FUZZING ON TARGET DETECTION SYSTEM OF UNMANNED AERIAL VEHICLE

Hepeng Dai[1], Zerui Wang[1], Jie Ding[1], Guozhi Li[1], Chentao Wang[1], Leiming He[1] & Huai Liu[2]

[1]Chinese Aeronautical Establishment, Beijing, China
[2]Swinburne University of Technology, Melbourne, Australia

Abstract

Thanks to the rapid development of deep learning, unmanned aerial vehicles (UAVs) are able to quickly locate targets of interest in a wide aerial view, which is widely used in many fields (such as traffic monitoring, and electric power inspection). Existing deep learning (DL) model testing techniques are unable to generate test cases that match the UAV's perspective, which may result in inadequate testing target detection systems of UAV. In this paper, we propose a new testing framework based on fuzzing technique, called FTU, to generate realistic images for UAV and automatically verify the test results based on metamorphic relations. FTU is evaluated through empirical studies on one dataset and two DL models (namely YOLO_v5 and YOLO_v7). The experimental results show that FTU can generate effective adversarial inputs, which can help improve the robustness of UAV's target detection systems

Keywords: Unmanned Aerial Vehicle, Target Detection System, Fuzzing, Metamorphic Relation

1. Introduction

Computer vision technology endows unmanned aerial vehicles (UAVs) with decision-making, analysis, and autonomous perception capabilities, in which, target detection is one of the key techniques to improve the perception capability of UAVs. When combined with intelligent target detection techniques, UAVs can take full advantage of their high manoeuvrability to locate targets of interest in a wide aerial field of view, which in turn enables flexible and efficient data collection capabilities. Driven by intelligent target detection technique, UAVs have shown promising applications in various fields such as traffic monitoring [1], power inspection [2], crop analysis [3] and military [4].

Like traditional software systems, the deep learning (DL) based intelligent target detection system also contains some defects and thus show incorrect or unexpected behaviors. Software testing, a core method to improve the quality of software systems, should also be applied to the contemporary target detection systems.

Among recently proposed testing techniques for DL systems, fuzzing is a basic technique that could generate lots of the test cases based on existing test data [5]. Despite the simplicity in its concept, fuzzing has successfully generated the adversarial inputs that facilitate the defects detection. Tian et al. proposed DeepTest, which focuses on generating realistic synthetic images by applying image transformations (such as changing brightness, changing contrast, translation, scaling, horizontal shearing, rotation, blurring) on seed images [6]. Odena et al. generated realistic images by adding the white noises [7]. Xie et al. generated new test cases by using the Paul Cezanne painting style, the Claude Monet painting style, and the Vincentvan Gogh painting style [8]. These existing studies showed that these testing methods are quite effective for critical error detection in image- based target detection system. Nonetheless, all of these testing methods cannot generate images that match the UAVs' characteristics.

In this paper, we propose a testing framework based on fuzzing technique, FTU, to generate test data from the UAV's perspective and automatically verify the test result based on metamorphic relation (MR) [9, 10], which is one of the core concepts of metamorphic testing (MT) [11]. Each MR not only defines the method to generate the new test case (called follow-up test case) from the existing test case but also the expected relation between the outputs of the two test cases. Once two outputs violate the corresponding MR, it implies the detection of some defects. Currently, MT has been recognized as one of the most popular techniques for DL models testing due to its high capability to alleviate the test oracle problem. It allows the unlabeled test cases being efficiently used to extensively test DL systems. FTU uses MRs to mimic the characteristics of images taken by UAVs in different environments (such as brightness changing, weather changing, shooting viewpoint switching and motion). The contributions of this work include:

- A testing framework for target detection system of UAVs is proposed, which can generate test cases and automatically verify the test results based on MR. Thanks to MR's mechanism of automatically determining test results, testers can execute tests without data labels, which effectively improves test efficiency and reduces test overheads.
- We define several metamorphic relations (MRs) which can mimic the characteristics of images taken by UAVs in different environments.
- Empirical studies on one dataset and two target detection systems of UAVs, were conducted to evaluate the effectiveness of FTU. As observed from these experiments, FTU could generate adversarial inputs that are effective in detecting various errors and thus help improve the robustness of UAV's target detection systems.

