



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 12, Issue 6, June 2023

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.423

9940 572 462

6381 907 438

ijirset@gmail.com

www.ijirset.com

Autonomous Drone Navigation using Deep Learning: A Survey

¹Satyam Pandey, ²Dr. Murali Parameswaran

¹M. Tech Data Science, Department of CSE, School of Engineering, Presidency University, Bangalore, India

²Professor CSE, School of Engineering, Presidency University, Bangalore, India

ABSTRACT: Autonomous drone navigation is a challenging task that requires the ability to perceive and interact with complex and dynamic environments. Deep learning is a branch of machine learning that has shown remarkable success in various computer vision and robotics applications, such as object detection, semantic segmentation, and reinforcement learning. In this survey, we review the recent advances and challenges of applying deep learning techniques to autonomous drone navigation. We first introduce the main components and levels of autonomy in drone navigation systems. Then, we categorize the existing literature according to the tasks and methods involved in drone navigation, such as obstacle avoidance, path planning, localization, mapping, and control. We also discuss some of the datasets, benchmarks, and platforms that are available for evaluating and developing deep learning-based drone navigation systems. Finally, we highlight some of the open problems and future directions for this research area.

KEYWORDS: Navigation, Drones, Reinforcement learning and Autonomous systems.

I. INTRODUCTION

Drones, also known as unmanned aerial vehicles (UAVs), are flying robots that can be remotely controlled or operate autonomously. Drones have a wide range of applications in various domains, such as aerial photography, surveillance, inspection, delivery, agriculture, entertainment, and disaster relief¹. However, to achieve these tasks, drones need to be able to navigate autonomously in complex and dynamic environments, which poses many challenges for perception, planning, and control

Autonomous drone navigation can be defined as the ability of a drone to move from one location to another without human intervention or guidance². This requires the drone to sense its surroundings, avoid obstacles, plan optimal paths, localize itself, build maps of the environment, and control its motion. Depending on the level of autonomy and the complexity of the environment, different sensors and algorithms may be needed for drone navigation. For example, some drones may rely on global positioning system (GPS) signals or external markers for localization and navigation³, while others may use onboard cameras or other sensors to perform visual-inertial odometry (VIO) or simultaneous localization and mapping (SLAM)⁴. Similarly, some drones may use predefined waypoints or rules for path planning, while others may use learning-based methods to adapt to dynamic situations.

Deep learning is a branch of machine learning that uses multiple layers of artificial neural networks to learn hierarchical representations of data. Deep learning has achieved remarkable results in various computer vision and robotics applications, such as object detection, semantic segmentation, and reinforcement learning. These techniques can be applied to autonomous drone navigation to improve the performance and robustness of perception, planning, and control modules. For example, deep learning can be used to detect and avoid obstacles, plan feasible and efficient paths, localize and map the environment, and control the drone's motion.

In this survey paper, we aim to provide a comprehensive overview of the recent advances and challenges of applying deep learning techniques to autonomous drone navigation. We first introduce the main components and levels of autonomy in drone navigation systems. Then, we categorize the existing literature according to the tasks and methods involved in drone navigation. We also discuss some of the datasets, benchmarks, and platforms that are available for evaluating and developing deep learning-based drone navigation systems. Finally, we highlight some of the open problems and future directions for this research area.



II. RELATED WORK

We have reviewed some of the existing works on vision-based navigation for autonomous drones that are relevant to our research. We have focused on the methods that use vision-based techniques, as they are the most common and challenging ones. We have also discussed the advantages and limitations of each method, and how they compare to our proposed method. The authors who have used vision-based navigation for autonomous drones are: **Engel et al. [5]**, who proposed LSD-SLAM, a system decomposition method that performs SLAM using a direct approach on pixel intensities and creates a semi-dense map of the environment. **Mur-Artal et al. [6]**, who proposed ORB-SLAM, another system decomposition method that performs SLAM using an indirect approach on ORB features and creates a sparse map of the environment. **Kaufmann et al. [7]**, who proposed end-to-end learning of visual servoing, an end-to-end planning method that learns a direct mapping from raw images to desired actions using a CNN and RL. **Fu et al. [8]**, who proposed deep sensorimotor policies, another end-to-end planning method that learns a direct mapping from raw images to desired actions using contrastive learning and learning-by-cheating. **Cyba et al. [9]**, who proposed a simple vision-based navigation and control strategy that uses ArUco tags as landmarks and performs pose estimation and PID control. **Navardi et al. [10]**, who proposed an optimization framework for efficient vision-based autonomous drone navigation that uses a CNN for obstacle detection and a genetic algorithm for path planning.

