IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 May 2024, accepted 3 June 2024, date of publication 5 June 2024, date of current version 13 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3409780

RESEARCH ARTICLE

Autonomous UAV Visual Navigation Using an Improved Deep Reinforcement Learning

HUSSEIN SAMMA[1] **AND SAMI EL-FERIK**[2,3]
[1]SDAIA-KFUPM Joint Research Center for Artificial Intelligence (JRC-AI), King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
[2]Control and Instrumentation Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
[3]Research Center for Smart Mobility and Logistics, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

Corresponding author: Hussein Samma (hussein.binsamma@kfupm.edu.sa)

This work was supported by Saudi Data and AI Authority (SDAIA) and King Fahd University of Petroleum and Minerals (KFUPM) under the SDAIA-KFUPM Joint Research Center for Artificial Intelligence Grant JRCAI-RG-07.

ABSTRACT In recent years, unmanned aerial vehicles (UAVs) have grown in popularity for a variety of purposes, including parcel delivery, search operations for missing persons, and surveillance. However, autonomously navigating UAVs in dynamic environments is a challenging task due to the presence of moving objects like pedestrians. In addition, traditional deep reinforcement learning approaches suffer from slow learning rates in dynamic situations and they need substantial training data. To improve learning performance, the present study proposed an enhanced deep reinforcement learning approach that encompasses two distinct learning stages namely the reinforced and self-supervised. In the reinforced learning stage, the deep Q-learning network (DQN) has been implemented and trained guided by the loss in the bellmen equation. On the other hand, the self-supervised stage is responsible for fine-tuning the backbone layers of DQN and it was directed by the contrastive loss function. The main benefit of incorporating the self-supervised stage is to speed up the encoding of the input scene captured by the UAV camera. To further enhance the navigation performances, an obstacle detection model was embedded to reduce UAV collisions. For experimental analysis, we have utilized an outdoor UAV-simulated environment called Blocks. This environment contains stationary objects that mimic buildings, as well as moving pedestrians. The study undertaken indicates that the implementation of the self-supervised stage led to significant improvements in navigation performance. Specifically, the simulated UAV was able to navigate longer distances in the correct direction toward the goal point. Moreover, the conducted analysis shows a significant navigation performance as compared with other DQN-based approaches like double DQN and dueling DQN.

INDEX TERMS UAV, visual navigation, deep reinforcement learning, obstacle avoidance, DQN.

I. INTRODUCTION

UAVs are increasingly used for several applications due to their high mobility, ease of deployment, and low maintenance costs. However, UAV autonomous navigation is considered as one of the most challenging tasks to implement. Vision-based models are extensively employed in the field of UAV navigation owing to their cost-effectiveness and adaptability, enabling operation in diverse environments such as indoors, outdoors, and under various weather conditions.

For vision-based approaches, using deep reinforcement learning models has emerged as the predominant methodology employed for UAV navigation [1], [2], [3], [4], and [5]. These models utilize computer vision techniques to encode the captured scenes of the navigated environment with the aim of navigating toward the desired destination. The work of Zhang et al. [1] implemented the Twin Delayed Deep Deterministic Policy Gradients (TD3) algorithm for UAV navigation in multi-obstacle environments. Their model was trained in a simulated environment, where the UAV learns to navigate to a target destination while avoiding obstacles. However, their virtual environment was simple and did not consider real 3D moving objects like a human. Further study

The associate editor coordinating the review of this manuscript and approving it for publication was Yu-Da Lin.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

was conducted in [2] where they adopted a DQN model for UAV navigation. The main objective of the trained UAV is to visit all mobile targets with the least energy consumption. Their approach was evaluated in both simulation and real field and the results show that the DQN agent can achieve good performances. Nevertheless, DQN algorithms require a large amount of training data to learn encoding the trained environment. Shantia et al. proposed a two-stage visual navigation approach which has been shown to be effective in both simulated and real-world environments [3]. The first stage was used to estimate the robot's position and the second stage was trained to navigate the robot to a target destination using the estimated positions. A CNN-based scheme for automatic obstacle avoidance was investigated in [4]. The central concept is that a CNN model was trained using the input image captured by the frontal camera of the UAV to forecast both the steering angle as well as the collision probability of the UAV path.

Nevertheless, the approaches mentioned above did not work in dynamic environments that contain moving obstacles. It should be noted that handling a dynamic environment is a challenge due to the fact that deep reinforcement learning models will need a larger number of training trials to comprehend the navigational environment if obstacles are relocated. To mitigate this difficulty, this work introduces an improved approach where a self-supervised stage is embedded to fine-tune the backbone network of the DQN agent based on collected data in the replay buffer. In addition, an obstacle avoidance model is added to reduce UAV collisions. To our knowledge, no work has been done on combining self-supervised learning with reinforcement learning. The subsequent points provide a concise overview of the primary contribution of this work.

1) Better navigation performance by incorporating a self-supervised learning stage that enhances the navigation's performance.
2) Enhanced obstacle avoidance during the UAV navigation by embedding an obstacle detection model.
3) A cost-effective visual navigation system, which exclusively relies on depth images acquired by a UAV camera.
4) A comprehensive evaluation of the proposed approach through navigation in a dynamic environment with both static and moving objects.

This paper is organized as follows. In Section II, the related work is described. The proposed approach is explained in Section III. Section IV details the conducted experiments. Finally, the conclusions and future works are given in Section V.

II. RELEATED WORK

A. UAV NAVIGATION METHODS

Deep reinforcement learning techniques have been utilized for several UAV applications such as obstacle avoidance [6], [4], [7] as well as navigation [8]. For example, to help a UAV avoid crashes, researchers in [6] have developed a reinforcement learning strategy based on saliency detection for a flying obstacle. Their work focused on using a convolution neural network to get reliable estimates of the positions of obstacles. In their experiment, they tested their approach in a semi-physical aircraft simulation with different scenarios including flying in unpredictable trajectories towards the UAV, flying in the central positive orientation, or flying towards the right or left. However, the effectiveness of the proposed approach is heavily dependent on the accuracy of the saliency detection algorithm and not the navigation performance.

A strategy for UAV obstacle avoidance that is based on CNN was introduced in [4]. They have used Airsim simulation environment to train the proposed CNN model. A reward-driven obstacle avoidance approach was demonstrated in [7] where the implemented U-Net-based network predicts the subsequent motion of the UAV. Nevertheless, in both studies, they did not account for moving pedestrians as a dynamic obstacle which makes navigation more difficult.

Zhang et al. investigated the challenge of navigating within a multi-obstacle environment [1]. They have proposed to use of an actor-critic network to extract observational features from the surrounding environment. They set up ten static obstacles five cubes and five cylinders for UAVs training. During the testing phase, the obstacles were moved in a random manner and the results showed that the proposed actor-critic network outperformed DDPG and TD3. However, the researchers did not consider the presence of moving pedestrians as a dynamic obstacle that increases the difficulty of navigation.

Indoor localization using tags and a visual-inertial system for UAV navigation in GPS-denied environments were studied in [8]. Their concept relies solely on visual cues manually placed inside the indoor workplace. Unfortunately, their method is based on visual information, which might be unreliable to locate landmarks in the presence of occlusion. A CNN-based navigation model was trained offline using Udacity dataset images to learn both steering angles and the probability of obstacles [9].

