



Identifying threatening drone based on deep learning with SWIR camera

Kyung-Soo Park¹ · Jung-Kn Kim² · Jae-Hoon Kim³ · Seong-Dae Lee[†]

(Received December 8, 2022 ; Revised December 19, 2022 ; Accepted December 19, 2022)

Abstract: Small drones of various sizes are used in numerous fields, including commerce, reconnaissance, and offensive attacks. Major facilities such as security areas of port, power, and offshore plants urgently need to develop solutions for detecting drones as an active countermeasure against small drone attacks because small drones used for military and terrorism pose a significant threat. It is not easy to detect various drones such as invasive or threatening ones, though recent developments have made it possible to detect them using three-dimensional radar. Therefore, this paper develops threatening drone identification system, which consists of two components: One is a software component for identifying threatening drones among various ones and the other is a hardware component for the system. The former uses well-known YOLO(You Look Only Once) (v7) model and the latter comprises a PC for running the model and an SWIR (Short-Wave InfraRed) camera for surveillance. Datasets for training and evaluation are constructed by hand from airborne videos taken drones including threatening one and is labelled by two types: normal and threatening. The datasets are comprised of 3,992 color images and 4,410 thermal images, which are trained separately. Through experiments, we have shown that mAP@.5 and mAP@.95 are 0.999 and 0.753 (0.999 and 0.760) for color images (for thermal images), respectively. Consequently the proposed system is helpful in identifying threatening drones.

Keywords: Drone, Deep Learning, Drone Identification, YOLO, SWIR camera

1. Introduction

Drones are radio wave-guided small, unmanned aerial vehicles in fixed or multi-copter forms and have recently been utilized for a range of applications including leisure, video shooting, delivery, and so on although they developed for military purposes at first [1]. In particular, small military drones used for surveillance, penetration, and terror attacks are making global issue [2], but active countermeasures against reconnaissance or attacks through small drones are very lack. Therefore, it is urgent to develop technologies related to dangerous or threatening drone identification, which follows drone detection. Most existing anti-drone systems and many researches, however, only detect drones using many sensors with radar and LiDAR (Light Detection And Ranging) [3]. The anti-drone systems have shortcomings in that they are complex and expensive because of their numerous sensors.

In order to alleviate the problem, in this paper, we develop

threatening drone identification system, which consists of two components: One is a software component for identifying threatening drones among various ones and the other is a hardware component for the system. The former uses well-known YOLO (v7) model and the latter comprises a SWIR (Short-Wave Infra-Red) camera for surveillance and a PC for running the model. The YOLO model is based on CNN (Convolutional Neural Network) [4] and can detect objects in real time with localization and classification at 1-stage [5]. The camera takes color and thermal images on a fine day (or daytime) and a bad weather (low light), respectively.

Datasets for identifying threatening drones are not publicly available and so we construct them by hand by taking airborne videos and labeling two types with normal and threatening. The datasets are comprised of 3,992 color images and 4,410 thermal images, which are trained separately. Through experiments, we have shown that mAP@.5 and mAP@.95 are 0.999 and 0.753

[†] Corresponding Author (ORCID: <http://orcid.org/0000-0002-8133-535X>): Research Professor, Department of Artificial Intelligence Engineering, Korea Maritime & Ocean University, 727, Taejong-ro, Yeongdo-gu, Busan 49112, Korea, E-mail: omega@kmou.ac.kr, Tel: 051-410-5294

1 M. S. Candidate, Department of Computer Engineering, Korea Maritime & Ocean University, E-mail: monotic1301@naver.com, Tel: 051-410-4896

2 President, Intelligent System Technology, E-mail: jungkn.ist@gmail.com, Tel: 070-7347-6678

3 Professor, Department of Artificial Intelligence Engineering and Interdisciplinary Major of Maritime AI Convergence, Korea Maritime & Ocean University, E-mail: jhoon@kmou.ac.kr, Tel: 051-410-4574

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

(0.999 and 0.760) for color images (for thermal images), respectively. In terms of time performance, the identification time for a frame is 11.9 ms and the proposed system can identify about 84 frames per second. Consequently, it seems to be real-time and is helpful in identifying threatening drones.