2. Background and Related Work

2.1 Deep Learning Model for Target Detection System of UAVs

Object detection algorithms based on convolutional neural networks (CNNs) directly extract features for classification and regression, greatly enhancing the efficiency of object detection and meeting the real-time requirements of UAVs. The YOLO_v5 model, known for its ease of training and deployment, is extensively applied in edge object detection devices. It maintains operational efficiency while improving detection and recognition accuracy. The YOLO_v5 model [12] consists of four parts: Input, Backbone, Neck, and Prediction. The overall model is illustrated in Figure 1. The Input module uses a Mosaic data augmentation technique, synthesizing new images through random cropping, scaling, and combination, thereby enriching the dataset and enhancing the algorithm's consistency. The Backbone module includes five Conv, CSPdarknet (C3), and SPPF structures for feature extraction [13]. The Neck module focuses on feature fusion, incorporating the Feature Pyramid Network (FPN) [14] and Path Aggregation Network (PANet). The Prediction module is responsible for predicting bounding boxes, computing the loss function, and implementing non-maximum suppression.
The YOLO_v7 model [15] retains the ease of training and deployment characteristics of YOLO_v5 but introduces ELAN and MP layers in the Backbone for feature extraction and implements model reparameterization during deployment, further speeding up detection while also improving accuracy.

2.2 Fuzzing

The core idea of fuzzing is to generate a large number of random test cases by changing existing test cases (called seeds) based on muatation strategies [5]. Generally, the process of fuzzing is composed of the following two components.

- *Seed selection*, which is responsible to select seeds from the test set based on a certain selection strategy.
- *Seed mutation*, which uses some seed mutation strategies to generate mutated seeds that serve as test cases for executing the software under test.

2.3 Metamorphic Testing

MT is a popular technique to alleviate the oracle problem [16]. Instead of applying an oracle, MT uses MRs (the necessary properties of software under test) to verify the test results [17, 18] across multiple test cases. MT is normally implemented according to the following steps:

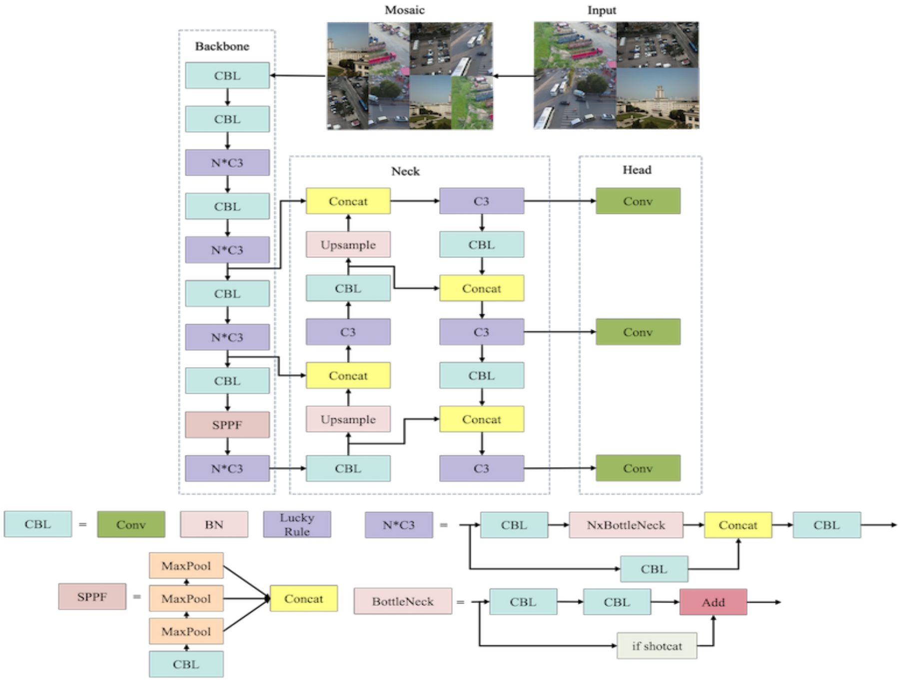


Figure 1 – Structural Chart of YOLO_v5 Network

Step I Identify an MR for software under test.

Step II Select/generate the source test case(s) using some test case selection/generation technique (such as APT [19], and DRT [20]).

Step III Construct the follow-up test case(s) from the selected source test case(s) based on the MR.

Step IV Execute source and follow-up test cases and obtaining their corresponding outputs.

Step V Verify the outputs of source and follow-up test cases against the used MR: If the MR does not hold, a fault is said to be detected.

In this paper, we use fuzzing to generate source test cases, and apply some MRs to guide the mutation of the source test cases as well as to verify the test results. Note that in this paper the source test case is denoted as the seed, and the follow-up test case is denoted as the test case.