Additional research was conducted in [3], wherein the authors explored and developed a two-stage visual navigation approach. In [3], Initially, the location estimator is trained using standard mapping positions, while the model-free reinforced learning (RL) agent learns about the environment using an approximated maze from the map. In a 2D simulated world, the RL agent learns how the robot controller affects its actions. Finally, in 3D, the agent uses CNN position estimations.

The research in [10] offers asynchronous curriculum experience replay (ACER) with deep reinforcement learning to solve the autonomous motion control (AMC) problem for UAVs in complex and unpredictable dynamic situations. The ACER exhibits a 5.59% improvement in convergence outcomes when compared to the twin delayed deep deterministic policy gradient (TD3) algorithm, which is considered as the current leading approach.

Navigation of UAV swarms were examined by many researchers. For example, a framework of fault-tolerant cooperation for UAV swarms was suggested in [11]. The proposed strategy, based on network graph theory and they have introduced a geometry-based collision avoidance approach utilizes onboard sensory information to detect and avoid potential collisions during the mission, ensuring the safety and integrity of the UAV swarm. Despite the required computational resources and processing power in [11] however their approach has many potential application such as search and rescue, cooperative exploration, and target surveillance. Another study [12] introduces an Information-Fusion based decentralized swarm decision which was designed to coordinate UAVs swarms in scenarios where communication is disrupted or fails. They have achieved this objective by integrating visual perception with communication-based information, enabling UAVs to maintain coordinated behaviors even when communication is interfered. An aspect of their methodology that presents a limitation pertains to the offline optimization of weight parameters using a heuristic genetic algorithm, which might restrict the system's adaptability to dynamic or evolving environments.

In 2024, several new UAV navigation methods were introduced. The study in [13] examined the navigation and landing processes of multiple UAVs within indoor settings. They have introduced a new approach for enabling the autonomous landing of UAVs in unfamiliar indoor environments, leveraging visual SLAM, semantic segmentation, terrain estimation, and a decision-making model. The paper in [14] introduces a novel method called Deep Learning-based Autonomous UAV Exploration (DLAE) to address the challenges of autonomous exploration for UAVs in complex and unknown environments. Unlike traditional autonomous ground robot exploration, DLAE utilizes cameras instead of radar sensors. To enhance the learning efficiency of autonomous UAV exploration, the method in [14] incorporates an invalid action masking scheme. Further recent work was given in [15] where they have introduced a computer vision-based collision avoidance system designed for autonomous drones within indoor environments. The proposed method aims to ensure safe navigation addressing the challenges of dynamic surroundings, particularly significant for the operation of multiple drones in urban settings. Nevertheless, this method is suitable for small-scale experiments involving two drones, scalability to larger drone swarms may necessitate further optimization and testing to ensure effective collision avoidance.

In [16], an extensive review was conducted, delving into the realm of AI-based approaches to UAVs navigation. This was further complemented by an additional review presented in [17], providing deeper insights into the subject matter. Furthermore, a recent review paper, documented in [18], provided a comprehensive explanation of path planning methodologies tailored specifically for autonomous UAVs.

In contrast to the aforementioned approaches, this study presents a new perspective on addressing UAV navigation complexities within dynamic environments, specifically targeting the obstacles posed by moving pedestrians and stationary buildings. The research introduces an advanced methodology integrating self-supervised techniques and an obstacle-detection model, detailed in Section II.

B. SELF-SUPERVISED METHODS FOR UAV APPLICATIONS

The concept of self-supervised learning was utilized for several UAV tasks such as path planning [19], depth estimation [20], and tracking [21].

In [19] UAV uses self-supervised learning to evaluate the expected surprise. This means it learns to predict and assess surprises based on its own experiences and data, without requiring labeled data or external supervision. As explained earlier that the traditional reinforcement learning often requires extensive trial and error, which can be time-consuming and inefficient. This method, by using expected surprise and world modeling, allows the UAV to make more informed decisions, leading to faster learning and better performance.

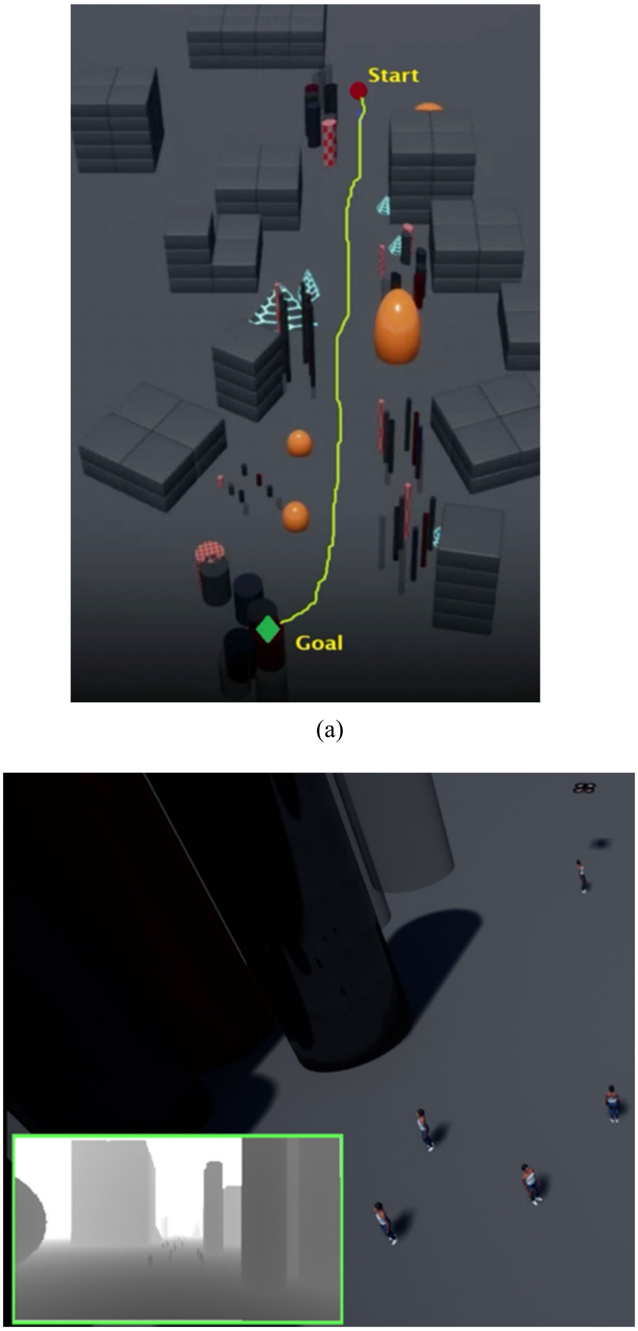
Self-supervised learning has been applied for depth estimation to facilitate UAV obstacle avoidance [20]. The method leverages the ease of unlabeled data collection and the adaptability of unsupervised learning, particularly self-supervised learning, allowing UAVs learn to estimate depth exclusively from images, without the necessity for ground truth depth or any supplementary data. This results in a more autonomous and flexible system capable of efficient and continuous learning in diverse environments.

