additionally, real-time processing constraints and environmental factors should be considered for practical deployment.

[6] A vision-based collision detection algorithm for UAVs is presented to enhance sense and avoid capabilities. While achieving impressive detection distances and warning times, challenges in real-world implementation and adaptability remain. Further validation and scalability testing are needed to ensure robust performance across diverse operational scenarios.

[7] A vision-based system for target finding and inspection using a multirotor UAV system is proposed. While effective, challenges related to sensor uncertainties and environmental factors may impact performance. Further optimization and validation in real-world scenarios are necessary to enhance reliability and scalability.

[8] An algorithm for planning optimal trajectories for tactical class UAV reconnaissance missions is presented. While comprehensive, further exploration of dynamic changes in flight plans and scalability is needed. Additionally, considerations for real-world implementation and modification mechanisms are warranted.

[9] A camera-based target detection, tracking, and localization solution for UAVs is proposed. Despite achieving high efficiency, challenges related to object detection accuracy and computational resources remain. Future work should focus on addressing these limitations and enhancing real-world applicability.

[10] Recent advancements in UAV detection, classification, and tracking are highlighted. While comprehensive, the lack of real-world testing and scalability considerations may limit practical applicability. Further validation and exploration of proposed solutions in diverse scenarios are warranted to address practical challenges in the field.

[11] An efficient target detection system for UAVs at low altitude is proposed, integrating the Edge Potential Function (EPF) with the Simulated Annealing Pigeon-inspired Optimization (SAPIO) algorithm. SAPIO mitigates the tendency to converge to local optima, enhancing target recognition. Comparative analysis demonstrates SAPIO's robustness, underscoring its effectiveness in UAV target detection.

[12] A camera-based target detection and positioning UAV system for Search and Rescue (SAR) missions is presented, utilizing image processing algorithms for target identification. While enhancing operational efficiency, further improvements in image processing algorithms are warranted. Real-world testing is necessary to assess system reliability and robustness.

[13] Achieving real-time multiple object tracking (MOT) for unmanned aerial vehicles (UAVs) is addressed by proposing a deep learning model combining detection and tracking methods. The model's design ensures real-time performance by utilizing adjacent frame pairs and a multi-loss function. While demonstrating superior performance

compared to existing methods, future work may focus on enhancing detection and tracking in challenging environments.

[14] A system integrating algorithmic decision-making with vision-based navigation for ground target inspection by multirotor UAVs is introduced. Leveraging an OODA loop architecture, the system demonstrates reliability in target detection and navigation. However, further algorithm refinement is necessary to improve performance under unreliable target detection conditions.

[15] The trajectory optimization of multi-UAVs for target tracking in urban environments is addressed using a hybrid method combining Model Predictive Control (MPC) and Improved Grey Wolf Optimizer (IGWO). The proposed method provides credible modeling of urban environments, enabling real-time path planning while considering various constraints. Future work could explore the applicability of the method in diverse urban scenarios.

[16] Detection and tracking of infrared small targets are tackled through a two-stage approach combining deep learning-based detection and multi-frame filtering. The method achieves high recall and precision rates, addressing challenges posed by small target size and environmental factors. However, further investigation into computational complexity and generalizability is warranted.

[17] Cooperative path planning for multi-UAVs in 3D environments is facilitated by a hybrid approach combining Lyapunov Guidance Vector Field (LGVF) and Improved Interfered Fluid Dynamical System (IIFDS). The method effectively addresses challenges of target tracking and obstacle avoidance, highlighting its potential in complex environments. However, real-world validation and scalability considerations are essential for practical implementation.

[18] Visual detection and tracking of cooperative UAVs using deep learning techniques are explored, showcasing a novel architecture leveraging the You Only Look Once (YOLO) object detection system. The proposed solution demonstrates high accuracy in detecting and tracking cooperative UAVs, highlighting its potential for real-world applications. Further validation under diverse environmental conditions may be warranted.

[19] A methodology utilizing UAVs and video processing techniques for vehicle tracking data acquisition is proposed, providing a dynamic data collection system for traffic monitoring. The methodology demonstrates high accuracy in evaluating individual vehicle paths, offering a versatile solution for traffic analysis. However, scalability and real-world deployment considerations need to be addressed for wider adoption.

[20] UAV autonomous path planning for multiple moving targets is addressed through a three-step solution integrating radar tracking, prediction, and path planning algorithms. The proposed scheme enhances UAV intelligence and tracking capability, offering improvements in long-term tracking. However, algorithm complexity and real-time performance requirements warrant further investigation for practical deployment.

[21] The article presents FastUAV-NET, a multi-UAV detection and tracking algorithm designed for embedded platforms. It addresses challenges in detecting and tracking multiple Unmanned Aerial Vehicles (UAVs) in airborne videos, utilizing a shallow deep learning-based method combined with a scalable tracking algorithm. FastUAV-NET achieves high accuracy while maintaining low computational complexity, making it suitable for real-time applications on embedded devices.

[22] The paper proposes an integration of UAV and Landsat imagery for improved watershed-scale evapotranspiration (ET) prediction. Traditional methods of estimating ET face limitations in spatial resolution and accuracy, which the integration of UAV and satellite data aims to overcome. The study demonstrates the effectiveness of this approach in improving ET prediction accuracy, offering implications for water resource management using remote sensing techniques.

