SURVEY ARTICLE

Recent Advancements in Deep Learning Applications and Methods for Autonomous Navigation: A Comprehensive Review

Arman Asgharpoor Golroudbari[1,2,3*] | **Mohammad Hossein Sabour**[1,4*]

[1]Department of Aerospace, Faculty of New Sciences & Technologies, University of Tehran, Tehran, Iran

[2]Students' scientific research center, Tehran University of Medical Sciences, Tehran, Iran

[3]Fuzzy Logic Lab Interest Group, Universal Scientific Education and Research Network, Tehran,Iran

[4]Department of Mechanical, Industrial and Aerospace Engineering, Concordia University, Montreal, QC, Canada

Correspondence
M.H. Sabour, Department of Aerospace, Faculty of New Sciences & Technologies, University of Tehran, Tehran, Iran
Email: sabourmh@ut.ac.ir

Abstract:
This review article presents recent advancements in deep learning methodologies and applications for autonomous navigation. It analyzes state-of-the-art deep learning frameworks used in tasks like signal processing, attitude estimation, obstacle detection, scene perception, and path planning. The implementation and testing methodologies of these approaches are critically evaluated, highlighting their strengths, limitations, and areas for further development. The review emphasizes the interdisciplinary nature of autonomous navigation and addresses challenges posed by dynamic and complex environments, uncertainty, and obstacles. With a particular focus on mobile robots, self-driving cars, unmanned aerial vehicles, and space vehicles to underscore the importance of navigation in these domains. By synthesizing findings from multiple studies, the review aims to be a valuable resource for researchers and practitioners, contributing to the advancement of novel approaches. Key aspects covered include the classification of deep learning applications, recent advancements in methods, general applications in the field, innovations, challenges, and limitations associated with learning-based navigation systems. This review also explores current research trends and future directions in the field. This extensive overview, initiated in 2020, provides a valuable resource for researchers of all levels, from seasoned experts to newcomers. Its main purpose is to streamline the process of identifying, evaluating, and interpreting relevant research, ultimately contributing to the progress and development of autonomous navigation technologies.

KEYWORDS
Deep Learning, Navigation, Inertial Sensors, Intelligent Filter, Sensor Fusion, Long-Short Term Memory, Convolutional Neural Network

[*] Equally contributing authors.

1 | INTRODUCTION

Autonomous navigation is a critical component of robotics which has application in various domains, such as medical Ramadhan et al. (2021), industrial Yang et al. (2020), aerial Aslan et al. (2022), space Zhou et al. (2022a), and agricultural Byun et al. (2022). It can expand robots versatility and functionality by enabling them to efficiently and securely move through dynamic environments without human intervention.

To enhance the performance of autonomous navigation systems, researchers have conducted various research and made constant progress in the past decades Geiger et al. (2012); Delmerico et al. (2019); Sun et al. (2018); Furfaro et al. (2018); Hu et al. (2018). As a result the expansive and ever-evolving nature of the literature surrounding autonomous navigation increased the need to conduct regular surveys in order to remain abreast of the latest advancements.

This paper provides a comprehensive overview of the current state-of-the-art in end-to-end deep learning methods for autonomous navigation, with a focus on catering to both experienced researchers and novices in the field. The key components of this review are as follows:

- A comprehensive collection of recent advancements in learning-based methods that can be utilized for autonomous navigation.
- A detailed classification of deep learning applications for autonomous navigation.
- An overview of recent innovations in autonomous navigation systems, highlighting the most promising approaches and their potential impact on the field.
- Coverage of general applications in the field, providing a comprehensive overview of the state-of-the-art in autonomous navigation systems.
- A discussion of the challenges and limitations of deep learning-based perception systems for autonomous navigation, including the potential for bias in training data, difficulties in complex and dynamic environments, and high computational requirements.
- A review of the current research trends and future directions in deep learning-based perception systems for autonomous navigation.

The focus of this review is on research conducted since 2020, with a brief historical overview provided for context to give a clear overview of the field. This paper covers a range of topics, including signal processing, perception, localization, mapping, and planning which can be utilized toward autonomous navigation. We also highlight advancements in each area and identifies key challenges and opportunities for future research. This paper aims to aid in the identification, evaluation, and interpretation of the available research relevant to a topic of interest.

Best of our knowledge, there are no surveys covering all aspects of learning-based autonomous navigation. Most of the previous studies focused on a specific part of the navigation process Carrio et al. (2017); Wong et al. (2017); Alom et al. (2019); Ejaz et al. (2019); Emmert-Streib et al. (2020); Grigorescu et al. (2020); Li et al. (2020); Ni et al. (2020); Chen et al. (2020a); Li et al. (2021b); Roy and Chowdhury (2021); El-Taher et al. (2021); Wang et al. (2022c); Silvestrini and Lavagna (2022); Qin et al. (2022); AlMahamid and Grolinger (2022); Badrloo et al. (2022); Wen and Jo (2022); Song et al. (2022a); Tang et al. (2022). The lack of such a comprehensive review to cover multiple aspects of utilizing deep learning toward the navigation system has motivated us to conduct this review. We believe that this review would be valuable for researchers and practitioners in the field, as well as for those interested in the latest advancements in deep learning for navigation.

A terminology is included in below to provide clear view of the technical vocabulary utilized throughout the article.

- **Autonomous Navigation**: Navigate and move through an environment without human intervention.
- **Perception**: Sense and understand the surroundings through the use of sensors.
- **Localization**: Determining the position within a known map or environment.
- **Mapping**: Creating a map of an environment using sensor data and other inputs.
- **Simultaneous Localization and Mapping (SLAM)**: Creating a map of an unknown environment while simultaneously localizing the robot or autonomous system within that environment.
- **Control**: Regulating the motion to follow a desired trajectory and achieve a specific task.

- **Obstacle Avoidance**: Navigate around obstacles in the path.
- **Collision Avoidance**: Using sensors and algorithms to detect potential collisions and take action to avoid them.
- **Path Planning**: Determining and constructing the path from a starting point to the end point.
- **Motion Planning**: The process to determine the set of actions to be executed in order to follow the planned path.
- **Sensor Fusion**: Combining data from multiple sensors to obtain a more accurate and comprehensive understanding of the environment.
- **Odometry**: Using sensory data to estimate the position and orientation by analyzing the movement over time.
- **Dead Reckoning**: Estimates the current position by using its previous position and velocity without using external sensor measurements.

Navigation refers to the process in which the movement is monitored and controlled, such as robots, autonomous systems, and unmanned aerial vehicles Britting (2010). This process requires perceiving the surroundings, planning the path, executing, and adapting as needed, all while avoiding collisions which is necessary to ensure safe, efficient, and accurate movement from the starting point to the endpoint without human intervention.

Recent advancements in technologies and algorithms have made navigation more reliable, effective, and efficient, broadening its applications in transportation, search and rescue Niroui et al. (2019), drones Lee et al. (2021), and space missions Gaudet et al. (2020).

Autonomous systems can be broadly categorized into two main topics: 1) reactive, 2) deliberative Muscettola et al. (2002). **Reactive systems**, known as behavior-based systems, designed to respond quickly to changes in the environment using predefined rules. It is used where the environment is relatively stable, and the system has a specific task, such as pick and place. **Deliberative systems** are designed to plan and reason about their actions before executing them and can handle more complex tasks and scenarios. They are commonly used in transportation, where the environment is complex, and the system must navigate efficiently and safely.

The deliberative systems can be further classified into two categories Haith and Krakauer (2013): 1) model-based systems, 2) model-free systems. **Model-based** systems use a computational or mathematical model to predict and analyze the outcomes for various decisions, such as using dynamic programming or graph search algorithms Wolek and Woolsey (2017). On the other hand, **model-free** systems do not rely on a model of the environment to make a decision Kontoudis and Vamvoudakis (2019). They are also known as model-agonist systems. It is noticeable that the choice of the system, whether reactive or deliberative and model-based or model-free, depends on the specific requirements and constraints.

Artificial Intelligence has been applied to navigation in various aspects, as evidenced by numerous studies and surveys (refer to Tables 1 and 2). While it holds great promise in enhancing navigation systems, there is a need to tackle the challenges and ethical considerations that come with its usage. Its ability to learn complex relationships in data, make them suitable for sophisticated tasks, such as signal filtering, image segmentation, object tracking, and sensor fusion. However, deep learning-based navigation systems must overcome challenges such as limited data, reliability, ethical concerns, such as privacy and bias. Techniques including transfer learning, multi-modal fusion, model uncertainty estimation, and safety-critical architectures can address these challenges. Additionally, differential privacy and fairness-aware machine learning techniques can address privacy and bias concerns, respectively.

Various surveys and reviews focused on the utilizing deep learning towards navigation domains, including urban navigationEl-Taher et al. (2021), vision-based navigation Wang et al. (2022c); O'Mahony et al. (2018), reinforcement learning AlMahamid and Grolinger (2022); Zhu and Zhang (2021), obstacle detection Badrloo et al. (2022), and spacecraft navigation Song et al. (2022a); Turan et al. (2022). Despite all these surveys, there is a lack of comprehensive review to provide a general overview of employing deep learning for navigation. This survey aims to fill this gap by presenting a general view of the applications of deep learning in navigation.

The paper is structured as follows: Section 2 provides an overview of deep learning and its methods. Section 3 presents an overview of navigation, autonomy, and autonomous navigation. Section 4 discusses the applications of deep learning in navigation, and delves into the various components of deep learning in autonomous navigation, such as perception, localization, mapping, and planning. Finally, Section 5 concludes the paper and highlights future directions. We also provided a detailed overview of the architecture and methods of deep neural networks in the appendix.

TABLE 1 Recent articles on deep learning for navigation

Title	Application
Hierarchical multi-robot navigation and formation in unknown environments via deep reinforcement learning and distributed optimization Chang et al. (2023)	Multi-robot Navigation
HINNet: Inertial navigation with head-mounted sensors using a neural network Hou and Bergmann (2023)	Inertial Navigation
Multi-sensor integrated navigation/positioning systems using data fusion: From analytics-based to learning-based approaches Zhuang et al. (2023)	Integrated Navigation
Study of convolutional neural network-based semantic segmentation methods on edge intelligence devices for field agricultural robot navigation line extraction Yu et al. (2023)	Visual Navigation
Goal-guided Transformer-enabled Reinforcement Learning for Efficient Autonomous Navigation Huang et al. (2023)	Autonomous Navigation
DeepNAVI: A deep learning based smartphone navigation assistant for people with visual impairments Kuriakose et al. (2023)	Visual Navigation
Monocular vision with deep neural networks for autonomous mobile robots navigation Sleaman et al. (2023)	Visual Navigation
URWalking: Indoor Navigation for Research and Daily Use Ludwig et al. (2023)	Indoor Navigation
A Simple Self-Supervised IMU Denoising Method For Inertial Aided Navigation Yuan and Wang (2023)	Inertial Navigation
Multi-Scale Fully Convolutional Network-Based Semantic Segmentation for Mobile Robot Navigation Dang and Bui (2023)	Visual Navigation
Drone Navigation Using Octrees and Object Recognition for Intelligent Inspections Martinez Acosta (2023)	Visual Navigation
Deep learning-enabled fusion to bridge GPS outages for INS/GPS integrated navigation Liu et al. (2022a)	Inertial Navigation
Deep learning based wireless localization for indoor navigation Ayyalasomayajula et al. (2020)	Indoor Navigation
Efficient and robust LiDAR-based end-to-end navigation Liu et al. (2021)	Terrain modeling
End-to-End Deep Learning Framework for Real-Time Inertial Attitude Estimation using 6DoF IMU Golroudbari and Sabour (2023)	Inertial Navigation

2 | A BRIEF OVERVIEW OF DEEP LEARNING

In this section we provide a brief overview of deep learning for a more detailed discussion, the reader is referred to Goodfellow et al. (2016). We begin by presenting a brief history of deep learning then discuss the different types and applications in the appendix.

Deep learning has its roots in the 1940s and 1950s with the introduction of the concept of artificial neurons by Warren McCulloch and Walter Pitts McCulloch and Pitts (1943) which was later extended by Frank Rosenblatt Rosenblatt (1958) to include a learning mechanism. In 1957, Rosenblatt introduced the perceptron Rosenblatt (1958) as single-layer artificial neural network, used for binary classification. However, the field did not gain significant traction until the late 1980s and early 1990s, when the back-propagation algorithm Rumelhart et al. (1986) was introduced for training ANNs.

In 2012, AlexNet Krizhevsky et al. (2017) achieved impressive results in the ImageNet visual recognition challenge which marked a turning point in deep learning. Since then, deep learning has been applied to various domains, including natural language processing Otter et al. (2020), speech recognition Deng and Platt (2014), self-driving cars

TABLE 2 Recent surveys on navigation or deep learning for navigation

Year	Topic	Highlight
2017	Unmanned Aerial Vehicles	Deep learning for UAV's navigation, motion control, and situational awareness Carrio et al. (2017).
2017	Adaptive Navigation	AI-based planetary rovers autonomous navigation Wong et al. (2017).
2019	Deep Learning Approaches	State-of-the-art deep learning approaches Alom et al. (2019).
2019	Visual Navigation	Reinforcement Learning for visual autonomous navigation Ejaz et al. (2019).
2020	Deep Learning Approaches	Major architectures and applications of deep learning Emmert-Streib et al. (2020).
2020	Autonomous Driving	Deep learning architectures for autonomous driving Grigorescu et al. (2020).
2020	Inertial Sensors	Enhancement of various aspects of inertial sensors (sensor fusion, calibration, and navigation) Li et al. (2020).
2020	Autonomous Driving	Deep learning architectures for autonomous driving Ni et al. (2020).
2020	Perception	Localization and Mapping Chen et al. (2020a)
2021	Convolutional Neural Network	Applications of CNNs Li et al. (2021b).
2021	Indoor Navigation	Machine learning for indoor localization Roy and Chowdhury (2021).
2021	Assistive navigation	Outdoor and urban navigation for people with visual impairments El-Taher et al. (2021).
2022	Visual Navigation	Agricultural robot navigation Wang et al. (2022c).
2022	Spacecraft Navigation	Deep learning for spacecraft dynamics control, guidance and navigation Silvestrini and Lavagna (2022).
2022	Visual Navigation	Unmanned underwater vehicles Qin et al. (2022).
2022	Unmanned Aerial Vehicle	Reinforcement Learning for UAVs AlMahamid and Grolinger (2022).
2022	Autonomous Driving	Visual obstacle detection for unmanned ground vehicles Badrloo et al. (2022)
2022	Autonomous Driving	Deep Learning for Perception Wen and Jo (2022)
2022	Spacecraft Navigation	Autonomous relative navigation for orbital applications Song et al. (2022a)
2022	Learning-based perception	Learning-based monocular approaches in perception and navigation in autonomous systems Tang et al. (2022)

Rao and Frtunikj (2018), healthcare Esteva et al. (2019), computational biology Angermueller et al. (2016), and gaming Justesen et al. (2019). It has continued to evolve with the development of new architectures and methods such as Generative Adversarial Networks (GANs) Aggarwal et al. (2021), and attention mechanisms Vaswani et al. (2017).

The increasing availability of large data sets and powerful computational resources has broadened its applications, and the field's future looks very promising. Some remarkable examples of recent advancements in deep learning include the development of the GPT-4 language model OpenAI (2023), which has shown impressive results in natural language processing tasks, as well as AlphaFold Ruff and Pappu (2021), which can predict the complex process of protein folding. DeepMind's AlphaGo Silver et al. (2016), which achieved groundbreaking results in the ancient Chinese board game, is also a notable example of deep learning's potential.

3 | AUTONOMOUS NAVIGATION

Navigation is the process of determining the position and orientation and finding a suitable path to reach a destination while executing proper actions to achieve that. Navigation systems can be broadly categorized into two main types: 1) **indoor** and 2) **outdoor**. **Indoor navigation** systems usually suffer from weak signals and other external references. On the other hand, **outdoor navigation** systems, despite the availability of external resources, are usually more challenging due to the presence of more obstacles and other factors that can affect the accuracy of the system.

The process of collecting data, perceiving the environment, planning, and actuating is known as the **autonomous navigation cycle** shown in figure 1.

Autonomous navigation systems can be found in a wide range of applications and can be further categorized based on the environment they operate in: **Land**, **Marine**, **Aerial**, **Space**, **Underwater**. Each type of system presents unique challenges and requirements, and specific constraints due to its environment's characteristics which must be taken into account when developing these systems.

Regardless of the environment, the main components of autonomous navigation systems include:

- **Environment sensing and perception** - Collect and process information about the environment.

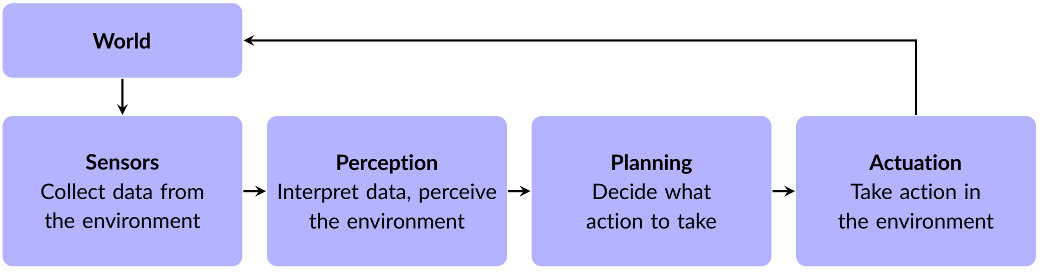

FIGURE 1 The Autonomous Navigation Cycle

- **Mapping** - Create a map of the environment.
- **Localization** - Determine its position and orientation in the environment.
- **Obstacle detection** - Detect and avoid obstacles in the environment.
- **Sensor fusion** - Combine data from multiple sensors to improve the overall performance of the navigation system.
- **Path planning and generation** - Plan a path to reach the destination and generate a trajectory to follow the path.
- **Trajectory optimization** - Adjust the trajectory to improve the efficiency and accuracy of the navigation.
- **Decision making and control** - Make decisions and take actions based on the information collected from the environment.
- **Motion control** - Control the motion of the vehicle to follow the trajectory.

These components work together to ensure the system can be safe and efficient. The subsequent sections will delineate the diverse domains in which deep learning techniques have been employed to attain autonomy in the navigation process.

3.1 | Applications of Deep Learning in Autonomous Navigation

Autonomous navigation encompasses various domains which have seen significant advancements in recent years, due to the integration of deep learning techniques. This subsection explores these applications in various domains from self-driving cars to autonomous drones and spacecraft, highlighting the key challenges and advancements in each field.

We will delve into some of these applications to give a general overview of each domain in the following. A detailed discussion of each application will be presented in the next section.

3.1.1 | Autonomous Drive

Automated driving systems could enable a safe, reliable, and efficient driving experience by combining various sensors and algorithms to control the vehicle without human intervention. Developing autonomous driving systems could reduce emissions, prevent accidents, and improve traffic efficiency by removing human error from the equation Yurtsever et al. (2020).

The development of self-driving cars has gained much attention in recent years as a result of technological advancements made by companies like Tesla, Google, and BMW. They use a combination of various algorithms and technologies to achieve autonomy, Figure 2. Deep learning methods and algorithms are at the forefront of recent progress in the field.

To bring autonomous driving into reality, it is imperative to develop dependable and resilient algorithms that can efficiently process data from sensors such as cameras, RADARs, and LiDARs in real-time, and make the right decision based on the obtained data. In this context, deep learning algorithms have been extensively employed in various aspects of autonomous driving, including signal processing Brossard et al. (2020), object detection Alaba and Ball

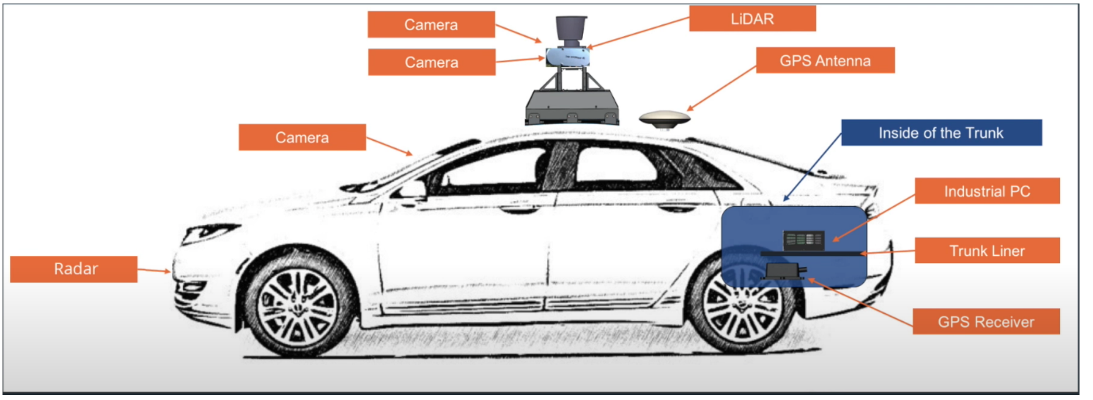

FIGURE 2 Sensors in Autonomous Vehicles Paravarzar and Mohammad (2020)

(2023), lane detection Lee and Liu (2023), and trajectory prediction Zhang et al. (2022c). Recent studies in this field and their key findings presented in Table 3.

The key components of self-driving cars are outlined as below:

1. **Sensors**: Devices that detect and measure physical phenomena, such as cameras, LiDAR, radar, and ultrasonic sensors.
2. **Perception algorithms**: Software that processes sensor data to extract meaningful information about the environment, such as object detection and tracking.
3. **Mapping and localization**: Algorithms that create a map of the environment and estimate the car's position within that map.
4. **Planning and decision-making**: Algorithms that determine the optimal path and actions.
5. **Control systems**: Systems that execute the planned actions and control the car's speed, acceleration, and steering.
6. **Human-machine interface**: Interface that allows passengers to interact with the car and monitor its performance.
7. **High-performance computing**: Systems to process large amounts of data in real-time and make decisions quickly.
8. **Machine learning**: Algorithms to improve their performance over time by learning from data and experience.
9. **Cybersecurity**: Systems to protect the car from cyber attacks which could compromise their safety and security.
10. **Redundancy and fail-safe systems**: Systems which ensure the car can operate safely even in the event of a failure or malfunction.

3.2 | Autonomous Flight

The use of autonomous drones has rapidly grown in recent years and is expected to continue its upward trend Clarke (2014); Merkert and Bushell (2020); Radoglou-Grammatikis et al. (2020); Alzahrani et al. (2020). The ability of these vehicles to operate without human intervention opens up a wide range of applications from delivering packages (Figure 4) to performing complex aerial photography and remote sensing. These vehicles are advancing various industries and make our lives easier and much convenient AlMahamid and Grolinger (2022).

One of the primary goals of autonomous drones is to improve safety by reducing the risk of human error which is especially important in hazardous scenarios such as search and rescue missions, where any missteps could be catastrophic Mohsan et al. (2022).

One of the backbones of autonomous flights is the Artificial Intelligence system that enables them to perceive the environment, make decisions and navigate securly. The AI core is responsible for processing data obtained from sensors to gain an accurate understanding of the surrounding and make decisions about the actions based on these in-

TABLE 3 Recent studies on autonomous driving

Ref.	Objectives	Key Findings
Zhang and Zhong (2023)	Lane estimation using Catmull-Rom curves	Achieved 93.81% accuracy rate on TuSimple dataset
Lyu et al. (2022)	Real time object tracking	Outperformed traditional methods by using Siamese framework and two feature extraction networks
Nie et al. (2022)	Object detection dataset	Proposed the Transnational Image Object Detection datasets (TDND) for nighttime driving
He et al. (2023)	Wheel odometry model for vehicle localization	Developed a generalized model based on Transformers
Gao et al. (2023)	Generate realistic and efficient scenarios	Developed heterogeneous driver model, trained on HighD dataset
Jain et al. (2023)	Facial Emotion Recognition	Achieved accuracy of 99.69% and 99.50% on KDEF and KMU-FED datasets
Wang et al. (2023c)	LiDAR-based 3D object detection	Achieved 81.46% accuracy on KITTI dataset using SC-RCNN network
Ji et al. (2023)	LiDAR-based shape reconstruction	Achieved state-of-the-art performance on KITTI dataset
Plascencia et al. (2023)	Pedestrian detection	Using SegNet for RGB, LiDAR, and RADAR fusing
Endo et al. (2023)	Enhance object detection in complex urban environments using HD maps.	Incorporating HD map information improves object detection accuracy by 37%
Mahaur and Mishra (2023)	Enhance detection of small objects	Improved performance in small-object detection with iS-YOLOv5 model
Acun et al. (2023)	Real-time drivable area detection with improved performance and runtime efficiency	D3NET achieves real-time drivable area detection with improved training and runtime performance
Zhao et al. (2023)	Lifelong framework for data-driven indoor positioning correction	Improves precision of indoor positioning for autonomous driving systems
Zhang et al. (2022a)	Optimize multi-objective urban driving tasks	Lexicographic actor-critic reinforcement learning improves urban scenarios
Jagtap and Kanade (2022)	Improve runtime efficiency of semantic video event classification	Achieves 99% accuracy in identifying car brake lights

formations. These actions include obstacle avoidance, autonomous movement, and autonomous take-off and landing. Deep learning algorithms have been used to develop perception and decision-making systems which aid to perform complex tasks with high accuracy, such as recognize and classify objects in real-time to identify and avoid obstacles Rezwan and Choi (2022).

The SAE's Six levels of autonomy, ranging from "no automation" to "full automation," were originally developed for autonomous ground vehicles but are applicable to any vehicle capable of autonomy, including drones Lee et al. (2021). It could be adapted to flying vehicles domain as follows:

- **Level 1**: Assisted features like GPS guidance and landing zone evaluation.
- **Level 2**: Pilot assistance with specific navigational operations such as "follow me" and "track target" commands.
- **Level 3**: Partial automation allowing autonomous navigation in certain identified environments.
- **Level 4**: Conditional automation where the drone can navigate autonomously in most use cases without human interaction.
- **Level 5**: Full automation, the theoretical ideal of autonomy in all possible use cases, environments, and conditions, with the system responsible for all flying tasks.

Table 4 shows the six levels of flying automation and the tasks that are performed at each level.

Implementing deep learning algorithms for autonomous drones could be a challenging task due to:

- **Limited computational resources**: Deep learning needs of significant computational resources is a challenge for drones with limited processing power and memory.

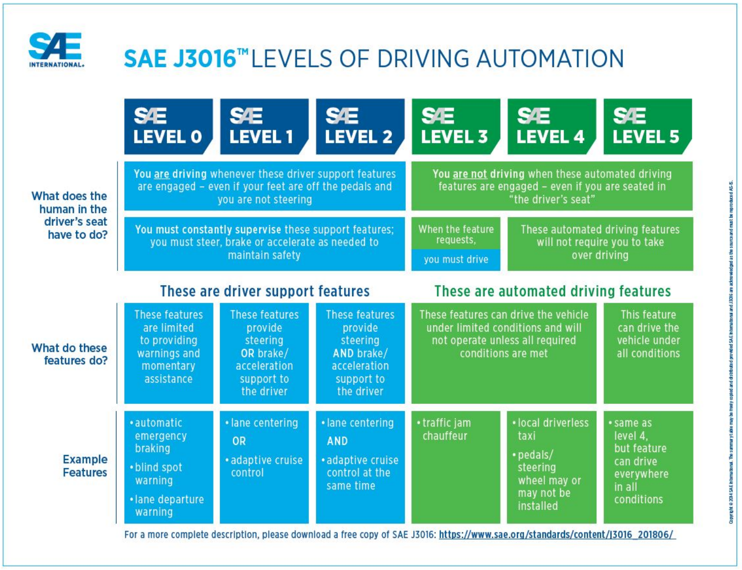

FIGURE 3 Levels of Automation, described by the Society of Automobile Engineers (SAE) Shuttleworth (2019)

FIGURE 4 Autonomous Delivery Drone Amazon (2022)

- **Limited data availability**: Access to low amount of accurate data is another challenge.
- **Real-time processing requirements**: Autonomous drones must process data in real-time, which can be a challenge for learning-based algorithms that may require significant processing time.
- **Environmental variability**: Drones operate in various environments, which can make it difficult for deep learning algorithms to generalize and perform accurately.
- **Safety and reliability**: Autonomous drones must operate safely and reliably, which can be a challenge for deep learning algorithms that may be prone to errors or unexpected behavior.

Addressing these challenges requires careful consideration of the operation application and environment, as well as the development of advanced algorithms and hardware that can meet the unique requirements of autonomous drone navigation.

TABLE 4 Levels of Automation for Flying Vehicles

Level	0	1	2	3	4	5
Obstacle Detection		✓	✓	✓	✓	✓
Adaptive Cruise Control		✓	✓	✓	✓	✓
Spatial Awareness		✓	✓	✓	✓	✓
Object Distinction			✓	✓	✓	✓
Obstacle Avoidance			✓	✓	✓	✓
Autonomous Movement			✓	✓	✓	✓
Environmental Awareness				✓	✓	✓
Non-planar Movement				✓	✓	✓
Autonomous Take off & Landing					✓	✓
Path Generation					✓	✓
Autonomous Flight						✓

3.3 | Autonomous Space Flight

In the past decades AI has made significant contributions to the space industry, especially in the field of Guidance, Navigation, and Control. Notably, deep learning algorithms have facilitated the development of sophisticated pose estimation Proença and Gao (2020); Park and D'Amico (2022b), path planning Santos et al. (2022), and collision avoidance Uriot et al. (2022) systems for autonomous spacecraft, such as satellites and planetary rovers. These AI-powered navigation systems have enabled spacecraft to operate autonomously, leveraging real-time data from onboard sensors and cameras to make informed decisions without human intervention.

AI has a broad range of application in space missions including orbit determination Mortlock and Kassas (2021), pose estimation Hu et al. (2021c), spacecraft health monitoring Mukhachev et al. (2021), and Space Situational Awareness (SSA) Lim et al. (2020). The utilization of AI in spacecraft navigation has enhanced the efficiency and precision of spacecraft operations, thereby enabling exploration of previously inaccessible domains in space. As space exploration continues to evolve, AI is expected to play an increasingly critical role in spacecraft navigation and other space-related applications.

Furthermore, AI has addressed several key challenges in space missions, such as low earth orbit attitude determination Sun et al. (2023), simultaneous navigation and characterization Stacey and D'Amico (2022), pose tracking Park and D'Amico (2022a), planetary landing Gaudet et al. (2020); Furfaro et al. (2018), crater detection Downes et al. (2021, 2020), deep space missions Tsukamoto et al. (2022); Donitz et al. (2022), gravity field modeling Neamati et al. (2022), motion planning Nakka (2021)

AI-based autonomous systems can process and interpret data much faster than traditional methods, enabling spacecraft to react quickly to dynamic situations, such as the detection of a hazard or a change in the terrain Moghe and Zanetti (2020). Additionally, autonomous spacecraft navigation systems are less susceptible to human error and can operate in hazardous environments, which is particularly useful in missions to explore unknown territories in space Song et al. (2022a). Table 5 summarizes the recent research on the main applications of AI in autonomous spacecraft.

Despite the considerable progress made in the field of AI-enabled spacecraft navigation, numerous challenges remain. A major challenge is the limited computational resources available in space. So, algorithms must be optimized for both accuracy and efficiency. Also, the quality of the collected data might be poor due to various factors including interferences, communication loss, and sensor noise.

As spacecraft operate in unpredictable and dynamic environment, the reliability and robustness of systems must be thoroughly tested and verified to ensure mission success. Therefore, the development of reliable and efficient AI-enabled spacecraft navigation systems requires a multifaceted approach, rigorous engineering design, testing and validation.

The integration of artificial intelligence (AI) technology within autonomous spacecraft has heralded a transformative era within the realm of space exploration, signifying a profound departure from conventional methodologies. By empowering spacecraft to operate autonomously and employ sensory data for instantaneous decision-making, AI has emerged as an indispensable asset within the space industry. As evidenced by the recent advancements achieved

TABLE 5 Applications of AI in Autonomous Spacecraft

Ref	Application	Method	Description
Scorsoglio et al. (2023)	Relative Motion Guidance	Reinforcement Learning	An actor-critical reinforcement learning algorithm based on lightweight feedback guidance was proposed for proximity operations in the cislunar environment.
Guthrie et al. (2022)	Relative Motion Guidance	Deep learning	Convolutional Siamese network used to estimate the attitude and rotational state of an uncooperative debris satellite with unknown geometry
Proença and Gao (2020)	Pose Estimation	Deep Learning	A CNN pose estimation of known uncooperative spacecraft approach built upon SPEED dataset and present a simulator called URSO built on Unreal Engine 4.
Abd-Elhay et al. (2022)	Fault Detection	Deep Learning	A one-dimensional convolutional neural network (1D-CNN) coupled with an LSTM network architecture is used for detecting and identifying anomalies in spacecraft reaction wheels.
Spantideas et al. (2021)	Magnetic Dipole modeling	Deep Learning	MDMnet, a deep-learning neural network, models the magnetic signature of spacecraft units using synthetic magnetic flux density data generated by virtual dipole sources and achieves high predictive accuracy.
AlDahoul et al. (2022)	Debris Detection	Deep Learning	A multi-modal learning approach is employed to detect space objects, including spacecraft and debris which integrates a vision transformer based on RGB and CNN-based network to classify 11 object categories.
Harris et al. (2022)	Procedure Generation	Reinforcement Learning	Examined the feasibility and safety of deep reinforcement learning for spacecraft decision-making, and the hybridization of DRL-driven policy generation algorithms with correct-by-construction control techniques.
Del Prete et al. (2022)	Navigation	Deep Learning	The Cascade Mask R-CNN was used for detecting craters from monocular images for autonomous spacecraft navigation, and for learning transfer functions from real lunar images acquired by the lunar reconnaissance orbiter.
Siew et al. (2022)	Sensor Tracking	Reinforcement Learning	A deep reinforcement learning agent to task a space-based narrow-FOV sensor in low Earth orbit for space situational awareness
Siew and Linares (2022)	Sensor Tracking	Reinforcement Learning	Detect, characterize, and track resident space objects in complex environments using ground-based optical telescopes.
Oestreich et al. (2021)	Control	Reinforcement Learning	Optimized proximal policies for 6 DoF closed-loop control for cooperating clients in spacecraft docking missions

thus far in space exploration missions, the escalating reliance on AI technology within this domain is poised to become increasingly paramount. Consequently, the trajectory of human exploration is poised to transcend conventional boundaries and embark upon hitherto uncharted frontiers in the cosmic expanse. With the steadfast evolution of AI-driven spacecraft navigation systems, a momentous paradigm shift within the space industry looms on the horizon, promising an era of unparalleled exploration characterized by limitless potential for groundbreaking discoveries and transformative innovations.

Table 6 summarizes recent research in AI-enabled space missions.