For enhancing UAV tracking self-learning was used in [21]. By utilizing contrasting instances and self-learning methodologies, their approach independently constructs meaningful feature representations, bypassing the necessity for human annotations. This not only reduces the time and effort involved in manual annotation but also enhances the scalability and adaptability of the tracking system.

III. THE PROPOSED APPROACH

This work proposed an improved deep reinforcement learning model for autonomous UAV visual navigation in dynamic environments. Figure 2 depicts the navigation environment for the simulated UAV, with the aim of flying from the starting location displayed in Figure 2 (a) to the target point shown in green. The ideal path is highlighted in green. Figure 2 (b) visually illustrates a snapshot of a flying UAV in front of several pedestrians as an obstacle. It should be noted that this environment consists of two categories of obstacles: fixed ones, represented by the buildings, and movable ones, represented by the pedestrians. Furthermore, the depth image obtained from the current UAV view is visible in the lower-left corner of Figure 2(b).

The main architecture of the proposed navigation system is illustrated in Figure 3, and it includes two training strategies where the first of which is reinforced learning, and the second



(a)

(b)

FIGURE 1. The navigation environment for UAV (a) a top view with the optimal navigation path, and (b) A snapshot of a navigated UAV with a depth view, acquired by the UAV camera.

of which is self-supervised learning from the collected depth images. These learning methods are explained as follows.

A. REINFORCED TRAINING OF DQN

During the reinforced phase, DQN receives information about the recent action's reward value as well as the current UAV scene, as shown by the depth of the scene captured by the front camera of the UAV. Based on the current state depth scene the DQN will produce the navigation action that needs to be executed by the UAV which includes forward movement, moving along an angle of 45 degrees, or moving along an angle of −45 degrees as described in Table 1 and illustrated in Figure 1. One should noted that for each action the movement of the UAV was set to a duration of 3 seconds. The reason for that is to make sure that the UAV is making small navigation steps and to avoid the collision of near obstacles. On the other hand, less than 3 seconds step size will increase the training time of UAV.

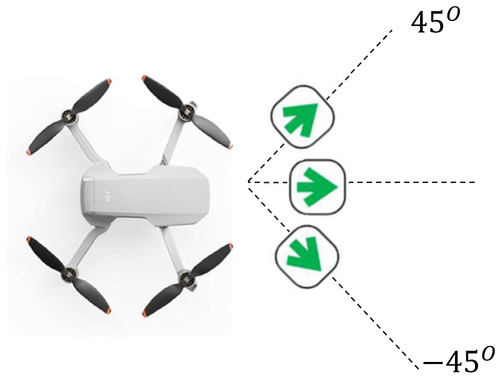


FIGURE 2. The action space of the simulated UAV.

TABLE 1. The implemented UAV actions.

No.	Action Description
1.	Move **forward** for a duration of **3 seconds**.
2.	Move **forward** along an **angle of 45** degrees for a duration of **3 seconds**.
3.	Move **forward** along an **angle of -45** degrees for a duration of **3 seconds**.

B. DQN ARCHITECTURE

In this study, the DQN model was implemented using a pre-trained ResNet50 network where the classification layer has been replaced with a fully connected network of 512 neurons. However, the last layer contains 3 neurons which will predict the Q-value (reward) of each action shown in Figure 3.

C. DQN REWARD FUNCTION

The reward function is the most important part of deep reinforcement learning. The formulated reward function in this work is defined as follows:

$$Reward = \begin{cases} 0 & \text{if collision occurred or UAV exceeds the limit of navigation area } (\textbf{terminationstate}) \\ \text{Use Equation (2)} & \textbf{otherwise} \end{cases} \tag{1}$$

As can be seen in Equation (1), when the UAV collides with a pedestrian or fixed obstacle, it is considered as a terminal

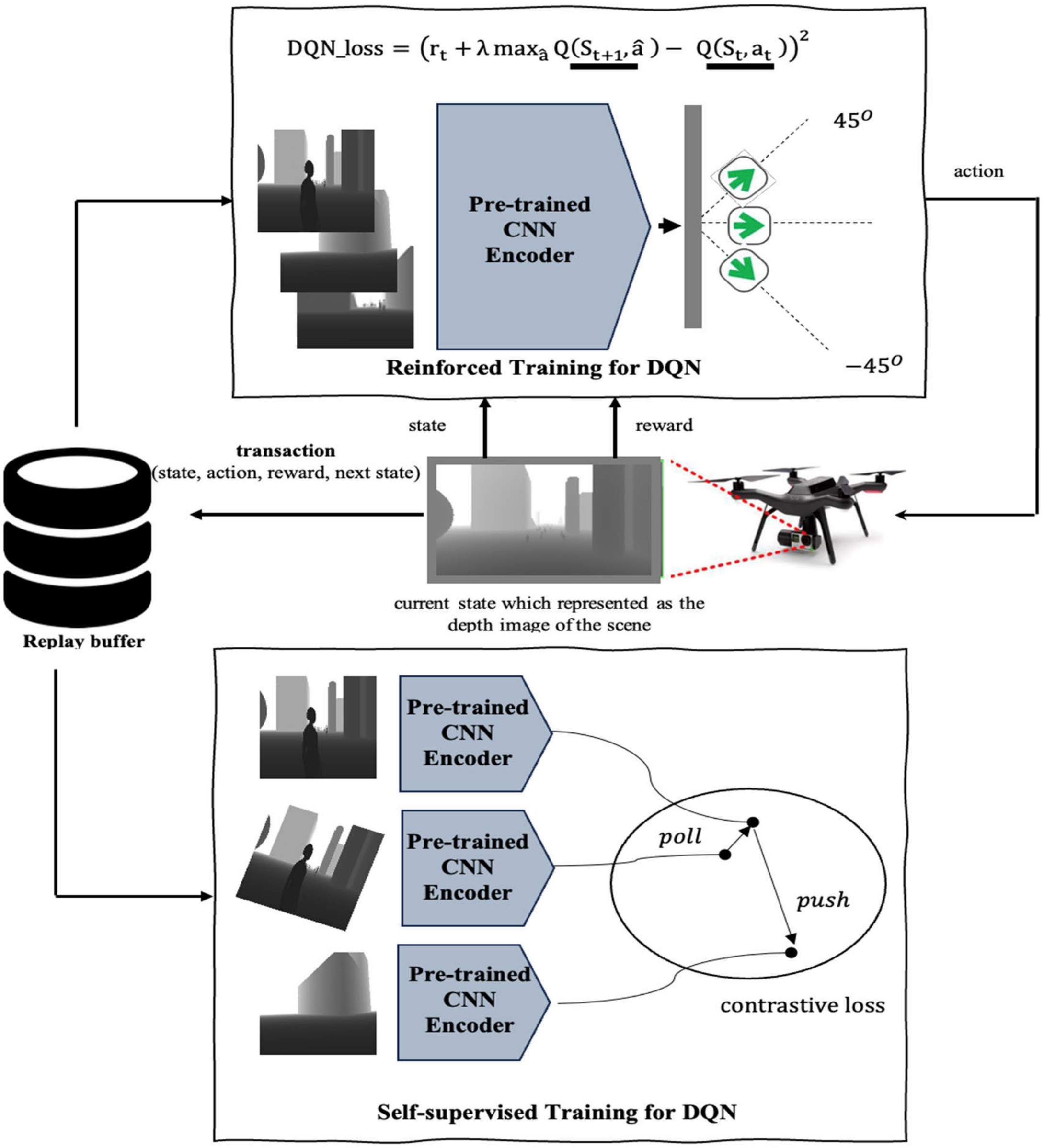

DQN_loss = (r_t + λ max_â Q(S_{t+1}, â) − Q(S_t, a_t))^2
45°
action
Pre-trained CNN Encoder
−45°
Reinforced Training for DQN
state
reward
transaction
(state, action, reward, next state)
Replay buffer
current state which represented as the depth image of the scene
Pre-trained CNN Encoder
Pre-trained CNN Encoder
Pre-trained CNN Encoder
poll
push
contrastive loss
Self-supervised Training for DQN