2.4 Related Work

Recently, a large number of techniques for DL models to generated test cases have been proposed. Among them, fuzzing and MT have been proven to be very effective in generating adversarial inputs and detecting faults. Closely related works which could be used to generated test cases for the target detection system of UAV, are described below.
DeepTest focuses on generating synthetic images by applying image transformations (including changing brightness, changing contrast, translation, scaling, horizontal shearing, rotation, blurring, fog effect, and rain effect) on seed images [6]. However, the images generated by DeepTest may not appear from a drone perspective or appear in very rare circumstances. TensorFuzz implemented two different types of mutation for generating new test cases [7]. The first is to just add white noise of a userconfigurable variance to the seeds. The second is to add white noise, but to constrain the difference between the test cases and the seeds from which it is descended to have a user-configurable L_∞ norm. These above mutation strategies may generate a large number of invalid test cases (the semantics of the image cannot be discerned). Xie et al. generated new test cases that describes the content of the seeds using the Paul Cezanne painting style, the Claude Monet painting style, and the Vincentvan Gogh painting style [8]. The images generated by those mutated strategies are not specific from a drone perspective.

3. Methodology

In this section, we first demonstrate the framework for implementing the FTU and the concrete algorithms for testing the target detection system of UAV, then describe the identified MRs.

3.1 Framework

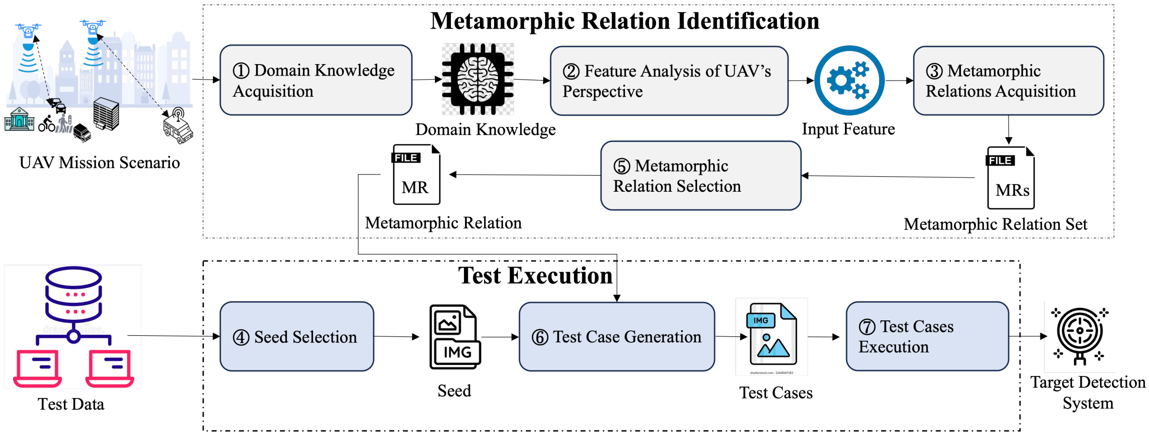


Figure 2 – The framework of FTU

Leveraging the principles and features of fuzzing and MT, we propose FTU framework, as illustrated in Figure 2. FTU assumes that there already exist test data and the mission scenario of the UAV under test is known. Note that test data is usually very easy to obtain, via the composition of data that are contained in the dataset for DL systems [21]. Interactions between FTU components are depicted in the framework. Details of the individual components in the framework are discussed as follows.
The top box of the FTU framework (Figure 2) is composed of the following four components.

- *Domain knowledge acquisition*. The tester analyses the application domain of the UAV to be tested (mainly considering the object to be detected and the characteristics presented by the target object).

- *Feature analysis of UAV's perspective*. The tester analyses the possible characteristics of the images captured during UAV operations (e.g. drastic changes in the weather of the operating environment or blurred motions, etc.)

- *Metamorphic relations acquisition*. MRs is the core component of MT, and thus have attracted lots of attentions [9]. The tester could use existing techniques to identify MRs such as search-based approach [22], data-mutation directed MR acquisition approach [23], METRIC+ [24].

- *Metamorphic relation selection*. MR is one of the key components that affects the fault detection efficiency of MT [9, 25, 26]. There are many MR selection approaches in academia. For example, Chen et al. [27] proposed that testers should choose good MRs which can make the multiple executions of the program as different as possible. Asrafi et al. [28] showed that the more different execution behaviors (measured by code coverage) the source and follow-up test cases had caused, the more effective MRs were. In this paper, we employ random approach to select MRs, which randomly select MRs from MRs set.

The bottom box of the FTU framework (Figure 2) is actually the process of fuzzing, which is composed of the following three components.

- *Seed selection*. Seed selection is a key component that affects the fault detection efficiency of fuzzing [5]. In this paper, we employ random approach to select seeds, which randomly select seeds from test data set.