This paper is organized as follows: In Chapter 2, we describe related works on YOLO, drone detection and identification; In Chapter 3 and Chapter 4, we illustrate the proposed system in detail and describe process of datasets construction and experiments, respectively; finally in Chapter 5, we draw conclusion against our observations and discuss future studies.

2. Related work

2.1 YOLO (You Only Look Once)

Object detection is to identify and locate objects in images or videos and is mainly based on CNN [4]. For object detection, a 2-stage detector is widely used for object classification and location, while a 1-stage detector carries out the two stages simultaneously. The 1-stage detector provides much faster detection speed in spite of lower accuracy than the 2-stage detector.

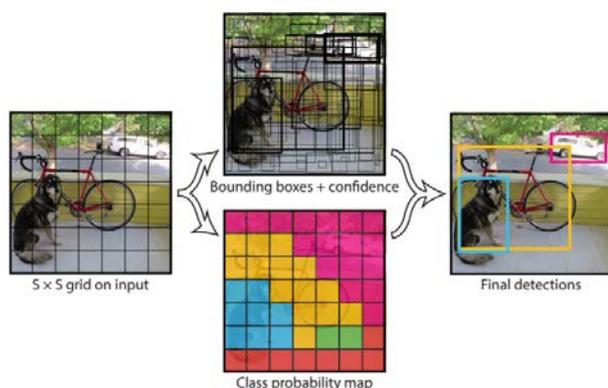


Figure 1: Example of a YOLO algorithm [5]

Figure 1 shows the process of the YOLO model, which is a 1-stage detector capable of real-time object detection [5]. YOLO divides input images into $S \times S$ size and then predicts anchor boxes based on confidence scores in a predefined form for given input images while passing through several levels. Next, YOLO chooses locations with high object reliability and predicts labels of objects.

YOLO has been continuously updated across various versions. YOLO v1 [5] had shown relatively-lower accuracy and many localization errors because boxes were identified based on a per cell basis, not using an anchor box. It was able to identify large

objects well, but not small objects. Regardless of this shortcoming, detection speed was fast. YOLO v2 [6] maintained the fast speed of the previous version, using the Darknet-19 model as a backbone and improved its performance using batch normalization, high resolution classifier, and anchor box. YOLO v3 [7] enabled multi-label classification for each class. Its speed was slow relatively although its accuracy is increased using Darknet-53 compared to Darknet-19. In YOLO v4 [8], CSPDarknet53 was used as the backbone, and BoF (Bag of Freebies) was used to obtain higher accuracy without increasing the inference time. To improve performance, BoS (bag of specials), SAM (Self-Attention Module), and PAN (Path Augmented) were also used. YOLO v7 [9] is the most advanced model of the YOLO series and proposed the trainable BoF method that can improve accuracy without increasing inference cost during real-time object detection.

2.2 Drone detection and identification

The work [10] on drone detection using sound signals complemented problems of existing studies on drone sound detection. It collected noises from the surrounding environment separately, and then detected drones even in a noisy environment by separating drone signals from environmental noise. Using the Mel Spectrogram feature extraction and CNN deep learning model, drones were detected with 98.55% accuracy. The work [11] took, divided, and preprocessed drone images and trained the deep learning model. The model detected drones and then located them through image-processing. And the other work [12] observed the RF communication band, while detecting communication protocols using the characteristics of using RF signals when controlling drones wirelessly. The work [13] that detected drones using only a camera and deep learning used the short-wave length infrared (SWIR) camera and YOLOv4 model as a deep learning model. The study obtained 2, 921 images of datasets using only thermal images, showing great performance for detecting drones with 98.17% precision and 98.65% recall. However, this works didn't include identification of various types of drones, focusing solely on simple drone detection.