FIGURE 3. The proposed navigation approach.

state and the simulation should start again from the initial point shown in Figure 2 (a). Another simulation termination condition is when the UAV flies in the opposite direction of the target or to an open area with no pedestrian (go to left or right direction). Here we set an empirical threshold for the limit to be 10 units from the optimal path shown in Figure 2(a). After each step, a decrease in the reward by −0.1 is applied to encourage the UAV to reach its destination with minimum steps. If the distance from the goal is less than 3 units, then terminate the episode and start a new trail. After each navigation step, the reward value is computed as the difference between the previous distance to the target and the current distance after the execution of each action. If the UAV moves in the opposite direction, then the distance will be negative. It is defined as follows.

$$Diff = (X_{target} - X_{old}) - (X_{target} - X_{new}) \tag{2}$$

where, X_{target} is the location of the target point, X_{old} is the old location before taking the action, and X_{new} is the new location of UAV after the action.

D. DQN LOSS FUNCTION

The loss function of DQN is computed based on the Q-learning formula given below:

$$DQN\_loss = \left(r_t + \lambda(max_{\hat{a}}(Q\left(S_{t+1}, \hat{a}\right)) - Q\left(S_t, a_t\right)\right)^2 \tag{3}$$

where r_t is the reward received by the DQN agent at time t, λ is the learning rate parameter that takes a value from 0 to 1.

Q (S_t, a_t) is the Q-value of the currently executed action a_t based on given state S_t (current depth), however $Q\left(S_{t+1}, \hat{a}\right)$ is the Q-value of the next state S_{t+1}. It is worth mentioning that the *DQN_loss* is used only during the updating step and it utilizes a replay buffer data that stores the transaction as tuples of (state, action, reward, next state) as can be seen in Figure 3.

E. SELF-SUPERVISED TRAINING

In this stage, the DQN backbone will be fine-tuned in a self-supervised manner based on the depth images stored in the replay buffer. To accomplish this goal, a triplet of three images, one positive, one augmented, and one negative, will provided to the DQN backbone, and a contrastive loss function will be used to the resultant embedding to estimate the loss value. Following that, the DQN backbone weights will be iteratively changed dependent on the number of self-supervised training epochs. It's important to mention that the choice of positive and negative images will be randomized during the fine-tuning phase. Therefore, as the training buffer size increases, the likelihood of obtaining diverse images also increase. The implemented contrastive loss function is demonstrated in the following equations.

$$\cos\_sim(u, v) = (\mathrm{u}.v)/(|(|u|)|.||v||) \tag{4}$$

where *cos_sim* is the cosine function that computes the similarity of two vectors u and v in the embedding space.

$$\begin{aligned} &\text{Contrast\_loss} \\ &\quad = -log(\exp(cos\_sim(u, v+)/\tau))/(\exp(cos\_sim(u, v+)/\tau) \\ &\qquad + \exp(cos\_sim(u, v-)/\tau)) \end{aligned} \tag{5}$$

where u represents the original positive image, $v+$ is the augmented image generated by either rotation, scaling, etc from the positive image as shown in Figure 3. $v-$ is the negative image. Finally, the τ is a hyper-parameter in the range [0.1 to 0.5] called the temperature coefficient that determines how much weight to give the computed similarity score. In this investigation, τ was set to 0.1.

F. OBSTACLE DETECTION MODEL

To further enhance the navigation performances of UAVs, an obstacle detection model has been implemented in this study. Figure 4 depicts the main idea, which is to feed the current scene from the UAV's input camera into another obstacle detection model so that it can determine if the object in question is an obstacle. The ResNet50 serves as the backbone network and it is coupled to a fully connected (FC) 512-neuron softmax classifier that produces two outputs: Yes or No. To train the obstacle avoidance model, the data saved in the depth image's reply buffer that resulted in a terminal state (crash) is considered an obstacle class. Other depth images, on the other hand, will be classified as no obstacle, as seen in Figure 5. As the implemented approach reliance on a single depth image rather than a video sequence, distinguishing between moving and stationary pedestrians poses a significant challenge. Therefore, all pedestrians will be perceived as potentially obstacles if their pose toward the UAV. In addition, leveraging deep reinforcement learning, the system refines its understanding of these moving obstacles through trial and error, ultimately determining the safest distance from a moving pedestrians based on several factors such as the depth image pixel values and the pedestrian's pose. Finally, during the training period, this process will become automated, and the integrated self-supervised model will boost the learning rate by encoding generic features.

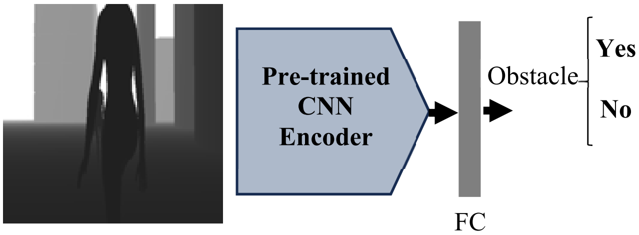


FIGURE 4. Obstacle detection model.

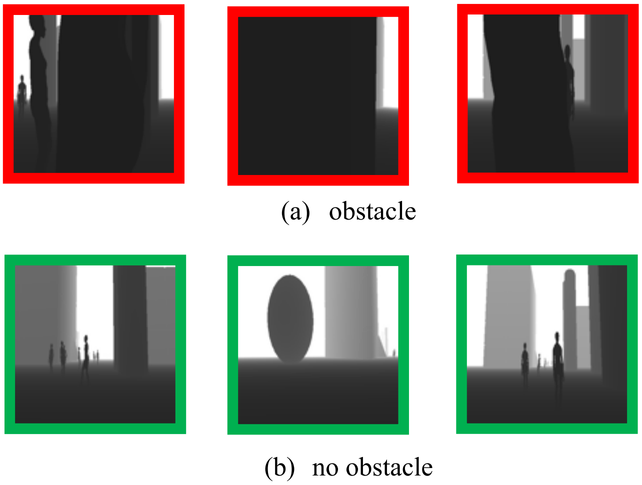


FIGURE 5. Sample images for training obstacle detection model (a) obstacle, (b) no obstacle.

A flowchart illustrating the key steps involved in integrating the navigation model with obstacle detection is presented in Figure 6.