- *Test case generation*. The test cases are constructed by transforming the selected seed based on the selected MR.
- *Seed execution*. This component receives the seeds and test cases, and executes them on the target detection system of UAV.

3.2 Algorithms for Testing the Target Detection System of UAV

Algorithm 1: The testing process of FTU

Input: Test data P, the seed selection strategy S_s, the MRs selection strategy S_m, the target detection system of UAV T, the threshold value of needed MRs n, the mission scenario of UAV MS, and a configurable total number of seed mutation K.

Output: The test report R.

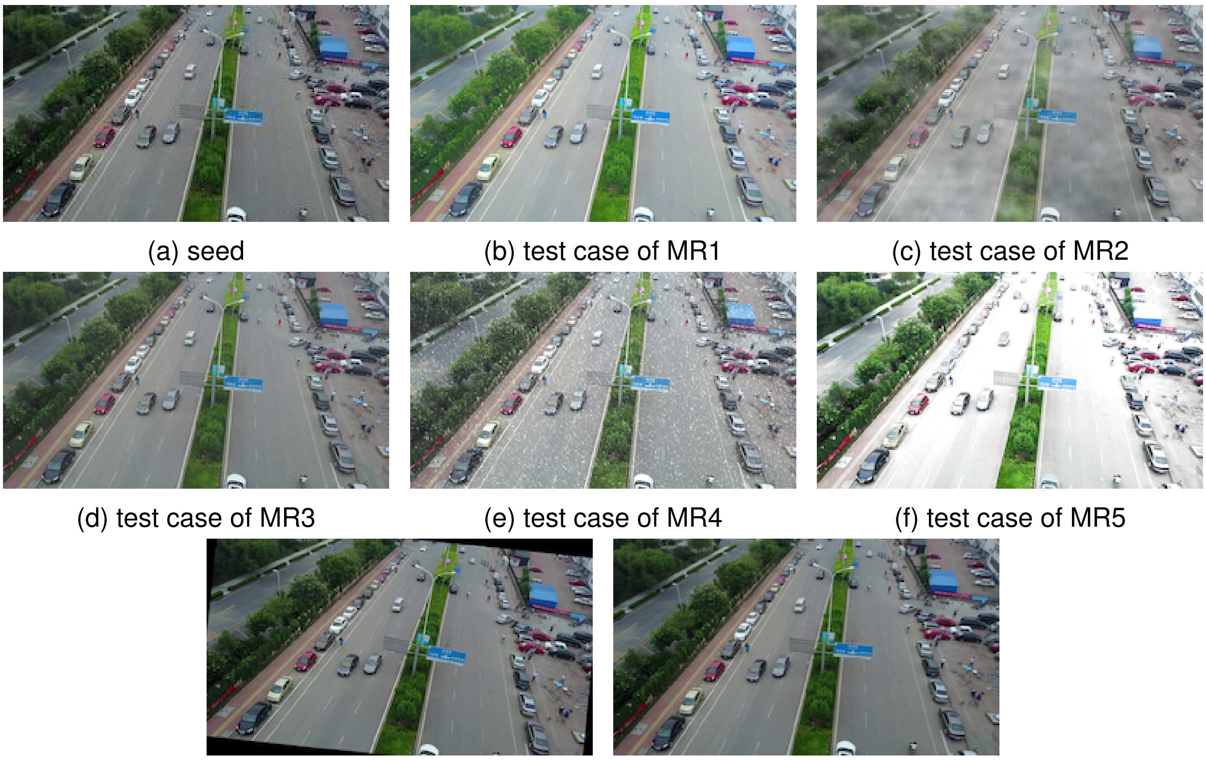
1 Initialize MR set $M = \emptyset$, and test cases set $S = \emptyset$.
2 **while** $n \neq size(M)$ **do**
3   Identify an MR $MR_i (1 \leq i \leq n)$ based on MS.
4   Put the identified MR MR_i into M: $M = M \cup \{MR_i\}$.
`/* Select a seed */`
5 Select a seed tc_i from P based on S_s.
6 **while** *termination condition is not satisfied* **do**
    `/* Select an MR */`
7   Select an MR MR_j from MS based on S_m.
    `/* Generate test cases */`
8   **while** $K < size(S)$ **do**
9   Generate a test case tc based on the selected tc_i and MR_j.
10   Put the generated test case tc into S: $S = S \cup \{tc\}$.
    `/* Execute test cases and verify the test result */`
11   **for** *each test case* tc *in* S **do**
12   Test the target detection system of UAV T using tc.
13   Verify the test result based on MR_j.
14   Make test cases set $S = \emptyset$.
15 Returen R.