The work [14] involved research on the development of EO/IR(electro-optical/Infrared)-connected radar to identify illegal drones of unmanned aerial vehicles. By reflecting the characteristic of gyro sensors of Micro Electro Mechanical Systems that are sensitive to specific frequency, the work [15] developed a system capable of identifying and sending airwaves to disturb sensors that control drones. For carrying out a drone ID project

monitoring receives many types of signals like normal drone, threatening drone, some other objects and displays them on the administrator screen. During monitoring, an alarm should alert the administrators to track the threatening drone continuously, which is displayed in a different color like red if a threatening drone would be identified. The drone monitoring tries to locate the drone in the next frame based on its location at the current frame and seeks around the neighborhood of the current drone's position. This function helps detect a drone in a certain region instead of the entire frame [17].

4. Experiments and performance evaluation

In this chapter, we describe how to construct datasets, experimental environments, and performance evaluation of threatening drones in the next subsection in subsequence.

4.1 Dataset

We assume that a threatening drone is attached to a dangerous object because actual threatening drones are very rare and it is not easy to get images of the threatening drones. For training the YOLO (v7) model, we take images of flying drones and then label them. For dataset, both color and thermal images were used as shown in **Table 1**.

Table 1: Statistics of image collection

	Color image	Thermal image
No. of frames	34,076	26,517
image size	1920 * 1080	1920 * 1080
Shooting time	18 min	15 min
No. of labelled images	3,992	4,410

Color images are suitable in the fine day (or daytime) with sufficient light, but it is difficult to identify drones with color images on cloudy days (or days with low light). So, we also use thermal images to complement this problem. Thermal images provide better quality than color images in case of cloudy days (or days with less light) because they use infrared light. We use three drones for every shooting and label with normal and threatening. We consider one drone with green box assumed to be dangerous object as threatening drone and the other two drones as normal drone. We label the drone with attached the green box as 'threatening', and the other two drones as 'normal'. We obtained 34,076 color images of 1920*1080 size for 18 minutes and 26,517 thermal images of same size for 15 minutes. Because frames shot too

close show little difference in image, we utilize images at intervals of five frames as data. We don't use images that were blurred or that were difficult to distinguish with the naked eye due to the cutting of drones.



Figure 3: Examples for images

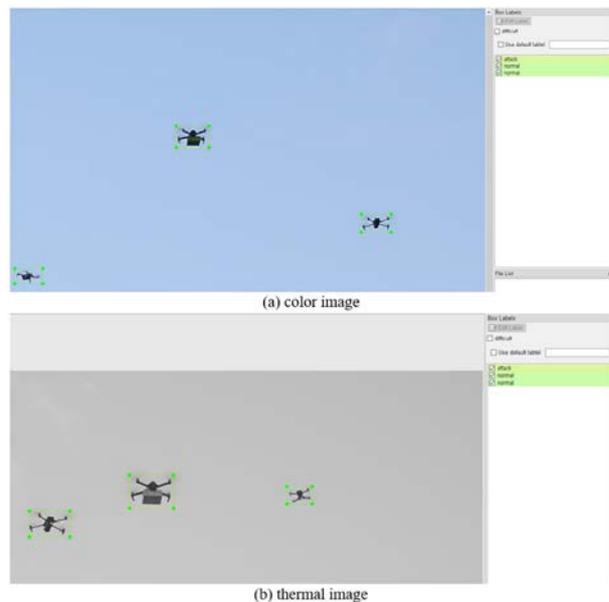


Figure 4: Labeling images

```
0 0.574609 0.445833 0.0828125 0.0986111
1 0.331836 0.582639 0.0730469 0.0833333
1 0.683984 0.502083 0.0375 0.0513889
```

Figure 5: A part of the data label file

Figure 3 is an example of color image (a) and thermal image (b) of drones. We let the drone rotate in place or move little by little around.

Data labeling was conducted in a bounding box form to fit the drone size as much as possible by hand. As shown in **Figure 4**, we label all images with the LabelImg [18] tool. We separately constructed the dataset of color images and thermal images.