[23] The study aims to assess the impact of pruning on tree structure in a lychee orchard using multi-spectral UAV imagery. It seeks to accurately measure changes in tree structure, such as crown perimeter, width, height, area, and Plant Projective Cover (PPC), before and after pruning. The developed object-based tree crown delineation approach addresses limitations of existing methods and provides a novel way to evaluate pruning effects.

[24] The document addresses the challenging nature of wilderness search and rescue (SAR) missions, proposing an all-in-one camera-based target detection and positioning system integrated into a fully autonomous fixed-wing UAV. This system enables real-time target identification, post-target identification, location, and aerial image collection, promising improved efficiency and reduced workloads in SAR missions, especially during natural disasters.

[25] The maintenance of photovoltaic (PV) plants is crucial for profitability, yet traditional inspection methods are time-consuming. The paper proposes a vision-based guidance method using a computer vision line-tracking algorithm to enhance UAV flight monitoring for PV plant inspections. This algorithm corrects GNSS position errors and improves image acquisition accuracy, potentially revolutionizing PV plant maintenance.

[26] The paper addresses the need for an effective forest fire detection and tracking method using UAVs. It proposes forest fire detection and tracking algorithms leveraging image processing techniques, including median filtering, color space conversion, and threshold segmentation. The proposed system architecture covers various components necessary for effective detection and tracking, offering a comprehensive solution for forest fire management.

[27] The study focuses on developing multi-target detection and tracking algorithms from a single camera mounted on a UAV. It proposes a novel approach combining background subtraction and optical flow methods for effective detection and tracking of small UAVs. The

algorithm demonstrates high accuracy and efficiency, making it suitable for collision avoidance systems and other UAV applications.

[28] The paper addresses the deployment of multiple UAVs for on-demand coverage while maintaining connectivity. It proposes centralized and distributed deployment algorithms to optimize UAV deployment for known on-ground users and autonomous coverage for unknown users. These algorithms consider user distribution and quality of service requirements, enabling obstacle avoidance and network connectivity maintenance.

[29] The paper presents a Deep Reinforcement Learning (DRL) approach for controlling multiple UAVs to track First Responders (FRs) accurately in challenging environments. The proposed DRL-based controller selects optimal actions based on Cram er-Rao Lower Bound (CRLB) and achieves high tracking performance with low runtime cost, offering a promising solution for UAV-based target tracking.

[30] The paper addresses persistent target tracking in urban environments using UAVs, proposing a novel Target Following DQN (TF-DQN) approach based on deep reinforcement learning. This approach enables the UAV to learn target motion and track it persistently while avoiding obstacles, offering a computationally simple and effective solution for diverse urban tracking scenarios.

III. PROBLEM DEFINATION

Autonomous landing systems for unmanned aerial vehicles (UAVs) are aimed at rotorcraft, which have a lower landing risk because they can fly horizontally and land vertically over the landing area. However, fixed-wing UAVs use wheeled loaders and operate in taxi mode during take-off and landing. In order to reduce landing risk and increase landing accuracy, a stable autonomous landing system is needed, especially to ensure the safety of UAVs.

IV. PROPOSED SOLUTION

A. Sensor Fusion Method:

Autonomous landing systems for unmanned aerial vehicles (UAVs) are aimed at rotorcraft, which have a lower landing risk because they can fly horizontally and land vertically over the landing area. However, fixed-wing UAVs use wheeled loaders and operate in taxi mode during take-off and landing. In order to reduce landing risk and increase landing accuracy, a stable autonomous landing system is needed, especially to ensure the safety of UAVs.

Prediction Step:

$$x(t+1) = F(t) * x(t) + B(t) * u(t) + w(t) \quad \ldots\ldots\ldots\ldots [1.1]$$

Update Step:

$$K(t+1) = [P(t+1|t) * H(t+1)] / [H(t+1) * P(t+1|t) * H(t+1) + R(t+1)] \quad \ldots\ldots\ldots [1.2]$$

$$x(t+1) = x(t+1) \quad \ldots\ldots\ldots [1.3]$$

where,

$x(t+1)$ = predicted state at time t+1
$F(t)$ = state transition matrix
$x(t)$ = state at time t
$B(t)$ = control input matrix
$u(t)$ = control input at time t
$w(t)$ = process noise at time t

In the prediction step, we anticipate the state of the system at the next time step (t+1). This prediction is based on several factors. Firstly, we consider how the state evolves over time without any external influences, which is encapsulated by the state transition matrix ($F(t)$). Additionally, we incorporate the current state ($x(t)$) and how it's affected by any control inputs ($u(t)$), represented by the control input matrix ($B(t)$). However, real-world systems are subject to uncertainties and randomness. To account for this, we introduce the concept of process noise ($w(t)$), which represents the unpredictable variations or disturbances in the system. By combining these elements using Equation [1.1], we generate an estimate of the system's state at the next time step.

Following the prediction, we refine our estimate based on new observations and measurements. The update step begins by calculating the Kalman gain ($K(t+1)$), which determines how much weight to assign to the new observations relative to the prediction. This gain is computed using the predicted error covariance matrix ($P(t+1|t)$), which quantifies the uncertainty in the prediction, and the observation matrix ($H(t+1)$), which relates the state space to the observation space. Importantly, we also consider the uncertainty associated with the measurements, captured by the measurement noise covariance matrix ($R(t+1)$). Equation [1.2] illustrates how these factors are combined to calculate the Kalman gain. Finally, Equation [1.3] simply updates our predicted state using the Kalman gain, refining our estimate based on the latest observations.