4 | DEEP LEARNING COMPONENTS IN AUTONOMOUS NAVIGATION

4.1 | Signal Processing

Signal processing refers to generating and processing sensory data to extract relevant information about the environment which is used to improve the quality of raw data. This could led to improve the decision-making process accuracy. Various signal processing techniques are used in navigation systems including filtering, segmentation, feature extraction, classification, and registration.

Deep learning techniques can be utilized toward multi-sensor data fusion, which involves combining data from multiple sensors to reduce uncertainty and obtain a more comprehensive representation of the environment. The use of deep neural networks in sensor fusion has enabled the system to learn more complex representations of the

TABLE 6 Recent Research in AI-enabled Space Missions

Ref	Method	Backbone	Dataset	Task
Mahendrakar et al. (2023)	Supervised	YOLO	Synthetic	Fault Detection
Park et al. (2019)	Supervised	YOLO + MobileNet	SPEED	Pose Estimation
Chen et al. (2019a)	Supervised	Faster R-CNN + HRNet	SPEED	Pose Estimation
Scorsoglio et al. (2020)	Reinforcement Learning	Deep Reinforcement Learning	✗	Landing
Harris et al. (2022)	Reinforcement Learning	Deep Reinforcement Learning	✗	Decision Making
Elkins et al. (2020)	Reinforcement Learning	Deep Reinforcement Learning	✗	Control
Singh and Junkins (2022)	Supervised	Gaussian Process Regression	Synthetic	Control
Hao et al. (2021)	Supervised	ResNet-50	OrViS	Pose Estimation
Yun et al. (2020)	Supervised	Dense	Synthetic	Motion Planning
Zhou et al. (2021)	Supervised	Track3D	Synthetic	Tracking
Huan et al. (2020)	Supervised	CNN	Synthetic	Pose Estimation
Oestreich et al. (2021)	Reinforcement Learning	Deep Reinforcement Learning	✗	Control
Roberts et al. (2021)	Reinforcement Learning	Deep Reinforcement Learning	✗	Space Situational Awareness
Roberts and Linares (2021)	Supervised	CNN	Synthetic	Maneuver Classification
Siew and Linares (2022)	Reinforcement Learning	Deep Reinforcement Learning	✗	Space Situational Awareness
Abd-Elhay et al. (2022)	Supervised	CNN-LSTM	Synthetic	Fault Detection
Liu et al. (2022b)	Supervised	DeepLabv3+	SPEED + URSO	Image Segmentation
Siew et al. (2022)	Reinforcement Learning	Deep Reinforcement Learning	✗	Space Situational Awareness
Rondao et al. (2022)	Supervised	RNN-CNN	Synthetic	Pose Estimation
AlDahoul et al. (2022)	Supervised	CNN	SPARK	Space Situational Awareness
Liu et al. (2023)	Supervised	TCN	SMAP/MSL	Anomaly Detection
Scorsoglio et al. (2023)	Reinforcement Learning	Deep Reinforcement Learning	✗	Relative Motion Guidance

environment and make more accurate decisions.

Signal denoising is a fundamental task in signal processing and has wide applications in various domains such as inertial navigation, attitude estimation, image processing, and biomedical signal analysis. Traditional methods for denoising such as wavelet transforms and filtering techniques have limitations in effectively removing noise while preserving the signal's fidelity. Recently, deep learning techniques have been applied to denoise signals, and shown promising results due to their ability to learn complex features and patterns directly from raw data.

The utilization of learning-based methodologies confers distinctive advantages by enabling the acquisition of intricate noise patterns that exhibit variability contingent upon the type and inherent characteristics of the input signal. Moreover, the realm of signal processing assumes a pivotal role in the advancement of autonomous navigation systems, encompassing multifarious tasks such as filtering, segmentation, feature extraction, classification, registration, and sensor data fusion. The deployment of deep learning techniques has yielded promising outcomes in the realm of signal denoising, thereby fortifying the veracity and dependability of sensor data across diverse applications, including inertial navigation, attitude estimation, and image processing. Given the capacity of deep learning algorithms to assimilate intricate noise patterns, their discernible appeal as a means of bolstering the fidelity and dependability of autonomous navigation systems becomes increasingly apparent.

Numerous research has been conducted on the utilization of deep learning techniques for denoising various types of data, such as images Pang et al. (2021); Cui et al. (2019), electrocardiogram (ECG) signals Arsene et al. (2019); Nurmaini et al. (2020), wind speed data Peng et al. (2020), sound Zhang and Li (2023); Xu et al. (2020), and magnetic resonance imaging (MRI) Tian et al. (2022a). However, there is a lack of research investigating the benefits of employing these techniques to enhance autonomous navigation.

Tian et. al. Tian et al. (2020a) conducted a comparative investigation of deep learning techniques for the purpose of image denoising. The study aimed to evaluate the effectiveness of various deep models in addressing noise-related challenges in image processing. The authors classified deep CNNs for four categories of noisy tasks, including additive white noisy images, blind denoising, real noisy images, and hybrid noisy images, and analyzed the motivations and principles of the different types of these methods. The authors also compared the state-of-the-art methods on public denoising datasets using quantitative and qualitative analysis.

Jiang et. al. Jiang et al. (2018) used LSTM network to filter MEMS-based gyroscope outputs to improve the accuracy of standalone INS. The results indicate that the denoising scheme was effective to improve MEMS INS accuracy, and the proposed LSTM method was preferable in this application. This study also compares LSTM with Auto-Regressive and Moving Average (ARMA) models with fixed parameters, and the LSTM method performed better in this application. Similarly Brossard et al. (2020) demonstrated that CNNs can denoise IMU measurements to achieve remarkable accuracy in attitude estimation, by filtering out noise caused by sensor drift, environmental disturbances, and dynamic effects.

Four hybrid models consist of LSTM-LSTM, GRU-GRU, and LSTM-GRU LSTM-LSTM, GRU-GRU, and LSTM-GRU, are proposed in Han et al. (2021), and static and dynamic experiments were conducted to validate the effectiveness of the proposed networks. The results show that the LSTM-GRU network has the best noise reduction effect, while GRU-LSTM network is overfitting the large amount of data. The study demonstrates the potential of using deep learning algorithms in MEMS gyroscope noise reduction and improving the accuracy of MEMS-IMU systems.

In Hu et al. (2021b) an end-to-end learning framework developed for denoising LiDAR signals. The method utilizes the encoding and decoding characteristics of the autoencoder to construct convolutional neural networks to extract deep features of LiDAR return signals. The method was compared with wavelet threshold and VMD methods and found to have significantly better denoising effects. It has strong adaptive ability and has the potential to eliminate complex noise in LiDAR signals while retaining the complete details of the signal. The article concludes that the method is feasible and practical and can be further improved by adding more data for training the network. Additionally, DLC-SLAM Liu and Cao (2023) is presented to address the issue of low accuracy and limited robustness of current LiDAR-SLAM systems in complex environments with many noises and outliers caused by reflective materials. The proposed system includes a lightweight network for large-scale point cloud denoising and an efficient loop closure network for place recognition in global optimization. The authors conducted extensive experiments and benchmark studies, which showed that the proposed system outperformed current LiDAR-SLAM systems in terms of localization accuracy without significantly increasing computational cost.

Chen et al.Chen et al. (2022c) utilized CNN to reduce noises from IMU sensors in autonomous vehicles. The proposed system does not need external measuring sources such as GPS or vehicle sensors, unlike EKF approaches, and the method can be applied for both high-grade and low-grade IMUs. Three CNNs were developed, and the results showed up to 32.5% error improvement in straight-path motion and up to 38.69% error improvement in oval motion compared with the ground truth. The authors' primary objective was to develop algorithms with higher performance and lower complexity that would allow implementation on ultra-low power artificial intelligence microcontrollers such as the Analog Devices MAX78000. Similarly, RNNs Engelsman and Klein (2022) have been used to remove noise from accelerometer measurements,

Further, in Yuan and Wang (2023), authors suggested a novel approach for IMU denoising called IMUDB using self-supervised learning and future-aware inference techniques. The method was evaluated using end-to-end navigation on two benchmark datasets, EuRoC and TUM-VI, and demonstrated promising results. The proposed method includes three self-supervised tasks and an uncertainty-aware inference method. The article highlights the efficiency and superiority of the proposed method over competing methods, as demonstrated through experiments on different datasets. Also, they identify the limitations of the proposed method, such as the sequence size and hyper-parameters, and suggest future works to address them.

The use of deep learning for denoising data in autonomous navigation is a promising area of research. By effectively removing noise from data collected by sensors such as IMU and radar, the accuracy and reliability of autonomous navigation systems can be greatly improved. For instance, in inertial-based navigation systems, the denoising of inertial data can enhance the accuracy of the navigation algorithm and reduce the bias and drift in the output. Similarly, the denoising of radar data can improve the performance of radar-based obstacle detection and tracking. Further research in the area of data denoising and preprocessing is needed to fully realize the benefits of these techniques for autonomous navigation.

4.2 | Perception

Learning-based perception methods used deep learning techniques to process and fuse data from multiple sensors, such as cameras, LiDARs, RADARs, and Infrared to obtain information about the surrounding environment. In the context of autonomous navigation systems, perception is a fundamental element that enables the system to detect and analyze the physical world around it, make decisions based on that information, and take actions accordingly. A more detailed discussion can be found in Thrun (2002).

In following, we will delve into utilizing deep learning methodologies for perception in autonomous navigation systems.

Deep learning algorithms and artificial neural networks, particularly convolutional neural networks, have demonstrated their capability for extracting features and image processing Hu et al. (2019); Matsumoto et al. (1990); Arena et al. (2003). They have uesed for road markings Tian et al. (2020b); Lima et al. (2022), obstacles detection Prabhakar et al. (2017); Hu et al. (2018); Lamberti et al. (2023), and traffic signs recognition Shustanov and Yakimov (2017); Dewi et al. (2022).

Perception encompasses multiple tasks. Initially, the system ingests data from diverse sources. Subsequently, the data undergoes preprocessing, wherein raw sensory data is filtered and processed to extract pertinent information and features. The ultimate phase entails interpretation, during which the system utilizes the extracted information and features to comprehend and perceive the surrounding environment, thus enabling the undertaking of apt actions.

The system is trained on a large amount of data to enable the system to learn the hidden relations beteween the input and output. This enables it to accurately recognize and classify objects and obstacles. Through ongoing operation, the system can continually refine and improve its perception abilities via continuous learning. The datasets mentioned in Table 7 contribute to this process.

TABLE 7 Some of the most popular publicly available datasets for autonomous navigation

Dataset (Year)Length	SensorAdditional Details	Scene / Application
KITTI (2012) Geiger et al. (2012)22 sequences	Camera + LiDAR + IMUstereo and optical flow	Urban / Ground Vehicle
EuRoC MAV (2016) Burri et al. (2016)11 sequences	Cameras + IMUstereo and monocular cameras	Indoor / Drone
NCLT (2016) Carlevaris-Bianco et al. (2016)27 sessions, 34.9h, 147.4 Km	Camera + LiDAR + IMU + GPS + WheelGPS and wheel odometry	Urban / Ground Vehicle
SUN RGB-D (2015) Song et al. (2015)10,000 RGB-D images	Camerasemantic segmentation and object detection	Indoor / Scene Understanding
Cityscapes (2016) Cordts et al. (2016)25,000 images	Camera + GPSsemantic segmentation	Urban / Ground Vehicle
LISA Traffic Light (2016) Jensen et al. (2016)43,000 images	Cameratraffic light annotations	Urban / Traffic light detection
Stanford Drone (2016) Robicquet et al. (2016)+60 video sequences	Cameraobject detection and tracking	Outdoor / UAV + Trajectory Pre
UAV123 dataset (2016) Mueller et al. (2016)123 video sequences	Cameraobject tracking	Indoor / Drone
Zurich Urban MAV (2017) Majdik et al. (2017)20 flights, 10 km	Camera + GPS + IMUGPS and IMU data	Outdoor / Drone
Oxford RobotCar (2017) Maddern et al. (2017)1000 km, 100 hours	Camera + LiDAR + RADAR + GPSsemantic segmentation and object detection	Urban / Ground Vehicle
DAVIS Dataset (2016) Perazzi et al. (2016)50 video sequences	Cameraobject tracking / Object Tracking	
BDD100K (2020) Yu et al. (2020b)100,000 video clips, 1.4 million images	Camera + GPS + IMUsemantic segmentation and object detection	Urban / Ground Vehicle
ApolloScape (2018) Huang et al. (2018)10 hours	Camera + LiDAR + GPS + IMUsemantic segmentation and object detection	Urban / Ground Vehicle
UPenn Fast Flight (2018) Sun et al. (2018)10 flights, 20 km	Camera + GPS + IMUGPS and IMU data	Outdoor / Drone
ScanNetV2 (2017) Dai et al. (2017)1,513 scans	Camera + Meshsemantic segmentation and object detection	Indoor / Scene Understanding
KAIST (2018) Choi et al. (2018)100 hours, 100 km	Camera + LiDAR + GPS + IMUpedestrian and vehicle detection	Urban / Ground Vehicle
UZH-FPV (2019) Delmerico et al. (2019)10 flights, 20 km	Camera + IMUGPS and IMU data	Indoor + Outdoor / Drone
Argoverse (2019) Chang et al. (2019)365 km	Camera + LiDAR + GPSHD maps and object detection	Urban / Ground Vehicle
H3D (2019) Patil et al. (2019)1 hour	Camera + LiDAR + GPS + IMUsemantic segmentation and object detection	Urban / Ground Vehicle
Lyft (2019) Kesten et al. (2019)1,000 scenes, 16,000 frames	Camera + LiDARsemantic segmentation and object detection	Urban / Ground Vehicle
A2D2 (2020) Geyer et al. (2020)	Camera + LiDARsemantic segmentation and object detection	Urban / Ground Vehicle
A3D (2020) Pham et al. (2020)39,000 frames	Camera + LiDAR3D object detection	Urban / Ground Vehicle
Carla Simulator (2021) Deschaud (2021)–	Camera + LiDARvarious scenarios and weather conditions	Simulation / Ground Vehicle
nuScenes (2020) Caesar et al. (2020)1,000 scenes, 400,000 objects	Camera + LiDAR + RADAR + GPS + IMUsemantic segmentation and object detection	Urban / Ground Vehicle
Waymo Open (2020) Sun et al. (2020)1,000 scenes, 200,000 frames	Camera + LiDARsemantic segmentation and object detection	Urban / Ground Vehicle
Blackbird (2020) Antonini et al. (2020)10 flights, 20 km	Camera + IMUGPS and IMU data	Indoor + Outdoor / Drone
VisDrone dataset (2021) Zhu et al. (2021)10,209 images	Cameraobject detection	Outdoor / Drone
WHUVID (2022) Chen et al. (2022d)10 hours	Camera + IMUGPS and IMU data	Urban / Ground Vehicle
HDIN (2022) Chang et al. (2022)10,000 images	Camerasemantic segmentation and object detection	Indoor / Drone

Although deep learning-based perception systems offer numerous benefits, they still several challenges which require attention including:

- The potential for bias in the training data, which can lead to inaccuracies in perception and negatively impact the performance of autonomous systems.
- The difficulty of operating in complex and dynamic environments, where there is a high degree of uncertainty and unpredictability, which can result in reduced accuracy and reliability.
- The significant computational requirements of deep learning-based perception systems, which may necessitate high-performance computing resources to achieve optimal performance.

Feature extraction, object/obstacle detection, and segmentation are fundamental tasks in computer vision that are closely interconnected and frequently employed in conjunction. Feature extraction entails identifying and extracting pertinent visual characteristics from an image or video frame, which can subsequently be leveraged for further analysis. On the other hand, Object/obstacle detection encompasses the process of identifying and localizing objects or obstacles within an image or video frame. But segmentation refers to the process for partitioning an image or video frame into distinct regions or segments based on their visual attributes.

In followings, we will delve into these three tasks and explore their recent advancements. Notably, due to the inherent interdependence and co-occurrence of these tasks, they are often implemented jointly in various research studies.

4.2.1 | Feature Extraction

Extracting features algorithms responsible to represent distinct patterns or characteristics from raw sensory data including images or image regions. These features serve as meaningful representations that enable higher-level analysis and interpretation of visual data. Some of conventional methods are:

- **Scale-Invariant Feature Transform (SIFT)**: Detects and describes local features in images that are invariant to scale changes.
- **Speeded Up Robust Features (SURF)**: Efficiently detects and describes local features, including those invariant to scale and rotation.
- **Oriented FAST and Rotated BRIEF (ORB)**: Combines the FAST keypoint detector and the BRIEF descriptor to extract and match features.
- **Histogram of Oriented Gradients (HOG)**: Captures information about the distribution of gradients in an image, commonly used in object detection tasks.
- **Local Binary Pattern (LBP)**: Encodes the local texture information by comparing the intensity values of pixels in a neighborhood.
- **Gabor Filters**: Extract features based on the responses of filters tuned to different frequencies and orientations, suitable for texture analysis.
- **Haar-like Features**: Simple rectangular filters that are used in the Viola-Jones object detection framework to identify object patterns.

The goal of extracting features is to derive meaningful representations that facilitate higher-level analysis and interpretation. Deep learning methodologies have gained significant traction in the field of feature extraction, employing diverse imaging technologies such as monocular RGB cameras, RGB-D sensors, and infrared sensors. CNN-based architectures such as residual networks (ResNet) He et al. (2016), DenseNet Huang et al. (2017), and attention mechanism Vaswani et al. (2017) showed their capability in extracting the feature from raw data, in particular visual data.

In fact, CNNs have emerged as the predominant feature extractors within the deep learning paradigm, exhibiting robust capabilities across a multitude of sensor types. Numerous studies conducted on feature extraction by implementing these type of networks Thakur and Maji (2023); Rajkumar (2022); Wang et al. (2023d, 2022b, 2023b); He (2020); Amosa et al. (2022).

This prevalent utilization of CNN-based feature extraction methods can be attributed to their efficacy in capturing and representing high-dimensional visual patterns, enabling effective information extraction from complex image data. By leveraging the hierarchical nature of CNN architectures, these feature extractors effectively learn discriminative features at multiple levels of abstraction, facilitating enhanced performance across a broad spectrum of applications. Nonetheless, while CNNs have achieved remarkable success in various image processing tasks, further research and exploration are warranted to investigate alternative feature extraction approaches and their compatibility with different sensor modalities. Such endeavors hold great potential for advancing the state-of-the-art in feature extraction using deep learning techniques, propelling innovation and breakthroughs across numerous domains reliant on image analysis and understanding.

CNN-based feature extraction techniques have been extensively utilized in robotics and computer vision, encompassing a wide range of applications such as scene classification and visual navigation systems. A notable contribution in this domain is the introduction of a CNN-GRU model that focuses on extracting behavioral features from inertial navigation trajectories. This model showcases the successful integration of RNNs with CNNs, enabling the capture of temporal dependencies and enhancing the accuracy of feature extraction Li et al. (2021a).

Furthermore, in the study conducted by Djenouri et al. Djenouri et al. (2022), the utilization of CNN-based networks for feature extraction from images in the context of a visual navigation system has been explored. This approach has proven effective in extracting discriminative visual cues, facilitating efficient and robust navigation.

Another popular approach is the attention mechanism, which is implemented in Mayo et al. (2021). This study aimed to enhance the feature extraction capability of channel state information (CSI) signals, address the limitations of non-adjacent feature aggregation and long-distance point mismatching in existing fingerprint-based localization algorithms, and validate the stability and accuracy of the Multi-Head Self-Attention mechanism combined with Effective Channel state information (MHSA-EC) algorithm. The proposed approach effectively aggregates long-distance CSI features and reduces mismatches of long-distance points in indoor localization. Another study Rajkumar (2022) used a similar approach based on attention mechanism for copy-move forgery detection in digital images. The attention-based DenseNet 121 model is employed to extract features from non-overlapping patches.

A deep multi-feature extraction framework based on attention mechanism (DMEFAM) is developed in Wang et al. (2023d) for human activity recognition (HAR) using wearable sensor data. This study aimed to address the existing challenges related to incomplete feature extraction and low utilization rate of features, which can lead to inaccurate activity recognition. This framework comprises the temporal attention feature extraction layer (TAFEL) and the channel and spatial attention feature extraction layer (CSAFEL) that leverage bidirectional gated recurrent units (Bi-GRU), self-attention (SA) mechanism, convolutional block attention module (CBAM), and residual network-18 (ResNet-18). By incorporating these components, the framework aimed to assign varying degrees of importance to features, increase the diversity of feature extraction, and enhance the overall accuracy of HAR.

In the context of signal feature extraction in aviation, an attention-based deep learning approach was investigated to improve the recognition of aircraft signals. The study aimed to address the challenges associated with radio signal recognition, particularly in the context of future communication systems and vehicular communication. The researchers focused on the extraction of electromagnetic fingerprints. To achieve this, a significant number of downlink aircraft communications addressing and reporting system (ACARS) signals were collected and generated. These signals, which facilitate communication between civil aircraft and airport towers, were used as the basis for feature extraction and recognition. A pre-transformation network was developed to convert the 1D signals into 2D feature maps for a more effective feature extraction. These feature maps were then fed into CNN-based models for classification. To enhance the classification accuracy, attention modules were integrated, resulting in improved performance compared to a baseline approach. The proposed method achieved a 94.1% accuracy rate for aircraft type identification and reduced the identification time compared to traditional methods.

Another study He (2020) utilized attention-based approach for image compressive sensing, addressing the challenge of reconstructing high-quality images from compressed measurements which employed a sampling network, an initial reconstruction network, and a residual reconstruction network. They integrated channel attention mechanism and multi-scale feature extraction to capture meaningful information for image reconstruction. The use of deep supervision during training further enhances the reconstruction quality and the experimental simulations validate the effectiveness of the proposed framework, demonstrating competitive performance compared to traditional methods.

In Wang et al. (2023b) a self-attention and self-cooperation YOLOX (SASC-YOLOX) method developed for smoke detection. The existing smoke detection methods face challenges in extracting accurate smoke features due to its transparency and variability. The SASC-YOLOX method addresses this issue by incorporating self-cooperation and self-attention mechanisms into the YOLOX architecture. These mechanisms filter redundancy, condense localization and semantic information, and emphasize meaningful features of smoke. The proposed method outperforms other methods on both a real smoke database and a synthetic database. The SASC-YOLOX method shows strong feature extraction ability and is a candidate for smoke detection in terms of its small model size, high accuracy, and frames per second.

A comparative study Amosa et al. (2022) focused on the feature extraction component of deep learning-based visual object trackers for industrial robots. The researchers compared the performance of CNN-based and Transformer-based feature extraction methods and evaluated their impact on the overall tracking performance. The study employed a Siamese-based tracker trained on ILSVRC15 datasets and a Transformer-based tracking architecture pre-trained on the ImageNet dataset. The authors also contributed a dataset of images of industrial robots for evaluating the trackers. They showed that the transformer-based feature extraction method significantly improved the overall tracking performance of the tracker.

A blind Gaussian denoising network presented in Thakur and Maji (2023), which utilizes the features of the negative input image to focus on enhancing the low contrast regions of the image. The multi-scale pixel attention (MSPA) block analyzes the features of the negative image at multiple scales and combines the most important features to generate a feature map. Similarly, the multi-scale feature extraction (MSFE) block analyzes the features of the input noisy image at different scales. These blocks contribute to the overall effectiveness of the denoising network. The study highlights that the proposed network is lightweight, fast, and outperforms state-of-the-art blind Gaussian denoising methods both quantitatively and qualitatively. The findings suggest that leveraging the features of the negative input image and incorporating multi-scale analysis can significantly improve the denoising performance. This network has potential applications in plug and play image restoration, real-time video denoising, and real image denoising.

The study presented in Zhang et al. (2021) developed a learning-based method to enhance the matching of infrared aerial images with satellite reference images. The main objectives of the research were to develop a learning-based approach for extracting local features in UAV navigation, improve the accuracy and robustness of image matching under challenging conditions, and address issues related to inconsistent image capture conditions and different imaging principles. To achieve these objectives, the authors proposed a network for local feature detection and description and introduced an iterative training scheme to optimize pseudo-label generation. The performance of the proposed method is evaluated and compared with state-of-the-art techniques, with a particular focus on its potential for UAV localization and trajectory reconstruction applications.

These advancements demonstrate the versatility of learning-based feature extraction techniques. By leveraging the power of deep learning, these methods offer a sophisticated means of processing complex visual data, enabling the extraction of meaningful features crucial for accurate perception and decision-making in real-world scenarios. The integration of CNNs with other architectures, such as RNNs or attention mechanisms, opens up further possibilities for enhancing feature extraction methods, empowering robots and computer vision systems to operate effectively in dynamic and visually diverse environments.

4.2.2 | Object Detection

The process of identifying an object's presence and class in an image or video frame and determining its location, also known as object detection, is widely used in computer vision applications including robotics, surveillance, and visual navigation. It involves creating bounding boxes around objects and labeling them with class labels Zou et al. (2023). Table 8 provides an overview of the most popular object detection algorithms. There are several types of object detection algorithms, including:

- **Feature-based methods**: Identify specific features, such as edges or corners, within an image. These methods are computationally efficient but may struggle with complex or cluttered scenes.
- **Template matching**: Compares an image to a predefined template to identify objects that match the template. This is simple and effective but may struggle with variations in object appearance or lighting conditions.
- **Region-based methods**: Divide an image into regions and analyze each region to detect objects. Region-based

methods are effective in detecting objects of varying sizes and shapes but may be computationally expensive.

- **Segmentation-based methods**: Segment an image into regions based on color, texture, or other properties and then analyze each segment to detect objects. Segmentation-based methods are effective in detecting objects in cluttered scenes but may struggle with objects that have similar properties to the background.
- **Motion-based methods**: Detect objects by analyzing changes in motion within an image or video stream. Motion-based methods are effective in detecting moving objects but may struggle with stationary objects or objects with slow motion.
- **Hybrid approaches**: Combine multiple techniques, such as feature-based methods and deep learning-based approaches, to improve object detection accuracy and efficiency.

TABLE 8 Object Detection Algorithms

Algorithm	Description
VJ Det. Viola and Jones (2004)	Viola-Jones detector, a real-time face detection algorithm
HOG Det. Dalal and Triggs (2005)	Histogram of Oriented Gradients, used to detect objects in images
DPM Felzenszwalb et al. (2008)	Deformable Parts Model, used for detecting objects with a combination of local parts
RCNN Girshick et al. (2014)	Region-based Convolutional Neural Networks, uses selective search for region proposals and a CNN to classify objects
SPPNet He et al. (2015b)	Spatial Pyramid Pooling Network, a CNN architecture that can handle images of arbitrary size and aspect ratio
Fast RCNN Girshick (2015)	Faster Region-based Convolutional Neural Networks, an improvement of RCNN that shares computation among the region proposals
Faster RCNN Ren et al. (2015)	A faster version of Fast RCNN that uses a Region Proposal Network to generate region proposals
YOLOv1 Redmon et al. (2016)	You Only Look Once version 1, a single-shot object detection model that predicts the bounding boxes and class probabilities directly from the image
YOLO9000 Redmon and Farhadi (2017)	An improved version of YOLOv1 that uses anchor boxes to improve the accuracy of object detection
YOLOv3 Redmon and Farhadi (2018)	An even more accurate version of YOLO that uses feature maps to predict object classes and bounding boxes
YOLOv4 Bochkovskiy et al. (2020)	You Only Look Once version 4 that uses several new techniques, including spatial pyramid pooling, path aggregation networks, and cross-stage partial connections
YOLOv5 Jocher et al. (2022)	A lightweight version of YOLO that achieves state-of-the-art accuracy and speed by using a novel CSPNet backbone and a hybrid approach to anchor boxes.
YOLOv6 Li et al. (2022a)	Used an efficient backbone with RepVGG or CSPStackRep blocks, a PAN topology neck, and an efficient decoupled head with a hybrid-channel strategy
YOLOv7 Wang et al. (2023a)	A couple of architecture changes and a series of bag-of-freebies, which increased the accuracy without affecting the inference speed.
YOLOv8 Jocher et al. (2023)	An anchor-free model with a decoupled head to independently process objectness, classification, and regression tasks.
SSD Liu et al. (2016)	Single Shot MultiBox Detector, a single-shot object detection model that uses a series of convolutional layers to predict object classes and bounding boxes
FPN Lin et al. (2017b)	Feature Pyramid Network, a CNN architecture that can handle images of different scales and resolutions by constructing a feature pyramid
RetinaNet Lin et al. (2017c)	A single-shot object detection model that uses a feature pyramid network and a focal loss function to handle the class imbalance problem
CornerNet Law and Deng (2018)	A keypoint-based object detection model that uses corners as key points to detect objects and their orientations
CenterNet Zhou et al. (2019)	A keypoint-based object detection model that predicts the center point of objects and their size and orientation
DETR Carion et al. (2020)	Detection Transformer, a transformer-based object detection model that uses an attention mechanism to directly predict the set of objects in an image
Cascade R-CNN Cai and Vasconcelos (2018)	A variant of Faster R-CNN that uses a cascade of detectors to improve accuracy.
EfficientDet Tan et al. (2020)	A family of object detection models that use EfficientNet as a backbone and incorporate various design changes to improve speed and accuracy.
Sparse R-CNN Sun et al. (2021a)	A variant of Faster R-CNN that uses sparse convolutional layers to reduce computation and memory requirements.
RepPoints Yang et al. (2019)	A method that represents object instances as a set of points and performs detection by matching these points to features in the image.
FCOS Tian et al. (2019)	A fully convolutional object detection method that predicts the bounding box, class, and objectness score for each pixel in the feature map.

4.2.3 | Obstacle Detection

Obstacle detection is a critical component and core technical problem in autonomous navigation systems, as it involves identifying and classifying static and dynamic objects such as vehicles, pedestrians, and other obstacles that may be present in the vehicle's path. In the past decades, various sensors have been utilized and developed for this purpose, including cameras, ultrasonic sensors, LiDARs, and RADARs. However, the most common sensor used for obstacle detection is the camera, as it is the most cost-effective and widely available sensor which is also capable of capturing a wide field of view, making it possible to detect objects that are far away from the vehicle.

4.2.4 | Segmentation

Image segmentation, involves identifying and labeling each pixel in an image or video as belonging to a particular object or class which is more precise than object detection as it provides a more detailed understanding and partitioning the digital image or video frames. It can be used for tasks such as satellite image analysis, medical image analysis, face recognition and detection, object tracking and scene understanding.

Segmentation algorithms use various techniques to segment the objects, including:

- **Thresholding**: Converts gray-scale images to binary images by setting a threshold value.
- **Clustering**: Groups pixels based on their similarity in color or intensity.
- **Edge-Based**: Detects edges in an image by finding abrupt changes in intensity.
- **Region-Based**: Segments an image into regions based on similar properties such as color or texture.
- **Watershed**: Separates objects in an image by treating the image as a topographic map and flooding it from local minima.
- **Neural Networks**: Uses artificial neural networks to classify or segment images based on patterns learned from training data.

Semantic segmentation is a type of image segmentation that assigns each pixel in an image a semantic label or category. The goal is to identify and classify the objects and regions of interest within an image, such as cars, buildings, people, or background. The output of semantic segmentation is a pixel-level map that associates each pixel with a specific object or region class Lin et al. (2014). In summary, while image segmentation aims to group pixels into visually similar parts, semantic segmentation goes further by assigning a semantic meaning to each of these parts.

The main difference between image segmentation and semantic segmentation is that image segmentation groups pixels into visually similar parts, while semantic segmentation assigns a semantic meaning to each of these parts. This means that semantic segmentation provides a more detailed understanding of the objects and regions of interest within an image, making it useful for tasks such as object recognition and scene understanding. In contrast, image segmentation is more focused on identifying visually similar parts of an image, making it useful for tasks such as image editing and compression.

Figure 5 shows the difference between image segmentation, semantic segmentation, and object detection.

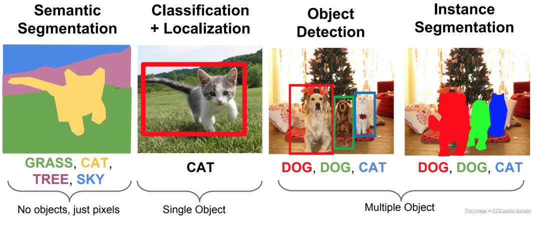

FIGURE 5 Image segmentation, semantic segmentation, and object detection. Li et al. (2017)

In Table 9, we summarize the most popular object segmentation algorithms.

TABLE 9 Image Segmentation Algorithms

Algorithm	Description
FCN Long et al. (2015)	Fully Convolutional Network, a CNN architecture that can output pixel-level predic
UNet Ronneberger et al. (2015)	A CNN architecture that combines convolutional and deconvolutional layers to pr tion masks
SegNet Badrinarayanan et al. (2017)	A CNN architecture that uses an encoder-decoder structure with skip connection
DeepLab Chen et al. (2017a)	A CNN architecture that uses atrous convolution and a global pooling layer to captu tation
PSPNet Zhao et al. (2017)	Pyramid Scene Parsing Network, a CNN architecture that uses spatial pyramid po gregation for semantic segmentation
ICNet Zhao et al. (2018)	A CNN architecture that performs real-time semantic segmentation by using a mult feature extractor
ENet Paszke et al. (2016)	An efficient CNN architecture for real-time semantic segmentation on mobile dev
DeepLabv3+ Chen et al. (2017b)	An improved version of DeepLab that uses a feature pyramid and an encoder-dec mentation
HRNet Wang et al. (2020a)	A CNN architecture that uses a high-resolution representation for semantic segm art performance on various datasets
Mask R-CNN He et al. (2017)	An extension of Faster R-CNN that adds a segmentation branch to predict the instance
LinkNet Chaurasia and Culurciello (2017)	A lightweight encoder-decoder architecture for fast and accurate segmentation, between the encoder and decoder to improve information flow
RefineNet Lin et al. (2017a)	A multi-path refinement network that employs residual connections and multi-sc curate and detailed segmentation
BiSeNet Yu et al. (2018)	A two-pathway network for real-time semantic segmentation, which combines sp ing path) and context path (long and slow processing path) for accurate segmentat

Autonomous navigation systems rely heavily on the aforementioned tasks to perceive the environment and make the proper decision. There are several limitations and constraints which the need to address including extensive datasets and computational resources. Despite these limitations, object detection and segmentation techniques have demonstrated significant potential in enabling autonomous navigation in a variety of applications. Deep learning algorithms have proven highly effective in object detection tasks, with one of their main advantages being their ability to learn rich and complex representations of the objects being detected which enable them to identify objects in challenging scenarios such as adverse weather conditions or low-light environments.