The details of FTU is given in Algorithm 1. The algorithm takes a set of test data P, the seed selection strategy S_s, the MRs selection strategy S_m, the target detection system of UAV T, the threshold value of needed MRs n, the mission scenario of UAV MS, and a configurable total number of seed mutation K. The output is the test report R. The key idea behind the algorithm is to generate test cases using fuzzing and automatically verify test result using MR. In FTU, testers need to analyze MS, and identify MRs based on existing techniques until the number of MRs reaches the threshold n (Lines 2 to 4). Then a seed tc_i and an MR MR_j is selected based on S_s and S_m (refer to Lines 5 and 7). The test cases S are generated based on the tc_i and MR_j (Lines 9 and 10). After the execution of source and follow-up test cases, the test reults are verified based on the selected MR_j (Lines 12 and 13). Such a process is repeated until the termination condition is met (refer to Line 6). Many termination conditions can be used; for example, “testing resource has been exhausted”, “a certain number of test cases have been executed”, and “a certain number of faults have been detected”.

3.3 MRs for the Target Detection System of UAV

The test cases generated by existing testing techniques may not be representative of the actual scenarios experienced in the UAV’s field [29]. In this work, we explore the usefulness of our method in enhancing the realism and representativeness of test cases and improving the capabilities of detecting violations on the target detection system of UAV. Given an image (e.g., Figure 3(a)), the model recognizes the target of the major objects in the image (e.g., car, people, and motor). To obtain the realistic images, we do not use the simple affine transformations like shift and shear. Instead, we design seven typical MRs to transfer the seed images into representative test cases based on the typical working scenarios of UAV. The seven MRs are as follows.

- **Brightness MR (MR1)** which adds brightness noise to the seed image to simulate the scenarios of UAV working on a sunny day or at night. The test case generated by this MR is illustrated in Figure 3(b).
- **Cloud MR (MR2)** which adds Perlin noise to the seed image to simulate the scenarios where UAV is working in the cloud and fog. The test case generated by this MR is illustrated in Figure 3(c).
- **Rain MR (MR3)** which simulates the scenarios of UAV working in rainy weather by adding raindrop noise to the seed image. The test case generated by this MR is illustrated in Figure 3(d).
- **Snow MR (MR4)** which adds a snow effect based on the seed image to simulate the scenarios of UAV working in a blizzard. The test case generated by this MR is illustrated in Figure 3(e).
- **Snow floor MR (MR5)** which adds snow effect based on the seed image to simulate the working scenarios of UAV after a blizzard. The test case generated by this MR is illustrated in Figure 3(f).
- **Rotation MR (MR6)** which simulates the change in shooting angle caused by the change of UAV's posture by rotating the seed image. The test case generated by this MR is illustrated in Figure 3(g).
- **Motion MR (MR7)** which simulates the scenarios of taking photos while UAV is maneuvering at high speed. The test case generated by this MR is illustrated in Figure 3(h).



(a) seed
(b) test case of MR1
(c) test case of MR2
(d) test case of MR3
(e) test case of MR4
(f) test case of MR5
(g) test case of MR6
(h) test case of MR7

Figure 3 – Illustration of the test cases for seven MRs

4. Evaluation

A series of empirical studies were conducted to evaluate the performance of FTU.

4.1 Research Questions

In our experiments, we focused on addressing the following two research questions.

RQ1 Can FTU generate adversarial inputs (which indicates the detection of faults)?

RQ2 Can FTU be used to retrain the corresponding deep learing models to improve the target detection system of UAV?

4.2 Experimental Design

We implemented FTU as a self-contained fuzzer, written in Python (ver.3.8) based on the deep learning framework Pytorch (ver.1.10.1).
In our experiments, we adopted one popularly used dataset: VisDrone [30]. We further employed the open-source DL models, including YOLO_v5 and YOLO_v7. Table 1 shows an informative summary of used models and dataset. The VisDrone dataset was collected by the AISKYEYE team at the Machine Learning and Data Mining Laboratory of Tianjin University, China, and is suitable for target detection tasks from the viewpoint of a drone. The dataset consists of 10,209 images with a resolution of 2,000 pixels × 1,500 pixels. The dataset is specifically annotated with target occlusion, truncated proportions, etc., in addition to target bounding box and category annotations. The data is labeled with fine-grained categorization for the targets, with labels covering 10 predefined categories and containing bounding boxes for 540k target objects. These categories are: pedestrian, people, bicycle, car, van, truck, tricycle, awning-tricycle, bus, and motor. The dataset uses 6471 images as a training subset, 548 images as a validation subset, and 1610 images as a test subset. The batch size is set to 8; a total of 200 epochs are trained without using the pre-trained weights. The gradient descent optimization strategy uses the Adam optimizer, and the initial learning rate is set to 0.01. In this experiment, the average precision are used as metrics for the experimental results. Training and testing were performed on a computer equipped with Win11 operating system, NVIDIA GeForce RTX 3080 Ti GPU, 12th Gen Intel(R) Core(TM) i7 processor, CUDA11.1.
To reduce the randomness effect of experiments, we randomly executed 30 times test set which is constructed by randomly selected 100 images from the test subset of VisDrone. The termination condition of testing models on VisDrone is the generation of 1400 test cases (each MR generated 2 images).