Label information for each image is stored as text files as shown in **Figure 5**. Each line consists of LABEL_ID, X_CENTER, Y_CENTER, WIDTH, and HEIGHT. LABEL_ID, values assigned numbers to labels, is numbered from 0 in the order of initially stored labels. We used with ‘threatening’ label numbered as 0 and ‘normal’ as 1. X_CENTER, the X-coordinate of the center of labeled objects, is relative values divided by image width. Y_CENTER, the Y-coordinate of the center of labeled objects, is relative values with coordinate values of objects divided by image height. WIDTH, bounding box width values of objects, is relative values with bounding box width values divided by image width. HEIGHT, bounding box height values, is relative values with bounding box width values divided by image height.

4.2 Experiment environments

Training was conducted on the same model with thermal and color datasets, respectively. **Table 2** and **3** show hyper-parameters for training and software versions on the software component respectively.

Table 2: Hyper-parameters

Hyper-parameters	
Epochs	300
Batch size	16
Input image size	416 * 416

Table 3: Software environments

Software	Version no.
Ubuntu	18.04.6
Matplotlib	3.2.2
Numpy	1.18.5
Opencv-python	4.1.1
Pillow	7.1.2
PyYAML	5.3.1
Requests	2.23.0
Scipy	1.4.1
Torch	1.7.0
Torchvision	0.8.1
Tqdm	4.41.0
Protobuf	4.21.3

As shown in **Table 2**, we trained model for 300 epochs and set batch size as 16. Images were rescaled to 416*416 in size before their input. For the sake of convenience, we used the same hyper-parameters for thermal and color images. The software environments used for training, as shown in **Table 3**, represents the version of each package. We use the same software environment for training color and thermal images.

Table 4 shows hardware environments of CPU and GPU. We also use the same hardware environments for training and inference. In the future, we would like to recommend different environments for training and inference because training usually takes a lot of time while inference do not take such much time in environments like embedding systems.

Table 4: Hardware environments

Hardware environment	
CPU	Intel® Xeon® CPU E5-1660 v3 @3.00GHz
GPU	TITAN RTX

We evaluate the trained model with precision, recall, $mAP@p$, where $p \in (0,1)$ is the IoU (Intersection over Union) as threshold. The precision is the ratio of drones that are correctly detected among true drones and the recall how accurately the model predicts images including drones. Mean Average Precision (mAP) [5] is a metric used to evaluate object detection models such as YOLO, etc. and is calculated by finding Average Precision (AP) for each class and then average over a number of classes. The AP is calculated as the weighted mean of precisions at each threshold; the weight is the increase in recall from the prior threshold p . Therefore, $mAP@.5$ and $mAP@.95$ mean the mAP for the IoU of 0.5 or less and between 0.5 and 0.95, respectively.

4.3 Performance Evaluation

Figure 6 shows curves of loss values and performance during training color and thermal images for each 300 epochs, respectively. For training and validating, losses tended to decrease, and precision, recall, and mAP tended to increase. This seems to be well trained.

Figure 7 is examples of the model’s identification results for color and thermal images, respectively. For both images, we can detect and identify all drones for each label. In particular, dangerous drones were identified with confidence scores of about 0.90 or more.