IV. EXPERIMENTAL RESULTS

A. SIMULATION ENVIRONMENT

To carry out experimentation of the proposed approach, we have used an existing 3D outdoor environment [22] shown in Figure 1. This environment was built using an open-source gaming engine called Unreal Engine. To connect with the Block environment and control the movement of the UAV, we have chosen to utilize the AirSim package. This package facilitates the integration of a Python-based navigator with the Unreal Engine through the use of APIs as demonstrated in Figure 7.

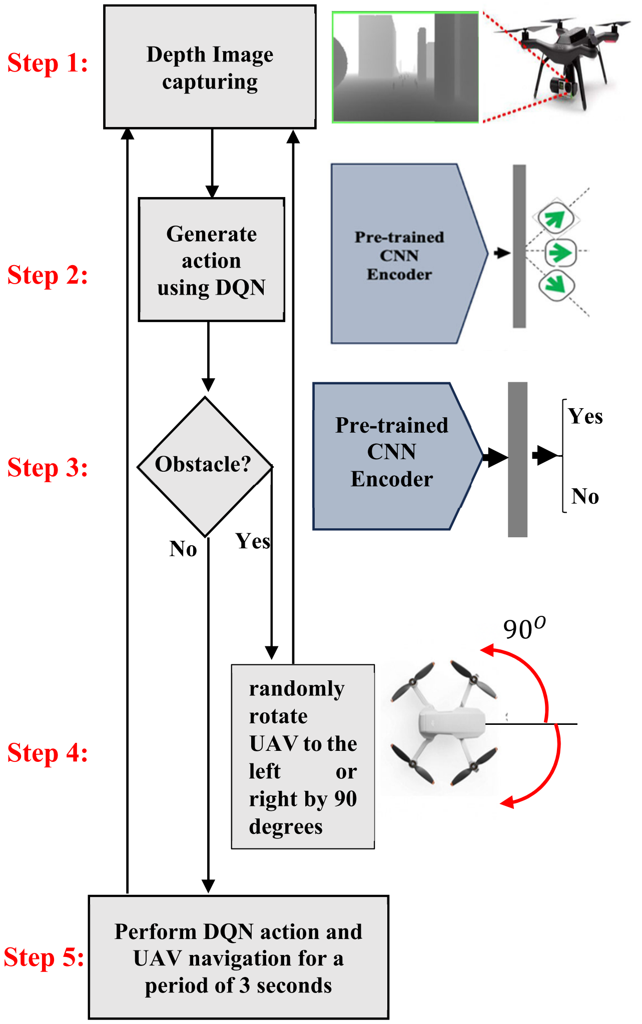

Step 1:
Depth Image capturing
Step 2:
Generate action using DQN
Pre-trained CNN Encoder
Step 3:
Obstacle?
No
Yes
Pre-trained CNN Encoder
Yes
No
Step 4:
randomly rotate UAV to the left or right by 90 degrees
90^0
Step 5:
Perform DQN action and UAV navigation for a period of 3 seconds

FIGURE 6. Flowchart of the main steps.

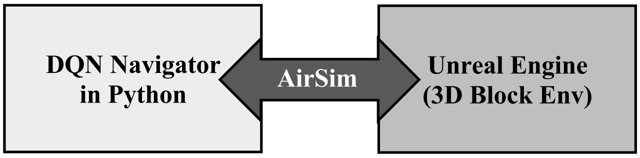


FIGURE 7. The adopted framework for UAV simulation.

B. PARAMETERS SETTINGS AND EVALUATION MEASURES

The implemented parameters related to deep reinforcement learning techniques that have been utilized in this study are given in Table 2. As can be seen, these settings are related to the maximum training episode; the size of the replay buffer, the batch training size which specifies how many samples are input to the model in a single batch, and the DQN parameters update time which controls the calling time for self-supervised and the reinforced training stages.

Moreover, the evaluation measures considered in this study are the average distance from the goal and the average number of collisions during the testing time, as well as the behavior of the loss function during the training time. These measures are widely utilized in other studies [1] and [5]. It is worth noting that the distance to the goal measure is highly dependent on the simulated environment and should be appropriately integrated into the designed reward function.

TABLE 2. Parameter settings.

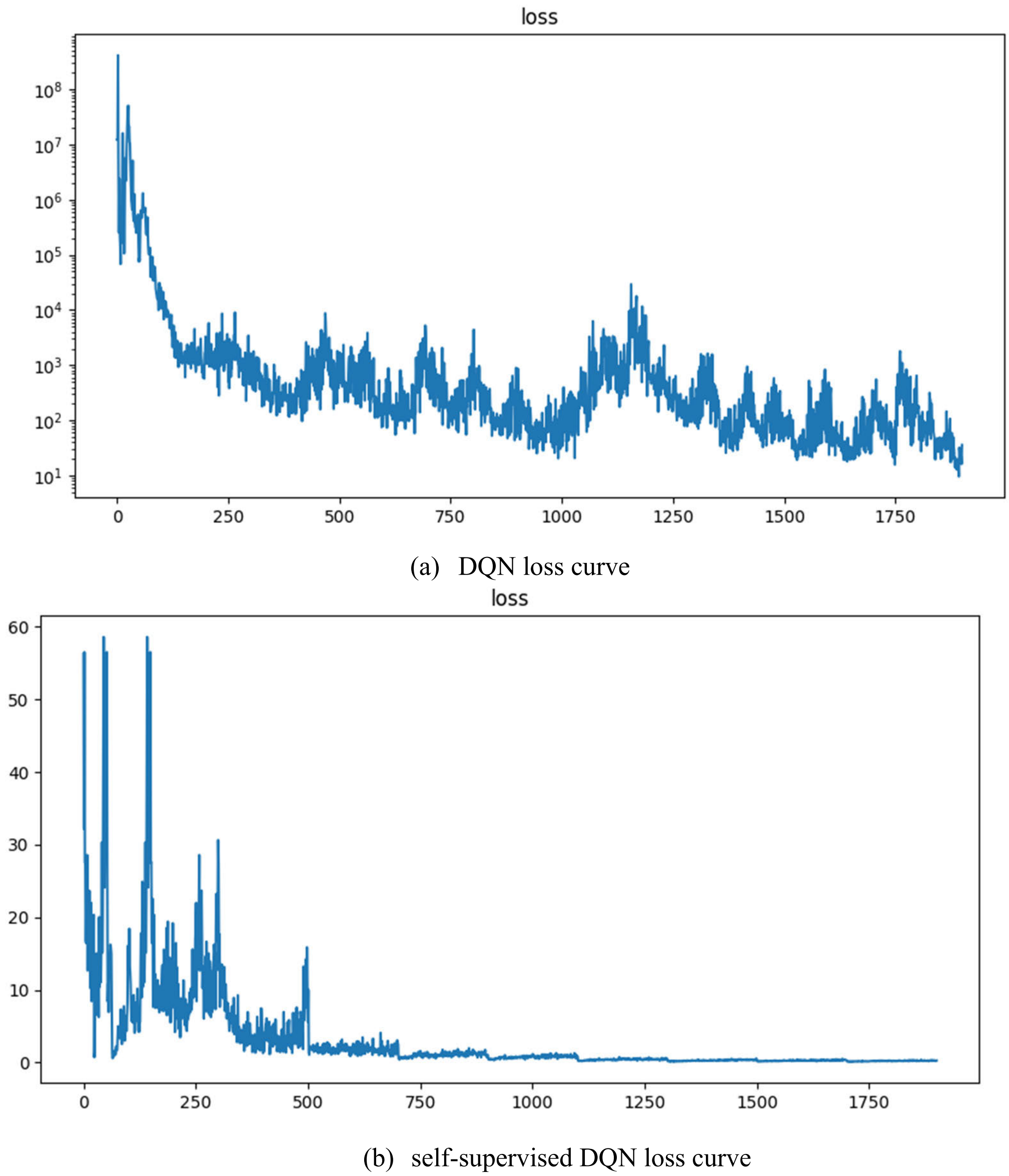
Parameters	value
Max epochs	2000
DQN update time	100 epochs
Batch size	16
Replay buffer size	10000
Learning rate	0.0001
UAV step duration	3 seconds
Self-supervised training epochs	100