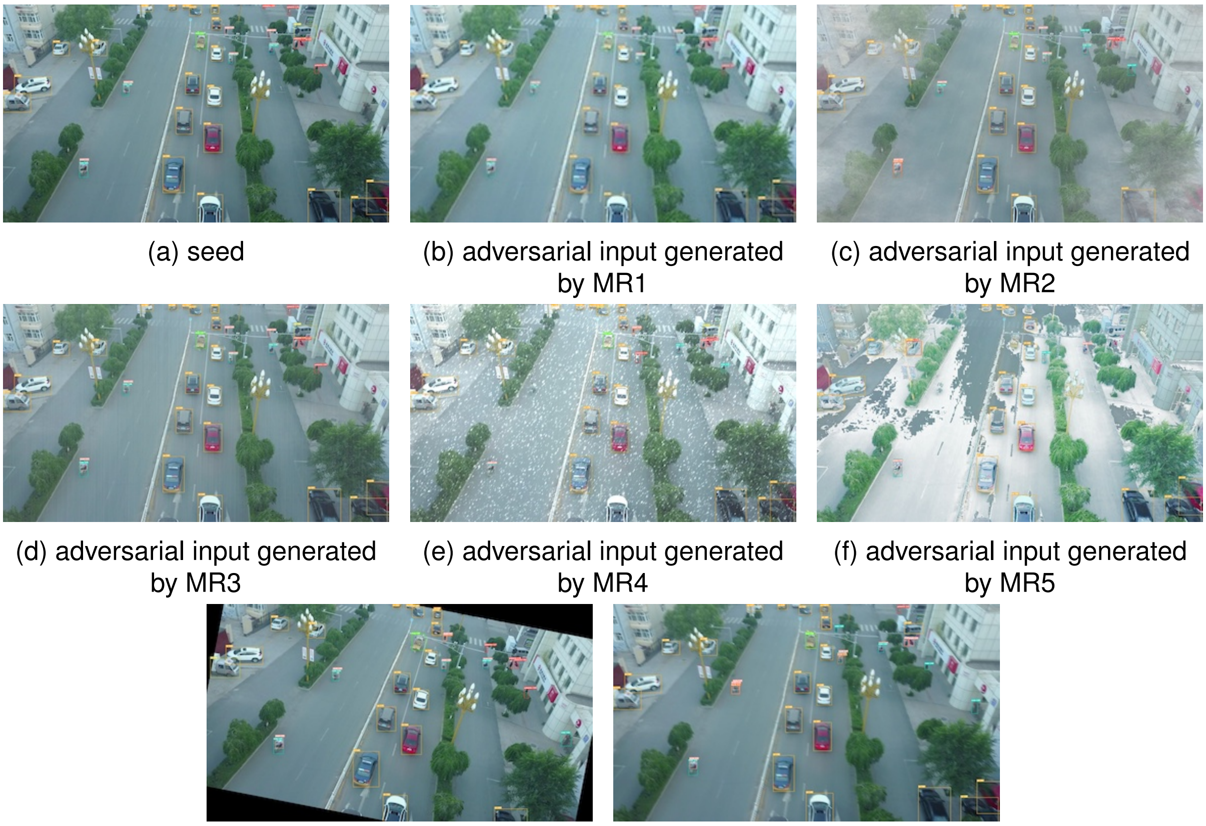
Table 1 – Subject datasets and DL models

Dataset	Dataset Description	Model	Parameters	Test acc(%)	New acc(%)
VisDrone	10 classes	YOLO_v5	20,907,687	25.70	27.8
	large scale	YOLO_v7	37,245,102	29.40	31.3

4.3 Results

4.3.1 RQ1: Generation of Adversarial Inputs

Table 2 reports the average number of adversarial inputs generated by FTU. It is clearly shown that FTU can generate effective adversarial inputs. We also can observe that different MRs have different abilities to generate adversarial inputs: MR1 generates the least adversarial inputs; MR4 generates the most adversarial inputs. This observation shows that the target recognition system of the UAV has strong anti-interference ability against brightness changes and low anti-interference ability against weather changes (especially snowy weather). Figure 4 shows an example of adversarial inputs generated by FTU on the YOLO_v5.