Object detection techniques, such as Faster R-CNN and YOLO, demonstrated high accuracy and speed which makes them ideal for real-time applications. These techniques use deep networks to analyze the digital image or video frames and identify the objects of interest. For example, in Mahendrakar et al. (2022), Faster R-CNN and YOLOv5 has been used for relative navigation tasks in on-orbit servicing and active space debris removal technology. They showed that in a formation flight simulation, while Faster R-CNN is more accurate than YOLOv5 but YOLO is 10 times faster. Visual SLAM has been used for semantic labeling and object detection by combining DeepCNN with Real-Time Graph-Based SLAM for an open-source ground robot Sadeghi Esfahlani et al. (2022).

In Shin et al. (2022) a multi-model based object detection with environment-context awareness introduced to solve the environment class imbalance problem for autonomous mobile robots. They used YOLOv5m Jocher et al. (2020) with Objects365 Shao et al. (2019) dataset for environment-specific object detection, ResNet He et al. (2016) trained by Places365 Zhou et al. (2017) for scene classification. Authors in Zhou et al. (2022b) use KITTI Geiger et al. (2012) and Lyft Houston et al. (2021) dataset to train a doubled stereo-guided monocular 3D (SGM3D) object detection framework based on monocular images for autonomous vehicle navigation.

Furthermore, deep learning has the potential to leverage robots' indoor navigation performance. In a recent study by Afif et al. Afif et al. (2022), lightweight EfcientDet Tan et al. (2020) was evaluated as an assistive navigation system for the visually impaired people. The study demonstrated that deep learning-based object detection and classification can effectively aid indoor navigation for individuals with visual impairments. By utilizing the lightweight and efficient object detection algorithm EfficientDet, mobile robots with limited resources can accurately detect obstacles and environmental features in real-time in indoor environments. This technology can greatly enhance the safety and efficiency of indoor navigation, particularly for visually impaired individuals.

In recent years, there have been many advances in the field of learning-based obstacle detection. For example, researchers in Talele et al. (2019) used TensorFlow Abadi et al. (2016) and OpenCV Bradski (2000) for real-time classification and detection of obstacles in each image pixel. Also, it provides a high resolution binary obstacle image. Also, A local and global defogging algorithm based on deep learning models for object detection in harsh weather scenarios proposed by Jiao and Wang Jiao and Wang (2021). The approach fuses the rich representation capability of visual sensors and the weather-resilient performance of inertial sensors to achieve robust positioning in all-weather conditions.

Markov random field model is used by fusing gradient, curvature prior, and depth variance potential for obstacle segmentation in Haris and Hou (2020). A CNN model is trained to predict the steering wheel angle and navigate the vehicle safely in a complex environment. LiteSeg Luu et al. (2020) introduced by Luu et al. which is a lightweight architecture for adaptive road detection method that combines lane lines and obstacle boundaries. This model simultaneously detects lanes and avoids obstacles. They used NVIDIA Jetson TX2 and Unity software to test their model in a simulated environment. Deep Simple Online and Realtime Tracking (deep SORT) was combined with YOLOv3 to detect and track the dynamic obstacles in Qiu et al. (2020). This model is used to detect and track buffaloes and humans in paddy field environments.

Authors in Cervera-Uribe and Mendez-Monroy (2022) have developed a very deep CNN, called U19-Net which improved encoder-decoder architecture for vehicle and pedestrian detection. The output of this model is a pixel-level mask which detects the presence of vehicles and pedestrians in each pixel.

Ci et al. presented a new unexpected obstacle detection method for Advanced Driving Assistance Systems (ADAS) based on computer vision Ci et al. (2022) which consists of two independent detection methods: a semantic segmentation method and an open-set recognition algorithm, that are combined in a Bayesian framework. This model is tested on the Lost and Found dataset Pinggera et al. (2016) and shows improved detection rate and relatively low false-positive rate, particularly when detecting long-distance and small-size obstacles.

Researchers in Wang et al. (2022a) used Deformable Detection Transformer to detect farmland obstacles in aerial images. In another study He et al. (2022), the ME Mask R-CNN is proposed to conduct fine-grained detection of 11 kinds of obstacles so that the detection accuracy of train obstacle identification can be improved.

Li et al. Li et al. (2022b) presented a real-time computational efficient object detection framework. They collected a real world stereo dataset which consists of indoor and outdoor scenes. The authors used a stereo camera to generate a 3D point cloud and then used a voxelization method to convert the point cloud into a voxel grid. The voxel grid is then fed into a 3D convolutional neural network to detect objects. The authors compared their method with other 3D object detection methods and showed that their method is more computationally efficient than other methods.

Further, Cortés et al. Cortés et al. (2022) presented an unsupervised domain adaptation training strategy for the semantic labeling of 3D point clouds. It bridges the domain gap caused by different LiDAR beams in 3D object detection tasks using LiDAR Distillation. This is done by training a model on data from one domain and then applying it to data from another domain, allowing the model to recognize objects in 3D point clouds even when they are misjudged as background points.

The improved YOLOv3 algorithm (YOLOv3-4L) is used in Wang et al. (2022d) to detect obstacles by simplifying darknet-53 and forming a four-scale detection structure. The output is the type and position of obstacles coincident with dangerous areas. In Franke et al. (2022), authors presented on-board and UAV-based vision systems for long-range obstacle detection in railways. A CNN-based Obstacle Detection and Collision Prevention developed in Devi et al. (2022) which has a Raspberry Pi 4 Model B as its control and processing unit.

A support system for visually impaired persons is developed in Hayashida and Miwa (2022) to detect obstacles and notify the person. This model used YOLOv3-tiny with seven CNNs to extract the features from the image and identify the obstacles. Byun et al. Byun et al. (2022) proposed an obstacle detection method for autonomous driving in agricultural environments based on point clouds from pulsed LiDAR technology. Deep learning models are used to detect property information of obstructions.

YOLO model is used for lane detection and obstacle identification in Huu et al. (2022) to Control Self-Driving Cars. They used real-time videos and the TuSimple dataset to test the proposed algorithm. Another study used YOLOv5 for real-time obstacle detection in maritime environments for autonomous berthing in Chen et al. (2022b). Authors improved YOLOv5s by Convolutional Block Attention Module (CBAM) also, they developed squeeze-and-excitation block called YOLOv5-SE. Additionally, in Ma et al. (2022) YOLO model is used to detect and track obstacles for trains.

In the context of self-driving cars, object detection plays a critical role in detecting and tracking other vehicles, pedestrians, and traffic signs for ensuring safe navigation. One crucial aspect of the object detection system is its ability to accurately determine the presence of objects and classify them into their respective categories. Achieving high accuracy in object detection is crucial for the overall safety and success of autonomous driving systems. To address this problem Gasperini et al. Gasperini et al. (2021) proposed CertainNet to estimate the uncertainty of the outputs (presence of object, and its class, location and size). They used the KITTI dataset to train the model.

In another study Hu et al. (2022), the researchers investigated how LiDARs placement effects on the performance of object detection models. They showed that sensor placement is significant in 3D point cloud-based object detection, contributing to 5% to 10% performance discrepancy.

Another method for object detection in low visibility conditions is to fuse mmWave radar with a visual sensor. The authors Chang et al. (2020) proposed a method for object detection in low visibility conditions by using a combination of mmWave radar and camera. This method was used to develop Spatial Attention Fusion (SAF) for merging the feature maps of radar and camera. Their model has been built upon the FCOS Tian et al. (2019) framework and trained on nuScenes dataset.

Autonomous Underwater Vehicles (AUVs) are one of the key components for ocean exploration and surveillance. However, the underwater environment due to the presence of turbid water makes it difficult for AUVs to detect and recognize objects. In Pranav et al. (2022) the authors proposed DeepRecog framework for improving the performance of AUV vision systems for underwater exploration. The framework employs a three-pronged approach for image deblurring using CNNs, adaptive, and transformative filters, in addition to an ensemble object detection and recognition module that surpasses existing algorithms in terms of precision, such as YOLOv3, FasterRCNN + VGG16, and FasterRCNN. This approach offers real-time detection and recognition. They provide a practical solution to tackle the challenges faced by AUV vision systems in underwater settings. The combination of image deblurring and object detection within a single framework is a noteworthy achievement. The experimental results demonstrate the superior performance of the proposed method and its potential for real-world applications.

To enhance the safety of train operations it is crucial to detect the objects in the rail region autonomously. Authors in Zhang et al. (2023b), used LiDAR and camera to detect the obstacle within the rail region by utilizing CNNs for semantic segmentation and pixel-wise object detection. An encoder-decoder architecture based on CNNs was used to segment images for the purpose of identifying rail regions. In addition, the residual network structure and depthwise separable convolution were used for obstacle detection. By using LiDAR, the model was able to cope with unpredictable weather conditions more robustly. The results show a precision of 99.994%.

The advancements in object detection and segmentation techniques have shown great potential in the field of autonomous navigation. However, there are several limitations that need to be addressed in order to ensure safe and reliable autonomous navigation. One of the primary challenges is the requirement for large and diverse datasets as well as significant computational resources for training and processing image or video frames in real-time. Obtaining sufficient training data in real-world environments is difficult and computational constraints can be limiting, particularly for embedded systems. Other challenges are listed as below:

1. **Limited Data and Resources**: Gathering diverse datasets and managing computational resources for training and real-time processing pose challenges. Obtaining sufficient real-world training data is difficult, and computational constraints can be limiting.
2. **Handling Environmental Variations**: Object detection models should handle variations in lighting, weather, and occlusions to ensure consistent performance across different conditions.
3. **Real-Time Performance**: Object detection and segmentation techniques need to operate in real-time, providing timely and accurate information for quick decision-making.

TABLE 10 Summary of Recent Studies on Feature Extraction, Object/Obstacle Detection, & Segmentation Techniques

Ref	Method
Mahendrakar et al. (2022)	Faster R-CNN and YOLOv5 for relative navigation in On-Orbit Servicing and Active Space Debris Removal Technology
Sadeghi Esfahlani et al. (2022)	Visual SLAM combined with deep CNN for semantic labeling and object detection in ground robots
Shin et al. (2022)	Multi-model based object detection with environment-context awareness
Zhou et al. (2022b)	Doubled stereo-guided monocular 3D (SGM3D) object detection framework for autonomous vehicles navigation
Afif et al. (2022)	Lightweight EfficientDet for assistive navigation for visually impaired individuals
Talele et al. (2019)	TensorFlow and OpenCV for real-time obstacle classification and detection
Jiao and Wang (2021)	Multi-model approach using visual and inertial sensors for robust positioning in all-weather conditions
Haris and Hou (2020)	Obstacle segmentation using Markov random field model
Luu et al. (2020)	LiteSeg for adaptive road detection combining lane lines and obstacle boundaries
Qiu et al. (2020)	Deep SORT combined with YOLOv3 for dynamic obstacle detection and tracking
Cervera-Uribe and Mendez-Monroy (2022)	U19-Net for vehicle and pedestrian detection with pixel-level mask output
Ci et al. (2022)	Bayesian framework combining semantic segmentation and open-set recognition for obstacle detection
Wang et al. (2022a)	Deformable Detection Transformer for farmland obstacle detection in aerial images
He et al. (2022)	ME Mask R-CNN for fine-grained detection of train obstacles
Li et al. (2022b)	Real-time 3D object detection using 3D convolutional neural network on a real-world stereo dataset
Cortés et al. (2022)	Unsupervised domain adaptation training strategy for semantic labeling of 3D point clouds
Wang et al. (2022d)	YOLOv3-4L for obstacle detection with simplified darknet-53 and four-scale detection structure
Franke et al. (2022)	On-board and UAV-based vision systems for long-range obstacle detection in railways
Devi et al. (2022)	CNN-based Obstacle Detection and Collision Prevention for Raspberry Pi-based system
Hayashida and Miwa (2022)	YOLOv3-tiny with CNNs for obstacle detection in a support system for visually impaired individuals
Byun et al. (2022)	Point cloud-based obstacle detection in agricultural environments using deep learning
Huu et al. (2022)	YOLO model for lane detection and obstacle identification in self-driving cars
Chen et al. (2022b)	YOLOv5 for real-time obstacle detection in maritime environments for autonomous berthing
Ma et al. (2022)	YOLO model for obstacle detection and tracking in trains
Gasperini et al. (2021)	CertainNet for estimating uncertainty of object detection outputs in self-driving cars
Hu et al. (2022)	Investigation of how LiDAR placement affects object detection performance
Chang et al. (2020)	Fusion of mmWave radar and camera for object detection in low visibility conditions
Pranav et al. (2022)	DeepRecog framework for improving AUV vision systems for underwater exploration
Zhang et al. (2023b)	CNNs with LiDAR and camera fusion for autonomous obstacle detection in rail regions

4. **Scalability**: Object detection models should handle complex scenes with a large number of objects, such as crowded urban environments or busy intersections.
5. **Privacy and Security**: Object detection techniques should address privacy concerns and comply with ethical data practices.
6. **Edge Computing and Energy Efficiency**: Processing data at the edge and optimizing energy usage are crucial for embedded systems.
7. **Domain Adaptation**: Deep learning-based object detection and segmentation techniques may not generalize well to new domains, which can lead to poor performance when applied to new datasets.
8. **Tiny Object Detection**: Detecting small objects can be challenging due to their low resolution and the presence of noise in the image.
9. **Integration with Domain Knowledge**: Deep learning-based object detection and segmentation techniques may not always be able to incorporate domain-specific knowledge, which can limit their performance in certain applications.
10. **Performance Evaluation**: Evaluating the performance of deep learning-based object detection and segmentation techniques can be challenging due to the lack of standardized evaluation metrics and datasets.
11. **Data Augmentation**: Generating diverse and representative training data through data augmentation techniques can be challenging, especially when dealing with complex objects and scenes.

Despite these challenges, object detection and segmentation techniques have proven to be highly effective in enabling autonomous navigation in various applications. As the technology continues to evolve, it is expected that these techniques will become more accurate and efficient, further improving the safety and reliability of autonomous navigation.

Deep learning algorithms, particularly convolutional neural networks, have demonstrated their effectiveness in object detection tasks by learning rich and complex representations of the objects being detected. This allows them to identify objects with greater accuracy and robustness, even in challenging scenarios such as adverse weather conditions or low-light environments. These deep learning algorithms can be trained on large and diverse datasets, enabling them to detect a wide range of objects. Moreover, these approaches can handle real-time data from various sensors, such as cameras, LiDARs, and radars, which can be processed and analyzed by deep learning algorithms in real-time, allowing the autonomous vehicle to respond quickly and accurately to any obstacles that may be present in its path.

Object detection techniques, such as Faster R-CNN and YOLO, have been shown to achieve high accuracy and speed in real-time object detection, making them ideal for various applications. For instance, in Mahendrakar et al. (2022), Faster R-CNN and YOLOv5 were utilized for relative navigation tasks in on-orbit servicing and active space debris removal technology. In another study, Sadeghi Esfahlani et al. (2022) combined Visual SLAM with deep CNN for semantic labeling and object detection in an open-source ground robot.

Several recent studies have demonstrated the potential of deep learning-based object detection and classification for indoor navigation. For example, Afif et al. Afif et al. (2022) evaluated the performance of lightweight EfficientDet as an assistive navigation system for individuals with visual impairments. The study showed that deep learning-based object detection and classification can effectively aid indoor navigation for visually impaired individuals. Furthermore, researchers in Talele et al. (2019) utilized TensorFlow and OpenCV for real-time obstacle detection in each image pixel, providing a high resolution binary obstacle image.

In summary, deep learning-based object detection and segmentation techniques have shown great potential in enabling safe and reliable autonomous navigation in various applications. Despite the challenges associated with obtaining sufficient training data and computational resources, learning-based methods have proven highly effective in identifying and locating objects in real-time. Future advancements in this field are expected to further improve the accuracy and efficiency, enhancing the safety and reliability of autonomous navigation. Also, there is a need to develop data augmentation methods to improve the model training and avoid overfitting.

4.3 | Localization and Mapping

By localizing and mapping, robots and vehicles can operate effectively and safely in dynamic and complex environments. This enable them to determine their position and orientation relative to the surroundings.

However, a significant challenge in localization and mapping is the presence of uncertainties in sensor measurements and motion models. Various probabilistic methods including Kalman filters Ullah et al. (2019), particle filters Raja et al. (2021), and graph-based SLAM Grisetti et al. (2010) developed to address this challenge which enable the system to estimate the robot's position and orientation and update the map of the environment as the robot moves.

Recent advancements in sensor technology, algorithms, and machine learning have led to significant progress in localization and mapping in terms of accuracy improvement, simplifying data collection, reducing the need for multiple, static set-ups of bulky, tripod-based systems, and enabling go-anywhere mapping. Deep learning-based methods, in particular, have shown promising results in visual localization and mapping tasks by leveraging large-scale datasets and powerful neural network architectures. Moreover, multi-sensor fusion and cooperative localization have contributed to more robust and accurate localization and mapping in complex environments.

4.3.1 | Simultaneous Localization and Mapping (SLAM)

Simultaneous Localization and Mapping is a method to obtain a 2D or 3D map of an unknown environment while providing the system with a consistent and accurate position and orientation information with respect to a reference frame and allowing it to navigate and make decisions based on its location. SLAM-based systems have applied to various domains from planetary rovers to drones and augmented reality. While there are various types of sensors to implement these systems, most research has been conducted on visual-SLAM as its simple configuration, low cost, low weight, and affected less by noise and bias.

SLAM suffers from various types of errors which impact the accuracy and reliability of the system and can be categorized into two main types:

- **Measurement errors**: Occur due to inaccuracies in sensor measurements, such as noise or calibration errors.
- **Modeling errors**: Occur due to the limitations of the mathematical models used to represent the system and the environment.

These errors can result in inaccurate state estimation which can further impact the accuracy. Also, it can be affected by environmental changes and dynamic objects, which can cause further errors and require the system to update the map in real-time. In general, SLAM comprises several key elements, namely landmark extraction, data association, state estimation, and state and landmark update. These components collectively contribute to the overall functioning of SLAM.

In the past decades, a variety of techniques has been implemented to solve the SLAM problem. Some of these methods listed in Table 11.

TABLE 11 SLAM Methods and Techniques

Category	Methods and Techniques
Feature-based	Extended Kalman Filter (EKF-SLAM)
	FastSLAM
	GraphSLAM
	ORB-SLAM
	SIFT-SLAM
	SURF-SLAM
Direct	Direct Sparse Odometry (DSO)
	LSD-SLAM
	DTAM (Dense Tracking and Mapping)
	DVO (Dense Visual Odometry)
Visual	MonoSLAM
	RGB-D SLAM
	VINS-Mono (Visual-Inertial Odometry)
	OKVIS (Optimized Keyframe-based Visual-Inertial SLAM)
	SVO (Semi-Direct Visual Odometry)
LiDAR-based	Iterative Closest Point (ICP) SLAM
	LeGO-LOAM
	LOAM (LiDAR Odometry and Mapping)
	GICP (Generalized ICP) SLAM
	NDT (Normal Distributions Transform) SLAM
Sensor fusion	VINS-Fusion (Visual-Inertial Navigation System)
	MSCKF (Multi-State Constraint Kalman Filter)
	Robust Pose Graph SLAM
Semantic	ORB-SLAM with semantic mapping
	A-LOAM (Adaptive LiDAR Odometry and Mapping) with semantics
Topological	Hierarchical SLAM
	Topological Graph SLAM
	Place Recognition-based SLAM
Hybrid	FAB-MAP (Fast Appearance-based Mapping)
	RTAB-Map (Real-Time Appearance-Based Mapping)

SLAM has been widely studied in domains such as autonomous vehicles Qin et al. (2020), unmanned aerial vehicles Bavle et al. (2020); Celik and Somani (2013), planetary rovers Geromichalos et al. (2020), spacecrafts Baldini et al. (2018), and robotics Tian et al. (2022b).

In the field of visual SLAM, traditional algorithms have encountered challenges such as scale ambiguity, scale drift, and pure rotation, which have impeded significant progress since 2017 Zhang (2022). In recent years, deep learning has emerged as a promising approach to enhance the accuracy and robustness of SLAM algorithms in various applications, including visual SLAM, LiDAR SLAM, and RGB-D SLAM.

Learning-based visual SLAM methods use deep neural networks to estimate depth information from monocular Amiri et al. (2019) or stereo camera inputs Li et al. (2019). On the other hand, learning-based LiDAR SLAM methods used to classify and segment the environment into different objects Xiao et al. (2019); Wang et al. (2021); Yu and Peng (2020); Langer et al. (2020), making it easier to build a map. Deep learning has also been utilized to address challenges such as handling dynamic objects, improving real-time performance, and dealing with large-scale environments.

One example of deep learning applied to SLAM is the DynaVINS approach presented by Song et al. Song et al. (2022b) which enhanced the robustness of visual-inertial SLAM in dynamic environments by enabling the system to continue building and updating the map even when the environment undergoes changes. This approach utilizes a deep learning-based framework to predict the motion and location of dynamic objects in the scene, enabling the system to accurately estimate its position and the environment's map.

A method for constructing object-oriented semantic maps developed in Sun et al. (2022) which combines the semantic information extracted from instance segmentation with RGB-D version of ORB-SLAM2 Mur-Artal and Tardós (2017). The method not only reduces the absolute positional errors (APE) and improves the positioning performance of the system, but also segments the point cloud model of objects in the environment with high accuracy. The proposed method that can help a robot better understand and interact with its environment, with potential applications in the field of human-computer interaction. The paper emphasizes that in order for a robot to have sophisticated interactions with its surroundings, it needs access to 3D semantic information about the environment. Also, it outlines the method of target tracking based on instance segmentation and provides experimental results on a TUM dataset Sturm et al. (2012).

A visual SLAM-based robotic mapping method, which utilizes a stereo camera system on a rover, has been proposed in Hong et al. (2021) to address the planetary construction problem. The effectiveness of the proposed method for local 3D terrain mapping has been evaluated with point-clouds from terrestrial LiDAR. However, the paper notes that the camera system on the rover is susceptible to varying illumination conditions, and global localization remains a concern for correcting the rover's trajectory estimate. To create a dense 3D point-cloud map, a self-supervised CNN model is trained using the stereo camera system to estimate a disparity map.

A robust monocular visual SLAM with direct Truncated Signed Distance Function (TSDF) mapping based on a Sparse Voxelized Recurrent Network, called SVR-Net presented in Lang et al. (2023). The proposed end-to-end pipeline utilizes a sparse voxelized structure to reduce the memory occupation of voxel features and gated recurrent units to search for optimal matches on correlation maps, resulting in enhanced robustness. Additionally, Gauss-Newton updates are embedded in iterations to impose geometrical constraints, ensuring accurate pose estimation. The method achieves state-of-the-art localization performance on the TUM-RGBD benchmark and provides direct TSDF mapping, a suitable dense map representation for downstream tasks such as navigation and planning.

Another visual-SLAM system based on a point-line-aware heterogeneous graph attention network presented in Lian et al. (2023). The paper proposes a point-line synchronous geometric feature extraction network (PL-Net Figure 6) to solve the problem of weak point-line extraction ability in complex industrial scenarios. To enhance the accuracy and efficiency of the point-line matching process, the paper presents a heterogeneous attention graph neural network (HAGNN), which uses an edge-aggregated graph attention network (EAGAT) to iterate the vertices of the heterogeneous graph constructed from points and lines. The paper also proposes a greedy inexact proximal point method for optimal transport (GIPOT) to calculate the optimal feature assignment matrix to find the global optimal solution for the point-line matching problem. The experiments on the KITTI dataset and a self-made dataset demonstrate that the proposed method is more effective, accurate, and adaptable than the state-of-the-art methods in visual SLAM. The paper also shows that the multi-feature fusion not only improves the accuracy of the algorithm but also avoids the degradation problem that may occur in the pose solution algorithm when using a single feature.

For fault detection in the scan-matching algorithm of SLAM in dynamic environments a CNN-based approach developed in Jeong and Lee (2023). In such environments, the presence of dynamic objects causes changes in the

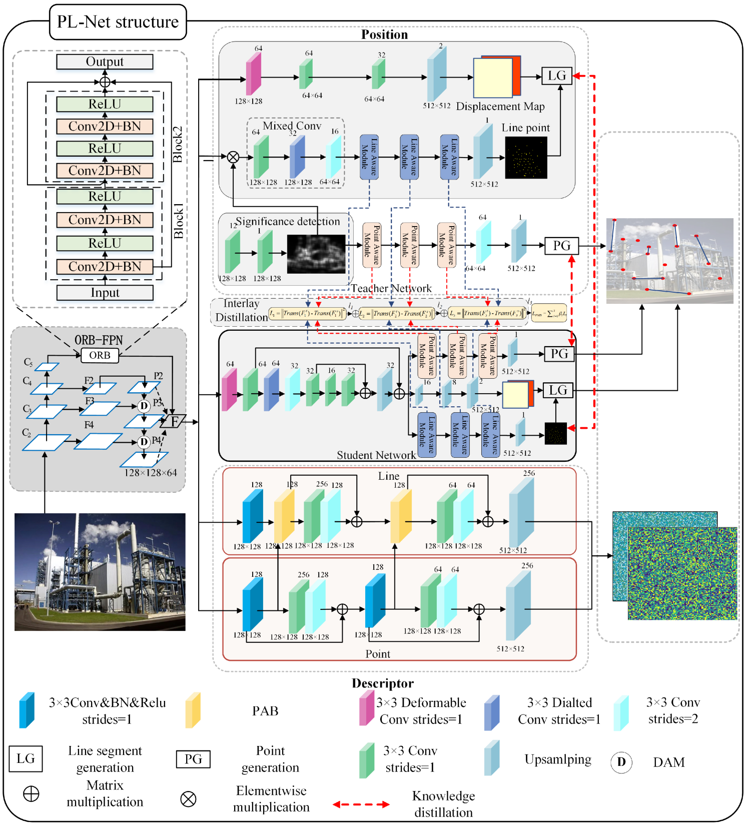

FIGURE 6 PL-Net, A point-line feature extraction network Lian et al. (2023)

environment detected by the LiDAR sensor, leading to faults in the scan-matching process. The proposed method acquires raw scan data from a 2D LiDAR and performs Iterative Closest Points scan matching. The matched scans are converted into images and fed to a CNN model for training to detect faults in scan matching. The proposed method has been evaluated in various dynamic environments, and the results show accuracy in detecting faults. The article highlights the significance of solutions for SLAM problems in dynamic environments with only a LiDAR, and deep learning applications for 2D SLAM. Its process includes raw scan data acquisition, scan matching, matched scan image generation, and fault detection of scan matching. The CNN model's architecture consists of five 2D convolutional layers with max-pooling layers to extract features and softmax activation to classify scan matching as normal or faulty.

While deep learning-based SLAM methods have shown promising results, they also come with some limitations and challenges. Their performance may degrade when operating in highly dynamic and unstructured environments, where the trained models may not generalize well. Another challenge is the potential for overfitting to specific environmental conditions, which can lead to poor generalization when deployed in different environments. This is particularly relevant for visual SLAM methods, where lighting conditions, camera hardware, and environmental texture can have a significant impact on the performance of the deep learning-based models. Also, the computational complexity of deep learning-based SLAM methods could be a barrier, which may require specialized hardware or cloud computing resources to operate in real-time. This can be a significant bottleneck for applications such as autonomous vehicles, where real-time operation is critical. Finally, deep learning-based SLAM methods may also be susceptible to adversarial attacks, where an attacker can manipulate the input data to cause the system to make incorrect predictions. This is a particular concern for safety-critical applications, such as autonomous vehicles.

The most important challenges could be summarized as below:

1. **Overfitting:** They are susceptible to overfitting, leading to poor generalization when deployed in diverse environments.
2. **Environmental conditions:** Visual SLAM methods can be affected by lighting, camera hardware, and environmental texture, significantly impacting deep learning-based models.
3. **Adversarial attacks:** They may be vulnerable to adversarial attacks, compromising their predictions in safety-critical applications.
4. **Interpretability:** Deep learning models in SLAM lack interpretability, hindering understanding of decision-making processes and behavior.
5. **Hardware limitations:** The computational requirements of They may exceed resource-constrained hardware capabilities, necessitating specialized hardware or cloud resources.
6. **Real-time performance:** Achieving real-time performance is crucial in They, especially in applications requiring prompt decision-making.
7. **Generalization:** They may struggle to generalize to new environments or sensor configurations, limiting their practical utility.
8. **Robustness in dynamic environments:** They face challenges in maintaining robust performance in highly dynamic and unstructured environments.
9. **Data scarcity:** Deep learning-based SLAM techniques require extensive labeled data, posing challenges in domains with limited data availability.

Addressing these challenges will contribute to enhancing the reliability, robustness, and practicality of deep learning-based SLAM techniques.

Overall, while deep learning-based SLAM methods have shown great promise, their limitations and challenges need to be carefully considered in order to ensure their practical applicability in real-world scenarios.

4.3.2 | Attitude Estimation

Attitude estimation refers to the process of determining the orientation of a robot with respect to a reference frame Asgharpoor Golroudbari (2022) which plays a critical role in the development of autonomous systems. Different types of sensors can be used to perform attitude estimation, including visual, inertial, and GNSS-based sensors Asgharpoor Golroudbari and Sabour (2023). Table 12, presents a brief overview of sensors which can be used in the attitude estimation process.

TABLE 12 **Attitude Estimation Sensors**

Sensor Type	Characteristics	Advantages	Disadvantages
Inertial Measurement Unit (IMU)	Combines accelerometers, gyroscopes, and magnetometers to measure linear and angular motion.	- Fast response and high update rate. - Suitable for real-time application. - Works well in GPS-denied environments.	- Subject to drift and r - Requires calibration. - Limited long-term ac
Global Navigation Satellite System (GNSS)	Utilizes satellite signals for position and attitude estimation.	- Provides absolute position and orientation information. - Suitable for outdoor environments.	- Limited accuracy canyons or under obst - Susceptible to multi ference. - Requires clear line c satellites.
Vision-based Sensors	Uses cameras and computer vision algorithms for feature tracking and estimation.	- High accuracy and resolution. - Provides rich visual information about the environment. - Suitable for mapping and navigation in complex environments.	- Affected by lighting and occlusions. - Requires computa sources. - Limited performanc texture or featureles ments.
Magnetic Sensors	Measures Earth's magnetic field for attitude estimation.	- Low cost and power consumption. - Complementary to other sensors for heading estimation. - Works well in indoor environments.	- Susceptible to magı ference. - Limited accuracy nea materials. - Prone to error accum
Light Detection and Ranging (LiDAR)	Uses laser beams to measure distances and create a 3D point cloud.	- Provides accurate and dense 3D point cloud data. - Works well in various lighting conditions. - Suitable for mapping and obstacle detection.	- Limited range and fie - Expensive comparec sensors. - Requires computa sources.

To ensure reliable and precise attitude estimation, sensor fusion techniques can be employed to combine the measurements from these sensors to minimize the impact of individual sensor errors Golroudbari and Sabour (2023). There are several techniques for attitude estimation, each with its own strengths and weaknesses.

Complementary Filter: A simple and computationally efficient method that combines the high-frequency response of the gyroscope with the low-frequency response of the accelerometer. This approach works well for estimating the attitude in the presence of low-frequency disturbances, but it may suffer from drift over time due to bias errors.

Extended Kalman Filter (EKF): An advanced version of the Kalman filter that uses a probabilistic model to estimate the true state of the system from noisy sensor measurements which is capable of handling nonlinear systems. The EKF can provide accurate estimates of attitude even in the presence of sensor noise and measurement errors. However, it is computationally expensive and requires a lot of memory to store the covariance matrix.

Unscented Kalman Filter (UKF): A more advanced version of the KF that uses a nonlinear transformation of the state variables to generate a set of sigma points which are propagated through the nonlinear system model. The UKF is more accurate than the EKF and requires less computation than others advanced filtering methods.

Particle Filter: A non-parametric filtering method that uses a set of weighted particles to represent the probability distribution of the robot's state. It is capable of handling nonlinear systems and can provide accurate estimates of attitude even in the presence of non-linearities and sensor noise. However, it is computationally expensive.

The development of advanced attitude estimation techniques has made it possible to build autonomous systems capable of operating in complex and dynamic environments, making attitude estimation a vital component of the robotics and autonomous systems field.

In recent decades, numerous MSDF techniques and Deep Learning models have been developed to enhance the accuracy and reliability of attitude estimation techniques. Attitude can be estimated using a minimum of a 6-Degree-of-Freedom (6DoF) Sensor Fusion Algorithm (SFA) in MSDF methods. In 6DoF SFAs, a three-axis accelerometer is fused with a three-axis gyroscope to estimate the altitude. However, 6DoF SFAs are not suitable for attitude and heading estimation/determination as the accelerometer cannot measure the yaw (heading) angle, and the gyroscope can only measure the yaw angle's rate Ding (2022). An alternative method is to fuse magnetometer readings with a 6DoF SFA to estimate the full orientation (attitude and heading). However, a magnetometer's primary disadvantage is the magnetic disturbances, which adversely affect its performance, mainly when used for indoor navigation. Several techniques have been developed to reduce the effect of magnetic disturbances on the filter performance, such as the Factorized Quaternion Algorithm (FQA) Lee and Low (2012).

Most SFAs are developed and parameterized based on the system's dynamic model, which requires a precise choice of model parameters Fauske et al. (2007). However, no algorithm can handle all types of motions, and it is challenging to design a generalizable algorithm for all possible scenarios. The traditional approach is to develop a model for a specific scenario and then adapt it to different situations. However, this approach is time-consuming and requires extensive domain knowledge.

In recent years, deep learning models have shown great potential in solving sequential data problems, including attitude estimation Asgharpoor Golroudbari (2022), signal de-noising Brossard et al. (2020). These models can learn the hidden patterns and relationships within the data and handle complex and nonlinear relationships between sensor measurements and attitude. Several studies have shown the potential of deep learning models for attitude estimation, including recurrent neural networks (RNNs) and convolutional neural networks Wang et al. (2020c); Xiao et al. (2018); Zulqarnain et al. (2020); Nevavuori et al. (2020); Bouktif et al. (2019).

The field of robotics, many studies conducted focusing on the development of odometry or the fusion of heterogeneous sensors for attitude estimation, including inertial-GPS or inertial-visual fusion, and learning-based frameworks coupled with conventional filtering methods. However, there remains a notable gap in research efforts devoted to end-to-end learning-based inertial attitude estimation and inertial data augmentation techniques.

Although relying solely on inertial data can result in significant drift or biases, the benefits of developing such models in scenarios where visual data is not available or GPS is denied cannot be understated. Therefore, there is a pressing need to design a learning-based model that can achieve end-to-end inertial attitude estimation, thereby providing highly accurate and reliable estimates of a system's orientation based solely on inertial sensor data. Such a model holds great potential for a range of applications, including robotics, navigation, and aerospace engineering.

To train, validate and test any neural network model, a comprehensive and accurate database is required. The performance of the Deep Learning model is directly affected by the quality of data used for its training. Table 13 presents some of the most commonly used IMU datasets.

