2024 International Conference on Unmanned Aircraft Systems (ICUAS)
June 4-7, 2024. Chania, Crete, Greece

DroneRanger: Vision-Driven Deep Learning for Drone Distance Estimation

Hamid Azad[1], Varun Mehta[2], Iraj Mantegh[2], Miodrag Bolic[1]

Abstract— **This paper introduces a novel approach to estimating the distance between a drone and a camera using deep learning techniques. The proposed method employs a low-complexity convolutional neural network (CNN), called DroneRanger, to analyze the captured 2D image and estimate the distance between the observer and target drones. Three types of input data for the CNN regression model are investigated, including extended bounding box, resized bounding box, and resized bounding box with additional size information. The effectiveness of the method is demonstrated through experiments conducted on both synthetic datasets built using AirSim© as well as real flight tests, showcasing its performance across various simulation conditions, including different weather and environments. Furthermore, experiments conducted on real-world data captured using camera-equipped drones validate the method's performance under practical conditions. To address uncertainties in training labels caused by imperfect localization information from GPS sensors, robust regression based on the Huber loss function is employed to improve accuracy (improvement of around 2 meters in RMSE compared to the MSE loss). These findings suggest promising prospects for accurately estimating 3D distances from 2D images (with RMSE of distance estimation error less than 5 meters and R^2 values of above 0.9 for the regression task), highlighting the potential of the proposed approach for real-world problems in drone applications such as collision avoidance between drones.**

I. INTRODUCTION

Uncrewed or Unmanned Aerial Vehicles (UAVs) or drones have gained significant attention and widespread use in various sectors, such as agriculture, surveillance, and transportation [1]–[4]. However, the surge in drone deployment has raised concerns about safety and security, leading to an increasing focus on counter-UAV measures including localization.

Localization in drone navigation refers to establishing the exact position of the drone within its environment, a form of information that is essential for autonomous navigation. Vision-based localization methods rely on visual information, like images or videos from onboard or external cameras, to determine the drone's position accurately. The inherent loss of depth information in 2D representations, compounded by factors like perspective distortion and lighting variations, contributes to the difficulty of distance estimation using vision data. These challenges necessitates innovative Deep Learning (DL)-based approaches for accurate distance estimation in drone localization. This paper investigates the application of the DL-based method in the problem of drone distance estimation using vision data. Accurate drone localization including distance estimation is essential for enabling autonomous navigation and self-guided flight, obstacle avoidance, and efficient mission planning. Additionally, having such information about any intruder drone allows for the assessment of necessary defensive actions depending on its closeness to the restricted zone. This allows for informed decisions regarding the necessary response level, and thus, improves the overall situational awareness.

A. Literature review

A limited number of works have employed vision cameras for the sake of UAV distance estimation. The authors in [5] addressed the development of a deep learning-based method for estimating distances to avoid mid-air collisions in UAVs. This method relied solely on a monocular camera to detect an approaching intruder drone (fixed-wing type) and estimate its distance. To estimate the distance of the detected drone, two distinct DL approaches, CNN and DNN (Convolutional/Deep Neural Network [6]), were employed. The CNN approach, chosen for its robustness, comprised a 5-layer VGG16-based network, followed by a 4-layer deep one. To evaluate the performance of the proposed networks, a synthetic dataset was generated using Blender. A notable drawback of this study lies in the utilization of a complex network for distance estimation which results in a high computational burden. In the work by Patel et al. [7], they introduced a hybrid framework for drone detection and distance estimation. This approach employed a linear regression network to estimate the distance, utilizing inputs such as bounding box coordinates, embedded features from the pre-step object detection network, and the mean RGB value of pixels within the bounding box. The authors of [8], [9] introduced an optical spatial localization system for UAVs, relying on a single camera. This system involved a blinking LED ring affixed to the UAV as a marker, alongside an event-based dynamic vision sensing camera and the developed 3D localization algorithm which was implemented on a base station. The algorithm could determine the location of the UAV by using the known physical parameters of the marker after detecting the blinking LED marker with the event-based camera. However, a notable limitation of this approach is its reliance on specific hardware installations like LEDs, making it impractical for scenarios involving non-owned drones.

Considering the constraints highlighted in the preceding

[1]H. Azad and M. Bolic are with the School of Electrical Engineering and Computer Science (SEECS), University of Ottawa, 800 King Edward, Ottawa, On., Canada {hamid.azad, miodrag.bolic}@uottawa.ca

[2]V. Mehta and I. Mantegh are with the National Research Council Canada (NRC), Montreal, QC., Canada {varunkumar.mehta, iraj.mantegh}@cnrc-nrc.gc.ca

2024 International Conference on Unmanned Aircraft Systems (ICUAS) | 979-8-3503-5788-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICUAS60882.2024.10557013

lines, such as the elevated computational burden or the need for additional hardware in existing algorithms, we introduce a novel shallow regression network. Our network comprises four convolutional layers, each containing 32 or fewer filters, facilitating low-level complex processing for distance estimation. The network takes input from the cropped image area within the bounding box, readily obtained from the output of the drone detection step. We have explored the impact of three types of inputs and have also introduced a robust regression method employing Huber loss to handle uncertain labels. The proposed network has demonstrated satisfactory performance in estimating the 3D distance of detected drones, using captured 2D images, both in synthetic and real-world scenarios.