(a) seed
(b) adversarial input generated by MR1
(c) adversarial input generated by MR2
(d) adversarial input generated by MR3
(e) adversarial input generated by MR4
(f) adversarial input generated by MR5
(g) adversarial input generated by MR6
(h) adversarial input generated by MR7

Figure 4 – Illustration of the generated adversarial inputs for seven MRs

Table 2 – Average number of adversarial inputs generated by FTU

Models	Adversarial Inputs Generated by FTU	Adversarial Inputs Generated by Each MR						
		MR1	MR2	MR3	MR4	MR5	MR6	MR7
YOLO_v5	899	73	150	121	162	125	122	146
YOLO_v7	1022	119	130	175	175	100	166	157

4.3.2 RQ2: Retraining the models

In order to verify whether the generated adversarial inputs can help the UAV's target detection systems improve the robustness, we use the generated adversarial inputs to retrain the models, and then use the test subset to obtain the accuracy of used each model.

The last column of Table 1 shows the accuracies of the retrained models: The accuracy of YOLO_v5 increased by 2.10%; The accuracy of YOLO_v7 increased by 1.90%.

5. Contact Author Email Address

Zerui Wang: wangzeruixx@163.com

6. Copyright Statement

References

[1] Igor Bisio, Chiara Garibotto, Halar Haleem, Fabio Lavagetto, and Andrea Sciarrone. A systematic review of drone based road traffic monitoring system. *IEEE Access*, 10:101537–101555, 2022.

[2] Zhaoyang Wang, Qiang Gao, Jianbin Xu, and Dahua Li. A review of uav power line inspection. In *Proceedings of 2020 International Conference on Guidance, Navigation and Control (ICGNC 2020)*, pages 3147–3159. Springer, 2022.

[3] Jose Cuaran and Jose Leon. Crop monitoring using unmanned aerial vehicles: A review. *Agricultural Reviews*, 42(2):121–132, 2021.

[4] Syed Agha Hassnain Mohsan, Nawaf Qasem Hamood Othman, Yanlong Li, Mohammed H Alsharif, and Muhammad Asghar Khan. Unmanned aerial vehicles (uavs): Practical aspects, applications, open challenges, security issues, and future trends. *Intelligent Service Robotics*, 16(1):109–137, 2023.

[5] Hepeng Dai, Chang-ai Sun, and Huai Liu. Deepcontroller: Feedback-directed fuzzing for deep learning systems. In *Proceedings of the 34 International Conference on Software Engineering and Knowledge Engineering (SEKE 2022)*, pages 531–536, 2022.

[6] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th International Conference on Software Engineering (ICSE 2018)*, pages 303–314, 2018.

[7] Augustus Odena, Catherine Olsson, David Andersen, and Ian Goodfellow. Tensorfuzz: Debugging neural networks with coverage-guided fuzzing. In *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, pages 4901–4911, 2019.

[8] Xiaoyuan Xie, Pengbo Yin, and Songqiang Chen. Boosting the revealing of detected violations in deep learning testing: A diversity-guided method. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering (ASE 2022)*, pages 1–13, 2022.

[9] Chang-Ai Sun, Hepeng Dai, Huai Liu, and Tsong Yueh Chen. Feedback-directed metamorphic testing. *ACM Transactions on Software Engineering and Methodology*, 32(1):1–34, 2023.

[10] Chang-Ai Sun, Hepeng Dai, Ning Geng, Huai Liu, Tsong Yueh Chen, Peng Wu, Yan Cai, and Jinqiu Wang. An interleaving guided metamorphic testing approach for concurrent programs. *ACM Transactions on Software Engineering and Methodology*, 33(1):1–21, 2023.

[11] Tsong Yueh Chen, Shing C Cheung, and Shiu Ming Yiu. Metamorphic testing: A new approach for generating next test cases. Technical report, Department of Computer Science , Hong Kong University of Science and Technology, Hong Kong, Tech. Rep. HKUST-CS98-01, 1998.

[12] Wentong Wu, Han Liu, Lingling Li, Yilin Long, Xiaodong Wang, Zhuohua Wang, Jinglun Li, and Yi Chang. Application of local fully convolutional neural network combined with yolo v5 algorithm in small target detection of remote sensing image. *PloS one*, 16(10):e0259283, 2021.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, 2015.