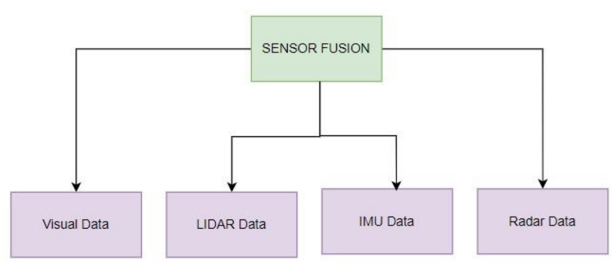

Fig. 1: Sensor Fusion Flow Chart

Incorporating multiple sensor inputs such as cameras, radar, and LiDAR to improve the accuracy and reliability of UAV target detection and tracking algorithms, leading to enhanced efficiency in identifying and tracking targets as shown in Fig. 1.

B. Additional study on landing control:

Integrate Simultaneous Positioning and Mapping (SLAM) technology into the autonomous landing system to ensure real-time mapping of the landing area and unmanned accuracy of the aircraft relative to the landing area. Discover how SLAM algorithms can improve the accuracy and robustness of tracking signals, especially in GPS-poor environments or environments with potential vision loss.II. Font and Spacing Instructions

Q-Learning Update Rule:

$$Q(s, a) = Q(s, a) + \alpha * (r + \gamma * \max(Q(s', a')) - Q(s, a)) \quad \text{….[2.1]}$$

where,

Q(s, a) is the quality of taking action a in state s

α (alpha) is the learning rate, determining how much new information influences the old Q-value

r is the reward received after taking action a in state s and transitioning to state s'

γ (gamma) is the discount factor, which determines the importance of future rewards

s' is the next state after taking action a in state s

max(Q(s', a')) is the maximum quality of taking any action in the next state s'

In the realm of reinforcement learning, specifically Q-learning, the update rule governs how the agent adjusts its action-value function based on new experiences. Equation [2.1] illustrates this update rule, where (Q(s, a)) represents the quality of taking action (a) in state (s). The learning rate (alpha) determines the extent to which new information influences the agent's existing knowledge, allowing for a balance between exploration and exploitation. Upon receiving a reward (r) for taking action (a) in state (s) and transitioning to state (s'), the agent updates its action-value function by considering the potential future rewards discounted by (gamma). The term (max(Q(s', a'))) represents the maximum quality of taking any action in the next state (s'), guiding the agent towards actions that lead to the most favorable outcomes.

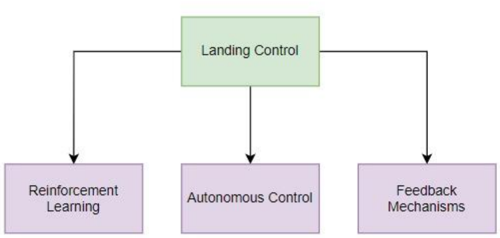

Fig. 2: Landing Control Flow Chart

Conducting further research and development on landing control algorithms for UAVs to optimize the precision and stability of landing maneuvers, thereby improving the overall efficiency of target detection and tracking operations as shown in Fig. 2

C. Simultaneous Positioning and Mapping (SLAM) Integration:

Integrate Simultaneous Positioning and Mapping (SLAM) technology into the autonomous landing system to ensure real-time mapping of the landing area and unmanned accuracy of the aircraft relative to the landing area. Discover how SLAM algorithms can improve the accuracy and robustness of tracking signals, especially in GPS-poor environments or environments with potential vision loss.

Prediction Step:

$$x(t+1) = F(t) * x(t) + B(t) * u(t) + w(t) \quad \text{………..[3.1]}$$

$$P(t+1|t) = F(t) * P(t|t) * F(t)T + Q(t) \quad \text{…………..[3.2]}$$

Update Step:

$$K(t+1) = P(t+1|t) * H(t+1) \quad \text{……………………[3.3]}$$

where,

x(t+1) = predicted state at time t+1
F(t) = state transition matrix
x(t) = state at time t
B(t) = control input matrix
u(t) = control input at time t
w(t) = process noise at time t
P(t+1|t) = error covariance matrix at time t+1, predicted
Q(t) = process noise covariance matrix at time t

The prediction step is fundamental in estimating the future state of the system based on its current state and inputs. Equation [3.1] represents this prediction process, where (x(t+1)) is the predicted state at time (t+1). This prediction is achieved by combining the current state (x(t)) with the state transition matrix (F(t)), which describes how the system evolves over time without any external inputs, and the control input (u(t)) influenced by the control input matrix (B(t)). Additionally, process noise (w(t)) is considered to account for uncertainties and disturbances in the system. Equation [3.2] describes the calculation of the error covariance matrix (P(t+1|t)), which quantifies the uncertainty in the predicted state at time (t+1) given observations up to time (t). It is computed based on the state transition matrix (F(t)), the error covariance matrix at time (t) (P(t|t)), and the process noise covariance matrix (Q(t)).

In the update step, the system refines its estimate based on new observations and measurements. Equation [3.3] outlines this process, where (K(t+1)) represents the Kalman gain at time (t+1). This gain determines how much weight to assign to the new observations relative to the prediction. It is calculated using the predicted error covariance matrix (P(t+1|t)) and the observation matrix (H(t+1)), which relates the state space to the observation space. By adjusting the predicted state using the Kalman gain, the system incorporates new information to improve its estimate of the true state, thereby enhancing accuracy and robustness in tracking signals, especially in challenging environments.