The rest of the paper is organized as follows. The next section details the motivation of this work including problem definition. Section III provides an overview of the proposed solution for drone distance estimation using vision data based on a deep regression network. The simulation results as well as results from practical tests are presented in section IV. The paper concludes with a summary of the findings.

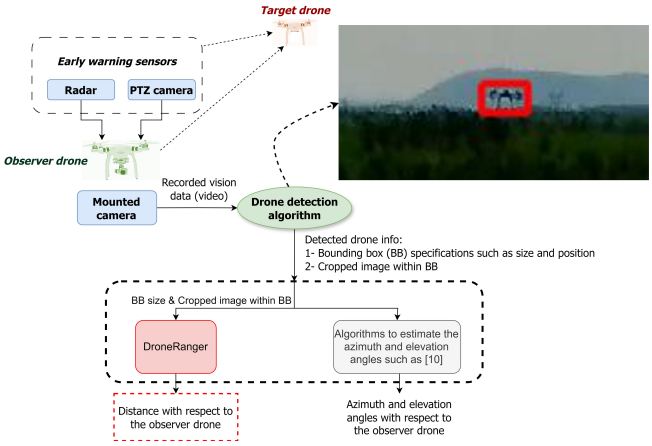


Fig. 1: Block diagram of the multi-sensor counter-UAV system (Blue blocks: deployed sensors, Green block: the drone detection block, the output of which is used in this study, Red block: proposed method for distance estimation (azimuth and elevation angles can be estimated using algorithms such as the one in [10]).

II. Problem Definition

The counter-UAV system in Fig. 1 is a multi-sensor one that includes a ground-mounted radar and Pan-Tilt-Zoom (PTZ) camera, which can be considered as early warning sensors to detect the intruder drones. However, these sensors have limitations in flexibility and coverage area for long distances, potentially hindering their effectiveness in addressing threats from target drones. To address this, an alternative approach is equipping an observer drone with a counter-UAV sensor to expand the defense coverage, and thus, enhance the effectiveness of the whole system. Given the constraints on payload (both size and weight) and power availability, mounting multiple sensors becomes quite challenging. Therefore, utilizing a vision camera emerges as a feasible solution due to its affordability, low power consumption, and potential for compact implementation. Therefore, as shown in Fig. 1, the vision camera serves as the primary and sole sensor on the observer drone for counter-UAV purposes. The images and videos captured by this sensor are crucial for identifying and neutralizing target drones by extracting vital information such as their location. After detecting an unknown drone using the radar and PTZ camera, the observer drone approaches it for closer inspection. The output of the onboard camera is continuously processed using object detection algorithms [11]–[13], likely YOLO-v5 [14]–[17] or its combination with Simple Online and Realtime Tracking (SORT) with a deep association metric (DeepSORT) tracker [18], [19], providing crucial data for analysis. These algorithms result in essential outputs such as the bounding box's position and size (in pixels), detection timestamp, and confidence level in identifying a drone. The extracted bounding box data enables the cropping of the region of interest from the image frame for subsequent processing. The distance estimation algorithm (here, DroneRanger) is then applied to the detected drone. The problem here is determining the distance in 3D space between the observing and target drones based on the 2D frames captured by the drone's camera. In the next section, the proposed method based on the deep regression network will be presented.

III. Proposed Method (DroneRanger)

Following drone detection in the captured image or video using tools like YOLO, the information about the resulting bounding box around the identified drone is forwarded to post-processing algorithms for tasks like drone distance estimation. This bounding box holds key information about the drone's size, position within the image, and its represented area. Utilizing this bounding box data is crucial for accurately estimating the 3D location of the target drone relative to the assumed reference coordinate frame on the camera (observer drone). To estimate the azimuth and elevation angles, acceptable results can be obtained by utilizing the bounding box position in the frame and assuming the simple pinhole model, as discussed in [10]. This study considers the estimation of the distance between the camera and the target drone based on the captured 2D images.

We employ regression with the developed CNN to estimate the distance from cropped images obtained via the object detector network (here, YOLO-v5 [14]). Training the regression network on data with ground truth distance labels enables it to associate visual features with varying distances [20], [21], and make accurate distance predictions. Visual features often correlate with the distance between the camera and the drone, with changes in size, perspective, or sharpness as the drone moves closer or farther away. Training on a diverse dataset enables the model to learn these correlations, leading to more effective distance estimation. The presence of convolutional layers facilitates the extraction of the non-linear relationship between visual features and distance by capturing relevant features across various scales and orientations.