C. LOSS CURVE ANALYSIS

This section will analyze the benefits of the incorporated self-supervised block with respect to the loss behavior. The superior convergence of the proposed self-supervised DQN is demonstrated by the loss curve in Figure 8, which yields reduced values. For instance, over the initial one thousand epochs, the loss value of DQN exceeds one hundred, as illustrated in Figure 8 (a). However, at epoch one thousand the loss value approaches zero with the implemented self-supervised DQN, Figure 8 (b). This is due to the increased efficacy of UAV scene encoding, which has accelerated the convergence of the loss curve.

D. NAVIGATION DISTANCE ANALYSIS

The analysis in this section was performed during the testing phase, subsequent to the training of both the self-supervised DQN and the DQN. Specifically, the UAV initiated its navigation from a new beginning point located approximately halfway along the best path depicted in Figure 2(a). In this analysis, two metrics for evaluation are used which are the average distance from the goal and the number of collisions. Furthermore, a maximum of 10 steps is permitted for each evaluated model. Table 3 displays the results of an experiment that was repeated ten times with the average value calculated. In terms of the average distance from the goal, the data demonstrate that the proposed self-supervised DQN achieved a better value of 157 which means the UAV is closer to the target. Nevertheless, the DQN model yielded a larger distance, indicating that it is still a long way from the target point. Nevertheless, the reported data in Table 3 demonstrated that the DQN has a smaller number of collisions. This is because the UAV is usually moved to an open area where no pedestrian is moving. This can be seen in Table 3 where the last raw shows the objects that the UAV hit during the 10 test runs. As can be seen, the DQN model moves UAV toward the Orange_Ball shown in Figure 1 (a). In contrast, the suggested self-supervised DQN guides the UAV toward the designated destination, resulting in an increased likelihood of collisions with pedestrians



(a) DQN loss curve

(b) self-supervised DQN loss curve

FIGURE 8. The training loss curves for a duration of 2000 epochs for both models (a) and (b).

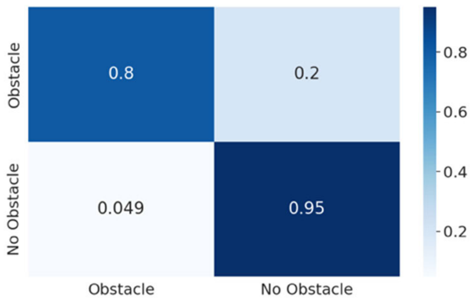


FIGURE 9. The confusion matrix for neural network.

TABLE 3. Performance analysis during testing phase.

	DQN	self-supervised DQN
The average distance from the goal	178	**157**
Average collisions	**2**	3
Collied with obstacles	Orange_Ball BP_person47	BP_person46 BP_person49 BP_person47 BP_person50

E. PERFORMANCE OF OBSTACLES DETECTION MODEL

To further minimize the occurrence of collisions, a model for detecting obstacles was integrated into the suggested self-supervised DQN, as depicted in Figure 4. The fundamental concept is that the captured depth image by the UAV is passed to the obstacle detection model to make a prediction regarding whether the UAV will crash or not. If the prediction is yes, the

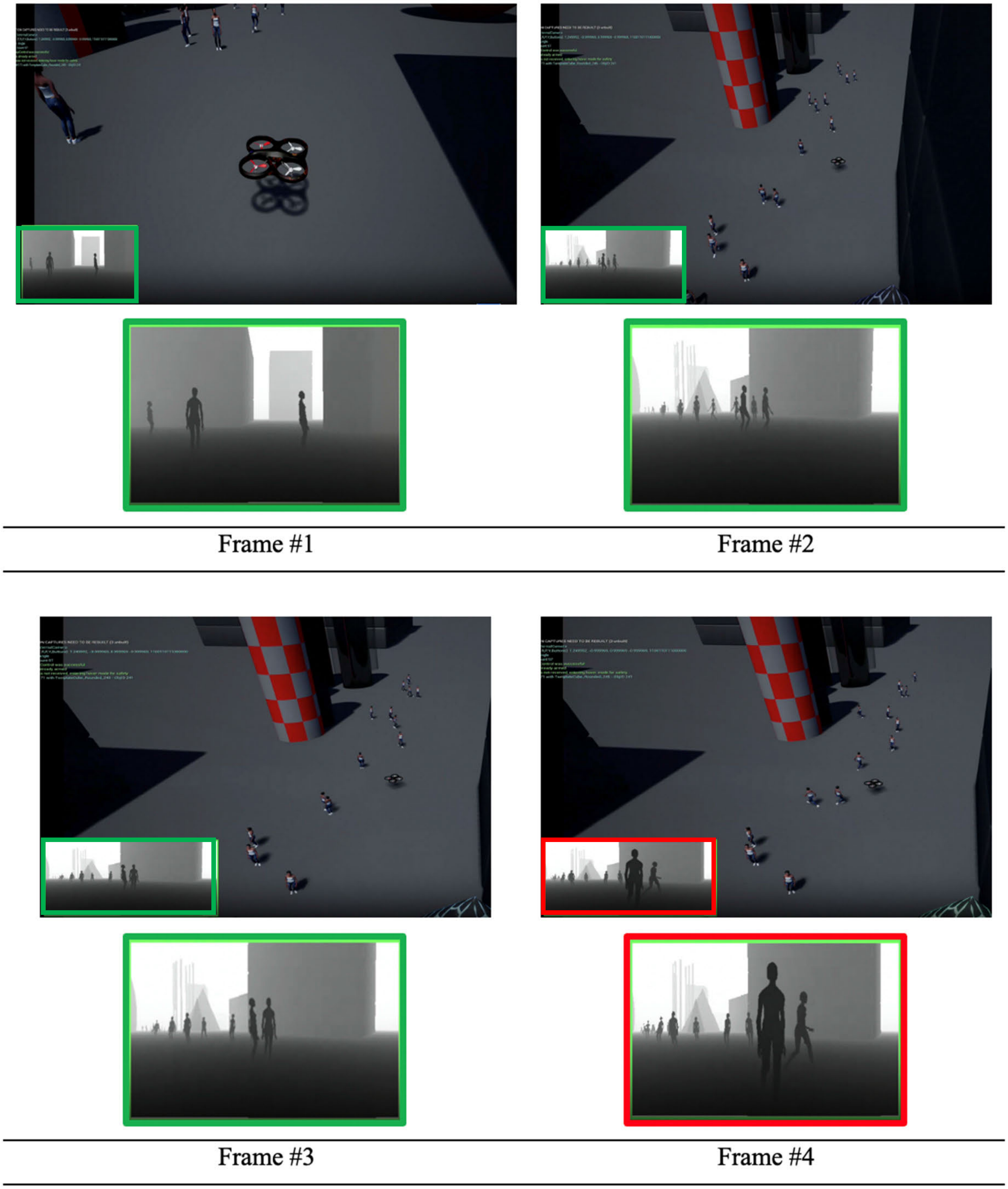


FIGURE 10. Snapshot frames during UAV test phase.