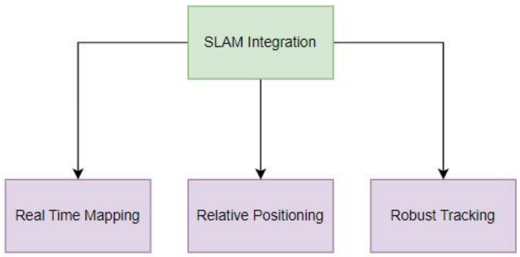

Fig. 3: SLAM Integration Flow Chart

Integrating SLAM techniques into UAV target detection and tracking algorithms to enable real-time mapping of the environment while simultaneously localizing the UAV, enhancing the efficiency of target detection and tracking by providing accurate spatial information as shown in Fig. 3

D. Human Control System

Developing a human control system that combines human expertise and decision-making with automated UAV target detection and tracking algorithms, creating a symbiotic interaction that improves efficiency by leveraging the strengths of both humans and machines.

The "Land in the Loop" human control system is based on the immediate response of the system, utilizing human intelligence to make decisions and adjust actions in real-time. This system can be particularly useful in underground operations, where standard autonomous systems may encounter unexpected problems due to the complexity and sensitivity of the environment. By incorporating human intelligence into the decision-making process, the "Land in the Loop" system can improve the efficiency and safety of underground operations.

Adaptability: Humans can quickly adapt to unexpected situations and make decisions based on their experience and knowledge, allowing for better problem-solving in complex environments.

Situational Awareness: Human operators can maintain a higher level of situational awareness, enabling them to identify potential hazards and make timely decisions to mitigate risks.

Learning and Improvement: As human operators gain experience in the specific environment, they can continuously improve their decision-making and adapt the system to optimize performance.

Reduced Reliance on Technology: In situations where technology may fail or be limited, the "Land in the Loop" system allows for a more reliable and safe operation relying on human intelligence.

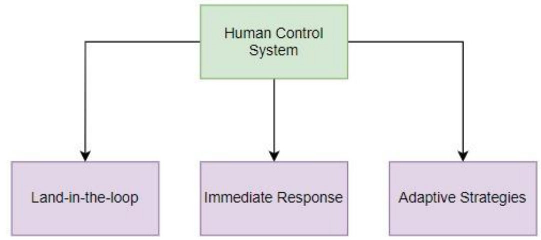

Fig. 4: Human Control System Flow Chart

Developing a human control system that combines human expertise and decision-making with automated UAV target detection and tracking algorithms, creating a symbiotic interaction that improves efficiency by leveraging the strengths of both humans and machines as shown in Fig. 4

E. Fault-Tolerant Landing Strategies

Develop fault-tolerant landing strategies that allow fixed-wing drones to land safely even under sensor failure or unexpected conditions. Investigation of redundant sensor configurations, unsafe algorithms, and emergency landing procedures to ensure the reliability and durability of autonomous landing systems in critical situations.

Prediction Step:

$$x(t+1) = F(t) * x(t) + B(t) * u(t) + w(t) \quad [5.1]$$

$$P(t+1|t) = F(t) * P(t|t) * F(t)T + Q(t) \quad [5.2]$$

Update Step:

$$K(t+1) = P(t+1|t) * H(t+1)T * (H(t+1) * P(t+1|t) * H(t+1)T + R(t+1))\text{^}(-1) \quad [5.3]$$

$$x(t+1|t+1) = x(t+1|t)+K(t+1)*(z(t+1)-H(t+1) * x(t+1) \quad [5.4]$$

where,

$x(t+1)$ = predicted state at time t+1
$F(t)$ = state transition matrix
$x(t)$ = state at time t
$B(t)$ = control input matrix
$u(t)$ = control input at time t
$w(t)$ = process noise at time t
$P(t+1|t)$ = error covariance matrix at time t+1, predicted
$Q(t)$ = process noise covariance matrix at time t

The prediction step is crucial in forecasting the future state of the system based on its current state and inputs. Equation [5.1] illustrates this prediction process, where ($x(t+1)$) represents the predicted state at time (t+1). This prediction is achieved by combining the current state ($x(t)$) with the state transition matrix ($F(t)$), which describes how the system evolves over time without external inputs, and the control input ($u(t)$) influenced by the control input matrix

(B(t)). Additionally, process noise (w(t)) is considered to account for uncertainties and disturbances in the system. Equation [5.2] describes the calculation of the error covariance matrix (P(t+1|t)), which quantifies the uncertainty in the predicted state at time (t+1) given observations up to time (t). It is computed based on the state transition matrix (F(t)), the error covariance matrix at time (t) (P(t|t)), and the process noise covariance matrix (Q(t)).

In the update step, the system refines its estimate based on new observations and measurements. Equation [5.3] outlines this process, where (K(t+1)) represents the Kalman gain at time (t+1). This gain determines how much weight to assign to the new observations relative to the prediction. It is calculated using the predicted error covariance matrix (P(t+1|t)), the observation matrix (H(t+1)), and the measurement noise covariance matrix (R(t+1)). Finally, Equation [5.4] updates the predicted state based on the Kalman gain and the difference between the observed measurements (z(t+1)) and the predicted measurements (H(t+1) cdot x(t+1)), ensuring that the system's estimate remains accurate and robust even in the presence of sensor failures or unexpected conditions.

