

Altitude-Based Automatic Tiling Algorithm for Small Object Detection

Mun Kyu Choi*, Soo Young Shin^o

ABSTRACT

In this paper, a method is proposed to perform consistent detection of small objects across all altitudes using UAVs. Unlike traditional tiling algorithms that detect small objects with a fixed number of tiles, our method dynamically adjusts the number of tiles based on the altitude and the size of the bounding box. This approach determines the proper number of tiles for detecting small objects at each altitude. Real-time object detection was conducted on the AI board mounted on the UAV, and for this purpose, the model was optimized using a lightweighting process. Using the Visdrone dataset, the performance of small object detection at each altitude was tested in real-world conditions. Compared to the conventional YOLOv5 and fixed-number tiling algorithms, our method consistently detected small objects at all altitudes.

Key Words : Deep Learning, Object Detection, Tiling, Small Object, UAV

I. Introduction

Computer vision is a technology that analyzes pixel information of images or videos to discern meaning or patterns, processing information in a manner similar to the human visual system^[1]. With the advent of deep learning, especially the Convolution Neural Network (CNN), there has been a significant improvement in the ability to autonomously learn features of images^[2]. As a result, deep learning has begun to be utilized in various fields, and in the realm of computer vision, topics such as object classification, object detection, segmentation, and natural language processing are being prominently researched^[3,4].

Object detection is a technology that simultaneously identifies the location and type of a

specific object within an image or video, with a wide range of applications^[5]. In security and monitoring systems, it is used to detect and track the movements of specific individuals or objects. In the medical field, it is applied to detect lesions or abnormalities within medical images. Especially in autonomous vehicles, it is essential for recognizing the surrounding environment and detecting obstacles or pedestrians. Due to the diverse applications requiring real-time and accurate information, high precision and swift processing speeds are demanded in object detection. Owing to its significance, the advancement of object detection technology is considered a critical research topic for many researchers, leading to continuous studies and developments^[6].

One application area that further emphasizes the importance of object detection technology is UGV

※ "This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the7 Ministry of Education"(2018R1A6A1A03024003) and by the MSIT(Ministry of Science and ICT), Korea, under the ICAN(ICT Challenge and Advanced Network of HRD) program(IITP-2022-RS-2022-00156394) supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation)"

♦ First Author : Kumoh National Institute of Technology, Department of IT Convergence Engineering, eo97@kumoh.ac.kr, 학생회원

° Corresponding Author : Kumoh National Institute of Technology, Department of IT Convergence Engineering, wdragon@kumoh.ac.kr, 종신회원

논문번호 : 202311-128-A-RU, Received November 3, 2023; Revised November 21, 2023; Accepted December 8, 2023

(Unmanned Ground Vehicle)^[7]. UGVs are unmanned vehicles that operate on the ground and can utilize object detection technology across various domains. Especially since UGVs operate in dynamic environments, the use of real-time object detection is essential. In military settings, they identify obstacles, enemy forces, or specific terrain features. In urban contexts, UGVs detect pedestrians, vehicles, traffic lights, and various signs in real-time. Conversely, object detection in UAVs (Unmanned Aerial Vehicles) holds even greater significance than in UGVs^[8]. Due to the enhanced mobility of UAVs, they require monitoring over vast areas from higher altitudes. Given these characteristics, they often detect objects in complex environments. For instance, in the military, they monitor enemy movements across expansive ranges. During disasters, they assess the overall situation of affected areas or search for missing persons. Thus, for the success of missions across diverse fields, the performance of object detection becomes a crucial factor. However, unlike UGVs, which are grounded and not heavily constrained by onboard weight limits, UAVs execute object detection with smaller onboards due to payload restrictions. Therefore, there's a heightened emphasis on achieving rapid inference speeds and high accuracy for real-time object detection on onboards with more limited computational capacities.

With the emphasized importance of real-time object detection in UAVs, various real-time object detection models and techniques (SSD, MobileNet, EfficientDet, YOLO) tailored for UAV environments have been researched. SSD (Single Shot MultiBox Detector) is structured to detect objects of various sizes in real-time and is widely used for real-time object detection in UAVs^[9]. MobileNet features a lightweight CNN structure and, when combined with SSD, is utilized for real-time object detection^[10]. EfficientDet is another model that garners attention, offering both fast speed and high accuracy through an efficient network structure^[11]. Among these, YOLO (You Only Look Once) stands out, with various versions from YOLOv2 to YOLOv8 that have been improved for optimal real-time processing^[12-15].