UAV will randomly rotate to the left or right by 90 degrees to avoid the predicted obstacle.

To train the obstacle detection model, a total of 443 images for obstacle type and 1374 images for no obstacle were used. After that, the dataset was split into train with 70% and test with 30%. Figure 9 displays the outcomes of calculating the confusion matrix. The trained model achieves an obstacle detection accuracy of 80% for obstacle classes and 95% for no-obstacle classes. In addition, the impact of incorporating the obstacle detection model into the suggested navigation system was examined. This experiment has been repeated 10 times and the findings are presented in Table 4. The integrated obstacle detection model demonstrates that the UAV can avoid more obstacles while covering a nearly equivalent distance. The last column in Table 4 lists the names of the objects that the UAV collided with during all

TABLE 4. Performance analysis of obstacle detection model.

	self-supervised DQN	self-supervised DQN (with obstacle detection)
The average distance from the goal	157	159
Average collisions	3	2
Collied with obstacles	BP_person46 BP_person49 BP_person47 BP_person50	BP_person46 BP_person50 BP_person49

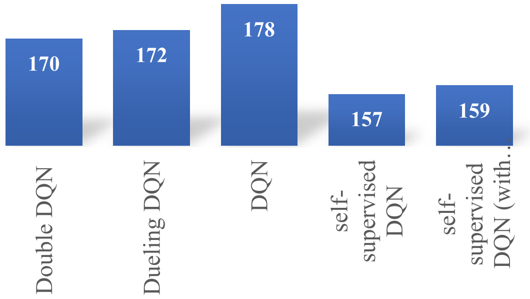


FIGURE 11. The average distance from the goal point for all implemented models.

trials. As can be seen with the embedded obstacle detection model UAV was able to avoid more pedestrians. Nevertheless, avoidance of moving obstacles remains a challenging task because they can move toward the UAV and hit it from the side or back. For visual illustration, a snapshot of the navigation environment, UAV depth view, and classification results of the proposed detection model is given in Figure 10. As can be seen, the suggested model clearly identifies a person in the front as an obstacle with red color.

F. COMPARE WITH OTHER DQN-BASED MODELS

Further analysis has been conducted by evaluating the performances against other well-known deep reinforcement learning algorithms such as Double DQN [23] and Dueling DQN [24]. The underlying principle of both networks is the utilization of an additional network, referred to as the target network. It is useful in simulated games to fix the issue of DQN overestimation. However, this case study is a simple visual navigation problem where the objective is to reach a target location in a GPS-denied environment while avoiding both moving and fixed obstacles. As such, in comparison to the other versions stated, DQN's simplicity and low complexity made it a viable option in this study.

In the analysis, we have examined the average distance from the goal for the UAV to navigate with a maximum of 10 steps for 10 trials. The results achieved by all algorithms are plotted in Figure 11. As can be seen, the proposed self-supervised DQN achieved the highest traveled distance among all algorithms. This is because of the self-supervised model which is beneficial to speed up the learning rate. Also, the obstacle detection model helps to reduce crashes caused by stationary and moving obstacles.

It should be noted that self-supervised learning often useful for dealing with variations and noise in the data. Therefore, the DQN network learns to be more robust to these variations, making it less prone to overfitting or errors on unseen data. In addition, the contrastive learning approaches makes the backbone to learn from the inherent structure and relationships within the unlabeled data, improving its ability to extract meaningful information and encoding a genetic feature. As such, by learning these generic features, the DQN backbone becomes better equipped to handle unseen data or variations within the target domain (i.e. during the navigation of UAV).

V. CONCLUSION AND FUTURE APPLICATIONS

An improved deep reinforcement learning for autonomous UAV visual navigation is introduced in this study. This has been accomplished by incorporating a self-supervised deep learning stage to enhance the encoding of DQN for the captured scene by the UAV. Further improvement has been introduced by implementing an obstacle detection model. The conducted results confirm the benefits of the incorporated self-supervised component for speeding up the learning curve and navigation performances. As compared with other methods, the numerical results showed that the proposed self-supervised DQN reported the maximum traveled distance with a minimum collision rate. This is because the extracted scene encoding and obstacle detection capabilities lead to more accurate and efficient navigation, as the UAV can better understand its surroundings and avoid collisions.

As a future work, the self-supervised DQN could be used for indoor visual navigation in GPS-denied environments to perform several tasks such as searching, inspection, counting. Additionally, several outdoor applications could be considered. One such application is navigation through difficult-to-access areas to collect data for research in fields like geology, meteorology, and marine biology. This enables scientists to gather critical information from remote or hazardous locations. Another important application is in search and rescue missions, where UAVs can autonomously navigate challenging terrains and provide real-time visual data to help locate missing persons more efficiently. Lastly, in precision agriculture, UAVs can autonomously fly over fields to monitor crop health, manage irrigation, and detect pests or diseases, thereby enhancing farming efficiency and productivity.

In addition, several ideas could be investigated for improving the proposed approach. For example, incorporating object detection models like YOLOv7 or DETR to enhance obstacle recognition accuracy and handles diverse object types. Further improvements could be explored, such as implementing a lightweight Deep Q-Network (DQN) architecture to enhance deployment efficiency and reduce inference time. Moreover, the integration of additional sensor inputs, such as thermal cameras, could yield valuable data across different conditions. By incorporating obstacle detection models utilizing ultrasound sensors, the navigation capabilities and safety of the UAV could be further enhanced.

Finally, the proposed approach could be deployed on a real UAV and validated in an actual outdoor environment, ensuring its effectiveness and reliability of the proposed approach.

REFERENCES

[1] S. Zhang, Y. Li, and Q. Dong, "Autonomous navigation of UAV in multi-obstacle environments based on a deep reinforcement learning approach," *Appl. Soft Comput.*, vol. 115, Jan. 2022, Art. no. 108194, doi: 10.1016/j.asoc.2021.108194.

[2] A. Soliman, A. Al-Ali, A. Mohamed, H. Gedawy, D. Izham, M. Bahri, A. Erbad, and M. Guizani, "AI-based UAV navigation framework with digital twin technology for mobile target visitation," *Eng. Appl. Artif. Intell.*, vol. 123, Aug. 2023, Art. no. 106318, doi: 10.1016/j.engappai.2023.106318.