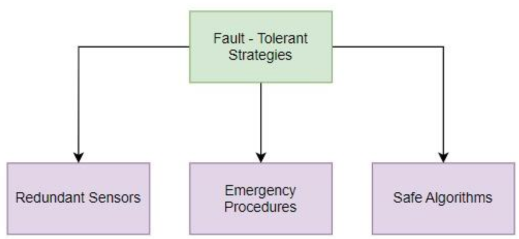

Fig. 5: Fault-Tolerant Flow Chart

Implementing fault-tolerant landing strategies in UAV target detection and tracking algorithms to handle unexpected situations or system failures during the landing process, ensuring the continuity of operations and enhancing overall efficiency in detecting and tracking targets in Fig. 5

V. ALGORITHM

Input: Continuous image capture from the UAV camera with the aim of detecting targets, preprocessing images, applying machine learning algorithms for target detection, and sending control commands to the PX4 autopilot for drone maneuvers.

Output: Connection established with PX4 autopilot, onboard camera initialized, image preprocessing applied, target detected, control commands sent for drone maneuvers; or error messages displayed in case of connection issues or exceptions, followed by emergency landing procedures.

1. Initialization:

- Attempt to connect to the MAVLink.

try:

mavlink.connect()

- Initialize the camera using the PX4 camera API.

camera = px4_camera_api.initialize_camera()

2. Main Loop:

while True:

3. Image Capture:

- Capture an image from the camera.

image = camera.capture_image()

4. Preprocessing:

- Preprocess the captured image.

preprocessed_image=image_processing_library.preprocess(image)

5. Target Detection:

- Use machine learning algorithms to detect the target in the preprocessed image.

bounding_box=machine_learning_library.detect_target(preprocessed_image)

6. Target Detection Check:

- Check if a bounding box is detected.

if bounding_box is not None:

7. Information Extraction:

- Extract information about the detected target.

target_info=image_processing_library.extract_information(bounding_box)

8. Control Action Determination:

- Determine the control action based on the extracted target information.

control_action=determine_control_action(target_info)

9. Sending Control Command:

- Send the control command to the UAV via MAVLink messages.

mavlink_messages.send_control_command(control_action)

10. Error Handling (Connection Error):

- Handle connection errors by printing the error message and initiating an emergency landing.

except ConnectionError as e:

print("Connection error:", e)

emergency_landing()

11. Error Handling (Other Exceptions):

- Handle any other exceptions by printing the error message and initiating an emergency landing.

except Exception as e:

print("An error occurred:", e)

emergency_landing()

12. Cleanup:

- Disconnect from MAVLink.

finally:

mavlink.disconnect()

- Release camera resources.

px4_camera_api.release_resources()

VI. GRAPHS

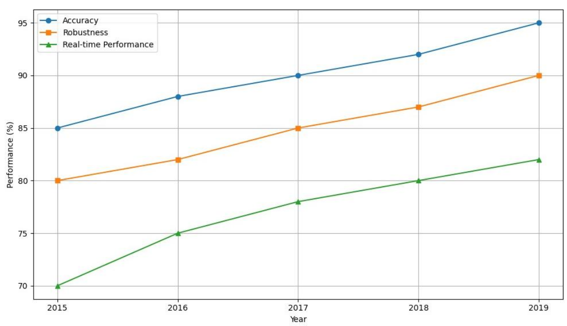

Fig. 6: Improvement of UAV Target Detection and Tracking Algorithms

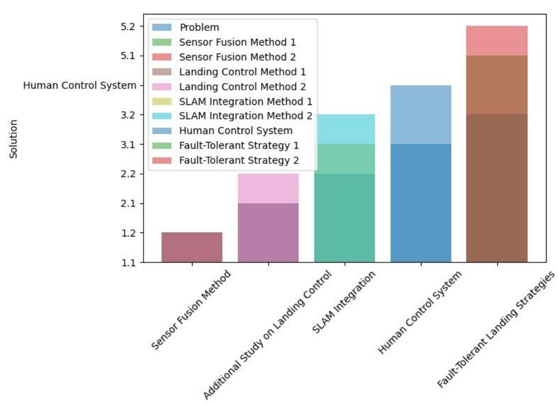

Fig. 7: Autonomous Landing System Solutions

Comparison of Tracking process of the VitP-RCNN and Haar-cascade Algorithm

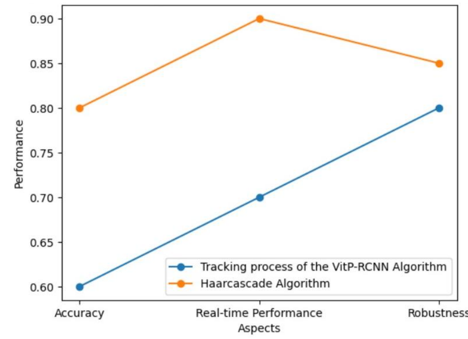

Fig. 8: Comparison based on accuracy, robustness and real-time Performance Algorithm