A. Regressor architecture

Fig. 2 depicts the CNN-based distance regressor architecture. It consists of four convolutional layers, each followed by batch normalization, ReLU activation, and average pooling. Starting with an input layer, the first convolutional layer

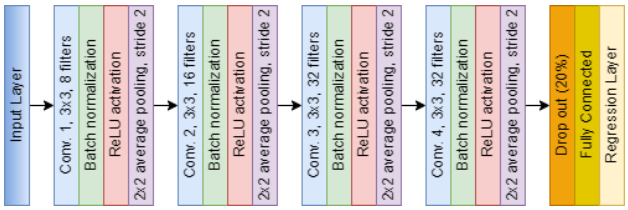


Fig. 2: The architecture of DroneRanger (CNN-based distance regressor)

utilizes 8 filters of size 3×3. Batch normalization enhances training stability and speed by normalizing previous layer activations. The nonlinearity of ReLU helps to capture complex data patterns. Following this, average pooling reduces feature map dimensions by half using a 2×2 operation with a stride of 2. This sequence repeats three times, with filter numbers increasing progressively (16, 32, and 32 filters in the three following layers, respectively). Dropout regularization, applied next, randomly sets 20% of activations to zero during training, preventing overfitting and enhancing generalization. The fully connected layer reduces output to a single scalar value, serving as the regression head. Finally, the regression layer computes the loss between the predicted and true distances and minimizes the regression error during training. We call this network as DroneRanger. Simulation results demonstrate the DroneRanger's acceptable performance in distance estimation, achieving low absolute error.

1) Network input: The initial layer sets the network's input size according to the dimensions of the training images (i.e., the size of cropped bounding box), ensuring uniformity in the expected input image size. This study explores three scenarios for input images:

- *Input Type 1*: In the first case, a fixed-size rectangular area measuring $V \times H$ pixels (representing the number of pixels along the vertical and horizontal axes, respectively) is cropped around the center of the bounding box. This cropped image is then fed into the network (Fig. 3).
- *Input Type 2*: In the second case, the input image is a cropped version of the bounding box area generated by the object detection network. When the bounding box size differs from $V \times H$ pixels, the cropped image is resized to match these dimensions using the Bicubic interpolation method (Fig. 3).
- *Input Type 3*: In the second input type, resizing all cropped bounding boxes to the fixed size of $V \times H$ pixels causes the actual size of the drone in the image to be lost. Given that this size information is indicative of the drone's distance to the camera (farther drones appear smaller in the captured image), the third input type addresses this by utilizing the same images as type 2. However, additional information is imported into the network, namely, the width and height of the bounding box in pixels.

(a) Input Type 1 (b) Input Type 2

Fig. 3: Illustration of input images to the network. (a) The drone maintains its actual size in the image, but the extended bounding box may contain various background forms. (b) In the resized image, the drone no longer maintains its actual size, leading to the loss of the inherent "3D distance-bounding box size relation". Consequently, the third input form is proposed to address this limitation.

Throughout this paper, V and H are set to 80 and 150 pixels, respectively. These values are chosen based on the maximum probable size of the detection bounding box, determined by the size of desirable targets and the resolution of the captured image. It is worth mentioning that all images are presumed to be in the compressed *JPEG* format.

2) Loss function and robust regression using Huber loss: As mentioned in the preceding section, accurate ground truth distance values are essential for training the regression network. This paper presents the training and testing of the proposed network using both simulated and experimental data. As it will be discussed in the results section, simulation data, synthesized by AirSim©, allows for the precise extraction of ground truth distance within the implemented AirSim© scenario. However, in the practical testing setup outlined in Section IV.B (Experimental Results), ground truth distance data is obtained through GPS sensors on both the observer and target drones. By utilizing GPS information for each drone, the distance between them can be easily estimated. Nevertheless, the inherent uncertainty in GPS data introduces corresponding uncertainty in the ground truth distance values, potentially resulting in outliers in the training data.

The two most commonly used loss functions in DL applications are the mean square error (MSE) and mean absolute error (MAE), as shown in equations (1) and (2) respectively:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{1}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{2}$$

Here, y_i and $\hat{y}_i$ represent the ground truth and estimated values (i.e., the output of the deep network) for the desired variable (in this case, the distance). While MSE loss exhibits a fast learning rate owing to its convex nature, it can be

sensitive to outlier values due to the square function. On the other hand, in MAE, all errors are weighted equally due to the linearity model, making it more robust to outliers than MSE. To leverage the advantages of both a fast learning rate and robustness, the Huber loss was introduced in [22]:

$$L_\delta(y_i, \hat{y}_i) = \begin{cases} \frac{1}{2}(y_i - \hat{y}_i)^2 & \text{if } |y_i - \hat{y}_i| \leq \delta, \\ \delta|y_i - \hat{y}_i| - \frac{1}{2}\delta^2 & \text{Otherwise.} \end{cases} \quad (3)$$

Within this framework for the loss function, a weighted MAE loss form is applied to the case of larger errors (i.e., outliers), whereas for smaller error values, the quadratic MSE is employed. This study investigates the impact of both MSE and Huber loss functions on the performance of the regression network.