There are currently a limited number of models that employ end-to-end learning-based approaches for inertial attitude estimation. The first model, RIANN, utilized a GRU-based approach, but lacked sufficient quantitative information Weber et al. (2021). Another study presented an LSTM-based model, but was restricted to a single sampling rate and did not mention publicly available inertial datasets Narkhede et al. (2021). A third study utilized a self-attention mechanism, but its model was trained and tested solely on the OxIOD dataset, limiting its applicability to specific motion types and sampling rates Brotchie et al. (2022). Asgharpoor et al. introduced three end-to-end learning frameworks that aimed to generalize across different environments and sensor sampling rates, although the use of IMU data windows in these frameworks may introduce delays Asgharpoor Golroudbari and Sabour (2023).

In the field of attitude estimation, the fully connected attitude detection network (FADN) has been developed as a comprehensive solution for estimating 3D attitude angles. FADN combines neural networks with traditional algorithms to achieve fast and accurate estimation Yang et al. (2020). The network utilizes deep learning to automatically learn the relationship between 2D images and 3D attitude angles, thereby reducing the manual workload involved in monocular visual estimation. A specialized dataset, consisting of one million training samples and 1,000 test samples, was constructed using rendering software to facilitate accurate attitude angle estimation in real-world scenarios. Experimental results demonstrate the high estimation accuracy and fast running speed of FADN. In Sun et al. (2021b),

TABLE 13 IMU Datasets

Dataset	Characteristics	Advantages	Disadvantages
RepoIMU T-stick Szczęsna et al. (2016)	- 9-axis IMU - Acc, Gyro, Mag - 100 Hz - 90 seconds per trial - 29 Trials	- Provides data for versatile applications.	- Some experiments may have clipping issues - Requiring preprocessing and exclusion of trials for accurate evaluation.
RepoIMU T-pendulum Szczęsna et al. (2016)	- Acc and Gyro - 90 Hz or 166 Hz	- Data from a triple pendulum with IMUs mounted.	- Not recommended for model training and evaluation due to duplicate samples and low effective sampling rate.
Sassari Caruso et al. (2020)	- 6 IMUs - Acc, Gyro, Mag - 100 Hz - 168 seconds - 18 Trials	- Provides data for parameter tuning approach based on orientation difference.	- Lacks robust variation in movement type and magnetic data diversity. - Short duration limits its use for training.
OxIOD Chen et al. (2018b)	- Acc, Gyro, and Mag - 100 Hz - 158 Trials	- Comprehensive collection of inertial data for evaluating inertial odometry models.	- Lack of comprehensive descriptions of certain essential parameters. - Irregularities in ground trace orientation.
MAV Dataset Lee et al. (2010)	- Camera - Acc, Gyro, and Mag - 5 Trials	- Data collected from a sensor array on a quadrotor platform for simultaneous localization and mapping.	- Short. - Lack of diversity.
EuRoC MAV Burri et al. (2016)	- Visual, Acc and Gyro - 6 Trials	- Large dataset for reconstructing visual-inertial 3D environment.	- Does not include magnetometer data. - Limited variety for model training.
TUM-VI Schubert et al. (2018)	- Camera - IMU - Acc and Gyro - 28 Trials	- Suitable for optical-inertial odometry. - Six-chamber experiments provide complete OMC data.	- Most experiments lack complete OMC data. - Does not include magnetometer data.
KITTI Geiger et al. (2012)	- Stereo camera and LiDAR - Acc and Gyro - 11 Trials (with ground truth)	- Large dataset for evaluating accuracy in the presence of optical flow.	- Does not include magnetometer data. - Limited IMU data and ground truth
RIDI Yan et al. (2018)	- Acc, Gyro, and Mag - +100 minutes - 98 Trials	- Provides ground-truth motion data for different placements of IMUs on human subjects.	- Ground truth error
RoNIN Herath et al. (2020)	- Acc, Gyro, and Mag - 100 Hz - 100 Trials - +40 h	- Provides data for evaluating gait analysis algorithms.	- Limited ground truth accuracy
BROAD Laidig et al. (2021)	- Acc, Gyro, and Mag - 39 Trials - 286.3 Hz	- Dataset containing realistic data from multiple sensors - Diverse motion characteristics	- Small gaps in ground truth

Sun et al. introduced a two-stage deep learning framework for inertial odometry based on a Long-Short-Term Memory and Feed-Forward Neural Network architecture. The first stage estimates the orientation, while the second stage estimates the position.

In the context of space missions, determining the attitude of non-cooperative targets presents a significant challenge. To address this, the article proposes a novel approach that combines ground and space access (GSA) scene radar images with deep learning techniques for attitude estimation Hou et al. (2023). By controlling the intersection of orbital planes between the target and observation satellites, collinearity with the target's position vector is ensured during radar accessibility, resulting in a series of GSA scenes. High-precision attitude values obtained from space-based optical images serve as training data for inverse synthetic-aperture radar (ISAR) images. The paper introduces an instantaneous attitude estimation method based on a deep network, which achieves robust estimation under varying signal-to-noise ratio (SNR) conditions. Validation using electromagnetic simulation data demonstrates an average attitude estimation error of less than 1°.

The paper by Kawai et al. Kawai and Kuroda (2023) introduces a camera attitude estimation network that improves the accuracy of attitude estimation in unknown environments. Unlike previous methods that focus on known environments, this network incorporates both camera and depth images to extract landscape and terrain information.

By leveraging cross-reference information and a classification-type output layer, the proposed network enhances attitude estimation accuracy in unknown environments.

In a different study, Ozaki et al. Ozaki et al. (2021) propose an extended Kalman filter (EKF) based self-attitude estimation method that combines a LiDAR deep neural network with gyroscope angular rates. The DNN learns landscape regularities from LiDAR data and estimates gravity direction. To overcome limitations of previous methods, the authors integrate gyroscope data and DNN outputs within the EKF framework, resulting in improved real-time estimation. Although the proposed method exhibits promising results, further work is required to address different environments and LiDAR data distortion.

Moving on to ship motion attitude prediction, Zhang et al. Zhang et al. (2023a) present a model called IWOA-TCN-Attention. This model utilizes an improved whale optimization algorithm (IWOA), temporal convolutional networks, and an attention mechanism to address the non-stationary, nonlinear, and stochastic characteristics of ship motion. By capturing long-term memory with TCN and assigning different weights to features using the attention mechanism, the IWOA-TCN-Attention model achieves higher prediction accuracy compared to other models. The proposed model holds potential for various time series prediction applications beyond ship motion.

In the domain of orientation estimation, Hu et al. Hu et al. (2021a) propose a deep reinforcement learning (DRL) algorithm that combines inertial sensors with a magnetometer. The algorithm utilizes Lyapunov's method from control theory to ensure convergence of orientation estimation errors. By parameterizing the estimator gains and a Lyapunov function through deep neural networks, the DRL estimator outperforms traditional methods in terms of accuracy. The algorithm's versatility and adaptability to challenging scenarios make it a significant advancement in orientation estimation.

Riden et al. Riden et al. (2021) introduce DeePOE, a deep learning framework for joint position and orientation estimation. DeePOE utilizes the in-phase (I) and quadrature-phase (Q) components of RF signal data and employs a convolutional neural network (CNN) to predict the position and orientation of a transmitter. Through transfer learning and comprehensive real-world experiments, DeePOE achieves impressive results, surpassing existing methods in accuracy. This research showcases the potential of deep learning in RF-based localization applications.

In the domain of attitude estimation, Ozaki et al. Ozaki and Kuroda (2021) propose a DNN-based approach for self-attitude estimation by learning landscape information. Their method predicts the gravity vector in the camera frame using a deep neural network trained on a dataset collected in a simulator. Unlike previous works, this study incorporates a larger dataset, reliable ground truth, L2 normalization, and outputs both mean and covariance. Validation results demonstrate the potential application of the proposed network in filtering out samples with large errors and integrating with Kalman filters. Future directions involve testing the method in different environments and applying it to real data, enhancing its adaptability and practicality.

For determining the in-orbit attitude of satellites without an active attitude determination system, Finance et al. Finance et al. (2021) present a new method using deep learning. They specifically developed the method for the UVSQ-SAT nanosatellite, aiming to monitor Earth Energy Imbalance (EEI) from space. The authors propose using housekeeping and scientific sensor measurements of the satellite, along with the deep learning algorithm, to estimate the satellite's altitude accurately. Ground-based calibration experiments validate the method's accuracy, showcasing its potential application to other satellites without active attitude determination systems and contributing to accurate EEI monitoring and climate change studies.

To address the challenge of accurately estimating the 3D position and orientation of users in indoor LiFi systems, Arfaoui et al. Arfaoui et al. (2021) propose two deep artificial neural network models based on multilayer perceptron (MLP) and convolutional neural network architectures. These models map the received signal-to-noise ratio (SNR) to the user's 3D position and orientation. Comparative evaluations against the widely used k-nearest neighbors (KNN) scheme demonstrate the significant gains and superiority of the proposed models in terms of estimation accuracy, precision, computational time, and bit error rate (BER). The models achieve centimeter-level positioning accuracy in real-time, with the inclusion of non-line-of-sight (NLOS) links further improving performance.

In the context of accurate extraction of building characteristics in high-resolution satellite images, Shahin et al. Shahin and Almotairi (2021) propose a novel approach for estimating the building orientation angle. Their proposed deep convolutional regression network (DCRN) architecture outperforms existing methods, achieving the lowest root mean square error (RMSE) and mean absolute error (MAE) values, as well as the highest adjusted R-squared value. The proposed approach exhibits stability in varying input building image contrast, demonstrating its superiority over traditional computer vision techniques and existing deep learning methods. This research presents a promising solution for building orientation angle estimation, with potential implications for urban planning and decision-making.

In the realm of cyber-attacks and robotic applications, Chumuang et al. Chumuang et al. (2022) introduce novel neural network models that leverage deep learning techniques to tackle these challenges. They utilize feature matching algorithms and deep learning models, including CNN and Recurrent Neural Network (RNN), to identify consistent features and predict quaternions for Micro Aerial Vehicles (MAVs). Comparative evaluations demonstrate the effectiveness of the proposed models in resolving hand self-occlusion challenges and achieving accurate 3D hand attitude estimation. This method holds promise for applications in augmented reality, virtual reality, and human-computer interaction, particularly in scenarios with limited degrees of freedom.

For 3D hand attitude estimation using dual-view RGB images, Ji et al. Ji et al. (2022) propose a method combining deep learning and ensemble learning. The approach involves constructing a 3D hand posture dataset using dual-view cameras and an IMU sensor, training CNN-based image feature extractors, and performing attitude regression using LightGBM. Experimental results demonstrate the method's effectiveness in resolving hand self-occlusion challenges and achieving accurate 3D hand attitude estimation with only two normal RGB cameras. This approach offers adaptability, ease of use, and minimal hardware requirements, making it a promising solution in the field.

Herrera et al. Herrera et al. (2023) introduce a deep learning based method for object attitude estimation in a Laser Beam Control Research Testbed (LBCRT). Their DL model, based on a modified Resnet 18 architecture using transfer learning, addresses the limited data challenge in DL applications. The experimental approach and validation results indicate the potential of DL-based object attitude estimation for laser beam control systems. Further generalization and cross-validation with a more representative dataset are recommended for future research, along with exploring multi-object attitude estimation and considering optical turbulence for improved model robustness.

Fu et al. Fu et al. (2023) present a novel hybrid deep learning approach, GCN-LSTM, for predicting the dynamic attitude and position of tunnel boring machines (TBMs) in tunnel construction. The proposed approach incorporates graph convolutional network (GCN) and long short-term memory (LSTM) networks to capture the spatial and temporal relationships among TBM parameters. The results demonstrate high prediction accuracy, outperforming state-of-the-art methods in most evaluation metrics. The research provides insights for improving TBM trajectory control and reducing mis-alignments in tunnel construction projects.

Zhou et al. Zhou et al. (2022a) propose the U-Linked Network (ULNet) for instantaneous attitude estimation from a single camera image. By utilizing the prior structural constraints of key points, the ULNet captures the relationship between 3D target attitude parameters and 2D images. Simulation results demonstrate the feasibility of this approach, showcasing its potential for real-time attitude estimation in on-orbit spacecraft with limited observation periods.

Kawai et al. Kawai et al. (2022) address the challenge of accurate camera pose estimation in scenarios with high maneuverability and uneven ground surfaces. They introduce a novel approach that leverages a classification neural network inspired by optical character recognition (OCR) techniques for robust camera pose estimation. The proposed method ensures high inference accuracy in pose estimation tasks, enhancing the practicality and accuracy of camera-based pose estimation for robots operating in complex environments.

Finally, Zaheer et al. Zaheer et al. (2023) propose an advanced solution for head orientation estimation in virtual and augmented reality systems. Their solution utilizes an instrumented helmet equipped with integrated inertial sensors and a magnetometer. Experimental evaluation demonstrates the superiority of the proposed CNN-based solution compared to existing methods, particularly in scenarios with high acceleration disturbance. The research emphasizes the need for accurate head orientation estimation in VR and AR applications and suggests future research directions for improved robustness and estimation performance.

The studies mentioned above make contributions to diverse domains, encompassing attitude estimation in various contexts, pose estimation, hand attitude estimation, and head orientation estimation. Through the application of deep learning techniques, these investigations introduce compelling solutions to address intricate challenges, thereby propelling advancements within their respective fields. The utilization of deep neural networks enables the accurate estimation of attitudes, positions, and orientations in different scenarios. These approaches demonstrate promising results, outperforming traditional methods and showcasing their potential for practical implementation. The integration of deep learning models with sensor data, such as IMU measurements, camera images, or LiFi signals, enhances the precision and robustness of the estimations. Moreover, the proposed methodologies exhibit adaptability, efficiency, and ease of implementation, making them valuable tools for applications in fields such as satellite technology, robotics, virtual and augmented reality, tunnel construction, and urban planning.

Indeed, there exists a substantial gap in the utilization of learning-based inertial attitude estimation techniques. Despite the advancements and successful application of deep learning in various domains, the integration of deep learning algorithms with inertial sensors for attitude estimation remains relatively underexplored. Conventional inertial attitude estimation methods heavily rely on mathematical models and signal processing techniques, which often pose challenges in their adoption and generalization across different scenarios. These traditional approaches require careful parameter tuning and may struggle to capture complex and nonlinear relationships present in real-world situations.

In contrast, deep learning algorithms can adapt and generalize well to diverse environments and conditions, making them highly suitable for real-world applications. But there is a need to develop new algorithms for inertial data augmentation methods which can directly affect the aforementioned aspects. The integration of deep learning with inertial sensors has the potential to revolutionize attitude estimation by improving accuracy, reliability, and adaptability. These learning-based approaches can effectively capture complex dynamics, compensate for sensor biases, and handle challenging scenarios such as occlusions, variable motion patterns, and uncertain environmental conditions. Furthermore, it can enable the fusion of multiple sensor modalities, such as combining inertial data with visual or LiDAR inputs, leading to enhanced estimation performance.

Despite these advances, there are still many challenges to overcome:

1. Difficulty in directly observing the attitude of a system, which requires estimation based on sensor measurements.
2. Poor generalization over different motion patterns and environmental disturbances.
3. Poor generalization over different sampling rates.
4. Inaccuracies associated with measurements of commercial-off-the-shelf IMUs, such as large drifts, biases, and unpredictable noise sequences.
5. Difficulty in choosing the appropriate deep learning topology for real-time attitude estimation, as it must be able to be used in real-time.
6. Need for a well-chosen model and proper loss function for orientation increments to achieve accurate attitude estimation.
7. Need for a training procedure to determine accelerometer weights in each axis, which can be time-consuming.
8. Difficulty in coping with the acceleration resulting from user or smartphone dynamics when estimating attitude for wearable devices and smartphones.
9. Difficulty in identifying key points when training with high-frequency inertial data.
10. Need to optimize filter parameters for each situation, which can limit filter performance.
11. Utilizing data augmentation techniques is an advanced approach in IMU-based attitude estimation. By generating synthetic data or augmenting existing datasets, the diversity and quantity of training samples can be increased. This improves the generalization and robustness of deep learning models, allowing them to adapt to different motion patterns, environmental conditions, and sensor characteristics. As a result, the accuracy and reliability of attitude estimation from IMU data are enhanced.

Addressing these challenges will contribute to improving the reliability, accuracy, and applicability of deep learning-based attitude estimation techniques in various domains, including robotics, aerospace, and wearable devices.

However, with increased attention and research focus on this area, it is expected that learning-based inertial attitude estimation will bridge the existing gap and pave the way for more accurate, reliable, and versatile attitude estimation techniques in various applications such as robotics, aerospace, navigation systems, and virtual reality.

4.3.3 | Odometry

Odometry is the process to determine the relative change in the position and orientation between two or more coordinate frames Aqel et al. (2016). There are various odometry techniques, including visual Nistér et al. (2004), inertial Solin et al. (2018), encoder Brossard and Bonnabel (2019), optical flow Muller and Savakis (2017), GPS/GNSS Ohno et al. (2004), and lots of others.

The most common odometry techniques are as follows Chen et al. (2020b):

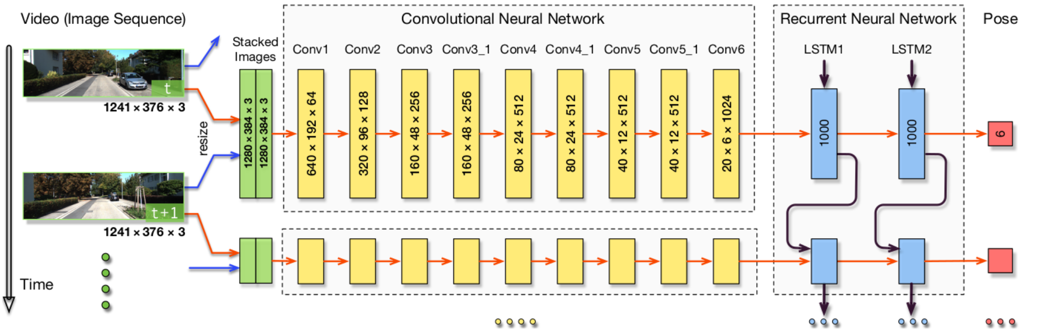

FIGURE 7 DeepVO, A typical structure of an End-to-End Visual Odometry Wang et al. (2017)

- **Inertial Odometry**: Relies on inertial data, such as accelerometers and gyroscopes, to estimate the position and orientation. It utilizes the principles of dead reckoning to track motion based on sensor readings. However, it is prone to accumulation of errors over time.
- **Visual Odometry**: Utilizes computer vision techniques to estimate the position and orientation of a camera relative to a fixed reference frame. By analyzing the changes in visual data captured by the camera, it can track the camera's motion.
- **Visual-Inertial Odometry**: Combines visual and inertial data to estimate pose and trajectory. It involves integrating measurements from an inertial measurement unit with visual data from a camera to reconstruct the 3D scene structure and determine the camera's motion within it. The visual data is captured from the camera's perspective, while the IMU data is referenced to the global coordinate system.

Further, learning-based visual odometry techniques can be broadly divided into two distinct categories:

- **Supervised**: The most popular VO techniques and used when the ground truth data is available which maps the extracted features from image to motion transformation Han et al. (2019).
- **Unsupervised**: The model is trained on unlabeled data, without the need for explicit ground truth information. It can be used to extract meaningful features or representations from visual data when ground truth labels are not available or too expensive to obtain.

One of the earliest works on the supervised VO, conducted by Konda et al. Konda and Memisevic (2015). They utilized CNN to predict velocity and changes in direction. DeepVO Wang et al. (2017) combined CNN and RNN to develop an End-to-End deep learning framework for pose estimation. Authors used deep neural networks to extract the features from images via CNN and learn the temporal dependencies through RNNs. They used the KITTI dataset Geiger et al. (2013) to train and test the model. To calculate the error between the prediction and true position and orientation they used the Mean Square Error (MSE) of all positions $\mathbf{p}$ and orientations ϕ in the sequence. The MSE of the position is defined as:

$$\theta^\star = \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\|\mathbf{p}_t^\star - \mathbf{p}_t\|_2^2 + \|\phi_t^\star - \phi_t\|_2^2 \tag{1}$$

where $\mathbf{p}_t^\star$ and $\phi_t^\star$ are the predicted position and orientation at time t, $\mathbf{p}_t$ and ϕ_t are the true position and orientation at time-step t, and N is the number of samples. Based on this architecture, various models have been developed since. Similarly, MagicVO Jiao et al. (2019) combined CNN and Bi-LSTM to 6 DoF pose estimation. They replaced LSTM layers with Bi-LSTM layers followed by a 256 unit dense layer. The results show an improvement in terms of

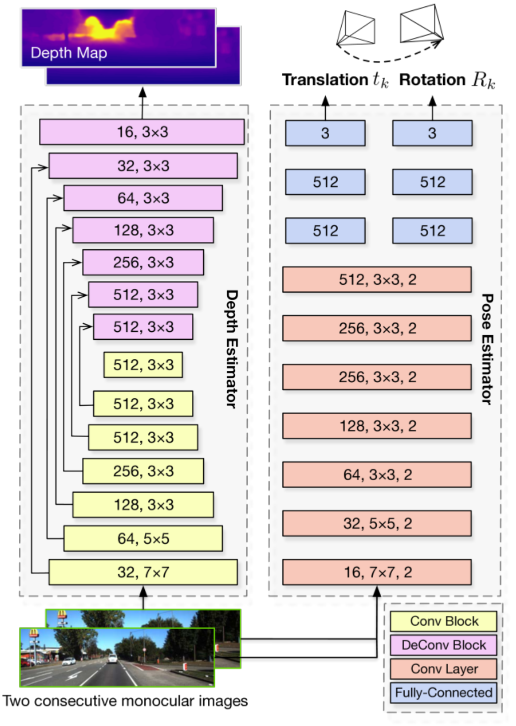

FIGURE 8 UnDeepVO Architecture for Unsupervised End-to-End Visual Odometry Li et al. (2018)

translational and rotational error. However, their comparison results consist of only five sequences.

In contrast with DeepVO, UnDeepVO Li et al. (2018), employed unsupervised techniques to develop an end-to-end VO framework. They used VGG-based CNN Simonyan and Zisserman (2014) as the pose estimator, and it trained using stereo images instead of consecutive monocular images

The monocular visual odometry algorithm UnDeepVO, proposed by Li et al. Li et al. (2018), offers an alternative approach to DeepVO. Unlike DeepVO, UnDeepVO is an end-to-end framework for visual odometry that employs unsupervised learning to estimate the camera's motion. It uses a VGG-based CNN Simonyan and Zisserman (2014) as the pose estimator, which is trained on stereo images, rather than consecutive monocular images (Figure 8). The authors use unsupervised learning to train the CNN, incorporating both spatial and temporal dense information in the loss function to improve accuracy and robustness.

By incorporating unsupervised learning and stereo images, UnDeepVO can estimate both pose and depth using a deep neural network. Compared to other monocular methods such as ORB-SLAM without loop closure, this model has been shown to be more accurate and robust. Overall, UnDeepVO provides a promising approach to visual odometry that does not rely on labeled data and can achieve comparable or superior performance to other monocular methods. Its use of unsupervised learning and stereo images provides a foundation for future research in this area.

GANVO Almalioglu et al. (2019) used Generative Adversarial Network for unsupervised learning to estimate pose and generate depth map for a monocular video sequences. Their result shows that the model outperforms conventional and other unsupervised methods. An end-to-end, Sequence-to-sequence Probabilistic Visual Odometry (ESP-VO) deep learning framework is presented by Wang et al. Wang et al. (2018) which used monocular camera and Recurrent Convolutional Neural Network (RCNN) for pose estimation directly from raw images.

VINet Clark et al. (2017) represents one of the pioneering End-to-End Visual-Inertial Odometry (VIO) deep learning models that addresses the odometry problem as a sequential learning task. Comprising a CNN and LSTM network, VINet leverages both monocular camera images and IMU measurements for pose estimation. The CNN processes pairs of sequential images to extract motion-related features, while the LSTM network captures temporal motion characteristics from the IMU sensor data. The fused features from both networks are input to a core LSTM network, enabling the prediction of pose information. The robustness of VINet is enhanced by the integration of visual and inertial data. This model was trained and evaluated on the KITTI dataset.

Similarly, HVIOnet Aslan et al. (2022) proposes a hybrid CNN-BiLSTM network specifically designed for Unmanned Aerial Systems. It incorporates multiple fully connected layers following the CNN and LSTM networks. HVIOnet was trained on the EuRoC dataset and evaluated in the ROS environment. In contrast to VINet, which used both visual and inertial data, HVIOnet focuses solely on visual information for pose estimation.

Yang et al. Yang et al. (2022) present an adaptive visual-inertial odometry system that intelligently deactivates the visual feature extraction process based on the motion state and IMU readings. By selectively deactivating the visual modality, the proposed method reduces computational redundancy and achieves significant computational complexity reduction of up to 78.8% for the KITTI dataset evaluation. The authors argue that visual data processing is more resource-intensive than inertial data processing and may not always contribute to improved pose estimation accuracy. The approach outperforms simple sub-optimal strategies and provides an interpretable, scenario-dependent adaptive behavior that can be easily implemented in other deep VIO systems.

In another study George et al. (2023), the feasibility of using VIO as a redundant positioning system for high-altitude drone operations was investigated. The collected data from this research has been published in George et al. (2022). The study reveals that stereo-VIO, with a baseline of 30 cm, outperforms mono-VIO for altitudes ranging from 40 to 100 meters. Additionally, the study highlights the impact of external conditions on estimation accuracy and demonstrates that fusing IMU data improves positioning performance and system robustness against external conditions. The authors suggest future research should focus on developing real-time estimation systems, integrating additional sensors to mitigate drift, evaluating the superiority of stereo-VIO over mono-VIO algorithms, and exploring the effects of system calibration models.

Tu et al. propose an innovative EMA-VIO framework Tu et al. (2022) that leverages external memory and attention-based fusion mechanisms for accurate position and translation estimation using visual-inertial data. The framework exhibits superior performance in various scenarios, including challenging ones like overcast days and water-filled ground, surpassing both traditional and learning-based VIO baselines. The introduction of a multi-state geometric constraint in the pose loss function enhances training convergence and yields more precise pose results. Experimental results on the KITTI dataset and the authors' own robot dataset validate the effectiveness of the proposed framework, particularly in challenging environments where traditional VIO methods struggle to extract and match visual features. EMA-VIO demonstrates superior performance compared to both traditional VIO models and other learning-based VIO models.

SelfVIO Almalioglu et al. (2022) is a deep learning-based monocular Visual-Inertial Odometry (VIO) system that utilizes self-supervised learning to estimate camera pose (position and orientation) and object depth in the environment. Unlike stereo or multi-camera systems, SelfVIO relies on a single camera and incorporates both visual information and IMU data for pose and depth estimation. Trained on unlabeled data, SelfVIO learns to predict camera motion and object depth based on its own observations.

Authors in Cortés et al. (2018) used a CNN-FFNN to estimate speed by utilizing IMU raw measurements over a 2-second window. IONet Chen et al. (2018a) is one of the first Inertial End-to-End deep learning networks for odometry. It consists of two stacked Bi-LSTM layers, each with 96 units. The first layer takes accelerometer and gyroscope readings in windows of 200 frames and a stride of 10 as input, and the second layer takes the output of the first layer as input. The output is the polar vector as the relative position and orientation of the sensor.

$$(\mathbf{a}, \omega)_{200,6} \rightarrow (\delta \mathbf{p}, \delta \phi)_1 \tag{2}$$

In Chen et al. (2019b), the authors used deep learning to estimate displacement in polar coordinates and the corresponding uncertainty over a window of 200 frames by utilizing IMU raw data. The model input shape is $(200, 6)$, where 6 is the number of features (accelerometer and gyroscope readings), and 200 is the window size. The input is fed into a fully connected layer with 128 units, followed by 128 and 256 units Bi-LSTM layers. The output of the last Bi-LSTM is fed into a 128-unit fully connected layer, followed by a double 2-unit fully connected layer to predict displacement in polar coordinates and uncertainty.

Lima et al. Silva do Monte Lima et al. (2019) proposed a deep inertial odometry method that utilizes a combination of CNN and bidirectional long short-term memory (BiLSTM) layers. The model takes two distinct inputs of shape $(200, 3)$, representing the window size of 200 and the number of axes. The CNN extracts features from the inertial measurement unit (IMU) sensor readings, which are then passed to two BiLSTM layers. The output of the final BiLSTM layer is fed into a fully connected layer with seven units, representing the orientation in quaternion form and position in metric form. Additionally, a multi-task learning framework is integrated to balance the pose distance metrics. The multi-task learning equation is as follows:

$$\mathbf{L}_{MTL} = \Sigma_{i=1}^{N} \exp(-\log \sigma_i^2) L_i + \log \sigma_i^2 \tag{3}$$

where σ_i and L_i represent the variance and loss function of task i, respectively. The multi-task learning framework is implemented by training the model to predict both the device's translation and rotation. The loss function is a weighted sum of the translation and rotation error, and the weights of the two tasks are dynamically adjusted during training to balance the contributions of the translation and rotation errors. The additional output layer predicts the rotation quaternion between two sequential moments. The loss function for the multi-task learning is a combination of the mean absolute error of the predicted translation and rotation, weighted by two coefficients that control the relative importance of the two tasks. The coefficients are updated during training using a technique called dynamic weighting, which balances the contributions of the two tasks based on their performance.

The multi-task learning approach enables the model to balance the importance of rotation and translation, leading to improved accuracy in estimating the device's position and orientation. The proposed method outperforms recent inertial odometry methods in both qualitative and quantitative evaluations.

However, Lima et al.'s method has a primary constraint in its generalizability. This model demonstrates optimal performance solely on the dataset on which it was trained. Furthermore, due to the model's inability to handle variations in gyroscope sampling rates, it is incapable of accurately estimating the position and orientation of the sensor when exposed to diverse sampling rates Asgharpoor Golroudbari (2022).

Another deep inertial odometry model, called Tightly-coupled Extended Kalman Filter framework (TILO), was developed by Liu et al. Liu et al. (2020). This model uses IMU measurements to estimate relative position and orientation by using a deep neural network coupled by an EKF. Two Bidirectional-LSTM is used to estimate the foot trajectory in gait analysis by using the accelerometer and gyroscope readings with a window size of 256 Guimarães et al. (2021). In Soyer et al. (2021), authors presented a real-time pedestrian position estimation based on smartphone internal IMU measurements. They used a window size of 150, where the first 100 frames are selected from the past measurements and the rest of 50 frames are selected from the future measurements. This method helps them reduce the latency of the system compared to other methods such as IONet. The model is similar to Lima's network, consisting of two distinct inputs (accelerometer and gyroscope), each input followed by two CNN layers and a max pooling layer concatenated and fed into two-Bidirectional-LSTM layers.

Uno et al. Uno et al. (2022) introduced Deep Inertial Underwater Odometry (DIUO) system for underwater 6-DOF odometry based on TLIO network. Convolutional block attention modules are combined with Res2Net in Chen et al. (2022a) to estimate the velocity in x and y axis. They test their model on the RONIN and OxIOD dataset, and the results show that the proposed model outperforms the other models in most scenarios.

A LSTM-based model for visual-inertial odometry has been proposed for UAVs in GNSS denied environments by Deraz et al. Deraz et al. (2022). Wang et al. introduced A2DIO model, which used a hybrid CNN-LSTM architecture with a temporal attention block for pedestrian odometry based on a window of 2-second IMU measurements Wang et al. (2022e).

One of the major drawbacks of using GNSS and GPS for navigation is their weakness against signal obstruction, such as tall buildings or natural obstacles like mountains. To address the lack of reliable indoor positioning alternatives to GNSS/INS, authors presented L5IN+ Shoushtari et al. (2022). The proposed methods consist of two approaches, namely deep inertial odometry and Kalman Filtering, to predict velocity vector elements and relative positions with noisy labeled data from 5G networks. The authors also develop an analytical platform for data collection and a simulation website for researchers to generate ground truth trajectories and simulate cellular measurements with assigned quality and exact error values. The major contribution of this paper is the development of a novel combination of DNN and KF for the relative and absolute positioning. The DNN model is designed to be robust to noisy labeled data, for example, data from a 5G network. The authors also introduce a semantic error generation approach to their simulation model.

The errors and time-varying biases present in the measurements of MEMS-based IMUs cause large drift in pose estimates, and only relying on the integration of the inertial measurements for state estimation is infeasible. In Cioffi et al. (2022), authors proposed a learning-based odometry algorithm for autonomous drone racing tasks that uses an IMU as the only sensor modality. The proposed algorithm aims to overcome this problem by combining an EKF, which is driven by the IMU measurements, with a learning-based module, which has access to the control commands, in the form of commanded collective thrust. The learning-based module is a TCN that takes as input a buffer of commanded collective thrust and gyroscope measurements and outputs an estimate of the distance traveled by the quadrotor, which is used to update the filter. The proposed algorithm is compared to state-of-the-art filter-based and optimization-based visual-inertial odometry algorithms, as well as the state-of-the-art learned-inertial odometry algorithm TLIO, and achieves superior performance.

Another study Zhang et al. (2022b) conducted on improving quadrotor state estimation. The proposed framework, Dido, employs GRU-based networks to learn both IMU and dynamic properties for better performance in quadrotor state estimation. The paper demonstrates that the system provides precise observations with only raw IMU and tachometer data and combines quadrotor dynamic constraints and network observations within a two-stage EKF to jointly estimate kinematic, dynamic states, and extrinsic parameters.

Legged robots offer advantages over wheeled robots in traversing uneven terrain, but controlling and navigating them is challenging due to the complex dynamics of legged locomotion. Navigation and odometry are crucial, involving determining the robot's position and movement through sensor data analysis. Legged robots commonly use multiple sensors such as cameras, IMUs, and rangefinders to improve accuracy and reliability. Ongoing research focuses on the development of advanced algorithms and sensors to enhance legged robot navigation and odometry. Legged robots have the potential to play a vital role in exploring remote and hazardous environments, disaster response, and search and rescue operations.

To address the challenges in legged robots navigation, Buchanan et al. Buchanan et al. (2022) proposed a new method to improve proprioceptive state estimation. The approach is based on using a learned displacement measurement from IMU readings, which reduces the drift in state estimation caused by unreliable leg odometry on challenging terrains such as slipping and compressible surfaces. The displacement measurement from IMU readings is combined with traditional leg odometry, and results from real robot experiments show a reduction in relative pose error by 37% in challenging scenarios and a 22% reduction in error when used with vision systems in visually degraded environments. The article's contributions include a kinematics-inertial estimator based on learned IMU displacement measurements, integration with both a filter-based estimator and a factor graph, and extensive testing on a quadruped robot, including field experiments navigating a mine. The study concludes that the addition of the learned measurement significantly improved state estimation compared to a traditional kinematic-inertial estimator without the learned measurement.