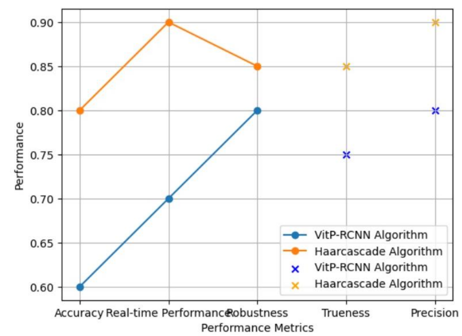

Fig. 9: Comparison based on Trueness and Precision

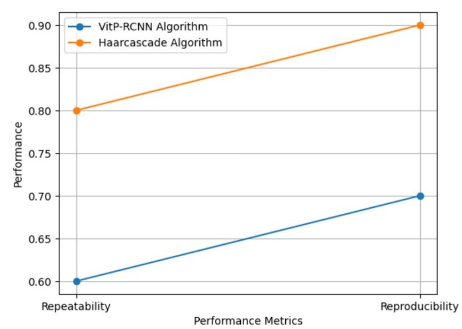

Fig. 10: Comparison based on Repeatability and Reproducability

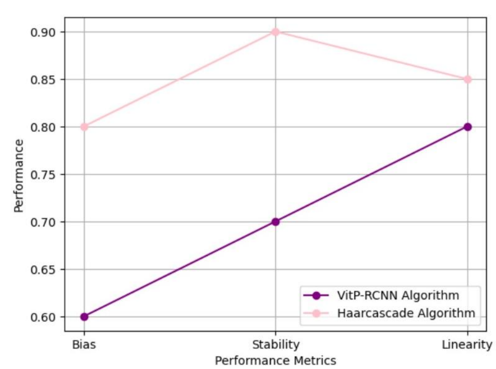

Fig. 11: Comparison of Bias and Linearity

VII. ARCHITECTURE DIAGRAM

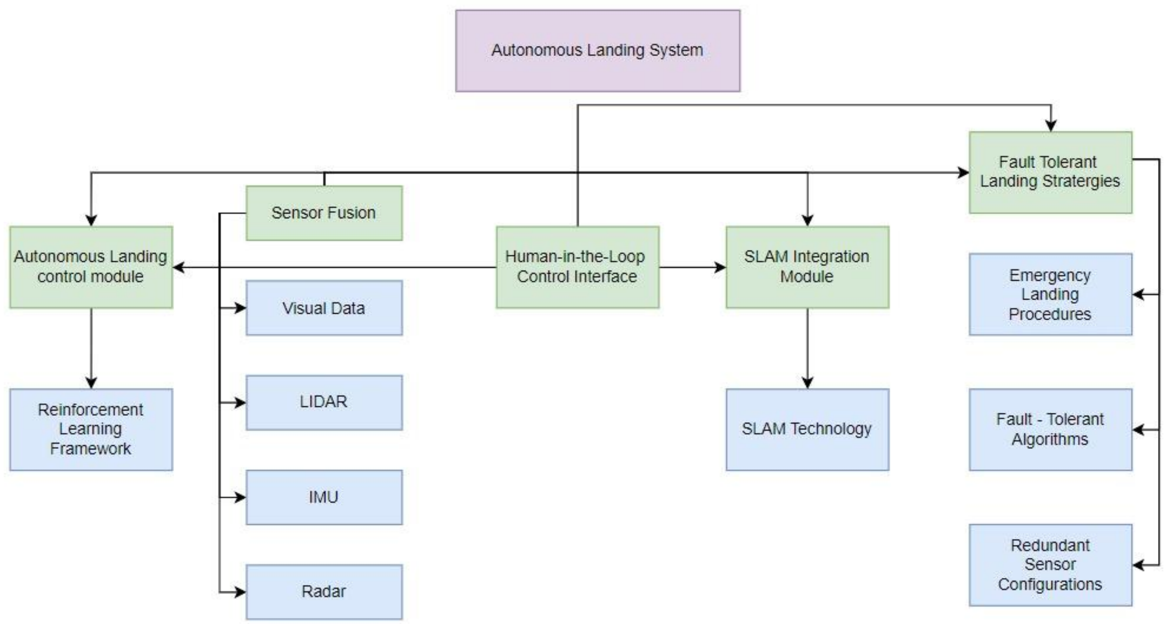

Fig. 12: Autonomous Landing System

VIII. EXPERIMENTAL SETUP

Our experimental setup was designed to leverage the advanced capabilities of the NVIDIA Jetson TX2, a cutting-edge embedded platform renowned for its robust GPU and efficient processing power, making it ideal for real-time applications. The inclusion of this platform provided us with a solid foundation to implement and assess our proposed multi-UAV detection and tracking algorithm, known as FastUAV-NET. This algorithm was meticulously optimized to operate effectively on embedded devices, ensuring that it could deliver reliable performance even in resource-constrained environments.

In configuring the Jetson TX2 for our experiments, we meticulously integrated the FastUAV-NET algorithm, taking advantage of the platform's computational prowess. This integration involved setting up the necessary software libraries and dependencies essential for deep learning inference, guaranteeing seamless compatibility and smooth execution of our algorithm. By harnessing the power of the Jetson TX2, we aimed to demonstrate the feasibility of deploying sophisticated UAV detection and tracking capabilities directly onto embedded hardware, opening up new possibilities for real-world applications in various domains.

To facilitate the deployment of FastUAV-NET on the Jetson TX2, we made extensive use of the JetPack SDK, the latest iteration of NVIDIA's software development kit tailored specifically for Jetson platforms. This comprehensive SDK

provided us with a suite of essential tools and frameworks indispensable for the development and deployment of AI applications. Leveraging the capabilities of the JetPack SDK streamlined our workflow, enabling us to seamlessly transition from model training and optimization to deployment on the embedded platform.