[3] A. Shantia, R. Timmers, Y. Chong, C. Kuiper, F. Bidoia, L. Schomaker, and M. Wiering, "Two-stage visual navigation by deep neural networks and multi-goal reinforcement learning," *Robot. Auto. Syst.*, vol. 138, Apr. 2021, Art. no. 103731, doi: 10.1016/j.robot.2021.103731.

[4] X. Dai, Y. Mao, T. Huang, N. Qin, D. Huang, and Y. Li, "Automatic obstacle avoidance of quadrotor UAV via CNN-based learning," *Neurocomputing*, vol. 402, pp. 346–358, Aug. 2020, doi: 10.1016/j.neucom.2020.04.020.

[5] A. Anwar and A. Raychowdhury, "Autonomous navigation via deep reinforcement learning for resource constraint edge nodes using transfer learning," *IEEE Access*, vol. 8, pp. 26549–26560, 2020, doi: 10.1109/ACCESS.2020.2971172.

[6] Z. Ma, C. Wang, Y. Niu, X. Wang, and L. Shen, "A saliency-based reinforcement learning approach for a UAV to avoid flying obstacles," *Robot. Auto. Syst.*, vol. 100, pp. 108–118, Feb. 2018, doi: 10.1016/j.robot.2017.10.009.

[7] S.-Y. Shin, Y.-W. Kang, and Y.-G. Kim, "Reward-driven U-Net training for obstacle avoidance drone," *Expert Syst. Appl.*, vol. 143, Apr. 2020, Art. no. 113064, doi: 10.1016/j.eswa.2019.113064.

[8] N. Kayhani, W. Zhao, B. McCabe, and A. P. Schoellig, "Tag-based visual-inertial localization of unmanned aerial vehicles in indoor construction environments using an on-manifold extended Kalman filter," *Autom. Construct.*, vol. 135, Mar. 2022, Art. no. 104112, doi: 10.1016/j.autcon.2021.104112.

[9] M. A. Arshad, S. H. Khan, S. Qamar, M. W. Khan, I. Murtza, J. Gwak, and A. Khan, "Drone navigation using region and edge exploitation-based deep CNN," *IEEE Access*, vol. 10, pp. 95441–95450, 2022, doi: 10.1109/ACCESS.2022.3204876.

[10] Z. Hu, X. Gao, K. Wan, Q. Wang, and Y. Zhai, "Asynchronous curriculum experience replay: A deep reinforcement learning approach for UAV autonomous motion control in unknown dynamic environments," *IEEE Trans. Veh. Technol.*, vol. 72, no. 11, pp. 13985–14001, Nov. 2023, doi: 10.1109/TVT.2023.3285595.

[11] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin, "Fault-tolerant cooperative navigation of networked UAV swarms for forest fire monitoring," *Aerosp. Sci. Technol.*, vol. 123, Apr. 2022, Art. no. 107494, doi: 10.1016/j.ast.2022.107494.

[12] Z. Wang, J. Li, J. Li, and C. Liu, "A decentralized decision-making algorithm of UAV swarm with information fusion strategy," *Expert Syst. Appl.*, vol. 237, Mar. 2024, Art. no. 121444, doi: 10.1016/j.eswa.2023.121444.

[13] L. Yang, J. Ye, Y. Zhang, L. Wang, and C. Qiu, "A semantic SLAM-based method for navigation and landing of UAVs in indoor environments," *Knowl.-Based Syst.*, vol. 293, Jun. 2024, Art. no. 111693.

[14] Y. Zhao, J. Zhang, and C. Zhang, "Deep-learning based autonomous-exploration for UAV navigation," *Knowl.-Based Syst.*, vol. 297, Aug. 2024, Art. no. 111925.

[15] J. Estevez, E. Nuñez, J. M. Lopez-Guede, and G. Garate, "A low-cost vision system for online reciprocal collision avoidance with UAVs," *Aerosp. Sci. Technol.*, vol. 150, Jul. 2024, Art. no. 109190, doi: 10.1016/j.ast.2024.109190.

[16] S. Rezwan and W. Choi, "Artificial intelligence approaches for UAV navigation: Recent advances and future challenges," *IEEE Access*, vol. 10, pp. 26320–26339, 2022, doi: 10.1109/ACCESS.2022.3157626.

[17] A. P. Kalidas, C. J. Joshua, A. Q. Md, S. Basheer, S. Mohan, and S. Sakri, "Deep reinforcement learning for vision-based navigation of UAVs in avoiding stationary and mobile obstacles," *Drones*, vol. 7, no. 4, p. 245, Apr. 2023, doi: 10.3390/drones7040245.

[18] G. Gugan and A. Haque, "Path planning for autonomous drones: Challenges and future directions," *Drones*, vol. 7, no. 3, p. 169, Feb. 2023, doi: 10.3390/drones7030169.

[19] A. Krayani, K. Khan, L. Marcenaro, M. Marchese, and C. Regazzoni, "Self-supervised path planning in UAV-aided wireless networks based on active inference," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2024, pp. 13181–13185.

[20] L. Madhuanand, F. Nex, and M. Y. Yang, "Self-supervised monocular depth estimation from oblique UAV videos," *ISPRS J. Photogramm. Remote Sens.*, vol. 176, pp. 1–14, Jun. 2021.

[21] X. Wang, D. Zeng, Y. Li, M. Zou, Q. Zhao, and S. Li, "Enhancing UAV tracking: A focus on discriminative representations using contrastive instances," *J. Real-Time Image Process.*, vol. 21, no. 3, p. 78, May 2024.

[22] *Blocks Environment*. Accessed: Jan. 2024. [Online]. Available: https://microsoft.github.io/AirSim/unreal_blocks/

[23] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. AAAI Conf. Artif. Intell.*, Mar. 2016, vol. 30, no. 1, pp. 2094–2100, doi: 10.1609/aaai.v30i1.10295.

[24] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, "Dueling network architectures for deep reinforcement learning," 2015, *arXiv:1511.06581*.

HUSSEIN SAMMA received the bachelor's degree in computer engineering from Yarmouk University, Jordan, in 2006, the master's degree in computer engineering from Jordan University of Science and Technology, in 2009, and the Ph.D. degree in computer vision and machine learning from Universiti Sains Malaysia (USM), in 2016. From 2018 to 2022, he was a Senior Lecturer with the Faculty of Engineering, School of Computing, Universiti Teknologi Malaysia (UTM), and the School of Electrical Engineering, Universiti Sains Malaysia (USM). He is currently a Researcher with the SDAIA-KFUPM Joint Research Center for Artificial Intelligence, King Fahd University of Petroleum and Minerals. His research interests include computer vision, deep learning, reinforcement learning, swarm intelligence, and soft biometrics.

SAMI EL-FERIK received the B.Sc. degree in electrical engineering from Laval University, Québec City, QC, Canada, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from Polytechnique Montréal, Canada. After the completion of the Ph.D., and postdoctoral positions, he was with the Research and Development Center of Systems, Controls, and Accessories, Pratt and Whitney, Canada. He is currently a Professor of control and instrumentation engineering with the Department of Systems Engineering, King Fahd University of Petroleum and Minerals. He is also the Director of the Interdisciplinary Research Center for Smart Mobility and Logistics. His research interests include sensing, monitoring, multiagent systems, and nonlinear control, with strong multidisciplinary research and applications.