The heading/yaw angle could be calculated by the magnetometer; however, this method suffers from magnetic interference, drift, and bias. The authors of Wang et al. (2022f) proposed a system to estimate the magnetometer bias online to provide a globally consistent heading estimation in magnetic disturbances. They used a graph-optimization-based system that fused learning-based inertial odometry and magnetometer readings. The proposed system showed a 50% improvement in positioning accuracy in an indoor scenario. They simplified the model to perform a real-time estimation and compensation procedure and achieved superior positioning performance when compared to calibrated magnetometer data and raw magnetometer data. The proposed method is suitable for consumer-grade devices as it does not require calibration before use. The study combined network and traditional empirical models to form a robust pedestrian positioning solution. The testing stage fuses the information from IMU, the pre-trained network, and magnetometer to provide a trajectory without heading drift and is not affected by local magnetic field disturbance. The system uses the long-term mean value of the geomagnetic field to obtain a consistent global heading direction and stores magnetic field observations for a sufficiently long time or keyframes to estimate magnetometer biases.

The research conducted by Saha and colleagues Saha et al. (2022) proposes a new framework for neural inertial navigation that addresses the challenge of deploying machine learning models on ultra-resource-constrained

(URC) devices in GPS-denied environments. The framework, named TinyOdom, utilizes a hardware-in-the-loop (HIL) automated machine learning (AutoML) framework based on Bayesian Optimization (BO) to create lightweight inertial odometry models that are suitable for deployment on URC devices. The framework incorporates hardware and quantization-aware Bayesian neural architecture search (NAS) and a TCN backbone to develop lightweight models. Moreover, the proposed framework incorporates a magnetometer, physics, and velocity-centric sequence learning formulation to enhance the localization performance of the lightweight models, along with a barometric g-h filter that tracks altitude with high accuracy and is resistant to disturbances and dynamics. The authors evaluated their framework on four different URC hardware platforms for four different applications, and the results show that it is an effective and efficient solution for neural inertial navigation on URC devices. They also suggest that their work has the potential to enhance the cost and energy footprint of embedded odometry while being adaptable to any reduced footprint hardware.

Over the years, the utilization of deep learning techniques for odometry has garnered significant interest and has witnessed notable advancements. However, one common limitation observed in many deep learning-based odometry approaches is the inherent delay associated with their sequential nature and the requirement of a window frame of data.

Sequential processing often introduces a time lag in estimating the device's motion which arises from the need to process data in a window frame. Consequently, the estimation of position and orientation is not instantaneous and may fall behind real-time data acquisition. This delay can be problematic in scenarios where real-time, on-board odometry is crucial. Many deep learning odometry techniques rely on capturing a sequence of sensor measurements over a fixed time window. While this approach can provide accurate results, it introduces a temporal dependency that requires a certain amount of historical data to be collected before a reliable estimation can be made.

To address these limitations, there is a growing need for real-time, on-board odometry methods that can leverage instantaneous measurements or require only a short window frame. Such methods would enable immediate estimation of position and orientation based on the current sensor data, eliminating the delay associated with sequential processing or the need for extensive historical data.

Developing real-time deep learning-based odometry techniques poses unique challenges. It requires designing models that can efficiently process and analyze sensor data in a continuous and time-sensitive manner which should be capable of handling the complexities of real-time data streams while maintaining accuracy and reliability. Additionally, considerations should be given to optimizing computational efficiency to ensure that the real-time odometry can be performed within the constraints of onboard computational resources.

In conclusion, while deep learning-based odometry techniques have made remarkable progress, the demand for real-time, on-board methods that can estimate position and orientation instantaneously or with minimal delay remains a pressing challenge. Advancements in model architectures, data processing techniques, and computational efficiency are paving the way for the development of real-time deep learning odometry approaches, which hold great promise for a wide range of applications requiring accurate and timely motion estimation.

4.4 | Path Planning

Path planning is an approach to solve the planning problems by finding a set of actions to reach the destination point Otte (2015). The aim of this task is to guide a moving object from a starting point to a destination while avoiding collisions with static or dynamic obstacles. Path planning algorithms are widely used in various domains such as robotics, autonomous vehicles, and computer graphics to find optimal or feasible paths Basiri et al. (2022).

These algorithms employ different strategies and techniques to navigate through the search space efficiently, which can be categorized based on their "Category" and "Approach". In the context of path planning algorithms, these terms refer to different aspects of classifying and describing the algorithms. In summary, Category provides a broad classification based on properties exhibited by the algorithms, while Approach describes the specific technique or methodology employed by the algorithm to solve the path planning problem. The distinction between category and approach lies in their levels of classification and the aspects they focus on.

Path planning methods can be generally divided into these classes:

- **Global Path Planning**: Assumes complete knowledge of the environment at the beginning.

- **Local Path Planning**: Obtains the majority of information during the process.
- **Multi-Robot Path Planning**: Planning paths for multiple robots to achieve a common goal.

Some the most popular approaches used for path planning task are as follows:

- **Particle Swarming Optimization (PSO)**: A population-based optimization technique that simulates the behavior of a swarm of particles to find the optimal solution.
- **Ant Colony Optimization (ACO)**:A probabilistic technique inspired by the behavior of ants to find the shortest path between two points.
- **Hybrid PSO and ACO**:A combination of PSO and ACO algorithms to improve the performance of path planning tasks.
- **Machine Learning Methods**: A set of techniques that use statistical models to learn patterns in data and make predictions or decisions without being explicitly programmed.
- **Genetic Algorithm**: A search heuristic that mimics the process of natural selection to find the optimal solution.
- **Greedy Heuristic Approach**: A simple and fast algorithm that makes locally optimal choices at each step to find a suboptimal solution.
- **Multi-Population Algorithm**: An optimization technique that uses multiple populations to explore the search space and find the optimal solution.

The Table 14 covers the most common path planning algorithms.

Many researchers have been discussed to solve the path planning problems in the past decades Lozano-Pérez and Wesley (1979); Thompson (1977); Mac et al. (2016); Ge and Cui (2000); Zhang et al. (2018); Sariff and Buniyamin (2006). They aimed to consume less computation resources and time to generate optimal or sub-optimal path. They have explored its application in domains including, self-deriving cars, autonomous spacecrafts, planetary rovers, indoor navigation, and drones.

For example in Kurdi et al. (2018), authors put forward an algorithm based on neural networks to address the challenges of navigation and positioning in UAVs. They combined GPS signals with visual data and fed them into an ANN model to generate output in the form of UAV localization.

In Dhuheir et al. (2022) to address the challenges of real-time applications involving UAVs, a model developed that combines distributed collaborative inference requests and trajectory path planning in a UAV swarm, while considering resource constraints. The authors highlight the limitations of relying on remote servers for inference and data processing tasks due to connection instability, limited bandwidth, and latency issues. The model is formulated as an optimization problem, which is known to be NP-hard. To tackle this complexity, the authors introduce a real-time and dynamic solution using deep reinforcement learning. Extensive simulations are conducted, comparing the proposed model to state-of-the-art studies, and demonstrating its superior performance. This work contributes to the field by addressing the trajectory planning aspect and latency reduction in UAV systems, considering the distances between devices and their impact on interference.

The article Bian et al. (2022) proposes a CNN-based path-planning algorithm for autonomous vehicles that considers traversability cost. It offers faster and more efficient path planning on large maps or long paths. The method introduces traversability cost, enabling calculation of the lowest-cost path while considering costs and the shortest path without considering costs. The FCNN model generates probability maps, and the highest probability method reconstructs the optimal path. Contributions include introducing traversability cost, expanding map sizes and shapes, training a general model, and optimizing path reconstruction strategy. The proposed method outperforms traditional algorithms with average optimal rates of 72.7% and 78.2% for lowest-cost and shortest paths, respectively. The article discusses data requirements, FCNN design, training procedure, and evaluation measures. The work presents a promising solution for path planning on grid maps, offering speed advantages, flexibility, and improved performance compared to existing methods.

On the other hand, in Zhang et al. (2023c) deep reinforcement learning and transformer networks combined for efficient and scalable multi-objective global path planning in UAV-assisted sensor data collection. Traditional algorithms are limited by resource constraints and lack scalability for dynamic scenarios, making them inefficient. By decomposing the problem into sequencing subproblems and utilizing a modified transformer network and DRL algorithm, the proposed approach outperforms existing methods in terms of robustness, convergence, diversity of solutions, and temporal complexity.

Simillarly, a UAV path planning approach using inverse reinforcement learning and human demonstrations presented in Xu et al. (2023). In this work, the trained policy is demonstrated to be feasible in tasks with unknown environments.

Authors in Visca et al. (2022) developed an adaptive deep meta-learning energy-aware path planner that uses a 1D convolutional neural network to analyze terrains sequentially, as experienced by an autonomous mobile robot (AMR), to provide energy estimates in real-time. The paper provides evidence of the method's improved robustness to provide more informed energy estimations and energy-efficient paths when navigating challenging uneven terrains. The method integrates the adaptive energy estimator into a state-lattice A* path planner to consider actual robot mobility constraints. The experimental results show the model's performance in simulation over several typologies of natural terrains and unstructured geometries, and it is compared with alternative state-of-the-art deep learning solutions. The method's main novelty is the development of a deep meta-learning architecture that efficiently adapts its energy estimates to new terrains based on a small number of local measurements. The paper is well-structured, and the proposed method is adequately explained with diagrams and figures. The results are promising and show the method's potential in improving energy efficiency in autonomous mobile robots.

Chehelgami et al. Chehelgami et al. (2023) proposed a novel approach for global path planning using RNNs and a new loss function to predict safe paths. The authors compare and evaluate their proposed method with existing techniques in various environments, and validate its effectiveness using the AI2THOR simulation environment. The paper presents a comprehensive overview of related studies in path planning and describes the training data generation process based on the Bug2 algorithm. The LSTM network is introduced for path planning, along with the training and inference phases. Additionally, a new loss function, MSE-NER, is proposed to improve the success rate of predicted paths by creating a safe margin around obstacles. The results and discussion section demonstrates the superior performance of the proposed method compared to classical approaches such as RRT and A*, particularly in complex environments with convex and non-convex obstacles.

A multi-objective global path planning in UAV-assisted sensor data collection (SDC) in wireless sensor networks (WSNs) has presented in Zhang et al. (2023d). The study formulates the problem as a multi-objective optimization problem (MOP) aiming to maximize data collection and minimize UAV flight time. Leveraging a deep reinforcement learning (DRL) framework, the MOP is decomposed into sub-problems, modeled as neural networks, and solved using an actor-critic algorithm and modified pointer network. The proposed method outperforms traditional multi-objective evolutionary algorithms in terms of convergence, solution diversity, and time complexity. Furthermore, it exhibits robustness in accommodating changing sensor numbers without retraining. This work presents a promising advancement in UAV-assisted SDC and demonstrates the effectiveness of DRL in tackling multi-objective optimization problems.

A novel approach for Energy-Efficient Multitier Cooperative Computing in wireless sensor networks (WSNs) by utilizing UAV-assisted computing introduced in Guo et al. (2023). The paper proposes a three-tier WSN model that includes sensor nodes, a UAV equipped with computing resources, and a sink node. The goal is to optimize the UAV path planning to maximize energy efficiency while considering node energy consumption and task deadlines. To address the time-varying indeterminate environment, a deep Q network (DQN)-based path planning algorithm is introduced. Simulation studies demonstrate that the proposed algorithm outperforms competitive algorithms, significantly improving energy efficiency and achieving energy consumption balance. The paper concludes by discussing potential research opportunities in the field, including the extension to multi-UAV cooperative computing and the development of multiagent machine learning algorithms to handle more complex scenarios. Overall, this work presents a promising direction for UAV-assisted cooperative computing in WSNs and highlights the potential for further advancements in the field.

Shan et al. Shan et al. (2023) developed a charging path planning scheme for multi-UAV wireless rechargeable sensor networks (WRSNs) by combining the improved HEED clustering algorithm and IA-DRL intelligent algorithm. The study addresses the challenge of enhancing energy utilization in the charging process of UAVs in WRSNs. The proposed scheme optimizes the network model using the improved dynamic clustering algorithm of HEED and incorporates an intelligent path optimization algorithm based on IA-DRL. Experimental results demonstrate that IA-DRL outperforms traditional heuristic methods in solving small and medium-scale UAV path planning problems, with a

3.8% improvement in minimum AVG performance and faster convergence speed of about 550 episodes compared to neural network models without PSO. The paper also discusses related works and future research directions. Overall, this advanced and professional research contributes to the field of multi-UAV WRSN charging path planning.

Model developed by Asante et al. Asante et al. (2023) takes visual data obtained by RGB-D camera to navigate through indoor environment while avoiding obstacles. This dynamic path planning approach

| Challenges Associated with Deep Learning-Based Path Planning Techniques

Deep learning-based path planning techniques have been domonstrated their strength in various domains, including autonomous driving, precision agriculture, and UAVs. However, these techniques also face several challenges. Here are some of the challenges associated with deep learning-based path planning techniques:

- **Efficiency and safety in various driving scenarios:** Local planning is a core component of the autonomous vehicle motion planning module. Challenges in this problem include solving the local planning problem efficiently while keeping high vehicle safety in various driving scenarios.
- **Continuous action space in real-world applications:** Constructing agents with planning capabilities has long been one of the main challenges in the pursuit of artificial intelligence. Tree-based planning methods from AlphaGo to Muzero have enjoyed huge success in discrete domains, such as chess and Go. Unfortunately, in real-world applications like robot control and inverted pendulum, whose action space is normally continuous, those tree-based planning techniques will be struggling.
- **Boundary extraction and segmentation performance:** In some applications, such as photovoltaic plants inspection, it is necessary to extract the boundary of the facility with some techniques. A deep neural network model with a U-net structure has been proposed to automatically extract the boundary of PV plants from imagery. The experimental results evaluated on the Amir dataset show that the proposed approach can significantly improve the boundary and segmentation performance in the test stage.
- **Environmental complexity and uncertainty:** Autonomous navigation requires the ability to perceive the surroundings, plan a path, execute it, and adapt as needed, all while avoiding obstacles and collisions to ensure safe, efficient, and accurate travel. Recent developments in deep learning have made navigation more reliable, effective, and efficient, enabling its use in a wide range of applications, including transportation, search and rescue, and delivery. However, these techniques face challenges due to environmental complexity, uncertainty, obstacles, dynamic environments, and the need to plan paths for multiple agents.
- **Dynamic environments and multi-agent planning:** Autonomous navigation in dynamic environments, such as crowded streets or busy airports, is a challenging task. Deep reinforcement learning-based methods have been proposed for real-time path planning and dynamic obstacle avoidance. These methods use a deep neural network as the optimizer and train it by the reward through the interaction with the environment. Once the deep neural network is trained, it can rapidly give the results of path planning without the convergence process during the optimization.
- **Computationally expensive navigation systems:** In the previous complete coverage path planning (CCPP) research, autonomous vehicles need to consider mapping, obstacle avoidance, and route planning simultaneously during operating in the workspace, which results in an extremely complicated and computationally expensive navigation system. A new framework has been developed in light of a hierarchical manner with the obtained environmental information and gradually solving navigation problems layer by layer, consisting of environmental mapping, path generation, CCPP, and dynamic obstacle avoidance.
- **Optimizing path and energy consumption for multi-UAV data collection:** To collect data of distributed sensors located at different areas in challenging scenarios through artificial way is obviously inefficient, due to the numerous labor and time. Unmanned Aerial Vehicle (UAV) emerges as a promising solution, which enables multi-UAV collect data automatically with the preassigned path. However, without a well-planned path, the required number and consumed energy of UAVs will increase dramatically. Thus, minimizing the number of UAVs and optimizing the path are essential in order to ensure efficient data collection. A deep learning-based method has been proposed to improve the multi-UAV path planning for data collection.

Addressing these challenges requires future research in various directions:

- Efficient path planning algorithms that consider obstacles, dynamic changes, and optimization criteria, while addressing computational complexity and real-time performance limitations.
- Exploration of alternative energy sources and efficient power management techniques to enhance energy efficiency in autonomous systems.
- Designing robust wireless communication networks for autonomous systems, considering mobility, dynamic topologies, and low-latency communication, while addressing signal interference, scalability, reliability, and security concerns.
- Developing robust security mechanisms to protect autonomous systems from cyber threats, ensure integrity and privacy, and strike a balance between security measures and system performance.
- Cross-domain integration frameworks that enable seamless communication, coordination, and cooperation among autonomous systems, addressing technological heterogeneity, regulatory constraints, and efficient resource allocation.
- Overcoming limitations in deep learning, such as the need for diverse training datasets, complexity of models, real-time processing, robustness against environmental variations, and interpretability of deep learning-based decision-making.

Addressing these challenges and exploring future research directions will advance the development of efficient, robust, and reliable deep learning-based path planning techniques across various domains of autonomous systems.

5 | CONCLUSION

The studies mentioned above have made significant contributions to a diverse range of domains. Through the application of deep learning techniques, these investigations introduce compelling solutions to address intricate challenges, thereby propelling advancements within their respective fields. The utilization of deep neural networks enables to navigate atonomously in different scenarios without human interventions. These approaches demonstrate promising results, outperforming traditional methods and showcasing their potential for practical implementation. The integration of deep learning models with sensor data, such as IMU measurements, camera images, or light fidelity (LiFi) signals, enhances the precision and robustness of the estimations. Moreover, the proposed methodologies exhibit adaptability, efficiency, and ease of implementation, making them invaluable tools for applications in fields such as satellite technology, robotics, virtual and augmented reality, tunnel construction, and urban planning.

Indeed, there exists a substantial gap in the utilization of learning-based inertial navigation techniques. Despite the advancements and successful application of deep learning in various domains, the integration of deep learning algorithms with inertial sensors for autnomous navigation remains relatively underexplored while much works have focused on using visual data. Conventional methods heavily rely on mathematical models and signal processing techniques, which often pose challenges in their adoption and generalization across different scenarios. These traditional approaches require careful parameter tuning and may struggle to capture the complex and nonlinear relationships present in real-world situations.

In contrast, deep learning algorithms can adapt and generalize well to diverse environments and conditions, making them highly suitable for real-world applications. However, the development of new algorithms for inertial data augmentation methods is crucial to address the aforementioned challenges. By generating synthetic data or augmenting existing datasets, the diversity and quantity of training samples can be increased, improving the generalization and robustness of deep learning models. As a result, these models become more adept at adapting to different motion patterns, environmental conditions, and sensor characteristics, thereby enhancing the accuracy and reliability of attitude estimation from IMU data.

Regarding deep learning-based odometry, one of the primary challenges is the demand for real-time, on-board methods capable of instantaneously or minimally delaying the estimation of position and orientation. Traditional odometry approaches typically rely on sequential processing, which can introduce significant delays and hinder real-time applications. Deep learning-based odometry methods aim to mitigate these limitations by leveraging the power of neural networks to process sensor data in a parallel and efficient manner.

However, developing models that can efficiently process sensor data while optimizing computational efficiency and eliminating the delay associated with sequential processing remains a critical challenge. The real-time performance of deep learning-based odometry systems relies on the ability to process sensor data at high frequencies and make instantaneous predictions. Achieving this requires careful architectural design, efficient implementation, and optimization techniques to ensure that the models can handle the computational demands in real-time scenarios.

Another challenge is the integration of various sensor modalities to enhance odometry performance. Deep learning models have demonstrated the capability to fuse information from different sensors such as cameras, LiDAR, and IMUs to improve odometry accuracy and robustness. However, effectively integrating these sensors and exploiting their complementary characteristics in a deep learning framework is an ongoing research area.

Furthermore, addressing the limitations of deep learning-based odometry techniques in handling challenging environments is crucial. Factors such as dynamic lighting conditions, sensor noise, occlusions, and abrupt changes in the scene can adversely affect the performance of odometry systems. Developing robust algorithms that can handle these challenges and adapt to varying environmental conditions is a critical objective for future research.

To improve the practicality and reliability of deep learning-based odometry, it is essential to explore strategies for handling sensor failures and recovering from temporary tracking losses. Robustness and fault tolerance are crucial in autonomous systems operating in real-world scenarios where sensor malfunctions or occlusions can occur. Investigating techniques to ensure the resilience of deep learning-based odometry methods in the face of sensor failures or environmental disturbances is an important research direction.

Moreover, developing methods to quantify the uncertainty associated with deep learning-based odometry estimates is essential for safety-critical applications. Uncertainty estimation allows for better decision-making, risk assessment, and integration with higher-level planning and control algorithms. Approaches such as Bayesian deep learning or ensembling techniques can be explored to provide probabilistic estimates and capture the uncertainty in odometry predictions.

By addressing these challenges and further refining deep learning-based odometry methods, we can enable more accurate, real-time, and robust estimation of position and orientation, thus enhancing the navigation capabilities of autonomous systems across a wide range of applications.

Deep learning-based path planning techniques have shown remarkable success in various domains, including autonomous driving, robotics, and aerial navigation. However, several challenges remain to be addressed to further improve the efficiency, safety, and applicability of these techniques in complex and dynamic environments.

Efficiency is a crucial aspect of path planning, particularly in real-time applications such as autonomous vehicles. Deep learning-based path planning methods often involve computationally intensive operations. Ensuring the computational efficiency of these models is essential to meet the real-time requirements of navigation systems. Techniques such as model compression, network pruning, or specialized hardware implementations can be explored to optimize the efficiency of deep learning models for path planning.

Safety is another critical consideration in path planning. Deep learning models for path planning should be able to handle complex driving scenarios, including interactions with other vehicles, pedestrians, and infrastructure. Ensuring the safe behavior of autonomous systems requires methods that can effectively capture and reason about uncertainties, adapt to novel situations, and generalize well across different environments. The development of robust deep learning architectures and training techniques that explicitly address safety concerns is of utmost importance.

Furthermore, interpretability and explainability are vital aspects of deep learning-based path planning algorithms. Being able to understand the decision-making process of these models and provide transparent explanations for their actions is crucial for building trust and ensuring accountability. Efforts should be made to develop techniques that enable interpretability, such as attention mechanisms, saliency maps, or rule-based post-hoc explanations, to enhance the transparency and reliability of deep learning-based path planning systems.

Generalization to unseen scenarios and adaptation to dynamic environments pose additional challenges. Deep learning models trained on a specific dataset or environment might struggle to generalize well to novel situations or adapt to changing conditions. Methods such as domain adaptation, transfer learning, or continual learning can be explored to improve the generalization capabilities of deep learning-based path planning algorithms. These approaches aim to leverage previously learned knowledge and adapt it to new environments or tasks, reducing the need for extensive retraining and improving the overall performance and applicability of the models.

Addressing ethical considerations and societal impacts is also essential when developing deep learning-based path planning methods. Autonomous systems equipped with deep learning-based path planning algorithms interact with the environment and other agents, raising concerns regarding fairness, privacy, and potential biases. Striving for fairness, transparency, and inclusiveness in the design and deployment of these algorithms is critical to ensure that they benefit society as a whole and avoid exacerbating existing biases or inequalities.

Despite the afpormentioned advances, there are still numerous challenges that need to be addressed in the field of deep learning-based navigation:

1. **Generalization over different motion patterns and environmental disturbances**: Deep learning models should be able to generalize well across diverse motion patterns and environmental disturbances. Ensuring robustness in various scenarios is crucial for practical deployment.

2. **Generalization over different sampling rates**: Inertial sensors may operate at different sampling rates, and deep learning models should be capable of handling variations in sampling rates without compromising estimation performance.

3. **Real-time performance**: Real-time attitude estimation is critical for applications such as robotics and navigation systems. Deep learning models should be designed and optimized to ensure real-time processing capabilities.

4. **Model selection and loss function**: Choosing the appropriate deep learning topology for navigation purposes is essential. Additionally, designing an effective loss function for orientation increments is crucial to achieve accurate results.

Overcoming these challenges will unlock the full potential of deep learning-based navigation methods, paving the way for their adoption in a wide range of applications.

In conclusion, while deep learning-based navigation techniques have demonstrated impressive capabilities, there are still challenges to overcome to further improve their efficiency, safety, interpretability, generalization, and ethical aspects. By addressing these challenges, we can unlock the full potential of deep learning in authonmous navigation and pave the way for the widespread adoption of autonomous systems in various domains.

Acknowledgements

The study presented in this paper is based on A. Asgharpoor Golroudbari's M.Sc. Thesis ("Design and Simulation of Attitude and Heading Estimation Algorithm", Department of Aerospace, Faculty of New Sciences & Technologies, University of Tehran).

We sincerely thank Professor Parvin Pasalar, Dr. Farsad Nourizade, and Dr. Maryam Karbasi Motlagh from the Students' Scientific Research Center for their invaluable guidance and support. Their contributions have greatly enhanced the quality and success of our research.

Conflict of Interest

The authors declare no competing interests.

A | ARTIFICIAL NEURON

An artificial neuron is a mathematical function that is used to model the behavior of biological neurons. It was first introduced by Warren McCulloch and Walter Pitts in 1943. The idea of artificial neurons is that an input signal weighted by a weight w with a bias b is passed through an activation function f to produce an output signal y Schmidhuber (2015). Figure 9 and Equation 4 shows the structure of an artificial neuron.

$$y = f(w \cdot x + b) \tag{4}$$

where the activation function f performs a non-linear transformation on the weighted sum of the inputs.

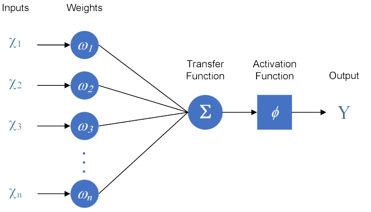

FIGURE 9 Structure of an artificial neuron

A.1 | Deep Feedforward Neural Networks

A deep feedforward neural network is composed of multiple layers of artificial neurons, each of which is fully connected to the next layer. This type of neural network is also known as a multilayer perceptron (MLP) Goodfellow et al. (2016). All data is passed through the network in a forward direction. The input is passed through the first layer, which is then passed through the second layer, and so on until the output of the network is produced. The output is determined by the output of the last layer. Figure 10 and Equation 5 shows the structure of a deep feedforward neural network.

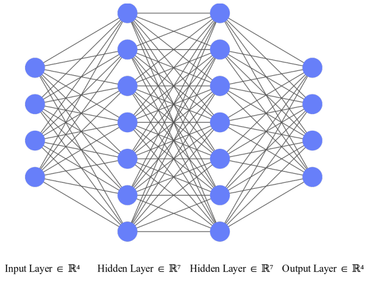

FIGURE 10 Structure of a deep feedforward neural network

$$
\begin{aligned}
&\text{Input:} \quad \mathbf{x} \in \mathbb{R}^n \\
&\text{Hidden layers:} \quad \mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_L \\
&\text{Weights:} \quad \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L \\
&\text{Biases:} \quad \mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_L \\
&\text{Activation function:} \quad f \\
&\text{Output:} \\
&\quad \mathbf{h}_1 = f(\mathbf{w}_1 \cdot \mathbf{x} + \mathbf{b}_1) \\
&\quad \mathbf{h}_i = f(\mathbf{w}i \cdot \mathbf{h}i - 1 + \mathbf{b}_i), \quad i = 2, \ldots, L \\
&\quad \mathbf{y} = \mathbf{w}L \cdot \mathbf{h}L - 1 + \mathbf{b}_L \\
&\quad \text{Output:} \quad \mathbf{y} \in \mathbb{R}^m
\end{aligned}
\tag{5}
$$

A.2 | Autoencoder

An autoencoder is designed to learn a compressed representation of input data and composed of two main parts: (1) encoder and (2) decoder. The encoder takes in the input data and learns to compress it into a lower-dimensional representation, typically called a latent representation or bottleneck. The decoder then takes this compressed representation and learns to reconstruct the original input data from it. The goal is to learn a compressed representation that captures the most important features of the input data while discarding the unnecessary or redundant information. They can be used for a variety of tasks such as dimensionality reduction, data denoising, and generative modeling. There are different variations of autoencoders such as convolutional autoencoder, variational autoencoder, and more. Figure 11 shows the structure of a standard feedforward autoencoder:

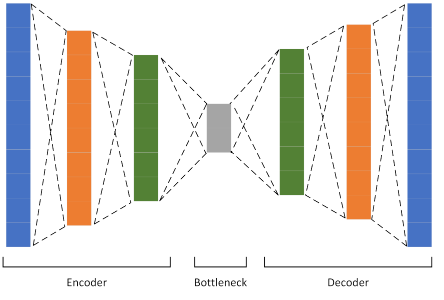

FIGURE 11 Structure of an autoencoder

A.3 | Convolutional Neural Network (CNN)

A CNN used convolution operation to extract features from data. This operation involves combining two sets of data, such as an image and a filter, in order to extract meaningful features from the input. By using this technique, CNNs can identify patterns in complex datasets that would otherwise be difficult or impossible for traditional algorithms to detect. CNNs are particularly useful for tasks such as image recognition, object detection, and image segmentation. It is composed of multiple layers which are arranged in a hierarchical structure, with the lower layers extracting simple features such as edges, and the higher layers extracting more complex features such as patterns and objects. Figure 12 shows the structure of a CNN:

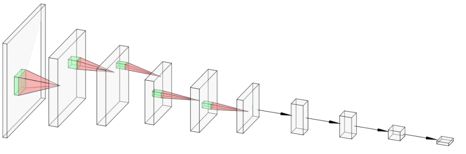

FIGURE 12 Structure of a CNN

A.4 | Deep Belief Networks (DBN)

Deep belief network was first introduced in 2006 by Hinton et al. Hinton et al. (2006) which is composed of multiple layers of hidden units, where each layer is a Restricted Boltzmann Machine (RBM). RBMs are shallow, two-layer neural networks where the visible units are connected to the hidden units, but not to each other. In a DBN, the RBMs are stacked on top of each other to form a deep network. The equation for an RBM can be represented as follows:

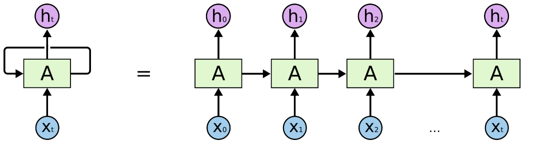

FIGURE 13 A simple RNN

$$P(v,h) = \frac{1}{Z} \exp(-E(v,h)) \tag{6}$$

$$E(v,h) = -\sum_{i}^{m} b_i v_i - \sum_{j}^{n} c_j h_j - \sum_{i}^{m}\sum_{j}^{n} v_i w_{ij} h_j \tag{7}$$

where v and h are the visible and hidden layer activations respectively, b and c are the biases of the visible and hidden layer, w is the weight matrix between the visible and hidden layer, and Z is the normalizing constant.

DBNs can be trained using an unsupervised learning algorithm, where the input data is used to learn the generative model. DBNs have been applied in several domains, including computer vision, speech recognition, natural language processing, and bioinformatics, for tasks such as image classification, feature extraction, and language recognition.

A.5 | Recurrent Neural Network (RNN)

Recurrent neural networks have gained popularity in recent years due to their ability to process sequential data, such as time series, speech, and text. They are characterized by the presence of recurrent connections, which allow information to flow through the network over multiple time steps. This allows RNNs to maintain a hidden state that can be updated at each time step, allowing them to remember previous input and use that information to inform their current output Goodfellow et al. (2016).

The most basic form of an RNN is the simple recurrent neural network (SRN), which has a single hidden layer with recurrent connections. The SRN's hidden state at time t, h_t, is a function of the current input x_t and the previous hidden state $h_t - 1$. The output y_t is then computed based on the hidden state h_t. The basic equation of a Recurrent Neural Network can be represented as follows:

$$\begin{aligned}
&\text{Input:} \quad x \in \mathbb{R}^n \\
&\text{Output:} \quad y \in \mathbb{R}^m \\
&\text{Hidden state:} \quad h \in \mathbb{R}^k \\
&\text{Weights:} \quad W(hh), W(hx) \\
&\text{Biases:} \quad b(h) \\
&\text{Activation function:} \quad f \\
&\text{Output:} \quad y = f(W(yh)h + b(y))
\end{aligned} \tag{8}$$

where h is the hidden state, x is the input, $W(hh)$, $W(hx)$ and $b(h)$ are the weights and bias for the hidden state, $W(yh)$ and $b(y)$ are the weights and bias for the output, and f is the activation function. The hidden state at the current time step is a function of the previous hidden state and the current input, with the weights and bias being learned during training. The output of the RNN at each time step is a function of the current hidden state, with the weights and bias for the output also being learned during training. The hidden state at time step t is defined as follows:

$$h(t) = f(W(hh)h(t-1) + W(hx)x(t) + b(h)) \tag{9}$$

where $h(t-1)$ is the hidden state at the previous time step, $x(t)$ is the input at time step t, and f is the activation function. The output of the RNN at each time step is a function of the current hidden state, with the weights and bias for the output also being learned during training.

Long short-term memory (LSTM) and gated recurrent units (GRUs) are two variations of RNNs that have been developed to address the problem of vanishing gradients which occur when the gradients of the parameters of the network become very small during the backpropagation process, making it difficult to train the network Hochreiter (1998). LSTM and GRUs both introduce gating mechanisms that allow the network to selectively choose which information to forget or remember at each time step, thus addressing the vanishing gradients problem.

Another variant of RNNs is the bidirectional RNN (BRNN), which allows the network to take into account future context in addition to past context. In a BRNN, two RNNs are used to processes the input in the forward direction and reverse direction. The output of the two RNNs is then concatenated and used as the final output.

RNNs have been used in a variety of applications, including natural language processing, speech recognition, and time series prediction and used for tasks such as language translation, text summarization, sentiment analysis, and forecast future values based on past values.

One of the main benefits of RNNs is their ability to handle sequential data, which allows them to model temporal dependencies and patterns. However, RNNs also have some limitations. One limitation is that they can be sensitive to the order of input data, which can be a problem when dealing with permuted data. Additionally, RNNs can be computationally expensive and may require a large amount of memory to store the hidden states. Figure 13 shows a diagram of a simple RNN.

A.6 | Gate Recurrent Unit (GRU)

GRU is introduced by Cho et al. in 2014 Cho et al. (2014). The main difference between a traditional RNN and a GRU is that a GRU has two gates, called the update gate and the reset gate, which control the flow of information between the previous hidden state and the current hidden state.

The update gate controls how much of the previous hidden state should be kept, and the reset gate controls how much of the previous hidden state should be forgotten. This allows a GRU to effectively handle long-term dependencies in the input sequence, as it can selectively choose which information to keep and which to discard.

The mathematical equation for the hidden state update in a GRU is as follows:

$$h_t = z_t * h_{t-1} + (1-z) * \tilde{h}_t \tag{10}$$

where h_t is the hidden state at time t, h_{t-1} is the hidden state at time $t-1$, z_t is the update gate, and $\tilde{h}_t$ is the candidate hidden state.