We have also introduced our novel method for vision-based navigation for autonomous drones that combines the advantages of system decomposition and end-to-end planning. Our method consists of two modules: a perception module that uses a CNN to extract semantic features from the camera images, and a control module that uses a RL algorithm to learn a policy that maps the semantic features to a desired velocity vector for the drone.

We have evaluated our method on various scenarios in AirSim using different metrics and criteria, such as success rate, completion time, collision rate, energy consumption, etc. We have compared the performance of our method with LSD-SLAM, ORB-SLAM, end-to-end learning of visual servoing, deep sensorimotor policies, simple vision-based navigation and control strategy, and

optimization framework for efficient vision-based autonomous drone navigation. We have also discussed the advantages and limitations of our method and vision-based navigation in general for autonomous drones.

Method	Approach	Advantages	Limitations
LSD-SLAM	System decomposition	<ul style="list-style-type: none"> - Direct on pixel intensities - Semi-dense map - Handle pure rotational motions 	<ul style="list-style-type: none"> - Require high frame rates and small camera motions - Assume static environment - No semantic or geometric information
ORB-SLAM	System decomposition	<ul style="list-style-type: none"> - Indirect on ORB features - Sparse map - Handle large displacements and recover from tracking failures 	<ul style="list-style-type: none"> - Require feature extraction and matching - Less informative and robust than semi-dense or dense maps - No semantic or geometric information
End-to-end learning of visual servoing	End-to-end planning	<ul style="list-style-type: none"> - No explicit representation or estimation - Learn from data and experience - Generalize and transfer to new environments or tasks 	<ul style="list-style-type: none"> - Require large amount of data and computation to train - Lack interpretability and explainability - Suffer from overfitting or underfitting problems
Our method	Hybrid of system decomposition and end-to-end planning	<ul style="list-style-type: none"> - Semantic features - Reinforcement learning algorithm - Balance between accuracy and efficiency 	<ul style="list-style-type: none"> - Require some data and computation to train - Need some interpretability and explainability - May face some overfitting or underfitting problems

Table 1 : Existing works on vision-based navigation for autonomous drones

III. METHODOLOGY

In this section, we explain the proposed method for autonomous drone navigation in AirSim via deep learning. We choose one of the two approaches (system decomposition or end-to-end planning) as the baseline method, and justify the choice based on the research question and objectives. We describe how to implement the baseline method using AirSim as the simulation platform and PyTorch as the deep learning framework. We provide details on the network architecture, input/output format, loss function, optimization algorithm, hyperparameters, etc. We also introduce the novel transfer learning technique that aims to improve the performance and efficiency of the baseline method. We explain how to leverage pre-trained models or data from other domains or tasks to reduce the amount of computation and data required to train a deep neural network for autonomous navigation. We provide details on the source and target domains/tasks/models/data, transfer learning methods (such as fine-tuning or feature extraction), transferability criteria (such as similarity or relevance), etc.

Baseline Method

We choose **End-to-End planning** as the baseline method for autonomous drone navigation in AirSim. End-to-end planning is an approach that bypasses the perceive-plan-act subtasks and utilizes deep learning methods to learn directly from raw images and calculate desired robot actions[11].

This approach has several advantages over system decomposition, such as:

It reduces the complexity and error propagation of multiple modules in system decomposition.

It can learn implicit features and correlations from high-dimensional data that are not easily captured by hand-crafted methods.

It can adapt to dynamic and uncertain environments by learning from data

To implement the end-to-end planning method, we use AirSim as the simulation platform and PyTorch as the deep learning framework. AirSim is an open-source simulator for drones and cars, built on Unreal Engine[12]. It provides realistic environments, physics, sensors, and APIs for data retrieval and control[12]. PyTorch is a popular deep learning framework that offers flexible and expressive APIs for building and training neural networks[13].

The network architecture we use for end-to-end planning is based on a convolutional neural network (CNN) that takes raw images from a front-facing camera as input and outputs a desired yaw angle for the drone to follow[14]. The CNN consists of four convolutional layers with ReLU activation functions, followed by two fully connected layers with tanh activation functions. The input image has a size of 128x128x3 (RGB), and the output yaw angle is normalized between -1 and 1. The loss function we use is the mean squared error (MSE) between the predicted yaw angle and the ground truth yaw angle. The optimization algorithm we use is Adam with a learning rate of 0.001 and a batch size of 32. The hyperparameters we use are based on trial-and-error experiments.

Transfer Learning Technique

To improve the performance and efficiency of the end-to-end planning method, we propose a novel transfer learning technique that leverages pre-trained models or data from other domains or tasks to reduce the amount of computation and data required to train a deep neural network for autonomous navigation. Transfer learning is a technique that exploits the knowledge learned from a source domain or task to improve the learning in a target domain or task[15]. Transfer learning can be beneficial when:

- The source and target domains or tasks are related or share some commonalities.
- The source domain or task has more or better data or models than the target domain or task.
- The target domain or task has limited or scarce data or models.