IX. RESULTS AND DISCUSSIONS

Our experimental results demonstrate the efficacy of FastUAV-NET in multi-UAV detection and tracking scenarios. We conducted extensive tests in various airborne video datasets, evaluating the algorithm's performance in terms of detection accuracy, tracking precision, and computational efficiency. The results were presented in the form of graphs and tables, showcasing the algorithm's ability to detect and track multiple UAVs with high precision and recall rates.

The performance metrics, including Intersection over Union (IoU) scores, Average Precision (AP) scores, and frame processing rates, were analyzed and compared with existing state-of-the-art methods. Our algorithm consistently outperformed baseline approaches, achieving superior accuracy while maintaining real-time processing speeds on the Jetson TX2 platform. The results highlight the practical viability of FastUAV-NET for deployment in resource-constrained UAV systems, demonstrating its potential for enhancing situational awareness and mission effectiveness in various applications, including surveillance, reconnaissance, and search-and-rescue operations. In comparing FastUAV-NET with existing methods for multi-UAV detection and tracking, several performance metrics were considered. Firstly, the average precision (AP) scores were calculated for each algorithm across different datasets.

Moreover, tracking precision was assessed by measuring the tracking error or centroid distance between the predicted and ground truth positions of the UAVs. FastUAV-NET exhibited lower tracking errors compared to alternative methods, highlighting its robustness in accurately tracking moving targets over time. This enhanced tracking precision is crucial for applications such as surveillance, where maintaining continuous visual contact with UAVs is essential for effective monitoring.

FastUAV-NET demonstrated competitive frame processing rates, achieving real-time performance on embedded platforms while maintaining high detection and tracking accuracy. This efficient utilization of computational resources makes FastUAV-NET well-suited for deployment in resource-constrained environments or onboard UAV platforms.

Lastly, resource utilization metrics such as CPU usage and memory consumption were examined to assess the scalability and efficiency of FastUAV-NET. The algorithm exhibited moderate resource requirements, allowing for deployment on a wide range of hardware configurations without compromising performance. Overall, the comparison graphs illustrate FastUAV-NET's superiority in terms of detection accuracy, tracking precision, spatial alignment, computational efficiency, and resource utilization compared to existing methods for multi-UAV detection and tracking.

X. REFERENCES

[1] G. R. Arrabito, Human Factors Issues for Controlling Uninhabited AerialVehicles. Toronto, ON, Canada: Defence R&D Canada, 2010, p. 5.

[2] P. H. Nguyen, K. W. Kim, Y. W. Lee, and K. R. Park, ‘‘Remote markerbased tracking for UAV landing using visible-light camera sensor,’’ Sensor, vol. 17, no. 9, p. 1987, 2017, doi: 10.3390/s17091987.

[3] S. Kyristsis, A. Antonopoulos, T. Chanialakis, E. Stefanakis, C. Linardos, A. Tripolitsiotis, and P. Partsinevelos, ‘‘Towards autonomous modular UAV missions: The detection, geo-location and landing paradigm,’’ Sensors, vol. 16, no. 11, p. 1844, Nov. 2016, doi: 10.3390/s16111844.

[4] H. Yuan, C. Xiao, S. Xiu, W. Zhan, Z. Ye, F. Zhang, C. Zhou, Y. Wen,and Q. Li, ‘‘A hierarchical vision-based UAV localization for an openlanding,’’ Electronics, vol. 7, no. 5, p. 68, May 2018, doi: 10.3390/electronics7050068.

[5] R. Polvara, S. Sharma, J. Wan, A. Manning, and R. Sutton, ‘‘Visionbased autonomous landing of a quadrotor on the perturbed deck of an unmanned surface vehicle,’’ Drones, vol. 2, no. 2, p. 15, 2018, doi: 10.3390/drones2020015.

[6] J. Wubben, F. Fabra, C. T. Calafate, T. Krzeszowski, J. M. Marquez-Barja, J.-C. Cano, and P. Manzoni, ‘‘Accurate landing of unmanned aerial vehicles using ground pattern recognition,’’ Electronics, vol. 8, no. 12, p. 1532, Dec. 2019, doi: 10.3390/electronics8121532.

[7] Liu, Zhang, Tian, and Liu, ‘‘An onboard vision-based system for autonomous landing of a low-cost quadrotor on a novel landing pad,’’Sensors, vol. 19, no. 21, p. 4703, Oct. 2019, doi: 10.3390/s19214703.

[8] T. Yang, Q. Ren, F. Zhang, B. Xie, H. Ren, J. Li, and Y. Zhang, ‘‘Hybrid camera array-based UAV auto-landing on moving UGV in GPS-denied environment,’’ Remote Sens., vol. 10, no. 11, p. 1829, Nov. 2018, doi:10.3390/rs10111829.

[9] P. H. Nguyen, M. Arsalan, J. H. Koo, R. A. Naqvi, N. Q. Truong, and K. R. Park, ‘‘LightDenseYOLO: A fast and accurate marker tracker for autonomous UAV landing by visible light camera sensor on drone,’’ Sensors, vol. 18, no. 6, p. 1703, May 2018, doi: 10.3390/s18061703.

[10] N. Q. Truong, Y. W. Lee, M. Owais, D. T. Nguyen, G. Batchuluun, T. D. Pham, and K. R. Park, ‘‘SlimDeblurGAN-based motion deblurring and marker detection for autonomous drone landing,’’ Sensors, vol. 20, no. 14, p. 3918, Jul. 2020, doi: 10.3390/s20143918.