The update gate z_t is computed using the following equation:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{11}$$

where W_z is the weight matrix for the input, U_z is the weight matrix for the previous hidden state, x_t is the input at time t, and b_z is the bias term for the update gate.

The candidate hidden state $\tilde{h}_t$ is computed using the following equation:

$$\tilde{h}_t = tanh(W_h x_t + U_h(h_{t-1} \circ r_t) + b_h) \tag{12}$$

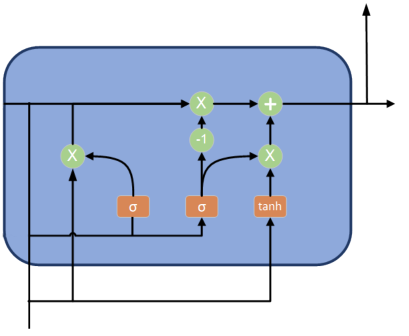

FIGURE 14 A GRU

where W_h is the weight matrix for the input, U_h is the weight matrix for the previous hidden state, r_t is the reset gate, and b_h is the bias term for the candidate hidden state. $\circ$ is the element-wise multiplication operator.

The reset gate r_t is computed using the following equation:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{13}$$

where W_r is the weight matrix for the input, U_r is the weight matrix for the previous hidden state, x_t is the input at time t, and b_r is the bias term for the reset gate.

The output of the GRU at time t is calculated by using the following equation:

$$y_t = \sigma(W_y h_t + b_y) \tag{14}$$

where W_y is the weight matrix for the hidden state and by is the bias term for the output. Figure 14 shows a diagram of a GRU.

A.7 | Long Short-Term Memory (LSTM)

An LSTM is introduced in 1997 by S. Hochreiter and J. Schmidhuber Hochreiter and Schmidhuber (1997). It has a memory cell that is used to retain information from previous inputs and use it to process the current input. LSTMs are consists of three gates, an input gate, an output gate, and a forget gate. The input gate is used to decide which values from the current input should be added to the memory cell. The forget gate is used to decide which values from the memory cell should be removed. The output gate is used to decide which values from the memory cell should be used to produce the output. LSTMs are particularly useful for tasks such as language modeling, machine translation, and speech recognition. The main advantage of LSTMs over other RNNs is that they can learn long-term dependencies. Its equation is as follows:

Input Gate:

$$i_t = \sigma(W_{xi} x_t + W_{hi} h_{t-1} + b_i) \tag{15}$$

Forget Gate:

$$f_t = \sigma(W_{xf} x_t + W_{hf} h_{t-1} + b_f) \tag{16}$$

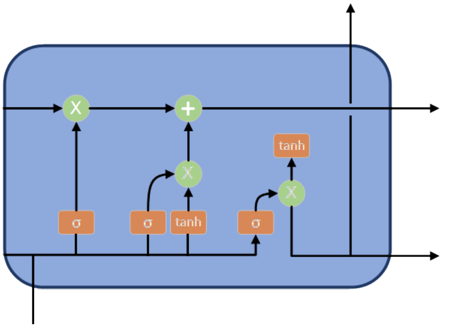

FIGURE 15 An LSTM

Output Gate:

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{17}$$

Cell State:

$$\tilde{C}_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{18}$$

Figure 15 shows a diagram of an LSTM.

A.8 | Transformer

Transformer is a neural network architecture introduced by Vaswani et al. in 2017 Vaswani et al. (2017). It is a type of encoder-decoder architecture that utilizes self-attention mechanisms to process sequential data.

The transformer architecture consists of an encoder and a decoder, each of which is made up of multiple layers. The encoder takes in a sequence of input data, such as a sentence or a document, and produces a set of hidden representations called the "keys" and "values". The decoder then takes in the keys and values and generates a new sequence of output data, such as a translated sentence or a summary of the input.

One of the key features of the this architecture is its use of self-attention mechanisms which allows the model to weigh the importance of different parts of the input sequence when generating the output. This is done by computing a set of attention weights for each element in the input sequence, which are used to weigh the contribution of each element to the final output.

Another important feature is the use of multi-head attention. Multi-head attention allows the model to attend to different parts of the input sequence in parallel, rather than serially. This allows the model to learn more complex relationships between different parts of the input sequence.

This architecture has been used in a wide range of natural language processing tasks, such as machine translation, text summarization, and language modeling. It has also been adapted for other types of data, such as images and time series. It has proven to be very effective at these tasks and has become a popular choice among researchers and practitioners.

A.9 | Generative Adversarial Network (GAN)

The GAN was first introduced in 2014 by Goodfellow and his colleagues Goodfellow et al. (2020). It is designed to generate new, previously unseen examples from a given dataset, such as images, text, or audio. GANs consist of two main components: a generator network and a discriminator network.

The generator network is responsible for creating new examples that are similar to the examples in the given dataset which takes a random noise input and maps it to a sample from the target distribution. The generator is typically a deep neural network with multiple layers, such as a fully connected or convolutional neural network.

The discriminator network is responsible for distinguishing between the examples generated by the generator and the examples from the real dataset. It takes an example as input and produces a probability value indicating the likelihood that the example is real. The discriminator is also typically a deep neural network with multiple layers.

The two networks are trained together in a two-player minimax game, where the generator is trying to produce examples that can fool the discriminator into thinking they are real, while the discriminator is trying to correctly identify which examples are real and which are fake. As the training progresses, the generator becomes better at producing realistic examples, and the discriminator becomes better at identifying them.

The goal of training is to reach a point where the generator can produce examples that are indistinguishable from real examples, and the discriminator is unable to make a clear distinction between the two. At this point, the generator can be used to generate new examples from the target distribution, such as new images that look like photographs of faces, or new audio samples that sound like speech.

GANs have been used in a wide range of applications, such as image synthesis, image-to-image translation, video synthesis, and text-to-speech synthesis. They have also been used for unsupervised learning tasks, such as anomaly detection and feature learning.

It's worth noting that GANs are considered one of the most difficult models to train and stabilize due to the adversarial nature of the training process and the potential for the generator and discriminator to get stuck in equilibrium where neither network makes progress. Therefore, multiple techniques have been proposed to stabilize the training process like Wasserstein GAN Arjovsky et al. (2017), Improved WGAN Gulrajani et al. (2017), and BEGAN Berthelot et al. (2017).

Table 15 summarizes the main architectures discussed in this section and their main properties and applications.

B | ACTIVATION FUNCTIONS

Activation functions introduce non-linearity into the network. Without non-linearity, a neural network would be limited to linear operations and would not be able to learn complex representations Goodfellow et al. (2016).

One of the earliest activation functions used in neural networks is the step function, invented by W. McCulloch and W. Pitts in 1943 McCulloch and Pitts (1943). The step function outputs a value of 1 if the input is greater than a certain threshold and 0 otherwise. This function is not used in modern neural networks, as it is not differentiable and therefore not useful for backpropagation.

Another early activation function is the sigmoid function, invented by Frank Rosenblatt in 1958 Rosenblatt (1958) which maps the input to a value between 0 and 1, making it useful for binary classification tasks. However, it is not recommended for use in modern neural networks, as it can saturate and produce vanishing gradients. It takes an input between $-\infty$ and $+\infty$ and outputs a value between 0 and 1. This can be considered as a likelihood or probability of the input being in a certain class. It is defined as follows:

$$f(x) = \frac{1}{1+e^{-x}} \tag{19}$$

A more recent activation function is the rectified linear unit (ReLU) function, invented by Fukushima Fukushima (1975) in 1975. The ReLU function outputs the input if it is positive and 0 otherwise, making it computationally efficient and effective in avoiding the vanishing gradients problem. It is defined as follows:

$$f(x) = \max(0, x) \tag{20}$$

Leaky ReLU is an extension of the ReLU function, which introduces small negative slope in the negative part, to overcome the dying neurons problem Maas et al. (2013). The difference between Leaky ReLU and ReLU is that Leaky ReLU is how to handle negative values. Instead of setting them to zero, Leaky ReLU sets them to a small negative value. This small negative value is called the leaky coefficient which is a hyperparameter that can be tuned. It is defined as follows:

$$f(x) = \max(0, x) + \alpha \min(0, x) \tag{21}$$

where α is the leaky coefficient. Another popular activation function is the hyperbolic tangent (tanh) function, which maps the input to a value between -1 and 1. This function is similar to the sigmoid function but is symmetric around the origin, which can be useful in certain situations. Tanh could be used instead of sigmoid function to avoid the vanishing gradient problem. It is defined as follows:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{22}$$

The exponential linear unit (ELU) function, invented by Clevert Clevert et al. (2015) in 2015, which is similar to ReLU but it is defined as negative when the input is less than zero. This function can be useful to help the network to learn faster and overcome the problem of dying neurons.

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1) & \text{if } x \leq 0 \end{cases} \tag{23}$$

where α is the negative slope coefficient, which is a hyperparameter that can be tuned.

Softmax function is another activation function that is commonly used in the output layer of a neural network for multi-class classification problems Bridle (1990, 1989). The softmax function maps the input to a probability distribution over multiple classes. It is defined as follows:

$$f(x) = \frac{e^x}{\sum_{i=1}^{n} e^x} \tag{24}$$

Softplus function is another activation function that is commonly used in the output layer of a neural network for multi-class classification problems Dugas et al. (2000). The softplus function maps the input to a probability distribution over multiple classes. It is defined as follows:

$$f(x) = \log(1 + e^x) \tag{25}$$

Swish function is commonly used in the output layer of a neural network for multi-class classification problems Ramachandran et al. (2017) which maps the input to a probability distribution over multiple classes. It helps to overcome the problem of vanishing gradient making it suitable for deep neural networks. It is defined as follows:

$$f(x) = x \cdot \text{sigmoid}(\beta x) \tag{26}$$

$$f(x) = x \cdot \frac{1}{1+e^{-\beta x}} \tag{27}$$

where β is the slope coefficient, which is a hyperparameter that can a constant or a learnable parameter.

Randomized ReLU (RRelu) is introduced by Xu et al. in 2015Xu et al. (2015) which is a variation of the traditional ReLU. With ReLU, the negative part of the input is dropped or set to zero, whereas with RReLU, this negative part is assigned a non-zero slope. This means that a randomly determined slope is assigned to the negative part, and can update during the training process. The advantage of using this is that it improves the performance of a convolutional neural network and can lead to better results on image classification tasks. It is defined as follows:

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if } x \leq 0 \end{cases} \tag{28}$$

In 2019 Misra et al. Misra (2019) introduced a new activation function called Mish, which is self-regularized non-monotonic activation function. It combines the advantages of ReLU and tanh functions to create a more robust non-linearity. It can be mathematically defined as:

$$f(x) = x \cdot \tanh(\text{softplus}(x)) = x \cdot \tanh(ln(1+e^{x})) \tag{29}$$

In conclusion, activation functions play a crucial role in deep neural networks by introducing non-linearity and allowing the network to learn complex representations. Different activation functions have their own advantages and disadvantages and are suitable for different types of problems. ReLU and its variants are widely used in modern neural networks due to their effectiveness and computational efficiency.

Some of popular activation functions has been listed in the table below: In Table 16, we summarized some related works in the navigation field using deep learning.

In figures 16 and 17, the most common activation functions have been shown.

In conclusion, activation functions play a crucial role in deep learning as they add non-linearity to the model and determine the output of each neuron in a network. The choice of activation function depends on the type of problem being solved and its desired output so there is no ultimate answer for the question "What is the best activation function?". However, it is important to consider the characteristics and limitations of each activation function when designing a deep learning model.

references

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. et al. (2016) Tensorflow: a system for large-scale machine learning. In *Osdi*, vol. 16, 265–283. Savannah, GA, USA.

Abd-Elhay, A.-E. R., Murtada, W. A. and Youssef, M. I. (2022) A reliable deep learning approach for time-varying faults identification: Spacecraft reaction wheel case study. *IEEE Access*, **10**, 75495–75512.

Acun, O., Küçükmanisa, A., Genç, Y. and Urhan, O. (2023) D3net (divide and detect drivable area net): deep learning based drivable area detection and its embedded application. *Journal of Real-Time Image Processing*, **20**, 16.

Afif, M., Ayachi, R., Said, Y. and Atri, M. (2022) An evaluation of efficientdet for object detection used for indoor robots assistance navigation. *Journal of Real-Time Image Processing*, **19**, 651–661.

Aftf, M., Ayachi, R., Said, Y., Pissaloux, E. and Atri, M. (2019) Indoor object c1assification for autonomous navigation assistance based on deep cnn model. In *2019 IEEE International Symposium on Measurements & Networking (M&N)*, 1–4. IEEE.

Aggarwal, A., Mittal, M. and Battineni, G. (2021) Generative adversarial network: An overview of theory and applications. *International Journal of Information Management Data Insights*, **1**, 100004.

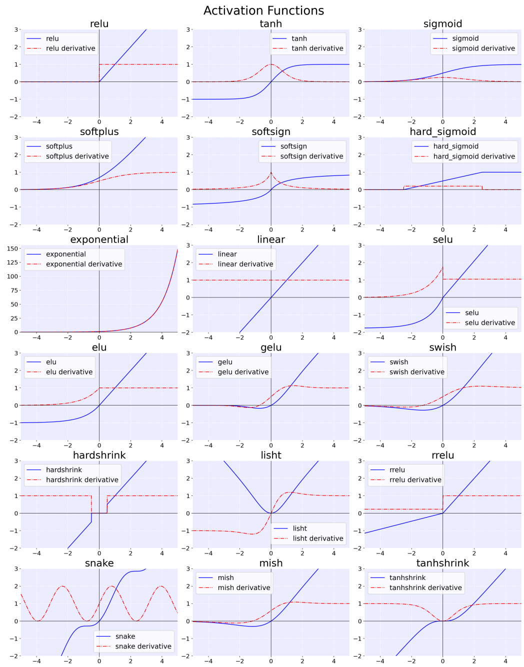

FIGURE 16 Activation Functions

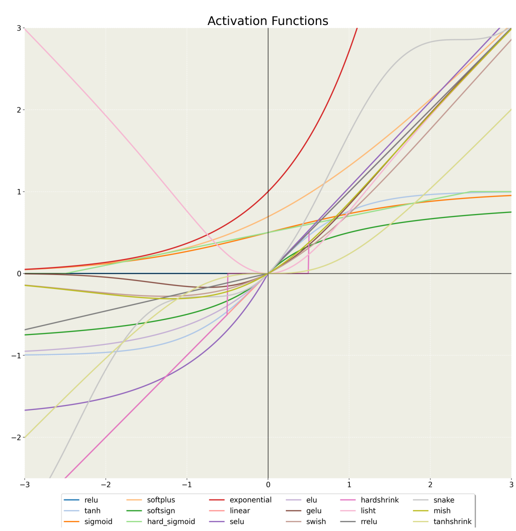

FIGURE 17 Compare Activation Functions

Alaba, S. Y. and Ball, J. E. (2023) Deep learning-based image 3d object detection for autonomous driving. *IEEE Sensors Journal*.

AlDahoul, N., Karim, H. A. and Momo, M. A. (2022) Rgb-d based multi-modal deep learning for spacecraft and debris recognition. *Scientific Reports*, **12**, 3924.

AlMahamid, F. and Grolinger, K. (2022) Autonomous unmanned aerial vehicle navigation using reinforcement learning: A systematic review. *Engineering Applications of Artificial Intelligence*, **115**, 105321.

Almalioglu, Y., Saputra, M. R. U., De Gusmao, P. P., Markham, A. and Trigoni, N. (2019) Ganvo: Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks. In *2019 International conference on robotics and automation (ICRA)*, 5474–5480. IEEE.

Almalioglu, Y., Turan, M., Saputra, M. R. U., de Gusmão, P. P., Markham, A. and Trigoni, N. (2022) Selfvio: Self-supervised deep monocular visual–inertial odometry and depth estimation. *Neural Networks*, **150**, 119–136.

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., Van Essen, B. C., Awwal, A. A. and Asari, V. K. (2019) A state-of-the-art survey on deep learning theory and architectures. *electronics*, **8**, 292.

Alzahrani, B., Oubbati, O. S., Barnawi, A., Atiquzzaman, M. and Alghazzawi, D. (2020) Uav assistance paradigm: State-of-the-art in applications and challenges. *Journal of Network and Computer Applications*, **166**, 102706.

Amazon (2022) Mk4, amazon prime air drone. URL: `https://www.aboutamazon.com/news/transportation/amazon-prime-air-prepares-for-drone-deliveries`.

Amiri, A. J., Loo, S. Y. and Zhang, H. (2019) Semi-supervised monocular depth estimation with left-right consistency using deep neural network. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 602–607. IEEE.

Amosa, T. I., Sebastian, P., Izhar, L. I. B. and Ibrahim, O. (2022) A comparative study on deep feature extraction approaches for visual tracking of industrial robots. In *2022 IEEE Industrial Electronics and Applications Conference (IEACon)*, 122–127. IEEE.

Angermueller, C., Pärnamaa, T., Parts, L. and Stegle, O. (2016) Deep learning for computational biology. *Molecular systems biology*, **12**, 878.

Antonini, A., Guerra, W., Murali, V., Sayre-McCord, T. and Karaman, S. (2020) The blackbird uav dataset. *The International Journal of Robotics Research*, **39**, 1346–1364.

Aqel, M. O., Marhaban, M. H., Saripan, M. I. and Ismail, N. B. (2016) Review of visual odometry: types, approaches, challenges, and applications. *SpringerPlus*, **5**, 1–26.

Arena, P., Basile, A., Bucolo, M. and Fortuna, L. (2003) Image processing for medical diagnosis using cnn. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, **497**, 174–178.

Arfaoui, M. A., Soltani, M. D., Tavakkolnia, I., Ghrayeb, A., Assi, C. M., Safari, M. and Haas, H. (2021) Invoking deep learning for joint estimation of indoor lifi user position and orientation. *IEEE Journal on Selected Areas in Communications*, **39**, 2890–2905.

Arjovsky, M., Chintala, S. and Bottou, L. (2017) Wasserstein generative adversarial networks. In *International conference on machine learning*, 214–223. PMLR.

Arsene, C. T., Hankins, R. and Yin, H. (2019) Deep learning models for denoising ecg signals. In *2019 27th European Signal Processing Conference (EUSIPCO)*, 1–5. IEEE.

Asante, I., Theng, L. B., Tsun, M. T. K., Jo, H. S. and McCarthy, C. (2023) Segmentation-based angular position estimation algorithm for dynamic path planning by a person-following robot. *IEEE Access*.

Asgharpoor Golroudbari, A. (2022) *Design and Simulation of Heading Estimation Algorithm*. Ph.D. thesis, University of Tehran.

Asgharpoor Golroudbari, A. and Sabour, M. H. (2023) Generalizable end-to-end deep learning frameworks for real-time attitude estimation using 6dof inertial measurement units. *Available at SSRN 4387238*.

Aslan, M. F., Durdu, A., Yusefi, A. and Yilmaz, A. (2022) Hvionet: A deep learning based hybrid visual–inertial odometry approach for unmanned aerial system position estimation. *Neural Networks*, **155**, 461–474.

Ayyalasomayajula, R., Arun, A., Wu, C., Sharma, S., Sethi, A. R., Vasisht, D. and Bharadia, D. (2020) Deep learning based wireless localization for indoor navigation. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 1–14.

Badrinarayanan, V., Kendall, A. and Cipolla, R. (2017) Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, **39**, 2481–2495.

Badrloo, S., Varshosaz, M., Pirasteh, S. and Li, J. (2022) Image-based obstacle detection methods for the safe navigation of unmanned vehicles: A review. *Remote Sensing*, **14**, 3824.

Baldini, F., Harvard, A., Chung, S.-J., Nesnas, I. and Bhaskaran, S. (2018) Autonomous small body mapping and spacecraft navigation via real-time spc-slam.

Basiri, M. A., Chehelgami, S., Ashtari, E., Masouleh, M. T. and Kalhor, A. (2022) Synergy of deep learning and artificial potential field methods for robot path planning in the presence of static and dynamic obstacles. In *2022 30th International Conference on Electrical Engineering (ICEE)*, 456–462. IEEE.

Bavle, H., De La Puente, P., How, J. P. and Campoy, P. (2020) Vps-slam: Visual planar semantic slam for aerial robotic systems. *IEEE Access*, **8**, 60704–60718.

Berthelot, D., Schumm, T. and Metz, L. (2017) Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*.

Bian, T., Xing, Y. and Zolotas, A. (2022) End-to-end one-shot path-planning algorithm for an autonomous vehicle based on a convolutional neural network considering traversability cost. *Sensors*, **22**, 9682.

Bochkovskiy, A., Wang, C.-Y. and Liao, H.-Y. M. (2020) Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Bouktif, S., Fiaz, A., Ouni, A. and Serhani, M. A. (2019) Single and multi-sequence deep learning models for short and medium term electric load forecasting. *Energies*, **12**, 149.

Bradski, G. (2000) The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, **25**, 120–123.

Bridle, J. (1989) Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. *Advances in neural information processing systems*, **2**.

Bridle, J. S. (1990) Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing: Algorithms, architectures and applications*, 227–236. Springer.

Britting, K. R. (2010) *Inertial navigation systems analysis*. Artech House Print on Demand.

Brossard, M. and Bonnabel, S. (2019) Learning wheel odometry and imu errors for localization. In *2019 International Conference on Robotics and Automation (ICRA)*, 291–297. IEEE.

Brossard, M., Bonnabel, S. and Barrau, A. (2020) Denoising imu gyroscopes with deep learning for open-loop attitude estimation. *IEEE Robotics and Automation Letters*, **5**, 4796–4803.

Brotchie, J., Shao, W., Li, W. and Kealy, A. (2022) Leveraging self-attention mechanism for attitude estimation in smartphones. *Sensors*, **22**, 9011.

Bryson, A. E. and Ho, Y.-C. (1975) *Applied optimal control: Optimization, estimation, and control*. CRC Press.

Buchanan, R., Camurri, M., Dellaert, F. and Fallon, M. (2022) Learning inertial odometry for dynamic legged robot state estimation. In *Conference on Robot Learning*, 1575–1584. PMLR.

Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M. W. and Siegwart, R. (2016) The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, **35**, 1157–1163.

Byun, S.-W., Noh, D. and Lee, H.-M. (2022) Design of obstacle detection method for autonomous driving in agricultural environments. In *2022 Thirteenth International Conference on Ubiquitous and Future Networks (ICUFN)*, 494–496. IEEE.

Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G. and Beijbom, O. (2020) nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11621–11631.

Cai, Z. and Vasconcelos, N. (2018) Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6154–6162.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A. and Zagoruyko, S. (2020) End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, 213–229. Springer.

Carlevaris-Bianco, N., Ushani, A. K. and Eustice, R. M. (2016) University of michigan north campus long-term vision and lidar dataset. *The International Journal of Robotics Research*, **35**, 1023–1035.

Carrio, A., Sampedro, C., Rodriguez-Ramos, A. and Campoy, P. (2017) A review of deep learning methods and applications for unmanned aerial vehicles. *Journal of Sensors*, **2017**.

Caruso, M., Sabatini, A. M., Knaflitz, M., Gazzoni, M., Della Croce, U. and Cereatti, A. (2020) Orientation estimation through magneto-inertial sensor fusion: A heuristic approach for suboptimal parameters tuning. *IEEE Sensors Journal*, **21**, 3408–3419.

Celik, K. and Somani, A. K. (2013) Monocular vision slam for indoor aerial vehicles. *Journal of electrical and computer engineering*, **2013**, 4–4.

Cervera-Uribe, A. A. and Mendez-Monroy, P. E. (2022) U19-net: a deep learning approach for obstacle detection in self-driving cars. *Soft Computing*, **26**, 5195–5207.

Chang, L., Shan, L., Zhang, W. and Dai, Y. (2023) Hierarchical multi-robot navigation and formation in unknown environments via deep reinforcement learning and distributed optimization. *Robotics and Computer-Integrated Manufacturing*, **83**, 102570.

Chang, M.-F., Lambert, J., Sangkloy, P., Singh, J., Bak, S., Hartnett, A., Wang, D., Carr, P., Lucey, S., Ramanan, D. et al. (2019) Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8748–8757.

Chang, S., Zhang, Y., Zhang, F., Zhao, X., Huang, S., Feng, Z. and Wei, Z. (2020) Spatial attention fusion for obstacle detection using mmwave radar and vision sensor. *Sensors*, **20**, 956.

Chang, Y., Cheng, Y., Murray, J., Huang, S. and Shi, G. (2022) The hdin dataset: A real-world indoor uav dataset with multi-task labels for visual-based navigation. *Drones*, **6**, 202.

Chaurasia, A. and Culurciello, E. (2017) Linknet: Exploiting encoder representations for efficient semantic segmentation. In *2017 IEEE visual communications and image processing (VCIP)*, 1–4. IEEE.

Chehelgami, S., Ashtari, E., Basiri, M. A., Masouleh, M. T. and Kalhor, A. (2023) Safe deep learning-based global path planning using a fast collision-free path generator. *Robotics and Autonomous Systems*, **163**, 104384.

Chen, B., Cao, J., Parra, A. and Chin, T.-J. (2019a) Satellite pose estimation with deep landmark regression and nonlinear pose refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 0–0.

Chen, B., Zhang, R., Wang, S., Zhang, L. and Liu, Y. (2022a) Deep-learning-based inertial odometry for pedestrian tracking using attention mechanism and res2net module. *IEEE Sensors Letters*, **6**, 1–4.

Chen, C., Lu, C. X., Wahlström, J., Markham, A. and Trigoni, N. (2019b) Deep neural network based inertial odometry using low-cost inertial measurement units. *IEEE Transactions on Mobile Computing*, **20**, 1351–1364.

Chen, C., Lu, X., Markham, A. and Trigoni, N. (2018a) Ionet: Learning to cure the curse of drift in inertial odometry. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, ,.

Chen, C., Wang, B., Lu, C. X., Trigoni, N. and Markham, A. (2020a) A survey on deep learning for localization and mapping: Towards the age of spatial machine intelligence.

Chen, C., Zhao, P., Lu, C. X., Wang, W., Markham, A. and Trigoni, N. (2018b) Oxiod: The dataset for deep inertial odometry. *arXiv preprint arXiv:1809.07491*.

— (2020b) Deep-learning-based pedestrian inertial navigation: Methods, data set, and on-device inference. *IEEE Internet of Things Journal*, **7**, 4431–4441.

Chen, G., Qi, J. and Dai, Z. (2022b) Real-time maritime obstacle detection based on yolov5 for autonomous berthing. In *Bio-Inspired Computing: Theories and Applications: 16th International Conference, BIC-TA 2021, Taiyuan, China, December 17–19, 2021, Revised Selected Papers, Part II*, 412–427. Springer.

Chen, H., Taha, T. M. and Chodavarapu, V. P. (2022c) Towards improved inertial navigation by reducing errors using deep learning methodology. *Applied Sciences*, **12**, 3645.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A. L. (2017a) Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, **40**, 834–848.

Chen, L.-C., Papandreou, G., Schroff, F. and Adam, H. (2017b) Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

Chen, T., Pu, F., Chen, H. and Liu, Z. (2022d) Whuvid: A large-scale stereo-imu dataset for visual-inertial odometry and autonomous driving in chinese urban scenarios. *Remote Sensing*, **14**, 2033.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y. (2014) Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Choi, Y., Kim, N., Hwang, S., Park, K., Yoon, J. S., An, K. and Kweon, I. S. (2018) Kaist multi-spectral day/night data set for autonomous and assisted driving. *IEEE Transactions on Intelligent Transportation Systems*, **19**, 934–948.

Chumuang, N., Farooq, A., Irfan, M., Aziz, S. and Qureshi, M. (2022) Feature matching and deep learning models for attitude estimation on a micro-aerial vehicle. In *2022 International Conference on Cybernetics and Innovations (ICCI)*, 1–6. IEEE.

Ci, W., Xu, T., Lin, R. and Lu, S. (2022) A novel method for unexpected obstacle detection in the traffic environment based on computer vision. *Applied Sciences*, **12**, 8937.

Cioffi, G., Bauersfeld, L., Kaufmann, E. and Scaramuzza, D. (2022) Learned inertial odometry for autonomous drone racing. *arXiv preprint arXiv:2210.15287*.

Clark, R., Wang, S., Wen, H., Markham, A. and Trigoni, N. (2017) Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, ,.

Clarke, R. (2014) Understanding the drone epidemic. *Computer Law & Security Review*, **30**, 230–246.

Clevert, D.-A., Unterthiner, T. and Hochreiter, S. (2015) Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S. and Schiele, B. (2016) The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3213–3223.

Cortés, I., Beltrán, J., De La Escalera, A. and García, F. (2022) Dali: Domain adaptation in lidar point clouds for 3d obstacle detection. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 3837–3842. IEEE.

Cortés, S., Solin, A. and Kannala, J. (2018) Deep learning based speed estimation for constraining strapdown inertial navigation on smartphones. In *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, 1–6. IEEE.

Cui, J., Gong, K., Guo, N., Wu, C., Meng, X., Kim, K., Zheng, K., Wu, Z., Fu, L., Xu, B. et al. (2019) Pet image denoising using unsupervised deep learning. *European journal of nuclear medicine and molecular imaging*, **46**, 2780–2789.

Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T. and Nießner, M. (2017) Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5828–5839.

Dalal, N. and Triggs, B. (2005) Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1, 886–893. Ieee.

Dang, T.-V. and Bui, N.-T. (2023) Multi-scale fully convolutional network-based semantic segmentation for mobile robot navigation. *Electronics*, **12**, 533.

Del Prete, R., Saveriano, A. and Renga, A. (2022) A deep learning-based crater detector for autonomous vision-based spacecraft navigation. In *2022 IEEE 9th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, 231–236. IEEE.

Delmerico, J., Cieslewski, T., Rebecq, H., Faessler, M. and Scaramuzza, D. (2019) Are we ready for autonomous drone racing? the UZH-FPV drone racing dataset. In *IEEE Int. Conf. Robot. Autom. (ICRA)*.

Deng, L. and Platt, J. (2014) Ensemble deep learning for speech recognition. In *Proc. interspeech*, ,.

Deraz, A. A., Badawy, O., Elhosseini, M. A., Mostafa, M., Ali, H. A. and El-Desouky, A. I. (2022) Deep learning based on lstm model for enhanced visual odometry navigation system. *Ain Shams Engineering Journal*, 102050.

Deschaud, J.-E. (2021) Kitti-carla: a kitti-like dataset generated by carla simulator. *arXiv preprint arXiv:2109.00892*.

Devi, Y. S., Sathvik, S., Ananya, P., Tharuni, P. and Vamsi, N. N. K. (2022) Vision-based obstacle detection and collision prevention in self-driving cars. In *Journal of Physics: Conference Series*, vol. 2335, 012019. IOP Publishing.

Dewi, C., Chen, R.-C., Jiang, X. and Yu, H. (2022) Deep convolutional neural network for enhancing traffic sign recognition developed on yolo v4. *Multimedia Tools and Applications*, **81**, 37821–37845.

Dhuheir, M., Baccour, E., Erbad, A., Al-Obaidi, S. S. and Hamdi, M. (2022) Deep reinforcement learning for trajectory path planning and distributed inference in resource-constrained uav swarms. *IEEE Internet of Things Journal*.

Dijkstra, E. W. (1959) A note on two problems in connection with graphs. *Numerische Mathematik*, **1**, 269–271.

Dimas, G., Ntakolia, C. and Iakovidis, D. K. (2019) Obstacle detection based on generative adversarial networks and fuzzy sets for computer-assisted navigation. In *Engineering Applications of Neural Networks: 20th International Conference, EANN 2019, Xersonisos, Crete, Greece, May 24-26, 2019, Proceedings 20*, 533–544. Springer.

Ding, W. (2022) Attitude estimation methods using low-cost gnss and mems marg sensors and their integration.

Djenouri, Y., Hatleskog, J., Hjelmervik, J., Bjorne, E., Utstumo, T. and Mobarhan, M. (2022) Deep learning based decomposition for visual navigation in industrial platforms. *Applied Intelligence*, 1–17.

Donitz, B. P., Mages, D., Tsukamoto, H., Dixon, P., Landau, D., Chung, S.-J., Bufanda, E., Ingham, M. and Castillo-Rogez, J. (2022) Interstellar object accessibility and mission design. *arXiv preprint arXiv:2210.14980*.

Downes, L., Steiner, T. J. and How, J. P. (2020) Deep learning crater detection for lunar terrain relative navigation. In *AIAA SciTech 2020 Forum*, 1838.

Downes, L. M., Steiner, T. J. and How, J. P. (2021) Neural network approach to crater detection for lunar terrain relative navigation. *Journal of Aerospace Information Systems*, **18**, 391–403.

Du, H., Yu, X. and Zheng, L. (2021) Vtnet: Visual transformer network for object goal navigation. *arXiv preprint arXiv:2105.09447*.

Dugas, C., Bengio, Y., Bélisle, F., Nadeau, C. and Garcia, R. (2000) Incorporating second-order functional knowledge for better option pricing. *Advances in neural information processing systems*, **13**.

Ejaz, M. M., Tang, T. B. and Lu, C.-K. (2019) Autonomous visual navigation using deep reinforcement learning: An overview. In *2019 IEEE Student Conference on Research and Development (SCOReD)*, 294–299. IEEE.

El-Taher, F. E.-Z., Taha, A., Courtney, J. and Mckeever, S. (2021) A systematic review of urban navigation systems for visually impaired people. *Sensors*, **21**, 3103.

Elkins, J., Sood, R. and Rumpf, C. (2020) Autonomous spacecraft attitude control using deep reinforcement learning. In *71st International Astronautical Congress (IAC)*, vol. 2020.

Emmert-Streib, F., Yang, Z., Feng, H., Tripathi, S. and Dehmer, M. (2020) An introductory review of deep learning for prediction models with big data. *Frontiers in Artificial Intelligence*, **3**, 4.

Endo, Y., Javanmardi, E., Gu, Y. and Kamijo, S. (2023) High definition map aided object detection for autonomous driving in urban areas. *International Journal of Automotive Technology*, **24**, 469–481.

Engelsman, D. and Klein, I. (2022) Data-driven denoising of accelerometer signals. *arXiv preprint arXiv:2206.05937*.

Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S. and Dean, J. (2019) A guide to deep learning in healthcare. *Nature medicine*, **25**, 24–29.

Fauske, K. M., Gustafsson, F. and Hegrenaes, O. (2007) Estimation of auv dynamics for sensor fusion. In *2007 10th International Conference on Information Fusion*, 1–6. IEEE.

Felzenszwalb, P., McAllester, D. and Ramanan, D. (2008) A discriminatively trained, multiscale, deformable part model. In *2008 IEEE conference on computer vision and pattern recognition*, 1–8. Ieee.