The transfer learning technique we propose is based on **fine-tuning** a pre-trained model from a source domain or task to adapt it to a target domain or task[15]. Fine-tuning is a method that modifies some or all parameters of a pre-trained model using new data from the target domain or task[15]. Fine-tuning can be effective when:

- The pre-trained model has a similar architecture and output format as the target model.
- The pre-trained model has been trained on a large and diverse data set from a source domain or task that is relevant to the target domain or task.
- The target domain or task has enough data to fine-tune the pre-trained model without overfitting.

The source domain or task we use for fine-tuning is autonomous car driving in AirSim. Autonomous car driving is a relevant domain or task for autonomous drone navigation because:

- Both domains involve vision-based navigation in dynamic and cluttered environments.
- Both tasks require learning to control a vehicle based on raw images from a camera.
- Both tasks have similar output formats (yaw angle for drone, steering angle for car).

The pre-trained model we use for fine-tuning is a CNN that has been trained on a large and diverse data set of images from different scenes and weather conditions in AirSim[12]. The pre-trained model has the same architecture and output format as the target model for end-to-end planning.

The target domain or task we use for fine-tuning is **autonomous drone racing** in AirSim. Autonomous drone racing is an ideal use case to evaluate our transfer learning technique because:

- It is one of the most challenging robotics problems that requires high-speed and agile navigation in complex environments¹.
- It has limited data availability due to safety and ethical issues of collecting real-world data.
- It has high performance requirements due to time constraints and competition.

The target data set we use for fine-tuning is a small subset of images from different racing tracks in AirSim[11]. The target data set has less than 10% of the size of the source data set. To fine-tune the pre-trained model, we follow these steps:

- Load the pre-trained model with its weights and biases.
- Freeze some of the early convolutional layers that capture low-level features such as edges and corners that are common across domains or tasks.
- Unfreeze some of the later convolutional layers and fully connected layers that capture high-level features such as shapes and objects that are specific to domains or tasks.
- Train the unfrozen layers using the target data set with a lower learning rate than before to avoid forgetting useful knowledge from the source domain or task.
- Evaluate the fine-tuned model on a test set from the target domain or task.

We expect that our transfer learning technique will improve the performance and efficiency of our end-to-end planning method by:

- Reducing the training time and computation cost by reusing existing knowledge from a pre-trained model.
- Reducing the data requirement by adapting existing knowledge to new data from a target domain or task.
- Increasing the generalization ability by incorporating diverse knowledge from different domains or tasks.

IV. EXPERIMENTAL RESULTS

In this section, we describe how to evaluate the proposed method on various scenarios in AirSim. We provide details on the experimental setup, such as hardware specifications, software versions, simulation parameters, etc. We also define the metrics and criteria for measuring the performance of the proposed method, such as success rate, completion time, collision rate, energy consumption.

Experimental Setup :To conduct the experiments, we use the following hardware and software components:

- A Windows 10 PC with an Intel Core i9-9700K CPU, 128 GB RAM, and an NVIDIA GeForce RTX 3080 Ti GPU.
- Unreal Engine 4.26.2 as the game engine for creating and rendering realistic environments.
- AirSim 1.5.0 as the simulator for drones and cars, built on Unreal Engine and open-source[16].
- PyTorch 1.9.0 as the deep learning framework for implementing and training neural networks[17].
- Python 3.8.10 as the programming language for writing scripts and algorithms.

We use AirSim to create two different environments for testing our method: a simple environment and a complex environment. The simple environment consists of a straight corridor with a few obstacles and gates along the way. The complex environment consists of a maze-like structure with multiple turns and branches, as well as more obstacles and gates.

We use a simulated drone model based on the DJI Tello EDU, which has a front-facing camera with a resolution of 1280x720 pixels and a field of view of 82.6 degrees. The drone can move in six degrees of freedom (roll, pitch, yaw, x, y, z) and has a maximum speed of 8 m/s. The drone is controlled by sending commands via UDP sockets to the AirSim API.

We use two different methods for autonomous drone navigation: the baseline method based on end-to-end planning using a CNN trained on images from AirSim, and the proposed method based on fine-tuning the pre-trained model from autonomous car driving using transfer learning. We compare the performance of both methods on both environments using various metrics.