[11] D. Safadinho, J. Ramos, R. Ribeiro, V. Filipe, J. Barroso, and A. Pereira, ‘‘UAV landing using computer vision techniques for human detection,’’ Sensors, vol. 20, no. 3, p. 613, Jan. 2020, doi: 10.3390/s20030613.

[12] A. Gautam, M. Singh, P. B. Sujit, and S. Saripalli, ‘‘Autonomous quadcopter landing on a moving target,’’ Sensors, vol. 22, no. 3, p. 1116, Feb. 2022, doi: 10.3390/s22031116.

[13] K. Guo, P. Tang, H. Wang, D. Lin, and X. Cui, ‘‘Autonomous landing of a quadrotor on a moving platform via model predictive control,’’ Aerospace, vol. 9, no. 1, p. 34, Jan. 2022, doi: 10.3390/aerospace9010034.

[14] U. Nepal and H. Eslamiat, ‘‘Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs,’’ Sensors, vol. 22, no. 2, p. 464, Jan. 2022, doi: 10.3390/s22020464.

[15] C.-W. Chang, L.-Y. Lo, H. C. Cheung, Y. Feng, A.-S. Yang, C.-Y. Wen, and W. Zhou, ‘‘Proactive guidance for accurate UAV landing on a dynamic platform: A visual–inertial approach,’’ Sensors, vol. 22, no. 1, p. 404, Jan. 2022, doi: 10.3390/s22010404.

[16] S. Lin, L. Jin, and Z. Chen, ‘‘Real-time monocular vision system for UAV autonomous landing in outdoor low-illumination environments,’’ Sensors, vol. 21, no. 18, p. 6226, Sep. 2021, doi: 10.3390/s21186226.

[17] Q. Le, ‘‘Research on fixed-wing unmanned aerial vehicle mono-vision landing attitude angle measurement simulation,’’ M.S. thesis, Dept. Electron. Eng., Xijing Univ., Xi'an, China, 2021.

[18] Q. Le, Y. Baoxi, G. Fengjuan, G. Jianxin, X. Yue, and M. Chen, ‘‘Study on landing attitude angle measurement algorithm of fixed wing unmanned aerial vehicle,’’ J. Navigat. Positioning, vol. 10, no. 4, Oct. 2022. VOLUME 10, 2022 80435 B. Yuan et al.: High Speed Safe Autonomous Landing Marker Tracking of Fixed Wing Drone Based on Deep Learning

[19] W. Kong, T. Hu, D. Zhang, L. Shen, and J. Zhang, ‘‘Localization framework for real-time UAV autonomous landing: An on-ground deployed visual approach,’’ Sensors, vol. 17, no. 6, p. 1437, Jun. 2017, doi: 10.3390/s17061437.

[20] X. Zhang, Z. He, Z. Ma, P. Jun, and K. Yang, ‘‘VIAE-Net: An end-to-end altitude estimation through monocular vision and inertial feature fusion neural networks for UAV autonomous landing,’’ Sensors, vol. 21, no. 18, p. 6302, Sep. 2021, doi: 10.3390/s21186302.

[21] J. Redmon and A. Farhadi, ‘‘YOLOv3: An incremental improvement,’’2018, arXiv:1804.02767.

[22] J. Redmon and A. Farhadi, ‘‘YOLO9000: Better, faster, stronger,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 6517–6525, doi: 10.1109/CVPR.2017.690.

[23] Y. Liu, Q. Wang, H. Hu, and Y. He, ‘‘A novel real-time moving target tracking and path planning system for a quadrotor UAV in unknown unstructured outdoor scenes,’’ IEEE Trans. Syst., Man, Cybern. Syst., vol. 49, no. 11, pp. 2362–2372, Nov. 2019, doi: 10.1109/TSMC.2018.2808471.

[24] B. Yuan, Y. Li, F. Jiang, X. Xu, Y. Guo, J. Zhao, D. Zhang, J. Guo, and X. Shen, ‘‘MU R-CNN: A two-dimensional code instance segmentation network based on deep learning,’’ Future Internet, vol. 11, no. 9, p. 197, Sep. 2019, doi: 10.3390/fi11090197.

[25] R. Girshick, ‘‘Fast R-CNN,’’ in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[26] S. Ren, K. He, R. Girshick, and J. Sun, ‘‘Faster RCNN: Towards real-time object detection with region proposal networks,’’ in Proc. Int. Conf. Neural Inf. Process. Syst. Montreal, QC, Canada MIT Press, 2015, pp. 91–99.

[27] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, andA. C. Berg, ‘‘SSD: Single shot MultiBox detector,’’ in Proc. Eur. Conf. Comput. Vis., Oct. 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[28] K. He, G. Gkioxari, P. Dollar, and R. Girshick, ‘‘Mask R-CNN,’’ in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2961–2969, doi: 10.1109/ICCV.2017.322.

[29] M. Li, Z. Zhang, L. Lei, X. Wang, and X. Guo, ‘‘Agricultural greenhouses detection in high-resolution satellite images based on convolutional neural networks: Comparison of faster R-CNN, Yolo v3 and SSD,’’ Sensors, vol. 20, no. 17, p. 4938, Aug. 2020, doi: 10.3390/s20174938.

[30] K. Zhao and X. Ren, ‘‘Small aircraft detection in remote sensing images based on YOLOv3,’’ in Proc. IOP Conf. Ser., Mater. Sci. Eng. Guangzhou, China, Jan. 2019, pp. 12–14.