Ferguson, D. and Stentz, A. (2006) Field d*: An intrinsically dynamic anytime search algorithm. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2552–2558. IEEE.

Finance, A., Meftah, M., Dufour, C., Boutéraon, T., Bekki, S., Hauchecorne, A., Keckhut, P., Sarkissian, A., Damé, L. and Mangin, A. (2021) A new method based on a multilayer perceptron network to determine in-orbit satellite attitude for spacecrafts without active adcs like uvsq-sat. *Remote Sensing*, **13**, 1185.

Franke, M., Reddy, C., Ristić-Durrant, D., Jayawardana, J., Michels, K., Banić, M. and Simonović, M. (2022) Towards holistic autonomous obstacle detection in railways by complementing of on-board vision with uav-based object localization. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7012–7019. IEEE.

Fu, X., Wu, M., Ponnarasu, S. and Zhang, L. (2023) A hybrid deep learning approach for dynamic attitude and position prediction in tunnel construction considering spatio-temporal patterns. *Expert Systems with Applications*, **212**, 118721.

Fukushima, K. (1975) Cognitron: A self-organizing multilayered neural network. *Biological cybernetics*, **20**, 121–136.

Furfaro, R., Bloise, I., Orlandelli, M., Di Lizia, P., Topputo, F., Linares, R. et al. (2018) Deep learning for autonomous lunar landing. *Advances in the Astronautical Sciences*, **167**, 3285–3306.

Gammell, J. D., Srinivasa, S. S. and Barfoot, T. D. (2014) Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1866–1873. IEEE.

— (2015) Batch informed trees (bit*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 3064–3071. IEEE.

Gao, L., Zhou, R. and Zhang, K. (2023) Scenario generation for autonomous vehicles with deep-learning-based heterogeneous driver models: implementation and verification. *Sensors*, **23**, 4570.

Gasperini, S., Haug, J., Mahani, M.-A. N., Marcos-Ramiro, A., Navab, N., Busam, B. and Tombari, F. (2021) Certainnet: Sampling-free uncertainty estimation for object detection. *IEEE Robotics and Automation Letters*, **7**, 698–705.

Gaudet, B., Linares, R. and Furfaro, R. (2020) Deep reinforcement learning for six degree-of-freedom planetary landing. *Advances in Space Research*, **65**, 1723–1741.

Ge, S. S. and Cui, Y. J. (2000) New potential functions for mobile robot path planning. *IEEE Transactions on robotics and automation*, **16**, 615–620.

Geiger, A., Lenz, P., Stiller, C. and Urtasun, R. (2013) Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, **32**, 1231–1237.

Geiger, A., Lenz, P. and Urtasun, R. (2012) Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, 3354–3361. IEEE.

George, A., Koivumäki, N., Hakala, T., Suomalainen, J. and Honkavaara, E. (2023) Visual-inertial odometry using high flying altitude drone datasets. *Drones*, **7**, 36.

George, A., Koivumäki, N., Hakala, T., Suomalainen, J. and Honkavaara, E. (2022) Fgi masala stereo-visual-inertial dataset 2021. `https://doi.org/10.23729/f4c5d2d0-eddb-40a1-89ce-601c252dab35`. National Land Survey of Finland, FGI Dept. of Remote sensing and photogrammetry.

Geromichalos, D., Azkarate, M., Tsardoulias, E., Gerdes, L., Petrou, L. and Perez Del Pulgar, C. (2020) Slam for autonomous planetary rovers with global localization. *Journal of Field Robotics*, **37**, 830–847.

Geyer, J., Kassahun, Y., Mahmudi, M., Ricou, X., Durgesh, R., Chung, A. S., Hauswald, L., Pham, V. H., Mühlegg, M., Dorn, S. et al. (2020) A2d2: Audi autonomous driving dataset. *arXiv preprint arXiv:2004.06320*.

Girshick, R. (2015) Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 1440–1448.

Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580–587.

Goldberg, D. E. (1989) *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Professional.

Golroudbari, A. A. and Sabour, M. H. (2023) End-to-end deep learning framework for real-time inertial attitude estimation using 6dof imu. *arXiv preprint arXiv:2302.06037*.

Gong, J., Ren, J. and Zhang, Y. (2021) Deepnav: A scalable and plug-and-play indoor navigation system based on visual cnn. *Peer-to-Peer Networking and Applications*, **14**, 3718–3736.

Goodfellow, I., Bengio, Y. and Courville, A. (2016) *Deep learning*. MIT press.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2020) Generative adversarial networks. *Communications of the ACM*, **63**, 139–144.

Grigorescu, S., Trasnea, B., Cocias, T. and Macesanu, G. (2020) A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, **37**, 362–386.

Grisetti, G., Kümmerle, R., Stachniss, C. and Burgard, W. (2010) A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, **2**, 31–43.

Guimarães, V., Sousa, I. and Correia, M. V. (2021) A deep learning approach for foot trajectory estimation in gait analysis using inertial sensors. *Sensors*, **21**, 7517.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. and Courville, A. C. (2017) Improved training of wasserstein gans. *Advances in neural information processing systems*, **30**.

Guo, Z., Chen, H., Li, S. et al. (2023) Deep reinforcement learning-based uav path planning for energy-efficient multitier cooperative computing in wireless sensor networks. *Journal of Sensors*, **2023**.

Guthrie, B., Kim, M., Urrutxua, H. and Hare, J. (2022) Image-based attitude determination of co-orbiting satellites using deep learning technologies. *Aerospace Science and Technology*, **120**, 107232.

Haith, A. M. and Krakauer, J. W. (2013) Model-based and model-free mechanisms of human motor learning. In *Progress in motor control: Neural, computational and dynamic approaches*, 1–21. Springer.

Han, L., Lin, Y., Du, G. and Lian, S. (2019) Deepvio: Self-supervised deep learning of monocular visual inertial odometry using 3d geometric constraints. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 6906–6913. IEEE.

Han, S., Meng, Z., Zhang, X. and Yan, Y. (2021) Hybrid deep recurrent neural networks for noise reduction of mems-imu with static and dynamic conditions. *Micromachines*, **12**, 214.

Hao, Z., Shyam, R. A., Rathinam, A. and Gao, Y. (2021) Intelligent spacecraft visual gnc architecture with the state-of-the-art ai components for on-orbit manipulation. *Frontiers in Robotics and AI*, **8**, 639327.

Haris, M. and Hou, J. (2020) Obstacle detection and safely navigate the autonomous vehicle from unexpected obstacles on the driving lane. *Sensors*, **20**, 4719.

Harris, A., Valade, T., Teil, T. and Schaub, H. (2022) Generation of spacecraft operations procedures using deep reinforcement learning. *Journal of Spacecraft and Rockets*, **59**, 611–626.

Hart, P. E., Nilsson, N. J. and Raphael, B. (1968) A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems, Science, and Cybernetics*, **4**, 100–107.

Hayashida, N. and Miwa, H. (2022) Obstacle detection support system using monocular camera. In *Advances in Intelligent Networking and Collaborative Systems: The 14th International Conference on Intelligent Networking and Collaborative Systems (INCoS-2022)*, 62–73. Springer.

He, C. (2020) Image compressive sensing via multi-scale feature extraction and attention mechanism. In *2020 International Conference on Intelligent Computing, Automation and Systems (ICICAS)*, 266–270. IEEE.

He, D., Qiu, Y., Miao, J., Zou, Z., Li, K., Ren, C. and Shen, G. (2022) Improved mask r-cnn for obstacle detection of rail transit. *Measurement*, **190**, 110728.

He, K., Ding, H., Xu, N. and Guo, K. (2023) Wheel odometry with deep learning-based error prediction model for vehicle localization. *Applied Sciences*, **13**, 5588.

He, K., Gkioxari, G., Dollár, P. and Girshick, R. (2017) Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

He, K., Zhang, X., Ren, S. and Sun, J. (2015a) Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 1026–1034.

— (2015b) Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, **37**, 1904–1916.

— (2016) Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hendrycks, D. and Gimpel, K. (2016) A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Herath, S., Yan, H. and Furukawa, Y. (2020) Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 3146–3152. IEEE.

Herrera, L., Jae Jun, K., Baker, J. and Agrawal, B. N. (2023) Deep learning based object attitude estimation for a laser beam control research testbed. *Applied Artificial Intelligence*, **37**, 2151191.

Hinton, G. E., Osindero, S. and Teh, Y.-W. (2006) A fast learning algorithm for deep belief nets. *Neural computation*, **18**, 1527–1554.

Hochreiter, S. (1998) The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, **6**, 107–116.

Hochreiter, S. and Schmidhuber, J. (1997) Long short-term memory. *Neural computation*, **9**, 1735–1780.

Hong, S., Bangunharcana, A., Park, J.-M., Choi, M. and Shin, H.-S. (2021) Visual slam-based robotic mapping method for planetary construction. *Sensors*, **21**, 7715.

Hou, C., Zhang, R., Yang, K., Li, X., Yang, Y., Ma, X., Guo, G., Yang, Y., Liu, L. and Zhou, F. (2023) Non-cooperative target attitude estimation method based on deep learning of ground and space access scene radar images. *Mathematics*, **11**, 745.

Hou, X. and Bergmann, J. H. (2023) Hinnet: Inertial navigation with head-mounted sensors using a neural network. *Engineering Applications of Artificial Intelligence*, **123**, 106066.

Houston, J., Zuidhof, G., Bergamini, L., Ye, Y., Chen, L., Jain, A., Omari, S., Iglovikov, V. and Ondruska, P. (2021) One thousand and one hours: Self-driving motion prediction dataset. In *Conference on Robot Learning*, 409–418. PMLR.

Hu, C., Wang, Y., Yu, G., Wang, Z., Lei, A. and Hu, Z. (2018) Embedding cnn-based fast obstacles detection for autonomous vehicles. *Tech. rep.*, SAE Technical Paper.

Hu, H., Liu, Z., Chitlangia, S., Agnihotri, A. and Zhao, D. (2022) Investigating the impact of multi-lidar placement on object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2550–2559.

Hu, L., Tang, Y., Zhou, Z. and Pan, W. (2021a) Reinforcement learning for orientation estimation using inertial sensors with performance guarantee. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 10243–10249. IEEE.

Hu, M., Mao, J., Li, J., Wang, Q. and Zhang, Y. (2021b) A novel lidar signal denoising method based on convolutional autoencoding deep learning neural network. *Atmosphere*, **12**, 1403.

Hu, Y., Speierer, S., Jakob, W., Fua, P. and Salzmann, M. (2021c) Wide-depth-range 6d object pose estimation in space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15870–15879.

Hu, Z., Fang, W., Gou, T., Wu, W., Hu, J., Zhou, S. and Mu, Y. (2019) A novel method based on a mask r-cnn model for processing dpcr images. *Analytical Methods*, **11**, 3410–3418.

Huan, W., Liu, M. and Hu, Q. (2020) Pose estimation for non-cooperative spacecraft based on deep learning. In *2020 39th Chinese Control Conference (CCC)*, 3339–3343. IEEE.

Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K. Q. (2017) Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Huang, W., Zhou, Y., He, X. and Lv, C. (2023) Goal-guided transformer-enabled reinforcement learning for efficient autonomous navigation. *arXiv preprint arXiv:2301.00362*.

Huang, X., Cheng, X., Geng, Q., Cao, B., Zhou, D., Wang, P., Lin, Y. and Yang, R. (2018) The apolloscape dataset for autonomous driving. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 954–960.

Huu, P. N., Pham Thi, Q. and Tong Thi Quynh, P. (2022) Proposing lane and obstacle detection algorithm using yolo to control self-driving cars on advanced networks. *Advances in Multimedia*, **2022**.

Jagtap, S. D. and Kanade, S. S. (2022) Improving run time efficiency of semantic video event classification. *International Journal of Intelligent Transportation Systems Research*, 1–14.

Jain, D. K., Dutta, A. K., Verdú, E., Alsubai, S. and Sait, A. R. W. (2023) An automated hyperparameter tuned deep learning model enabled facial emotion recognition for autonomous vehicle drivers. *Image and Vision Computing*, **133**, 104659.

Janson, N., Schmerling, E., Clark, A. and Pavone, M. (2015) Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 4597–4604. IEEE.

Jensen, M. B., Philipsen, M. P., Møgelmose, A., Moeslund, T. B. and Trivedi, M. M. (2016) Vision for looking at traffic lights: Issues, survey, and perspectives. *IEEE Transactions on Intelligent Transportation Systems*, **17**, 1800–1815.

Jeong, H. and Lee, H. (2023) Cnn-based fault detection of scan matching for accurate slam in dynamic environments. *Sensors*, **23**, 2940.

Ji, C., Wu, H. and Liu, G. (2023) Probabilistic instance shape reconstruction with sparse lidar for monocular 3d object detection. *Neurocomputing*, **529**, 92–100.

Ji, P., Wang, X., Ma, F., Feng, J. and Li, C. (2022) A 3d hand attitude estimation method for fixed hand posture based on dual-view rgb images. *Sensors*, **22**, 8410.

Jiang, C., Chen, S., Chen, Y., Zhang, B., Feng, Z., Zhou, H. and Bo, Y. (2018) A mems imu de-noising method using long short term memory recurrent neural networks (lstm-rnn). *Sensors*, **18**, 3470.

Jiao, J., Jiao, J., Mo, Y., Liu, W. and Deng, Z. (2019) Magicvo: An end-to-end hybrid cnn and bi-lstm method for monocular visual odometry. *IEEE Access*, **7**, 94118–94127.

Jiao, S. and Wang, L. (2021) Road obstacle detection in bad weather based on deep learning. In *Journal of Physics: Conference Series*, vol. 1881, 042041. IOP Publishing.

Jocher, G., Chaurasia, A. and Qiu, J. (2023) Yolo by ultralytics. URL: `https://github.com/ultralytics/ultralytics`.

Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., NanoCode012, Kwon, Y., Michael, K., TaoXie, Fang, J., imyhxy, Lorna, Yifu, Z., Wong, C., V, A., Montes, D., Wang, Z., Fati, C., Nadar, J., Laughing, UnglvKitDe, Sonck, V., tkianai, yxNONG, Skalski, P., Hogan, A., Nair, D., Strobel, M. and Jain, M. (2022) Yolov5 sota realtime instance segmentation. URL: `https://doi.org/10.5281/zenodo.7347926`.

Jocher, G., Stoken, A., Borovec, J., Chaurasia, A. and Changyu, L. (2020) ultralytics/yolov5. *Github Repository, YOLOv5*.

Justesen, N., Bontrager, P., Togelius, J. and Risi, S. (2019) Deep learning for video game playing. *IEEE Transactions on Games*, **12**, 1–20.

Karaman, S. and Frazzoli, E. (2011) Sampling-based algorithms for optimal motion planning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 961–976. IEEE.

Kavraki, L. E., Svestka, P., Latombe, J.-C. and Overmars, M. H. (1996) Probabilistic roadmaps for path planning in high-dimensional configuration spaces. In *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, 1663–1668. IEEE.

Kawai, H. and Kuroda, Y. (2023) Dnn based camera attitude estimation using aggregated information from camera and depth images. In *2023 IEEE/SICE International Symposium on System Integration (SII)*, 1–6. IEEE.

Kawai, H., Yoshiuchi, W., Hirakawa, Y., Shibuya, T., Matsuda, T. and Kuroda, Y. (2022) Camera attitude estimation by neural network using classification network method instead of numerical regression. In *2022 IEEE/SICE International Symposium on System Integration (SII)*, 401–407. IEEE.

Kesten, R., Usman, M., Houston, J., Pandya, T., Nadhamuni, K., Ferreira, A., Yuan, M., Low, B., Jain, A., Ondruska, P. et al. (2019) Lyft level 5 av dataset 2019. *urlhttps://level5. lyft. com/dataset*, **1**, 3.

Khatib, O. (1986) Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, **5**, 90–98.

Klambauer, G., Unterthiner, T., Mayr, A. and Hochreiter, S. (2017) Self-normalizing neural networks. *Advances in neural information processing systems*, **30**.

Koenig, S. and Likhachev, M. (2002) D* lite. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 2, 476–483.

— (2004) Lifelong planning a*. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 4, 1510–1515.

Konda, K. R. and Memisevic, R. (2015) Learning visual odometry with a convolutional network. *VISAPP (1)*, **2015**, 486–490.

Kontoudis, G. P. and Vamvoudakis, K. G. (2019) Kinodynamic motion planning with continuous-time q-learning: An online, model-free, and safe navigation framework. *IEEE transactions on neural networks and learning systems*, **30**, 3803–3817.

Kouris, A. and Bouganis, C.-S. (2018) Learning to fly by myself: A self-supervised cnn-based approach for autonomous navigation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1–9. IEEE.

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2017) Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, **60**, 84–90.

Kurdi, M. M., Dadykin, A. K., Elzein, I. and Ahmad, I. S. (2018) Proposed system of artificial neural network for positioning and navigation of uav-ugv. In *2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)*, 1–6. IEEE.

Kuriakose, B., Shrestha, R. and Sandnes, F. E. (2023) Deepnavi: A deep learning based smartphone navigation assistant for people with visual impairments. *Expert Systems with Applications*, **212**, 118720.

Laidig, D., Caruso, M., Cereatti, A. and Seel, T. (2021) Broad—a benchmark for robust inertial orientation estimation. *Data*, **6**, 72.

Lamberti, L., Bompani, L., Kartsch, V. J., Rusci, M., Palossi, D. and Benini, L. (2023) Bio-inspired autonomous exploration policies with cnn-based object detection on nano-drones. *arXiv preprint arXiv:2301.12175*.

Lang, R., Fan, Y. and Chang, Q. (2023) Svr-net: A sparse voxelized recurrent network for robust monocular slam with direct tsdf mapping. *Sensors*, **23**, 3942.

Langer, F., Milioto, A., Haag, A., Behley, J. and Stachniss, C. (2020) Domain transfer for semantic segmentation of lidar data using deep neural networks. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 8263–8270. IEEE.

LaValle, S. M. (1998) Rapidly-exploring random trees: A new tool for path planning. *Tech. rep.*, Carnegie Mellon University, Robotics Institute.

Law, H. and Deng, J. (2018) Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, 734–750.

Lee, D.-H. and Liu, J.-L. (2023) End-to-end deep learning of lane detection and path prediction for real-time autonomous driving. *Signal, Image and Video Processing*, **17**, 199–205.

Lee, G. H., Achtelik, M., Fraundorfer, F., Pollefeys, M. and Siegwart, R. (2010) A benchmarking tool for mav visual pose estimation. In *2010 11th International Conference on Control Automation Robotics & Vision*, 1541–1546. IEEE.

Lee, G. X. and Low, K.-S. (2012) A factorized quaternion approach to determine the arm motions using triaxial accelerometers with anatomical and sensor constraints. *IEEE Transactions on Instrumentation and Measurement*, **61**, 1793–1802.

Lee, T., Mckeever, S. and Courtney, J. (2021) Flying free: A research overview of deep learning in drone navigation autonomy. *Drones*, **5**, 52.

Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W. et al. (2022a) Yolov6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*.

Li, F., Li, Q., Zhang, T., Niu, Y. and Shi, G. (2019) Depth acquisition with the combination of structured light and deep learning stereo matching. *Signal Processing: Image Communication*, **75**, 111–117.

Li, F.-F., Johnson, J. and Yeung, S. (2017) Lecture 11: Detection and segmentation.

Li, H., Li, Z., Akmandor, N. U., Jiang, H., Wang, Y. and Padir, T. (2022b) Stereovoxelnet: Real-time obstacle detection based on occupancy voxels from a stereo camera using deep neural networks. *arXiv preprint arXiv:2209.08459*.

Li, R., Wang, S., Long, Z. and Gu, D. (2018) Undeepvo: Monocular visual odometry through unsupervised deep learning. In *2018 IEEE international conference on robotics and automation (ICRA)*, 7286–7291. IEEE.

Li, X., Liu, W. and Chen, Q. (2021a) Intelligent extraction method of inertial navigation trajectory behavior features considering road environment. In *Spatial Data and Intelligence: Second International Conference, SpatialDI 2021, Hangzhou, China, April 22–24, 2021, Proceedings 2*, 43–56. Springer.

Li, X., Xie, Z., Deng, X., Wu, Y. and Pi, Y. (2022c) Traffic sign detection based on improved faster r-cnn for autonomous driving. *The Journal of Supercomputing*, 1–21.

Li, Y., Chen, R., Niu, X., Zhuang, Y., Gao, Z., Hu, X. and El-Sheimy, N. (2020) Inertial sensing meets artificial intelligence: Opportunity or challenge? *arXiv preprint arXiv:2007.06727*.

Li, Z., Liu, F., Yang, W., Peng, S. and Zhou, J. (2021b) A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*.

Lian, Y., Sun, H. and Dong, S. (2023) Point–line-aware heterogeneous graph attention network for visual slam system. *Applied Sciences*, **13**, 3816.

Likhachev, M., Ferguson, D., Gordon, G. and Stentz, A. (2008) Anytime search-based motion planning with (unbounded) error bounds. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2362–2369. IEEE.

Lim, M., Mousavi, P., Sirovljevic, J. and You, H. (2020) Onboard artificial intelligence for space situational awareness with low-power gpus. In *Advanced Maui Optical and Space Surveillance Technologies Conference*.

Lima, A. A., Kabir, M. M., Das, S. C., Hasan, M. N. and Mridha, M. (2022) Road sign detection using variants of yolo and r-cnn: An analysis from the perspective of bangladesh. In *Proceedings of the International Conference on Big Data, IoT, and Machine Learning: BIM 2021*, 555–565. Springer.

Lin, G., Milan, A., Shen, C. and Reid, I. (2017a) Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1925–1934.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B. and Belongie, S. (2017b) Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2117–2125.

Lin, T.-Y., Goyal, P., Girshick, R., He, K. and Dollár, P. (2017c) Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C. L. (2014) Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. Springer.

Liu, C. and Reif, J. H. (2011) Lazy theta*: Any-angle path planning and path length analysis in 3d. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2075–2082. IEEE.

Liu, K. and Cao, M. (2023) Dlc-slam: A robust lidar-slam system with learning-based denoising and loop closure. *IEEE/ASME Transactions on Mechatronics*.

Liu, L., Tian, L., Kang, Z. and Wan, T. (2023) Spacecraft anomaly detection with attention temporal convolution networks. *Neural Computing and Applications*, 1–9.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. and Berg, A. C. (2016) Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, 21–37. Springer.

Liu, W., Caruso, D., Ilg, E., Dong, J., Mourikis, A. I., Daniilidis, K., Kumar, V. and Engel, J. (2020) Tlio: Tight learned inertial odometry. *IEEE Robotics and Automation Letters*, **5**, 5653–5660.

Liu, Y., Luo, Q. and Zhou, Y. (2022a) Deep learning-enabled fusion to bridge gps outages for ins/gps integrated navigation. *IEEE Sensors Journal*, **22**, 8974–8985.

Liu, Y., Zhu, M., Wang, J., Guo, X., Yang, Y. and Wang, J. (2022b) Multi-scale deep neural network based on dilated convolution for spacecraft image segmentation. *Sensors*, **22**, 4222.

Liu, Z., Amini, A., Zhu, S., Karaman, S., Han, S. and Rus, D. L. (2021) Efficient and robust lidar-based end-to-end navigation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 13247–13254. IEEE.

Long, J., Shelhamer, E. and Darrell, T. (2015) Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431–3440.

Lozano-Pérez, T. and Wesley, M. A. (1979) An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, **22**, 560–570.

Ludwig, B., Donabauer, G., Ramsauer, D. and Subari, K. a. (2023) Urwalking: Indoor navigation for research and daily use. *KI-Künstliche Intelligenz*, 1–8.

Luu, V.-T., Huynh, V.-C., Tran, V.-H., Nguyen, T.-H. et al. (2020) Traditional method meets deep learning in an adaptive lane and obstacle detection system. In *2020 5th International Conference on Green Technology and Sustainable Development (GTSD)*, 148–152. IEEE.

Lyu, P., Wei, M. and Wu, Y. (2022) High-precision and real-time visual tracking algorithm based on the siamese network for autonomous driving. *Signal, Image and Video Processing*, 1–9.

Ma, W., Wang, X. and Wang, B. (2022) Fast obstacle detection platform based on yolo. In *Proceedings of the 5th International Conference on Electrical Engineering and Information Technologies for Rail Transportation (EITRT) 2021: Rail Transportation System Safety and Maintenance Technologies*, 218–226. Springer.

Maas, A. L., Hannun, A. Y., Ng, A. Y. et al. (2013) Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, vol. 30, 3. Atlanta, Georgia, USA.

Mac, T. T., Copot, C., Tran, D. T. and De Keyser, R. (2016) Heuristic approaches in robot path planning: A survey. *Robotics and Autonomous Systems*, **86**, 13–28.

Maddern, W., Pascoe, G., Linegar, C. and Newman, P. (2017) 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, **36**, 3–15.

Mahaur, B. and Mishra, K. (2023) Small-object detection based on yolov5 in autonomous driving systems. *Pattern Recognition Letters*, **168**, 115–122.

Mahendrakar, T., Ekblad, A., Fischer, N., White, R., Wilde, M., Kish, B. and Silver, I. (2022) Performance study of yolov5 and faster r-cnn for autonomous navigation around non-cooperative targets. In *2022 IEEE Aerospace Conference (AERO)*, 1–12. IEEE.

Mahendrakar, T., White, R. T., Wilde, M. and Tiwari, M. (2023) Spaceyolo: A human-inspired model for real-time, on-board spacecraft feature detection. *arXiv preprint arXiv:2302.00824*.

Majdik, A. L., Till, C. and Scaramuzza, D. (2017) The zurich urban micro aerial vehicle dataset. *The International Journal of Robotics Research*, **36**, 269–273.

Martinez Acosta, A. (2023) Drone navigation using octrees and object recognition for intelligent inspections. In *AIAA SCITECH 2023 Forum*, 1075.

Matsumoto, T., Yokohama, T., Suzuki, H., Furukawa, R., Oshimoto, A., Shimmi, T., Matsushita, Y., Seo, T. and Chua, L. (1990) Several image processing examples by cnn. In *IEEE International Workshop on Cellular Neural Networks and their Applications*, 100–111. IEEE.

Mayne, D. Q., Rawlings, J. B., Rao, C. V. and Scokaert, P. O. (2014) *Model predictive control: Recent developments and future directions*. Springer.

Mayo, B., Hazan, T. and Tal, A. (2021) Visual navigation with spatial attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16898–16907.

McCulloch, W. S. and Pitts, W. (1943) A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, **5**, 115–133.

Merkert, R. and Bushell, J. (2020) Revolution or epidemic? a systematic literature review on the effective control of airborne drones.

Misra, D. (2019) Mish: A self regularized non-monotonic neural activation function. *arXiv preprint arXiv:1908.08681*, **4**, 10–48550.

Moghe, R. and Zanetti, R. (2020) On-line hazard detection algorithm for precision lunar landing using semantic segmentation. In *AIAA Scitech 2020 Forum*, 0462.

Mohammadi, M., Al-Fuqaha, A. and Oh, J.-S. (2018) Path planning in support of smart mobility applications using generative adversarial networks. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 878–885. IEEE.

Mohsan, S. A. H., Khan, M. A., Noor, F., Ullah, I. and Alsharif, M. H. (2022) Towards the unmanned aerial vehicles (uavs): A comprehensive review. *Drones*, **6**, 147.

Silva do Monte Lima, J. P., Uchiyama, H. and Taniguchi, R.-i. (2019) End-to-end learning framework for imu-based 6-dof odometry. *Sensors*, **19**, 3777.

Mortlock, T. and Kassas, Z. M. (2021) Assessing machine learning for leo satellite orbit determination in simultaneous tracking and navigation. In *2021 IEEE Aerospace Conference (50100)*, 1–8. IEEE.

Mueller, M., Smith, N. and Ghanem, B. (2016) A benchmark and simulator for uav tracking. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, 445–461. Springer.

Mukhachev, P. A., Sadretdinov, T. R., Pritykin, D. A., Ivanov, A. B. and Solov'ev, S. (2021) Modern machine learning methods for telemetry-based spacecraft health monitoring. *Automation and Remote Control*, **82**, 1293–1320.

Muller, P. and Savakis, A. (2017) Flowdometry: An optical flow and deep learning based approach to visual odometry. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 624–631. IEEE.

Mur-Artal, R. and Tardós, J. D. (2017) Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, **33**, 1255–1262.

Muscettola, N., Dorais, G. A., Fry, C., Levinson, R., Plaunt, C. and Clancy, D. (2002) Idea: Planning at the core of autonomous reactive agents. In *Sixth International Conference on AI Planning and Scheduling*, ,.

Nair, V. and Hinton, G. E. (2010) Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, 807–814.

Nakka, Y. K. (2021) *Spacecraft Motion Planning and Control under Probabilistic Uncertainty for Coordinated Inspection and Safe Learning*. Ph.D. thesis, California Institute of Technology.

Narkhede, P., Walambe, R., Poddar, S. and Kotecha, K. (2021) Incremental learning of lstm framework for sensor fusion in attitude estimation. *PeerJ Computer Science*, **7**, e662.

Nash, A. and Koenig, S. (2007) Theta*: Any-angle path planning on grids. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 7, 1177–1183.

Neamati, D., Nakka, Y. K. K. and Chung, S.-J. (2022) Learning-based methods to model small body gravity fields for proximity operations: Safety and robustness. In *AIAA SCITECH 2022 Forum*, 2271.

Nevavuori, P., Narra, N., Linna, P. and Lipping, T. (2020) Crop yield prediction using multitemporal uav data and spatio-temporal deep learning models. *Remote Sensing*, **12**, 4000.

Ni, J., Chen, Y., Chen, Y., Zhu, J., Ali, D. and Cao, W. (2020) A survey on theories and applications for self-driving cars based on deep learning methods. *Applied Sciences*, **10**, 2749.

Nie, C., Qadar, M. A., Zhou, S., Zhang, H., Shi, Y., Gao, J. and Sun, Z. (2022) Transnational image object detection datasets from nighttime driving. *Signal, Image and Video Processing*, 1–9.

Niroui, F., Zhang, K., Kashino, Z. and Nejat, G. (2019) Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments. *IEEE Robotics and Automation Letters*, **4**, 610–617.

Nistér, D., Naroditsky, O. and Bergen, J. (2004) Visual odometry. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, I–I. Ieee.

Nurmaini, S., Darmawahyuni, A., Sakti Mukti, A. N., Rachmatullah, M. N., Firdaus, F. and Tutuko, B. (2020) Deep learning-based stacked denoising and autoencoder for ecg heartbeat classification. *Electronics*, **9**, 135.

Oestreich, C. E., Linares, R. and Gondhalekar, R. (2021) Autonomous six-degree-of-freedom spacecraft docking with rotating targets via reinforcement learning. *Journal of Aerospace Information Systems*, **18**, 417–428.

Ohno, K., Tsubouchi, T., Shigematsu, B. and Yuta, S. (2004) Differential gps and odometry-based outdoor navigation of a mobile robot. *Advanced Robotics*, **18**, 611–635.

OpenAI (2023) Gpt-4 technical report. *arXiv*.

Otte, M. W. (2015) A survey of machine learning approaches to robotic path-planning. *University of Colorado at Boulder, Boulder*.

Otter, D. W., Medina, J. R. and Kalita, J. K. (2020) A survey of the usages of deep learning for natural language processing. *IEEE transactions on neural networks and learning systems*, **32**, 604–624.

Ozaki, R. and Kuroda, Y. (2021) Dnn-based self-attitude estimation by learning landscape information. In *2021 IEEE/SICE International Symposium on System Integration (SII)*, 733–738. IEEE.

Ozaki, R., Sugiura, N. and Kuroda, Y. (2021) Lidar dnn based self-attitude estimation with learning landscape regularities. *ROBOMECH Journal*, **8**, 1–12.

O'Mahony, N., Campbell, S., Krpalkova, L., Riordan, D., Walsh, J., Murphy, A. and Ryan, C. (2018) Deep learning for visual navigation of unmanned ground vehicles: A review. In *2018 29th Irish Signals and Systems Conference (ISSC)*, 1–6. IEEE.

Pang, T., Zheng, H., Quan, Y. and Ji, H. (2021) Recorrupted-to-recorrupted: unsupervised deep learning for image denoising. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2043–2052.

Paravarzar, S. and Mohammad, B. (2020) Motion prediction on self-driving cars: A review. *arXiv preprint arXiv:2011.03635*.

Park, T. H. and D'Amico, S. (2022a) Adaptive neural network-based unscented kalman filter for spacecraft pose tracking at rendezvous. *arXiv preprint arXiv:2206.03796*.

— (2022b) Robust multi-task learning and online refinement for spacecraft pose estimation across domain gap. *arXiv preprint arXiv:2203.04275*.

Park, T. H., Sharma, S. and D'Amico, S. (2019) Towards robust learning-based pose estimation of noncooperative spacecraft. *arXiv preprint arXiv:1909.00392*.

Paszke, A., Chaurasia, A., Kim, S. and Culurciello, E. (2016) Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*.

Patil, A., Malla, S., Gang, H. and Chen, Y.-T. (2019) The h3d dataset for full-surround 3d multi-object detection and tracking in crowded urban scenes. In *2019 International Conference on Robotics and Automation (ICRA)*, 9552–9557. IEEE.

Peng, Z., Peng, S., Fu, L., Lu, B., Tang, J., Wang, K. and Li, W. (2020) A novel deep learning ensemble model with data denoising for short-term wind speed forecasting. *Energy Conversion and Management*, **207**, 112524.

Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M. and Sorkine-Hornung, A. (2016) A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 724–732.

Pham, Q.-H., Sevestre, P., Pahwa, R. S., Zhan, H., Pang, C. H., Chen, Y., Mustafa, A., Chandrasekhar, V. and Lin, J. (2020) A 3d dataset: Towards autonomous driving in challenging environments. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2267–2273. IEEE.

Pinggera, P., Ramos, S., Gehrig, S., Franke, U., Rother, C. and Mester, R. (2016) Lost and found: detecting small road hazards for self-driving vehicles. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1099–1106. IEEE.

Plascencia, A. C., García-Gómez, P., Perez, E. B., DeMas-Giménez, G., Casas, J. R. and Royo, S. (2023) A preliminary study of deep learning sensor fusion for pedestrian detection. *Sensors*, **23**, 4167.

Prabhakar, G., Kailath, B., Natarajan, S. and Kumar, R. (2017) Obstacle detection and classification using deep learning for tracking in high-speed autonomous driving. In *2017 IEEE region 10 symposium (TENSYMP)*, 1–6. IEEE.

Pranav, M., Shreyas Madhav, A. and Meena, J. (2022) Deeprecog: Threefold underwater image deblurring and object recognition framework for auv vision systems. *Multimedia Systems*, 1–11.

Proença, P. F. and Gao, Y. (2020) Deep learning for spacecraft pose estimation from photorealistic rendering. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 6007–6013. IEEE.