IV. Performance Analysis

A. Simulation results

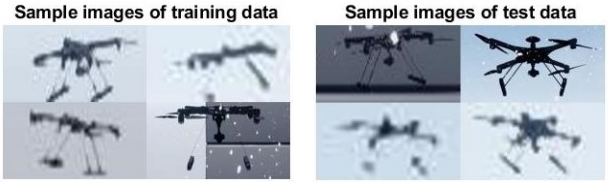
This section evaluates the performance of our proposed distance estimation method using simulated data from AirSim©, utilizing our previously published dataset [23]. The dataset includes the drone's Cartesian coordinates (XYZ values) in AirSim©'s internal frame, facilitating evaluation of the localization algorithm. AirSim© includes an internal "object detection" feature that directly offers bounding box information without requiring the use of a separate object detection network. This guarantees consistent access to bounding box data across all recorded frames, minimizing the chance of missed detections typical in DL-based detection networks. Furthermore, this feature provides ground-truth (GT) bounding box information, potentially eliminating the need for manual annotation (although this is not within the scope of the current study).

Analyzing the simulated dataset will provide a crucial benchmark for evaluating the accuracy of the localization algorithm. Assessing the performance of the simulated data offers valuable insights into the capabilities and constraints of the method before its deployment in real-world settings.

The stochastic gradient descent optimizer is utilized with an initial learning rate of 1×10^{-5} to train the regression network in Fig. 2. Also, the number of epochs is set to be 20. In each case, 80% of the dataset is allocated for training, while the remaining 20% is set aside for testing/validation.

To begin evaluating the performance of the distance estimation method, we first examine a basic scenario featuring one type of drone across different weather conditions. A Quadrotor model is simulated in the Blocks environment under sunny, rainy, and snowy conditions, each comprising around 800 samples of simulated data (equivalent to 1920 and 480 samples for the training and testing phases, respectively). Fig. 4 displays sample images from both the training and testing datasets.

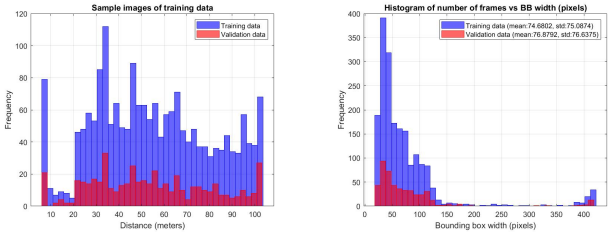
The histogram in Fig. 5 illustrates the distribution of the drone's distance and width (based on the size of the bounding box before resizing) in both the training and testing datasets. These figures indicate that the majority of drone distances fall within the 20 to 100-meter range, with a notable



(a) Training data (b) Testing data

Fig. 4: Some sample input images (*Input Type 2*) used to train and test the deep network in the synthetic dataset case (scenario 1)

concentration around 30 to 40 meters, suggesting diverse distance scenarios in the dataset. However, the distribution of data versus distance is not uniform, as depicted in Fig. 5, with some distances having less data. This discrepancy is attributed to the manual control of the drone during data capture in AirSim©, where human actions influenced key presses, resulting in fewer data instances for certain distances. Furthermore, most bounding boxes have a width below 150 pixels, aligning with the predetermined fixed size of input images for the CNN regression network (considering a camera with a field of view of 82 degrees and a resolution of 1080×1920 pixels).



(a) Distance (b) Bounding box width

Fig. 5: Distribution of the distance and the size of the bounding box in the first simulated scenario

In the case of considering the MSE loss, the Root Mean Square Error (RMSE) of distance estimation error values for the three input types mentioned in the preceding sections are 5.89, 5.87, and 3.49 meters, respectively. Although the performance of both the extended and resized bounding box inputs (Input Type 1 and 2) is almost the same in this scenario, it is important to note that the dataset used in this analysis solely consists of samples from the Blocks environment. In this environment, most samples feature a blue sky background (Fig. 4). In such instances, the extended bounding box (as in Input Type 1) contains minimal information, primarily the blue sky background in almost all the samples. Conversely, in environments with backgrounds other than blue sky, the extended bounding box area contains significantly varying backgrounds among dataset samples (see Fig. 3, for example). This, in turn, can potentially degrade the performance in the Input Type 1 case. However, this variation can adversely affect the performance of the output associated with Input Type 2. In that case, where only the bounding box area is cropped, the background

behind the drone does not occupy a substantial portion of the imported images to the network, minimizing its influence on the results. Finally, as evident from the comparison of results between Input Types 2 and 3, adding information about the actual size of the bounding box can help in enhancing distance estimation performance. As previously noted, the inherent information in the bounding box's actual size provides insights into the target drone's distance from the camera.