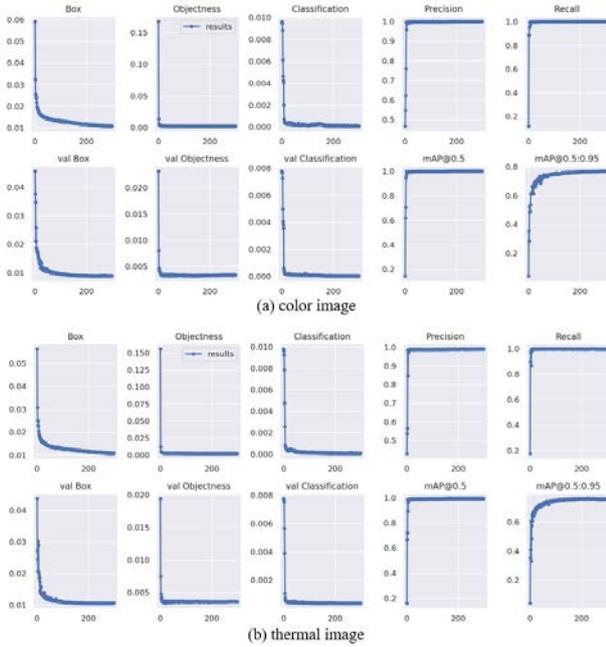


Figure 6: Curves of losses and measures for images

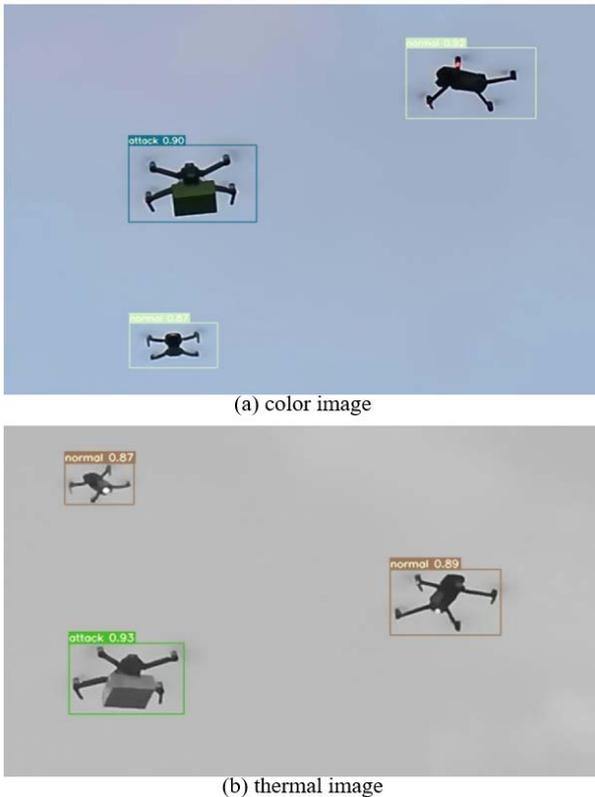


Figure 7: Examples of drone identification for image

Table 5 and 6 are mAP performance for color and thermal image test sets, respectively. All is the average of threatening and normal. We obtain that mAP@.5 and mAP@.95 are 0.999 and 0.753 (0.999 and 0.760) for color images (for thermal images).

Table 3: Performance evaluation for color test images

	mAP@.5	mAP@.95
all	0.999	0.753
threatening	0.999	0.809
normal	0.998	0.697

Table 4: Performance evaluation for thermal test images

	mAP@.5	mAP@.95
all	0.999	0.76
threatening	0.998	0.792
normal	0.999	0.729

Time taken to detect and identify drones per frame is 11.9 ms. We can identify drone at about 84 frames per second, making real-time surveillance possible.

A false alarm is a false positive which incorrectly indicates that malicious activity is occurring (also called a type I error) and is very important in threatening drone detection. In the case of the IOU of 0.5, mAP is almost 1.0 and then the false alarm is almost zero. Consequently the proposed model can achieve better accuracy with lower false alarm rate. As shown in Table 5 and 6, a loose IoU threshold results in a higher AP score than a strict IoU threshold as usual in most cases.

5. Conclusion

This paper presents a system for identifying drones using SWIR camera and YOLO (v7) model. Drones can be identified using color images in a fine day (or daytime with much light), but identifying them with color images in bad weather (or at night with little light) is challenging. Therefore, the thermal images were used. We collected and labeled drone for color images and thermal images. The datasets were separately learned for the same model.

For color and thermal image sets, we observed that mAP@.5 and mAP@.95 are 0.999 and 0.753 (0.999 and 0.760) for color images (for thermal images), respectively. A detection speed of approximately 84 fps enables real-time drone identification. Consequently, it seems to be real-time and is helpful in identifying threatening drones. However, as shown in Figure 6, loss tended to overfit rapidly. This limitation was caused by the inability to create a large amount of data, as data collecting and labeling take a long time.

In future works, experiments are expected to be conducted various environments to secure more datasets for better drone identification performance and to overcome the limitations. mAP can

contain several errors such as duplicate detection, misclassification, mislocalization, *etc.* and then the detailed analysis for errors should be required.

Acknowledgement

Following are results of a study on the "Leaders in INdustry-university Cooperation 3.0" Project, supported by the Ministry of Education and National Research Foundation of Korea.