Continuous research is being conducted to achieve

high object recognition rates and fast inference speeds, even in the rapidly changing environments of UAVs. However, many of these studies fail to consider the unique characteristic of UAVs, which is their ability to freely adjust their altitude. As the UAV ascends to higher altitudes, objects on the ground become smaller in terms of resolution relative to the altitude. This can lead to a decline in object detection accuracy, as smaller objects result in fewer pixels, making their features less distinct. As the number of pixels for an object changes with altitude, it becomes challenging for the model to classify or detect that object correctly.

To overcome these challenges, research for object detection in UAVs is moving towards developing techniques specifically designed for detecting small objects at high altitudes (FPN, Anchors Matching Strategies, Data Augmentation, Super Resolution, Tiling). FPN (Feature Pyramid Networks) aims to detect small objects in images using filters of various scales^[16]. Anchors Matching Strategies adjust the size and ratio of anchors to make them more suitable for detecting small objects^[17]. Data Augmentation involves cropping or resizing images to train the network to recognize small objects better^[18]. Super Resolution enhances the resolution of low-resolution objects, making small objects clearer and easier to recognize^[19]. Moreover, research is also being conducted to modify the network structure or the loss function to enhance the detection capability for small objects^[20]. There's also research using the tiling algorithm, which splits the image into multiple small tiles and detects objects within each tile, aiming to detect smaller objects^[21-23].

Research on the development of tiling algorithms for small object detection has introduced various approaches. The study [21] presents a new framework called Slicing Aided Hyper Inference (SAHI), which divides high-resolution images into overlapping patches for detecting small objects. This tiling approach has been extensively expanded in studies such as [25, 26, 27, 28, 30]. Additionally, research [22] has conducted small object detection focusing on pedestrians and vehicles in Micro Aerial Vehicles (MAVs) using a PeleeNet-based SSD network. To enhance the efficiency of tiling, adaptive tiling

methods like [24, 28] have been investigated. Adaptive tiling, which splits the image into non-overlapping tiles and applies dynamic overlapping rates, increases detection accuracy while reducing computational requirements. Such tiling and adaptive tiling algorithms contribute significantly to real-time object detection on mobile platforms like UAVs.

However, most tiling algorithms have a fixed number of tiles, leading to significant variability in detection performance as the altitude of the UAV changes. Specifically, while these algorithms are advantageous for detecting small objects when the UAV is flying at high altitudes, they might degrade the detection performance at lower altitudes. Therefore, research is being conducted to address these issues.

In contrast with traditional approach where a fixed number of tiles is utilized to detect small object, this paper proposed a different method. In this paper, a dynamic number of tiles is proposed based on altitude and the size of detected bounding box. Therefore, it is possible to detect an object consistently regardless the altitude of the UAV.

The structure of the paper is as follows: In Chapter 2, a detailed description of the system model is provided and algorithm proposed in this paper. Chapter 3 offers a comprehensive introduction to the equipment, environment, methods, and results used in the experiments. Finally, Chapter 4 presents the conclusions of this study and discusses future research directions.

II. Real-time Automatic Tiling

In this section, a method for detecting small objects through altitude-based automatic tiling is presented. To be specified, the system model and proposed algorithm are described

2.1 Proposed System Model

The system model of this paper is presented in Fig. 1. As shown in Fig. 1, UAV is equipped with a Flight Controller (FC), an AI board, and a camera. Real-time tiling is performed using the images received from

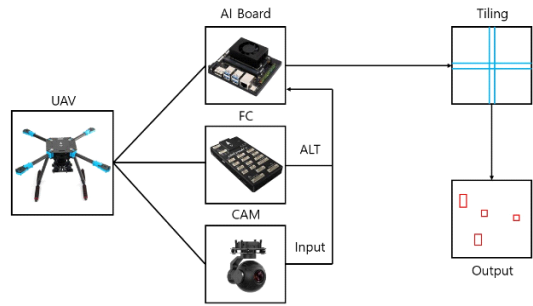


Fig. 1. System model

the camera. Afterward, a split image frame is used as network input to conduct inference on the AI board. To determine the proper number of tiles at each altitude, the altitude value is obtained from the FC using Robot Operation System (ROS), and the number of tiles is adjusted in real-time to detect small objects at different altitudes precisely.