To validate the claim about the difference between the Input Types 1 and 2, and compare the results with the previous case, the same drone model is considered in two Blocks and City Park environments. Considering 800 frames for each environment, with 80% allocated for training and 20% for testing/validation, we utilize 1280 frames for training and 320 frames for testing/validation, respectively. Examples of the captured images can be seen in Fig. 6 (the distribution of the data is depicted in Fig. 8).

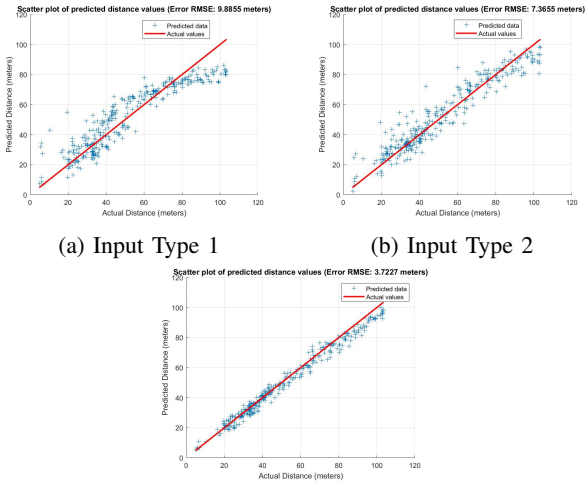
(a) Training data (b) Testing data

Fig. 6: Some sample input images (*Input Type 2*) used to train and test the deep network in the synthetic dataset case (scenario 2)

The scatter plot, illustrating the estimated distance versus the actual value, is presented in Fig. 7. As expected, the regression network exhibits improved performance in the case of Input Type 2 compared to Type 1, with the RMSE decreasing by approximately 25%, from 9.89 to 7.37 meters. Moreover, the favorable impact of incorporating information about the bounding box is confirmed again in this scenario through a comparison of the results between Input Types 2 and 3. In addition to RMSE, the coefficient of determination R^2 serves as a statistical metric, reflecting the portion of the variability in the dependent variable explained by the independent variable [24]. In scatter plots illustrating estimated distance against true values, incorporating the R^2 value offers an understanding of how well the regression line aligns with the data points, indicating the goodness of fit. This coefficient can be calculated using the following equation [24]:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{4}$$

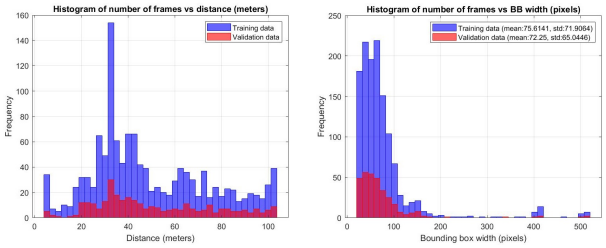
where, SS_{res} and SS_{tot} denote the sum of squared errors between the ground truth and estimated values, and the total sum of squares, respectively. The computed R^2 values for the three input types in Fig. 7 are 0.8419, 0.9122, and 0.9776, respectively. These findings highlight the superiority of Input Type 3 over the other two types. According to the findings presented, it seems that the third type of input (resized bounding box image area plus its size in pixels)



(a) Input Type 1 (b) Input Type 2

(c) Input Type 3

Fig. 7: Scatter plot depicting the estimated distance versus the true value for the second simulated scenario



(a) Distance (b) Bounding box width

Fig. 8: Distribution of the distance and the size of the bounding box in the second simulated scenario

yields the most favorable result. This input format effectively addresses the limitations of the other two types, overcoming the negative impact of the background in the extended bounding box case (Type 1) and mitigating the loss of the actual bounding box size (Type 2). In the upcoming section, the proposed method will be applied and analyzed using real data. Additionally, the impact of the chosen loss function will be examined.