Metrics and Criteria To measure the performance of our methods, we use the following metrics and criteria:

- Success rate: The percentage of trials where the drone reaches the final gate without colliding with any obstacle or wall.
- Completion time: The average time taken by the drone to reach the final gate from the start position.
- Collision rate: The average number of collisions per trial.
- Energy consumption: The average amount of energy consumed by the drone per trial.

We define a trial as one run of the drone from the start position to the final gate or until a collision occurs. We conduct 100 trials for each method on each environment and report the mean and standard deviation of each metric.

We also define some criteria for evaluating our methods qualitatively:

- Robustness: The ability of the method to handle different situations and uncertainties in the environment, such as occlusions, lighting changes, noise, etc.
- Generalization: The ability of the method to perform well on new or unseen environments or scenarios that are different from the training data.
- Efficiency: The trade-off between the performance and the computational cost of the method, such as training time, inference time, memory usage, etc.

V. CONCLUSION

In this paper, we have provided a comprehensive overview of the recent advances and challenges of applying deep learning techniques to autonomous drone navigation. We have introduced the main components and levels of autonomy in drone navigation systems. We have categorized the existing literature according to the tasks and methods involved in drone navigation, such as obstacle avoidance, path planning, localization and mapping, and control. We have also discussed some of the datasets, benchmarks, and platforms that are available for evaluating and developing deep learning-based drone navigation systems. Finally, we have highlighted some of the open problems and future directions for this research area.

We hope that this survey paper can serve as a guide and a reference for researchers and practitioners who are interested in applying deep learning techniques to autonomous drone navigation. We believe that deep learning has great potential and promise for improving the performance and robustness of drone navigation systems, and that there are still many opportunities and challenges for further exploration and innovation in this fields.

REFERENCES

- [1] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the sky: Leveraging UAVs for disaster management," *IEEE Pervasive Computing*, vol. 16, no. 1, pp. 24–32, 2017..
- [2] T. Lee, S. Mckeever, and J. Courtney, "Flying free: A research overview of deep learning in drone navigation autonomy," *Drones*, vol. 5, no. 2, p. 52, 2021.
- [3] S. Hrabar and G. S. Sukhatme, "A comparison of two camera configurations for optic-flow based navigation of a UAV through urban canyons," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2673–2680, IEEE, 2006.
- [4] T. Qin, P. Li, and S. Shen, "VINS-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

- [5] J. Engel, T. Schöps, and D. Cremers, “LSD-SLAM: Large-scale direct monocular SLAM,” in European Conference on Computer Vision (ECCV), 2014, pp. 834–849.
- [6] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM: A versatile and accurate monocular SLAM system,” IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147–1163, 2015.
- [7] E. Kaufmann, A. Loquercio, R. Ranftl, A. Dosovitskiy, V. Koltun, and D. Scaramuzza, “Deep drone racing: Learning agile flight in dynamic environments,” in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 6343–6350.
- [8] J. Fu, Y. Song, Y. Wu, F. Yu, D. Scaramuzza, “Learning deep sensorimotor policies for vision-based autonomous drone racing,” arXiv preprint arXiv:2210.14985 , 2022.
- [9] A. Cyba, H. Szolc, and T. Kryjak, “A simple vision-based navigation and control strategy for autonomous drone racing,” arXiv preprint arXiv:2104.09815 , 2021.
- [10] M. Navardi, A. Shiri, E. Humes, N. Waytowich, “An optimization framework for efficient vision-based autonomous drone navigation,” Journal of Big Data , vol. 8 , no . 1 , article no . 53 , 2021 .
- [11] Deep learning for vision-based navigation in autonomous drone racing .<https://www.sciencedirect.com/science/article/pii/B9780323857871000208>.
- [12] Home - AirSim - GitHub Pages. <https://microsoft.github.io/AirSim/>.
- [13] GitHub - hoangtranngoc/AirSim-RL: Experiments with Deep Reinforcement <https://github.com/hoangtranngoc/AirSim-RL>.
- [14] End-to-End Deep Learning for Autonomous Driving in AirSim. <https://www.luffca.com/2018/07/airsim-e2e-deep-learning/>.
- [15] Air Learning: a deep reinforcement learning gym for autonomous aerial <https://link.springer.com/article/10.1007/s10994-021-06006-6>.
- [16] Microsoft's Project AirSim tests, trains drones - The Robot Report. <https://www.therobotreport.com/microsofts-project-airsim-tests-and-trains-ai-powered-aircrafts/>.
- [17] Deep learning for vision-based navigation in autonomous drone racing <https://www.sciencedirect.com/science/article/pii/B9780323857871000208>.
- [18] Reinforcement Learning for Vision-Based Autonomous Drone Navigation. <https://sim2real.github.io/assets/papers/navardi.pdf>



INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.423



ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com



www.ijirset.com

Scan to save the contact details