Qin, J., Li, M., Li, D., Zhong, J. and Yang, K. (2022) A survey on visual navigation and positioning for autonomous uuvs. *Remote Sensing*, **14**, 3794.

Qin, T., Chen, T., Chen, Y. and Su, Q. (2020) Avp-slam: Semantic visual mapping and localization for autonomous vehicles in the parking lot. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5939–5945. IEEE.

Qiu, Z., Zhao, N., Zhou, L., Wang, M., Yang, L., Fang, H., He, Y. and Liu, Y. (2020) Vision-based moving obstacle detection and tracking in paddy field using improved yolov3 and deep sort. *Sensors*, **20**, 4082.

Radoglou-Grammatikis, P., Sarigiannidis, P., Lagkas, T. and Moscholios, I. (2020) A compilation of uav applications for precision agriculture. *Computer Networks*, **172**, 107148.

Raja, G., Suresh, S., Anbalagan, S., Ganapathisubramaniyan, A. and Kumar, N. (2021) Pfin: An efficient particle filter-based indoor navigation framework for uavs. *IEEE Transactions on Vehicular Technology*, **70**, 4984–4992.

Rajkumar, R. (2022) Deep learning feature extraction using attention-based densenet 121 for copy move forgery detection. *International Journal of Image and Graphics*, 2350042.

Ramachandran, P., Zoph, B. and Le, Q. V. (2017) Searching for activation functions. *arXiv preprint arXiv:1710.05941*.

Ramadhan, S. F., Taufik, M., Novita, D. and Turnip, A. (2021) Design of 2d lidar-based indoor slam for medical robot covid-19. In *2021 International Conference on Artificial Intelligence and Mechatronics Systems (AIMS)*, 1–5. IEEE.

Rao, Q. and Frtunikj, J. (2018) Deep learning for self-driving cars: Chances and challenges. In *Proceedings of the 1st international workshop on software engineering for AI in autonomous systems*, 35–38.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016) You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788.

Redmon, J. and Farhadi, A. (2017) Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7263–7271.

— (2018) Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Ren, S., He, K., Girshick, R. and Sun, J. (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, **28**.

Rezwan, S. and Choi, W. (2022) Artificial intelligence approaches for uav navigation: Recent advances and future challenges. *IEEE Access*.

Riden, A., Roy, D., Pasiliao, E. and Mukherjee, T. (2021) Deepoe: Deep learning for position and orientation estimation. In *2021 26th IEEE Asia-Pacific Conference on Communications (APCC)*, 235–242. IEEE.

Roberts, T. G. and Linares, R. (2021) Geosynchronous satellite maneuver classification via supervised machine learning. In *Advanced Maui Optical and Space Surveillance Technologies Conference. Maui, Hawaii*.

Roberts, T. G., Siew, P. M., Jang, D. and Linares, R. (2021) A deep reinforcement learning application to space-based sensor tasking for space situational awareness. In *Proceedings of the 2021 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS), Wailea Beach Resort, Maui, HI, USA*, 14–17.

Robicquet, A., Sadeghian, A., Alahi, A. and Savarese, S. (2016) Learning social etiquette: Human trajectory prediction. In *European Conference on Computer Vision (ECCV)*, vol. 3.

Rondao, D., Aouf, N. and Richardson, M. A. (2022) Chinet: Deep recurrent convolutional learning for multimodal spacecraft pose estimation. *IEEE Transactions on Aerospace and Electronic Systems*.

Ronneberger, O., Fischer, P. and Brox, T. (2015) U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, 234–241. Springer.

Rosenblatt, F. (1958) The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, **65**, 386.

Roy, D., Ishizaka, T., Mohan, C. K. and Fukuda, A. (2019a) Vehicle trajectory prediction at intersections using interaction based generative adversarial networks. In *2019 IEEE Intelligent transportation systems conference (ITSC)*, 2318–2323. IEEE.

Roy, P. and Chowdhury, C. (2021) A survey of machine learning techniques for indoor localization and navigation systems. *Journal of Intelligent & Robotic Systems*, **101**, 63.

Roy, S. K., Manna, S., Dubey, S. R. and Chaudhuri, B. B. (2019b) Lisht: Non-parametric linearly scaled hyperbolic tangent activation function for neural networks. *arXiv preprint arXiv:1901.05894*.

Ruff, K. M. and Pappu, R. V. (2021) Alphafold and implications for intrinsically disordered proteins. *Journal of Molecular Biology*, **433**, 167208.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986) Learning representations by back-propagating errors. *nature*, **323**, 533–536.

Sadeghi Esfahlani, S., Sanaei, A., Ghorabian, M. and Shirvani, H. (2022) The deep convolutional neural network role in the autonomous navigation of mobile robots (srobo). *Remote Sensing*, **14**, 3324.

Saha, S. S., Sandha, S. S., Garcia, L. A. and Srivastava, M. (2022) Tinyodom: Hardware-aware efficient neural inertial navigation. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, **6**, 1–32.

Santos, R. R., Rade, D. A. and da Fonseca, I. M. (2022) A machine learning strategy for optimal path planning of space robotic manipulator in on-orbit servicing. *Acta Astronautica*, **191**, 41–54.

Sariff, N. and Buniyamin, N. (2006) An overview of autonomous mobile robot path planning algorithms. In *2006 4th student conference on research and development*, 183–188. IEEE.

Schmidhuber, J. (2015) Deep learning in neural networks: An overview. *Neural networks*, **61**, 85–117.

Schubert, D., Goll, T., Demmel, N., Usenko, V., Stückler, J. and Cremers, D. (2018) The tum vi benchmark for evaluating visual-inertial odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1680–1687. IEEE.

Scorsoglio, A., D'Ambrosio, A., Ghilardi, L., Furfaro, R., Gaudet, B., Linares, R. and Curti, F. (2020) Safe lunar landing via images: A reinforcement meta-learning application to autonomous hazard avoidance and landing. In *Proceedings of the 2020 AAS/AIAA Astrodynamics Specialist Conference, Virtual*, 9–12.

Scorsoglio, A., Furfaro, R., Linares, R. and Massari, M. (2023) Relative motion guidance for near-rectilinear lunar orbits with path constraints via actor-critic reinforcement learning. *Advances in Space Research*, **71**, 316–335.

Shahin, A. I. and Almotairi, S. (2021) Dcrn: an optimized deep convolutional regression network for building orientation angle estimation in high-resolution satellite images. *Electronics*, **10**, 2970.

Shan, T., Wang, Y., Zhao, C., Li, Y., Zhang, G. and Zhu, Q. (2023) Multi-uav wrsn charging path planning based on improved heed and ia-drl. *Computer Communications*, **203**, 77–88.

Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J. and Sun, J. (2019) Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 8430–8439.

Shin, D., Cho, J. and Kim, J. (2022) Environment-adaptive object detection framework for autonomous mobile robots. *Sensors*, **22**, 7647.

Shoushtari, H., Kassawat, F., Harder, D., Venzke, K., Müller-Lietzkow, J. and Sternberg, H. (2022) L5in+: From an analytical platform to optimization of deep inertial odometry. In *12th International Conference on Indoor Positioning and Indoor Navigation-Work-in-Progress Papers (IPIN-WiP 2022), 5-7 September 2022, Beijing, China.*, ,. CEUR-WS.

Shustanov, A. and Yakimov, P. (2017) Cnn design for real-time traffic sign recognition. *Procedia engineering*, **201**, 718–725.

Shuttleworth, J. (2019) Sae standards news: J3016 automated-driving graphic update. *SAE International*. URL: `https://www.sae.org/news/2019/01/sae-updates-j3016-automated-driving-graphic`.

Siew, P. M., Jang, D., Roberts, T. G. and Linares, R. (2022) Space-based sensor tasking using deep reinforcement learning. *The Journal of the Astronautical Sciences*, 1–38.

Siew, P. M. and Linares, R. (2022) Optimal tasking of ground-based sensors for space situational awareness using deep reinforcement learning. *Sensors*, **22**, 7847.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M. et al. (2016) Mastering the game of go with deep neural networks and tree search. *nature*, **529**, 484–489.

Silvestrini, S. and Lavagna, M. (2022) Deep learning and artificial neural networks for spacecraft dynamics, navigation and control. *Drones*, **6**, 270.

Simonyan, K. and Zisserman, A. (2014) Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Singh, S. K. and Junkins, J. L. (2022) Stochastic learning and extremal-field map based autonomous guidance of low-thrust spacecraft. *Scientific Reports*, **12**, 17774.

Sleaman, W. K., Hameed, A. A. and Jamil, A. (2023) Monocular vision with deep neural networks for autonomous mobile robots navigation. *Optik*, **272**, 170162.

Solin, A., Cortes, S., Rahtu, E. and Kannala, J. (2018) Inertial odometry on handheld smartphones. In *2018 21st International Conference on Information Fusion (FUSION)*, 1–5. IEEE.

Song, J., Rondao, D. and Aouf, N. (2022a) Deep learning-based spacecraft relative navigation methods: A survey. *Acta Astronautica*, **191**, 22–40.

Song, S., Lichtenberg, S. P. and Xiao, J. (2015) Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 567–576.

Song, S., Lim, H., Lee, A. J. and Myung, H. (2022b) Dynavins: A visual-inertial slam for dynamic environments. *IEEE Robotics and Automation Letters*, **7**, 11523–11530.

Soyer, M. S., Abdel-Qader, A. and Onbaşlı, M. C. (2021) An efficient and low-latency deep inertial odometer for smartphone positioning. *IEEE Sensors Journal*, **21**, 27676–27685.

Spantideas, S. T., Giannopoulos, A. E., Kapsalis, N. C. and Capsalis, C. N. (2021) A deep learning method for modeling the magnetic signature of spacecraft equipment using multiple magnetic dipoles. *IEEE Magnetics Letters*, **12**, 1–5.

Stacey, N. and D'Amico, S. (2022) Robust autonomous spacecraft navigation and environment characterization. *Authorea Preprints*.

Sturm, J., Engelhard, N., Endres, F., Burgard, W. and Cremers, D. (2012) A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, 573–580. IEEE.

Sun, K., Mohta, K., Pfrommer, B., Watterson, M., Liu, S., Mulgaonkar, Y., Taylor, C. J. and Kumar, V. (2018) Robust stereo visual inertial odometry for fast autonomous flight. *IEEE Robotics and Automation Letters*, **3**, 965–972.

Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B. et al. (2020) Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2446–2454.

Sun, P., Zhang, R., Jiang, Y., Kong, T., Xu, C., Zhan, W., Tomizuka, M., Li, L., Yuan, Z., Wang, C. et al. (2021a) Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14454–14463.

Sun, S., Melamed, D. and Kitani, K. (2021b) Idol: Inertial deep orientation-estimation and localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 6128–6137.

Sun, Y., Hu, J., Yun, J., Liu, Y., Bai, D., Liu, X., Zhao, G., Jiang, G., Kong, J. and Chen, B. (2022) Multi-objective location and mapping based on deep learning and visual slam. *Sensors*, **22**, 7576.

Sun, Z., Simo, J. and Gong, S. (2023) Satellite attitude identification and prediction based on neural network compensation. *Space: Science & Technology*.

Szczęsna, A., Skurowski, P., Pruszowski, P., Pęszor, D., Paszkuta, M. and Wojciechowski, K. (2016) Reference data set for accuracy evaluation of orientation estimation algorithms for inertial motion capture systems. In *International Conference on Computer Vision and Graphics*, 509–520. Springer.

Talele, A., Patil, A. and Barse, B. (2019) Detection of real time objects using tensorflow and opencv. *Asian Journal For Convergence In Technology (AJCT) ISSN-2350-1146*.

Tan, M., Pang, R. and Le, Q. V. (2020) Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10781–10790.

Tang, Y., Zhao, C., Wang, J., Zhang, C., Sun, Q., Zheng, W. X., Du, W., Qian, F. and Kurths, J. (2022) Perception and navigation in autonomous systems in the era of learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*.

Thakur, R. K. and Maji, S. K. (2023) Multi scale pixel attention and feature extraction based neural network for image denoising. *Pattern Recognition*, **141**, 109603.

Thompson, A. M. (1977) The navigation system of the jpl robot. *Tech. rep.*

Thrun, S. (2002) Probabilistic robotics. *Communications of the ACM*, **45**, 52–57.

Tian, C., Fei, L., Zheng, W., Xu, Y., Zuo, W. and Lin, C.-W. (2020a) Deep learning on image denoising: An overview. *Neural Networks*, **131**, 251–275.

Tian, J., Yuan, J. and Liu, H. (2020b) Road marking detection based on mask r-cnn instance segmentation model. In *2020 international conference on computer vision, image and deep learning (CVIDL)*, 246–249. IEEE.

Tian, Q., Li, Z., Fan, Q., Polimeni, J. R., Bilgic, B., Salat, D. H. and Huang, S. Y. (2022a) Sdndti: Self-supervised deep learning-based denoising for diffusion tensor mri. *Neuroimage*, **253**, 119033.

Tian, Y., Chang, Y., Arias, F. H., Nieto-Granda, C., How, J. P. and Carlone, L. (2022b) Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems. *IEEE Transactions on Robotics*, **38**.

Tian, Z., Shen, C., Chen, H. and He, T. (2019) Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9627–9636.

Tong, Q., Li, X., Lin, K., Li, C., Si, W. and Yuan, Z. (2019) Cascade-lstm-based visual-inertial navigation for magnetic levitation haptic interaction. *IEEE Network*, **33**, 74–80.

Topini, E., Topini, A., Franchi, M., Bucci, A., Secciani, N., Ridolfi, A. and Allotta, B. (2020) Lstm-based dead reckoning navigation for autonomous underwater vehicles. In *Global Oceans 2020: Singapore–US Gulf Coast*, 1–7. IEEE.

Tsukamoto, H., Chung, S.-J., Donitz, B., Ingham, M., Mages, D. and Nakka, Y. K. (2022) Neural-rendezvous: Learning-based robust guidance and control to encounter interstellar objects. *arXiv preprint arXiv:2208.04883*.

Tu, Z., Chen, C., Pan, X., Liu, R., Cui, J. and Mao, J. (2022) Ema-vio: Deep visual–inertial odometry with external memory attention. *IEEE Sensors Journal*, **22**, 20877–20885.

Turan, E., Speretta, S. and Gill, E. (2022) Autonomous navigation for deep space small satellites: Scientific and technological advances. *Acta Astronautica*.

Ullah, I., Shen, Y., Su, X., Esposito, C. and Choi, C. (2019) A localization based on unscented kalman filter and particle filter localization algorithms. *IEEE Access*, **8**, 2233–2246.

Uno, T., Isoyama, N., Uchiyama, H., Sakata, N. and Kiyokawa, K. (2022) Deep inertial underwater odometry system. *CEUR Workshop Proceedings*.

Uriot, T., Izzo, D., Simões, L. F., Abay, R., Einecke, N., Rebhan, S., Martinez-Heras, J., Letizia, F., Siminski, J. and Merz, K. (2022) Spacecraft collision avoidance challenge: Design and results of a machine learning competition. *Astrodynamics*, **6**, 121–140.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I. (2017) Attention is all you need. *Advances in neural information processing systems*, **30**.

Viola, P. and Jones, M. J. (2004) Robust real-time face detection. *International journal of computer vision*, **57**, 137–154.

Visca, M., Powell, R., Gao, Y. and Fallah, S. (2022) Meta-conv1d energy-aware path planner for mobile robots in unstructured terrains.

Wang, C.-Y., Bochkovskiy, A. and Liao, H.-Y. M. (2023a) Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7464–7475.

Wang, D., Li, Z., Du, X., Ma, Z. and Liu, X. (2022a) Farmland obstacle detection from the perspective of uavs based on non-local deformable detr. *Agriculture*, **12**, 1983.

Wang, G., Zou, C., Zhang, C., Pan, C., Song, J. and Yang, F. (2022b) Aircarft signal feature extraction and recognition based on deep learning. *IEEE Transactions on Vehicular Technology*, **71**, 9625–9634.

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X. et al. (2020a) Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, **43**, 3349–3364.

Wang, J., Zhang, X., Jing, K. and Zhang, C. (2023b) Learning precise feature via self-attention and self-cooperation yolox for smoke detection. *Expert Systems with Applications*, 120330.

Wang, Q., Li, Z., Zhu, D. and Yang, W. (2023c) Lidar-only 3d object detection based on spatial context. *Journal of Visual Communication and Image Representation*, **93**, 103805.

Wang, S., Clark, R., Wen, H. and Trigoni, N. (2017) Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *2017 IEEE international conference on robotics and automation (ICRA)*, 2043–2050. IEEE.

— (2018) End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks. *The International Journal of Robotics Research*, **37**, 513–542.

Wang, T., Chen, B., Zhang, Z., Li, H. and Zhang, M. (2022c) Applications of machine vision in agricultural robot navigation: A review. *Computers and Electronics in Agriculture*, **198**, 107085.

Wang, W., Lin, H. and Wang, J. (2020b) Cnn based lane detection with instance segmentation in edge-cloud computing. *Journal of Cloud Computing*, **9**, 1–10.

Wang, W., Wang, S., Guo, Y. and Zhao, Y. (2022d) Obstacle detection method of unmanned electric locomotive in coal mine based on yolov3-4l. *Journal of Electronic Imaging*, **31**, 023032–023032.

Wang, W., You, X., Zhang, X., Chen, L., Zhang, L. and Liu, X. (2021) Lidar-based slam under semantic constraints in dynamic environments. *Remote Sensing*, **13**, 3651.

Wang, X., Zhao, Y. and Pourpanah, F. (2020c) Recent advances in deep learning. *International Journal of Machine Learning and Cybernetics*, **11**, 747–750.

Wang, Y., Cheng, H. and Meng, M. Q.-H. (2022e) A2dio: Attention-driven deep inertial odometry for pedestrian localization based on 6d imu. In *2022 International Conference on Robotics and Automation (ICRA)*, 819–825. IEEE.

Wang, Y., Kuang, J., Li, Y. and Niu, X. (2022f) Magnetic field-enhanced learning-based inertial odometry for indoor pedestrian. *IEEE Transactions on Instrumentation and Measurement*, **71**, 1–13.

Wang, Y., Xu, H., Liu, Y., Wang, M., Wang, Y., Yang, Y., Zhou, S., Zeng, J., Xu, J., Li, S. et al. (2023d) A novel deep multifeature extraction framework based on attention mechanism using wearable sensor data for human activity recognition. *IEEE Sensors Journal*, **23**, 7188–7198.

Weber, D., Gühmann, C. and Seel, T. (2021) Riann—a robust neural network outperforms attitude estimation filters. *AI*, **2**, 444–463.

Wen, L.-H. and Jo, K.-H. (2022) Deep learning-based perception systems for autonomous driving: A comprehensive survey. *Neurocomputing*.

Wolek, A. and Woolsey, C. A. (2017) Model-based path planning. *Sensing and Control for Autonomous Vehicles: Applications to Land, Water and Air Vehicles*, 183–206.

Wong, C., Yang, E., Yan, X.-T. and Gu, D. (2017) Adaptive and intelligent navigation of autonomous planetary rovers—a survey. In *2017 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, 237–244. IEEE.

Xiao, C., Choi, E. and Sun, J. (2018) Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review. *Journal of the American Medical Informatics Association*, **25**, 1419–1428.

Xiao, L., Wang, J., Qiu, X., Rong, Z. and Zou, X. (2019) Dynamic-slam: Semantic monocular visual localization and mapping based on deep learning in dynamic environment. *Robotics and Autonomous Systems*, **117**, 1–16.

Xu, B., Wang, N., Chen, T. and Li, M. (2015) Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*.

Xu, R., Wu, R., Ishiwaka, Y., Vondrick, C. and Zheng, C. (2020) Listening to sounds of silence for speech denoising. *Advances in Neural Information Processing Systems*, **33**, 9633–9648.

Xu, Z., Dong, J., Hua, Y., Dong, X., Li, Q. and Ren, Z. (2023) Uav path planning from human demonstrations using inverse reinforcement learning. In *Advances in Guidance, Navigation and Control: Proceedings of 2022 International Conference on Guidance, Navigation and Control*, 5542–5551. Springer.

Yan, H., Shan, Q. and Furukawa, Y. (2018) Ridi: Robust imu double integration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 621–636.

Yang, J., Xi, M., Jiang, B., Man, J., Meng, Q. and Li, B. (2020) Fadn: fully connected attitude detection network based on industrial video. *IEEE Transactions on Industrial Informatics*, **17**, 2011–2020.

Yang, M., Chen, Y. and Kim, H.-S. (2022) Efficient deep visual and inertial odometry with adaptive visual modality selection. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVIII*, 233–250. Springer.

Yang, Z., Liu, S., Hu, H., Wang, L. and Lin, S. (2019) Reppoints: Point set representation for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9657–9666.

Yu, C., Ma, X., Ren, J., Zhao, H. and Yi, S. (2020a) Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, 507–523. Springer.

Yu, C., Wang, J., Peng, C., Gao, C., Yu, G. and Sang, N. (2018) Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, 325–341.

Yu, C.-Y. and Peng, C.-C. (2020) Machining learning for 2d-slam object classification and recognition. In *2020 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan)*, 1–2. IEEE.

Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V. and Darrell, T. (2020b) Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2636–2645.

Yu, J., Zhang, J., Shu, A., Chen, Y., Chen, J., Yang, Y., Tang, W. and Zhang, Y. (2023) Study of convolutional neural network-based semantic segmentation methods on edge intelligence devices for field agricultural robot navigation line extraction. *Computers and Electronics in Agriculture*, **209**, 107811.

Yuan, K. and Wang, Z. J. (2023) A simple self-supervised imu denoising method for inertial aided navigation. *IEEE Robotics and Automation Letters*.

Yun, K., Choi, C., Alimo, R., Davis, A., Forster, L., Rahmani, A., Adil, M. and Madani, R. (2020) Multi-agent motion planning using deep learning for space applications. In *ASCEND 2020*, 4233.

Yurtsever, E., Lambert, J., Carballo, A. and Takeda, K. (2020) A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, **8**, 58443–58469.

Zaheer, M. H., Yoon, S. Y. and Higginson, B. K. (2023) Orientation estimation for instrumented helmet using neural networks. *Measurement and Control*, 00202940221149062.

Zhang, B., Wang, S., Deng, L., Jia, M. and Xu, J. (2023a) Ship motion attitude prediction model based on iwoa-tcn-attention. *Ocean Engineering*, **272**, 113911.

Zhang, H. (2022) Deep learning applications in simultaneous localization and mapping. In *Journal of Physics: Conference Series*, vol. 2181, 012012. IOP Publishing.

Zhang, H., Lin, Y., Han, S. and Lv, K. (2022a) Lexicographic actor-critic deep reinforcement learning for urban autonomous driving. *IEEE Transactions on Vehicular Technology*.

Zhang, H.-y., Lin, W.-m. and Chen, A.-x. (2018) Path planning for the mobile robot: A review. *Symmetry*, **10**, 450.

Zhang, J. and Zhong, H. (2023) Curve-based lane estimation model with lightweight attention mechanism. *Signal, Image and Video Processing*, 1–7.

Zhang, K., Jiang, C., Li, J., Yang, S., Ma, T., Xu, C. and Gao, F. (2022b) Dido: Deep inertial quadrotor dynamical odometry. *IEEE Robotics and Automation Letters*, **7**, 9083–9090.

Zhang, K., Zhao, L., Dong, C., Wu, L. and Zheng, L. (2022c) Ai-tp: Attention-based interaction-aware trajectory prediction for autonomous driving. *IEEE Transactions on Intelligent Vehicles*.

Zhang, Q., Yan, F., Song, W., Wang, R. and Li, G. (2023b) Automatic obstacle detection method for the train based on deep learning. *Sustainability*, **15**, 1184.

Zhang, R., Hao, J., Wang, R., Deng, H. and Wang, H. (2023c) Multi-objective global path planning for uav-assisted sensor data collection using drl and transformer. In *Web and Big Data: 6th International Joint Conference, APWeb-WAIM 2022, Nanjing, China, November 25–27, 2022, Proceedings, Part II*, 492–500. Springer.

Zhang, R., Hao, J., Wang, R., Wang, H., Deng, H. and Lu, S. (2023d) Global path planning for multi-objective uav-assisted sensor data collection: A drl approach. In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*, 163–174. Springer.

Zhang, X., He, Z., Ma, Z., Wang, Z. and Wang, L. (2021) Llfe: A novel learning local features extraction for uav navigation based on infrared aerial image and satellite reference image matching. *Remote Sensing*, **13**, 4618.

Zhang, Y. and Li, J. (2023) Birdsoundsdenoising: Deep visual audio denoising for bird sounds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2248–2257.

Zhao, C., Song, A., Zhu, Y., Jiang, S., Liao, F. and Du, Y. (2023) Data-driven indoor positioning correction for infrastructure-enabled autonomous driving systems: A lifelong framework. *IEEE Transactions on Intelligent Transportation Systems*.

Zhao, H., Qi, X., Shen, X., Shi, J. and Jia, J. (2018) Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European conference on computer vision (ECCV)*, 405–420.

Zhao, H., Shi, J., Qi, X., Wang, X. and Jia, J. (2017) Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2881–2890.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A. and Torralba, A. (2017) Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, **40**, 1452–1464.

Zhou, D., Sun, G. and Hong, X. (2021) 3d visual tracking framework with deep learning for asteroid exploration. *arXiv preprint arXiv:2111.10737*.

Zhou, X., Wang, D. and Krähenbühl, P. (2019) Objects as points. *arXiv preprint arXiv:1904.07850*.

Zhou, Y., Li, W., Ma, Y., Li, B., Wen, Z. and Zhang, L. (2022a) Attitude estimation of on-orbit spacecraft based on the u-linked network. *IEEE Geoscience and Remote Sensing Letters*, **19**, 1–5.

Zhou, Z., Du, L., Ye, X., Zou, Z., Tan, X., Zhang, L., Xue, X. and Feng, J. (2022b) Sgm3d: stereo guided monocular 3d object detection. *IEEE Robotics and Automation Letters*, **7**, 10478–10485.

Zhu, K. and Zhang, T. (2021) Deep reinforcement learning based mobile robot navigation: A review. *Tsinghua Science and Technology*, **26**, 674–691.

Zhu, P., Wen, L., Du, D., Bian, X., Fan, H., Hu, Q. and Ling, H. (2021) Detection and tracking meet drones challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **44**, 7380–7399.

Zhuang, Y., Sun, X., Li, Y., Huai, J., Hua, L., Yang, X., Cao, X., Zhang, P., Cao, Y., Qi, L. et al. (2023) Multi-sensor integrated navigation/positioning systems using data fusion: From analytics-based to learning-based approaches. *Information Fusion*, **95**, 62–90.

Ziyin, L., Hartwig, T. and Ueda, M. (2020) Neural networks fail to learn periodic functions and how to fix it. *Advances in Neural Information Processing Systems*, **33**, 1583–1594.

Zou, Z., Chen, K., Shi, Z., Guo, Y. and Ye, J. (2023) Object detection in 20 years: A survey. *Proceedings of the IEEE*.

Zulqarnain, M., Ghazali, R., Hassim, Y. M. M. and Rehan, M. (2020) A comparative review on deep learning models for text classification. *Indones. J. Electr. Eng. Comput. Sci*, **19**, 325–335.

TABLE 14 Path Planning Algorithms

Algorithm	Category	Comp[1]	Opt[2]	Heu[3]	Approach	Advantages
A* Hart et al. (1968)	Global	Yes	Yes	Yes	Graph-based	Faster than Dijkstra's with heuristic, completenes optimality
RRT LaValle (1998)	Local	No	No	No	Sampling-based	Efficient in high-dimensional spaces
D* Lite Koenig and Likhachev (2002)	Local	Yes	Yes	Yes	Graph-based	Can handle dynamic environments, efficient upda
PRM Kavraki et al. (1996)	Global	Yes	No	No	Sampling-based	Efficient in low-dimensional spaces, global planni stacle avoidance
Potential Fields Khatib (1986)	Local	No	No	No	Potential-based	Simple, real-time planning
Dijkstra's Dijkstra (1959)	Global	Yes	Yes	No	Graph-based	Guaranteed to find the shortest path, shortest p weighted graphs
Theta* Nash and Koenig (2007)	Global	Yes	Yes	Yes	Graph-based	Smoother paths than A*, improved efficiency ove
LQR Bryson and Ho (1975)	Global	No	Yes	No	Optimal Control	Optimal control for continuous systems
MPC Mayne et al. (2014)	Global	No	Yes	No	Optimal Control	Constraints handling, predictive behavior
Genetic Algorithms Goldberg (1989)	Global	No	No	No	Evolutionary Computation	Global search, exploration
RRT* Karaman and Frazzoli (2011)	Local	Yes	Yes	No	Sampling-based	Improved optimality over RRT
BIT* Gammell et al. (2015)	Global	Yes	Yes	Yes	Sampling-based	Improved efficiency over PRM
FMT* Janson et al. (2015)	Global	Yes	Yes	Yes	Sampling-based	Improved efficiency over RRT*
RRT# Gammell et al. (2014)	Local	Yes	Yes	Yes	Sampling-based	Improved efficiency over RRT*
Lazy Theta* Liu and Reif (2011)	Global	Yes	Yes	Yes	Graph-based	Improved efficiency over Theta*
LPA* Koenig and Likhachev (2004)	Global	Yes	Yes	Yes	Graph-based	Can handle dynamic environments
ARA* Likhachev et al. (2008)	Global	Yes	Yes	Yes	Graph-based	Can be stopped at any time with a valid solution
Field D* Ferguson and Stentz (2006)	Local	Yes	Yes	Yes	Graph-based	Can handle dynamic environments, efficient upda

[1] **Completeness** Whether the algorithm is guaranteed to find a solution if one exists, indicated by "Yes" or "No".
[2] **Optimality** Whether the algorithm is guaranteed to find the optimal solution, indicated by "Yes" or "No".
[3] **Heuristics** Whether the algorithm uses heuristics to guide the search, indicated by "Yes" or "No".

TABLE 15 Deep Learning Architectures, Properties, and Applications.

Arch	Properties	Applications
CNN	Uses convolutional layers Proper for extracting features from images	Obstacle detection He et al. (2022) Object classification Aftf et al. (2019) Lane detection Wang et al. (2020b) Traffic sign detection Li et al. (2022c) Outdoor Navigation Kouris and Bouga Indoor Navigation Gong et al. (2021)
RNN	Uses recurrent layers Memory and store hidden states Extracts features from sequential data Learn temporal dependencies	Attitude Estimation Chumuang et al. (Denoising Brossard et al. (2020) Dead Reckoning Topini et al. (2020) Inertial Navigation Tong et al. (2019)
Transformer	Uses self-attention mechanisms Uses multi-head attention	Visual Navigation Du et al. (2021) Pedestrian Navigation Yu et al. (2020a
GAN	Uses a generator network Uses a discriminator network	Trajectory Prediction Roy et al. (2019a Obstacle Detection Dimas et al. (2019 Path Planning Mohammadi et al. (201

TABLE 16 Activation Functions.

Activation	Equation	Gradient
Sigmoid McCulloch and Pitts (1943)	$\frac{1}{1+e^{-x}}$	$\frac{e^{-x}}{(1+e^{-x})^2}$
ReLU Nair and Hinton (2010)	$\max(0, x)$	$\begin{matrix} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{matrix}$
Leaky ReLU Maas et al. (2013)	$\max(ax, x)$	$\begin{cases} 1 & \text{if } x > 0 \\ a & \text{if } x \leq 0 \end{cases}$
Softmax Bridle (1989)	$\frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$	$\frac{e^{x_i}(1-\frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}})}{\sum_{j=1}^{n} e^{x_j}}$
Tanh	$\frac{e^x-e^{-x}}{e^x+e^{-x}}$	$\frac{4e^{2x}}{(e^x+e^{-x})^2}$
Softplus Dugas et al. (2000)	$\ln(1+e^x)$	$\frac{e^x}{1+e^x}$
Swish Ramachandran et al. (2017)	$x \cdot \text{sigmoid}(\beta x)$	$x \cdot \text{sigmoid}(\beta x) \cdot (1 - \text{sigmoid}(\beta x)) + \text{sigmoid}(\beta x)$
Mish Misra (2019)	$x \cdot \tanh(\ln(1+e^x))$	$x \cdot \tanh(\ln(1+e^x)) \cdot (1 - \tanh(\ln(1+e^x))) + \tanh$
ELU Clevert et al. (2015)	$\max(0, x) + \min(0, \alpha(e^x - 1))$	1
PReLU He et al. (2015a)	$\begin{matrix} ax & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{matrix}$	$\begin{matrix} a & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{matrix}$
GELU Hendrycks and Gimpel (2016)	$\frac{1}{2}x(1+\tanh(\sqrt{\frac{2}{\pi}}(x+\alpha x^3)))$ $\alpha = 0.044715$	$\frac{1}{2}(1+\tanh(\sqrt{\frac{2}{\pi}}(x+\alpha x^3))) + \frac{1}{2}x\sqrt{\frac{2}{\pi}}(1 - \frac{\tanh^2(\sqrt{\frac{2}{\pi}}(}{2}$
SELU Klambauer et al. (2017)	$\begin{matrix} \lambda x & \text{if } x \geq 0 \\ \lambda\alpha(e^x - 1) & \text{if } x < 0 \end{matrix}$	$\begin{matrix} \lambda & \text{if } x \geq 0 \\ \lambda\alpha(e^x) & \text{if } x < 0 \end{matrix}$
Softsign	$\frac{x}{1+\lvert x\rvert}$	$\frac{1}{(1+\lvert x\rvert)^2}$
Hard shrink	$\begin{matrix} x & \text{if } x > \lambda \\ x & \text{if } x < -\lambda \\ 0 & \text{otherwise} \end{matrix}$	$\begin{matrix} 1 & \text{if } x > \lambda \\ 1 & \text{if } x < -\lambda \\ 0 & \text{otherwise} \end{matrix}$
Soft shrink	$\begin{cases} x-\lambda & \text{if } x > \lambda \\ x+\lambda & \text{if } x < -\lambda \\ 0 & \text{otherwise} \end{cases}$	$\begin{cases} 1 & \text{if } x > \lambda \\ 1 & \text{if } x < -\lambda \\ 0 & \text{otherwise} \end{cases}$
Tanh shrink	$x - \tanh(x)$	$1 - \tanh^2(x)$
LiSHT Roy et al. (2019b)	$x \cdot \tanh(x)$	$x \cdot \tanh(x) \cdot (1 - \tanh(x)) + \tanh(x)$
Snake Ziyin et al. (2020)	$x + \frac{1}{a}\sin^2(ax)$ or $x - \frac{1}{2a}\cos(2ax) + \frac{1}{2a}$	$\frac{1}{a}\cos^2(ax)$ or $\frac{1}{2a}\cos(2ax) - \frac{1}{2a}$