Finally, as a critical analysis, the performance of the CNN regression network should be evaluated on unseen data. In all previous instances, the same dataset was utilized for both the training and testing phases, thereby maintaining identical data distributions. However, it is common to observe a domain shift between these two sets in most scenarios. To address this, the network was trained using Quadrotor and DJI Mavic drones but tested on a dataset recorded using DJI FPV for the final simulation in this section (an unseen drone type in the test data compared to the training one). The second type of input is utilized here, with all other simulation parameters remaining consistent with previous examples. The only modification made is to adjust the number of frames for the training and test/validation steps to 1600 (800 frames from both the Quadrotor and DJI Mavic

drones) and 800 (from the DJI FPV), respectively. The results are depicted in Fig. 9. As illustrated, the RMSE increased to approximately 8.2 meters, though it remains within an acceptable range. Future investigations should delve deeper into cases involving domain shift, where differences in the distributions of training and testing data are present. In terms

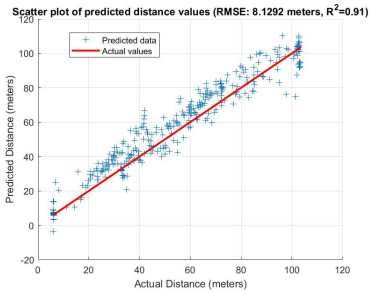


Fig. 9: Scatter plot depicting the estimated distance versus the true value for the case of domain shift (training: Quadrotor and DJI Mavic drones, testing: DJI FPV)

of computational load, it is worth mentioning that after conducting 5 independent inference runs of the DroneRanger network, each comprising 1330 frames, the average inference time per frame is under $5ms$ on an Ubuntu desktop equipped with a 16GB Nvidia RTX 4000 GPU.

B. Experimental results

While simulations offer controlled settings for training and testing deep networks, their real-world performance is crucial. Undesirable effects such as lighting variations, occlusions, and perspective shifts in real-world scenarios introduce complexities that affect the network's generalization. Hence, evaluating the performance of the regression network on the practical data is essential. This section outlines the network's application on the recorded dataset from the conducted real experiments and discusses its performance.

The experiments involved deploying a quadrotor drone as the target (Fig. 10), while another quadrotor drone with a camera captured images of drones at different distances (Fig. 11). The air-to-air configuration of the test introduced extra complexities because of the relative motion and perspective shifts between the camera and the target drone.

(a) Close view (b) Drone in the sky

Fig. 10: The target drone in the experimental test

Fig. 11: The drone equipped with the onboard camera as the observer drone

The observer drone had a camera with a 62.7-degree field of view, capturing frames at a resolution of 1080×1920 pixels and recording videos at a frame rate of 30 frames per second. Finally, it should be mentioned that there were GPS sensors mounted on both drones. The data from these two sensors were utilized to extract the GT distance values between the target and observer drones.

The practical dataset comprises of diverse scenarios, with the quadrotor positioned at distances ranging from 20 to approximately 100 meters from the camera, mirroring real-world situations. The drone's width varied from fewer than 20 pixels to over 120 pixels in the frames. Recorded during the observer drone's hovering phase, the camera captured videos of the target drone executing different maneuvers (Fig. 12), including approaches, retreats, lateral movements (equivalent to different ground truth azimuth angles relative to the camera), and altitude changes (various elevation angles). Fig. 13 displays the 3D trajectory of both the observer

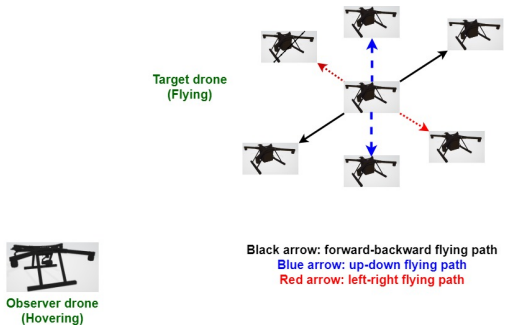
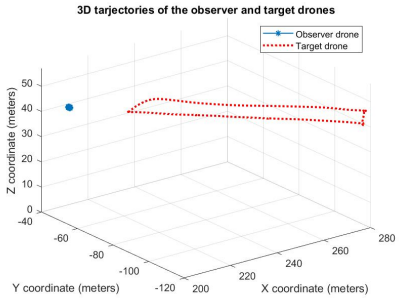


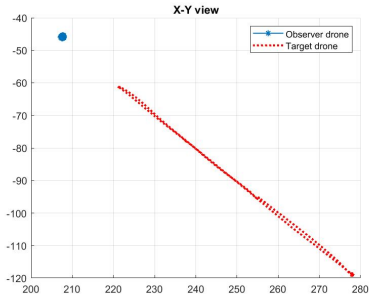
Fig. 12: The flying paths of the target drone in front of the observer one in the experimental test

and target drones during the training phase. It reveals that the target drone maintained distances between roughly 20 to 100 meters from the observer drone's camera. However, there were instances where the target drone paused at different distances, hovering in position. These stationary intervals are evident in the histogram representation of the recorded data shown in Fig. 14.

The training dataset comprised 1816 images. A subset of sample images from the training data, is depicted in Fig. 15. These images exhibit diverse backgrounds and lighting conditions, crucial for training DL-based methods. This diversity allows the model to learn robust features, enhancing its ability to handle real-world variations effectively.