Author Contributions

Conceptualization, J. -K. Kim and S. -D. Lee; Methodology, J. -H. Kim; Software, K. -S. Park; Formal Analysis, J. -H. Kim and K. -S. Park; Investigation, J. -K. Kim and K. -S. Park; Resources, J. -K. Kim and K. -S. Park; Data Curation J. -K. Kim and K. -S. Park; Writing-Original Draft Preparation, S. -D. Lee and K. -S. Park; Writing-Review & Editing, J. -H. Kim; Visualization, K. -S. Park; Supervision, S. -D. Lee; Project Administration, S. -D. Lee; Funding Acquisition, S. -D. Lee.

References

- [1] H. -G. Kim, "Analysis on patent trends for industry of unmanned aerial vehicle," Proceedings of the 2017 Fall Conference of the Korean Entertainment Industry Association, pp. 90-93, 2017 (in Korean).
- [2] S. H. Choi, J. S. Chae, J. H. Cha, and J. Y. Ahn, "Recent R&D trends of anti-drone technologies," Electronics and Telecommunications Trends, vol. 33, no. 3, pp. 78-88, 2018 (in Korean).
- [3] D. -H. Lee, "Convolutional neural network-based real-time drone detection algorithm," Journal of Korea Robotics Society, vol. 12, no. 4, pp. 425-431, 2017 (in Korean).
- [4] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," Neural Computation, vol. 1, no. 4, pp. 541-551, 1989.
- [5] J. Redmon, S. Divvala, R. Girchick, and A. Farhadi, "You only look once: Unified, real-time object detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779-788, 2016.
- [6] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6517-6525, 2017.
- [7] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," arXiv:1804.02767, 2018.
- [8] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optical speed and accuracy of object detection," arXiv:2004.10934, 2020.
- [9] C. -Y. Wang, A. Bochkovskiy, and H. -Y. H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-Time object detectors," arXiv:2207.02696, 2022.
- [10] H. Y. Kang and K. -H. Lee. "A general acoustic drone detection using noise reduction preprocessing" Journal of the Korea Institute of Information Security & Cryptology, vol. 32, no. 5, pp. 881-890, 2022 (in Korean).
- [11] K. Yi and K. Seo, "Image based identification and localization of drones," The Transactions of the Korean Institute of Electrical Engineers, vol. 69, no. 2, pp. 331-336, 2020 (in Korean).
- [12] P. Nguyen, H. Truong, M. Ravindranathan, A. Nguyen, R. Han, and T. Vu, "Cost-effective and passive RF-based drone presence detection and characterization," GetMobile: Mobile Computing and Communications, vol. 21, no. 4, 2017, pp. 30-34, 2017.
- [13] H. -M. Park, G. -S. Park, Y. -R. Kim, J. -K. Kim, J. -H. Kim, and S. -D. Lee, "Deep learning-based drone detection with SWIR cameras," Journal of Advanced Marine Engineering and Technology, vol. 44, no. 6, pp. 500-506, 2020.
- [14] I. Alnujaim, D. Oh, and Y. Kim, "Generative adversarial networks for classification of micro-Doppler signatures of human activity," IEEE Geoscience and Remote Sensing Letters, vol. 17, no. 3, pp. 396-400, 2020.
- [15] Y. Son, H. Shin, D. Kim, Y. Park, J. Noh, K. Choi, J. Cho, and Y. Kim, "Rocking drones with intentional sound noise on gyroscopic sensors," Proceedings of the 24th USENIX security symposium, pp. 881-896, 2015.
- [16] K. Jensen, K. Shriver, and U. P. Schultz, Drone Identification and Tracking in Denmark", Technical Report TR-2016-2, University of Southern Denmark, 2016.
- [17] Y. Chen, P. Aggarwal, J. Choi, and C. -C. Jay Kuo, "A deep learning approach to drone monitoring," Proceedings of APSIPA Annual Summit and Conference, pp. 686-591, 2017.
- [18] GitHub – heartexlabs/labelImg: Video/Image Labeling and Annotation Tool, <https://github.com/heartexlabs/labelImg>.