[14] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, pages 2117–2125, 2017.

[15] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the 36th IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR 2023)*, pages 7464–7475, 2023.

[16] Yang Xiao, Ning Zhang, Wenjing Lou, and Y Thomas Hou. A decentralized truth discovery approach to the blockchain oracle problem. In *Proceedings of IEEE Conference on Computer Communications (INFOCOM 2023)*, pages 1–10, 2023.

[17] Zenghui Zhou, Zheng Zheng, Tsong Yueh Chen, Jinyi Zhou, and Kun Qiu. Follow-up test cases are better than source test cases in metamorphic testing: A preliminary study. In *Proceedings of the 6th International Workshop on Metamorphic Testing (MET 2021)*, pages 69–74, 2021.

[18] Chengying Mao, Jifu Chen, Xiaorong Yi, and Linlin Wen. An empirical study on metamorphic testing for recommender systems. *Information and Software Technology*, 169:107410, 2024.

[19] Chang-ai Sun, Hepeng Dai, Huai Liu, Tsong Yueh Chen, and Kai-Yuan Cai. Adaptive partition testing. *IEEE Transactions on Computers*, 68(2):157–169, 2018.

[20] Chang-ai Sun, Hepeng Dai, Guan Wang, Dave Towey, Tsong Yueh Chen, and Kai-Yuan Cai. Dynamic random testing of web services: a methodology and evaluation. *IEEE Transactions on Services Computing*, 15(2):736–751, 2019.

[21] Hepeng Dai, Chang-Ai Sun, Huai Liu, and Xiangyu Zhang. Dfuzzer: Diversity-driven seed

queue construction of fuzzing for deep learning models. *IEEE Transactions on Reliability*, 2023, DOI:10.1109/TR.2023.3322406.

[22] Jie Zhang, Junjie Chen, Dan Hao, Yingfei Xiong, Bing Xie, Lu Zhang, and Hong Mei. Search-based inference of polynomial metamorphic relations. In *Proceedings of the 29th ACM/IEEE International Conference on Automated Software Engineering (ASE 2014)*, pages 701–712, 2014.

[23] Chang-ai Sun, Yiqiang Liu, Zuoyi Wang, and W. K. Chan. μMT: A data mutation directed metamorphic relation acquisition methodology. In *Proceedings of 2016 IEEE/ACM the 1st International Workshop on Metamorphic Testing (MET'16), Co-located with the 38th International Conference on Software Engineering (ICSE 2016)*, pages 12–18, 2016.

[24] Chang-ai Sun, An Fu, Pak-Lok Poon, Xiaoyuan Xie, Huai Liu, and Tsong Yueh Chen. METRIC+: A metamorphic relation identification technique based on input plus output domains. *IEEE Transactions on Software Engineering*, 47(9):1765–1786, 2021.

[25] Tsong Yueh Chen, Fei-Ching Kuo, Ying Liu, and Antony Tang. Metamorphic testing and testing with special values. In *Proceedings of the 5th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2004)*, pages 128–134, 2004.

[26] René Just and Franz Schweiggert. Automating software tests with partial oracles in integrated environments. In *Proceedings of the 5th Workshop on Automation of Software Test (AST 2010)*, pages 91–94, 2010.

[27] Tsong Yueh Chen, DH Huang, T.H. Tse, and Zhi Quan Zhou. Case studies on the selection of useful relations in metamorphic testing. In *Proceedings of the 4th IberoAmerican Symposium on Software Engineering and Knowledge Engineer-ing (JIISIC 2004)*, pages 569–583, 2004.

[28] Mahmuda Asrafi, Huai Liu, and Fei-Ching Kuo. On testing effectiveness of metamorphic relations: A case study. In *Proceedings of the 15th International Conference on Secure Software Integration and Reliability Improvement (SSIRI 2011)*, pages 147–156, 2011.

[29] Sajad Khatiri, Sebastiano Panichella, and Paolo Tonella. Simulation-based test case generation for unmanned aerial vehicles in the neighborhood of real flights. In *Proceedings of the 16th IEEE Conference on Software Testing, Verification and Validation (ICST 2023)*, pages 281–292, 2023.

[30] Yaru Cao, Zhijian He, Lujia Wang, Wenguan Wang, Yixuan Yuan, Dingwen Zhang, Jinglin Zhang, Pengfei Zhu, Luc Van Gool, Junwei Han, et al. Visdrone-det2021: The vision meets drone object detection challenge results. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV 2021)*, pages 2847–2854, 2021.