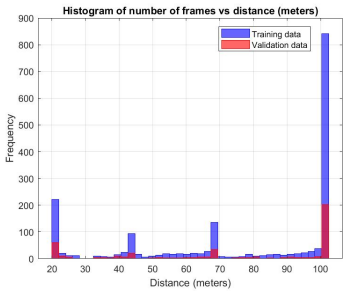


(a) 3D trajectory

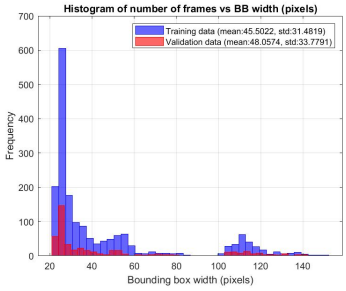


(b) X-Y view

Fig. 13: Flight trajectory of the drones in the real scenario (the one which used for extracting the training data)



(a) Distance



(b) Bounding box width

Fig. 14: Histogram displaying the distribution of training data in the experimental test

Based on the results outlined in the synthetic data section, Input Type 3 exhibits the most superior performance among the three suggested input types in the preceding sections. Consequently, the regression network was trained using this input format. Regarding the loss function, as detailed in

Fig. 15: Some sample input images (*Input Type 2*) used to train the regression network in the experimental test

Section III, since the GT values were derived from GPS sensors, both MSE and Huber losses were employed and compared. Other settings for training the networks were selected similar to the synthetic data section.

Several flight tests were run using the two drones. As an illustration of the test data, consider the sample input images depicted in Fig. 16. As evident from this figure, the drone exhibited various orientations throughout the test. Additionally, the camera captured the target drone from different viewing angles. The total number of sample images in this test amounts to 2243.

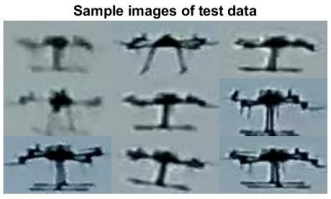


Fig. 16: A subset of input images (*Input Type 2*) provided to the regression network during the experimental testing phase

The results of applying the trained network to this dataset are presented in Fig. 17. To smooth the estimated distance output, a simple moving average (MA) filter with a length of 10 is applied to the output. It is worth noting that the recorded video has a frame rate of 30 frames per second, resulting in a time interval of $1/30$ seconds between consecutive frames. Considering this time resolution, applying an MA filter with 10 samples would correspond to an interval of approximately 0.33 seconds. Within such intervals, the location of the drone does not exhibit significant changes, allowing us to effectively utilize the MA filter to smooth out the data. As observed, the target drone covered distances ranging from approximately 40 to 100 meters to the camera. The RMSE for the MSE and Huber loss functions is 6.87 and 4.96 meters, respectively. The adoption of the robust regression network with the Huber loss resulted in an improvement of approximately 2 meters in the RMSE of the error. Finally, this curve illustrates the feasibility of estimating the 3D distance between the target drone and the camera using solely the 2D captured images.

V. CONCLUSIONS

This study introduced a novel method for drone distance estimation employing deep learning techniques, leveraging

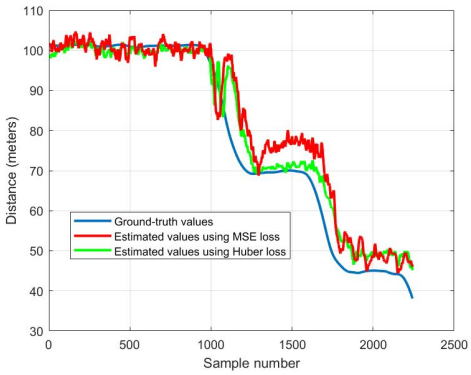


Fig. 17: Estimated distance for the test data in the experimental test by the trained regression network using two forms of the loss function (*Input Type 3*)

both simulated and real-world data. Our results indicate that incorporating actual bounding box size in addition to the cropped image within the bounding box, significantly improves distance estimation accuracy. These simulation results are further validated through practical flight tests and the analysis of aerial vision data captured by a drone. Despite challenges such as varying lighting conditions and perspective changes, our proposed method demonstrates feasibility and robustness in drone distance estimation based solely on using a cost-effective vision camera. The simple architecture of the CNN network, coupled with its low computational load, enables real-time implementation. Further investigations could explore additional factors such as domain shift (in the background, light/weather conditions, and even drone type) and integration with other sensor modalities to enhance performance and applicability in complex environments.

ACKNOWLEDGMENT

This work is supported by the Canadian Safety and Security Program, which is led by Defence Research and Development Canada's Centre for Security Science (DRDC CSS), in partnership with Public Safety Canada. The Canadian Safety and Security Program is a federally-funded program to strengthen Canada's ability to anticipate, prevent/mitigate, prepare for, respond to, and recover from natural disasters, serious accidents, crime, and terrorism through the convergence of science and technology with policy, operations, and intelligence. Partial funding is provided from the NSERC Discovery Grant RGPIN-2020-04417, and complementary support has also been provided by National Research Council Canada through Artificial Intelligence for Logistics and Integrated Aerial Mobility programs.

REFERENCES

[1] S. A. H. Mohsan, N. Q. H. Othman, Y. Li, M. H. Alsharif, and M. A. Khan, "Unmanned aerial vehicles (UAVs): Practical aspects, applications, open challenges, security issues, and future trends," *Intelligent Service Robotics*, vol. 16, no. 1, pp. 109–137, 2023.

[2] J.-P. Huttner and M. Friedrich, "Current challenges in mission planning systems for UAVs: A systematic review," in *2023 Integrated Communication, Navigation and Surveillance Conference (ICNS)*. IEEE, 2023, pp. 1–7.

[3] D. Fortune, H. Nitsch, G. Markarian, D. Osterman, and A. Staniforth, "Counter-unmanned aerial vehicle systems: Technical, training, and regulatory challenges," *Security Technologies and Social Implications*, pp. 122–148, 2022.

[4] S. A. H. Mohsan, M. A. Khan, F. Noor, I. Ullah, and M. H. Alsharif, "Towards the unmanned aerial vehicles (UAVs): A comprehensive review," *Drones*, vol. 6, no. 6, p. 147, 2022.

[5] Y.-C. Lai and Z.-Y. Huang, "Detection of a moving UAV based on deep learning-based distance estimation," *Remote Sensing*, vol. 12, no. 18, p. 3035, 2020.

[6] M. A. Haseeb, J. Guan, D. Ristic-Durrant, and A. Gräser, "DisNet: a novel method for distance estimation from monocular camera," *10th Planning, Perception and Navigation for Intelligent Vehicles (PPNIV18), IROS*, 2018.

[7] V. Patel, V. Mehta, M. Bolic, and I. Mantegh, "A hybrid framework for object distance estimation using a monocular camera," in *2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC)*. IEEE, 2023, pp. 1–7.

[8] H. Stuckey, A. Al-Radaideh, L. Escamilla, L. Sun, L. G. Carrillo, and W. Tang, "An optical spatial localization system for tracking unmanned aerial vehicles using a single dynamic vision sensor," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 3093–3100.

[9] H. Stuckey, A. Al-Radaideh, L. Sun, and W. Tang, "A spatial localization and attitude estimation system for unmanned aerial vehicles using a single dynamic vision sensor," *IEEE Sensors Journal*, vol. 22, no. 15, pp. 15 497–15 507, 2022.

[10] H. Azad, "Deep learning based drone localization and payload detection using vision data," Master's thesis, University of Ottawa, 2023.

[11] R. Jiang, Y. Zhou, and Y. Peng, "A review on intrusion drone target detection based on deep learning," in *2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, vol. 4. IEEE, 2021, pp. 1032–1039.

[12] Y. Liu, P. Sun, N. Wergeles, and Y. Shang, "A survey and performance evaluation of deep learning methods for small object detection," *Expert Systems with Applications*, p. 114602, 2021.

[13] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," *arXiv preprint arXiv:2104.11892*, 2021.

[14] F. Dadboud, V. Patel, V. Mehta, M. Bolic, and I. Mantegh, "Single-stage uav detection and classification with yolov5: Mosaic data augmentation and panet," in *2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2021, pp. 1–8.

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

[16] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, 2022.

[17] C.-Y. Wang, H.-Y. M. Liao, and I.-H. Yeh, "Designing network design strategies through gradient path analysis," *arXiv preprint arXiv:2211.04800*, 2022.

[18] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 3645–3649.

[19] N. Wojke and A. Bewley, "Deep cosine metric learning for person re-identification," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 748–756.

[20] L. Huang, T. Zhe, J. Wu, Q. Wu, C. Pei, and D. Chen, "Robust inter-vehicle distance estimation method based on monocular vision," *IEEE Access*, vol. 7, pp. 46 059–46 070, 2019.

[21] X. Zhang, L. Zhang, D. Xu, and H. Pei, "Multi-loss function for distance-to-collision estimation," in *2021 8th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS)*. IEEE, 2021, pp. 205–210.

[22] K. Gokcesu and H. Gokcesu, "Generalized huber loss for robust learning and its efficient minimization for a robust statistics," *arXiv preprint arXiv:2108.12627*, 2021.

[23] H. Azad, V. Mehta, F. Dadboud, M. Bolic, and I. Mantegh, "Air-to-air simulated drone dataset for AI-powered problems," in *2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC)*, 2023, pp. 1–7.

[24] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to linear regression analysis*. John Wiley & Sons, 2021.