2.2 Proposed Algorithm

As illustrated in Fig. 2, the tiling algorithm for image processing to find small objects has been studied. The tiling algorithm refers to the process of splitting a large dataset into multiple tiles. In this manner, each tile can be processed independently, in parallel, and distributed. Finding the proper tile size in the tiling algorithm is a crucial task. This is because memory issues arise when the number of tiles increases, leading to problems in inference speed. Therefore, it is essential to obtain the proper number of tiles in UAV small object detection task, where the altitude of the UAV is continuously changing. The algorithm, in theory, can continue tiling indefinitely as long as the altitude of the drone and the performance of the onboard AI board are supported.

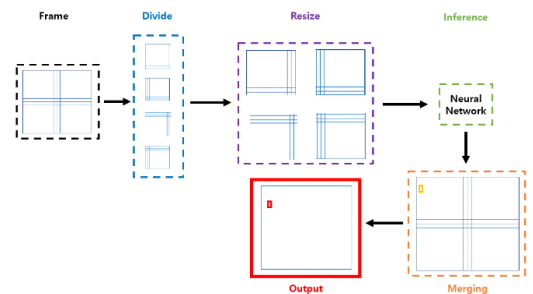


Fig. 2. Tiling Algorithm

However, in this paper, tiling was performed up to 3x3 due to the limited computational capacity of the onboard AI board.

In this paper, the existing tiling algorithm is integrated with YOLOv5. As mentioned in the introduction, there are various versions of YOLO. Although theoretical increases in accuracy have been observed with each new version, the real-time object detection in our considered onboard UAV environment does not translate these increases into practical differences. Therefore, based on our previous research experience, YOLOv5 was selected to ensure the reliability and consistency of our study. Inference is configured to operate an automatic tiling system according to Algorithm 1 at different altitudes that is integrated with YOLOv5 for object detection and is processed on an AI board mounted on a UAV. The real-time images received from the camera are split into rows and columns. The number of tiles is determined based on the altitude of the UAV and bounding box size from the object detection. The altitude value of UAV is obtained from the Pixhawk, an FC mounted on the UAV, using ROS and Mavlink ROS (MAVROS). If the bounding box size is small

at a specific altitude, the number of tiles is increased to enable small objects to be detected and thus increase the precision of the small object detection.

2.2.1 Image Splitting

The size of original image frame which contains height and width information is detected, and then the information is stored in the 'height' and 'width' variables. The size of the image frame is classified as follows; (i) In a 2D image, the height and width of 2D image are utilized directly; (ii) in a 3D image, the height and width of the image are stored in a variable and then utilized. To calculate the coordinates of the split image frame, the height, and width of each tile at time t can be defined as follows

$$h_i(t) = \frac{1}{R} \times h(t) \quad (1),$$

$$w_i(t) = \frac{1}{C} \times w(t) \quad (2),$$

where $h_i(t)$ denoted the height of i -th tile at time t , R denotes the rows of original image frame at time t , and $h(t)$ denotes the height of original image frame at time t . $w_i(t)$ denotes the width of i -th tile at time t , C denotes the columns of original image frame at time t , and $w(t)$ denotes the width of original image frame at time t . In these calculations, results are rounded, as rounding is generally utilized for providing the most accurate estimates, ensuring the dimensions of each tile closely match the actual pixel dimensions. Subsequently, an empty list named 'split images info' is then created. In this work, a corner-to-center method is utilized to determine the tile coordinates in the form of (y1, y2, x1, x2). The corner-to-center method is used to calculate the center point, height, and width of a bounding box using the coordinates of its two diagonal corners. At this step, an overlap part is added to prevent objects from being cut off when tiles division is performed. Iteratively, a dictionary named 'split image info' is created, then the image frame and coordinate tuple are stored using the keys 'image' and 'coords'. At each iteration, information on the split image frame is stored in the dictionary and appended to the 'split images info' list.

Algorithm. 1. Automatic Tiling Algorithm

Algorithm 1: Automatic Tiling Algorithm

```

Input : Real-time video, altitude, bbox size
Output: Object detection result
Initialize row (R), column (C);
while UAV is flying do
    Detect object through the real-time video;
    Get current altitude, image height (h), and
    image width (w);
    if current altitude > altitude threshold then
        Check bbox size;
        if bbox size < bbox size threshold then
            | Increase row (R) and column (C);
        else
            | Keep row (R), column (C);
        end
    else
        Check bbox size;
        if bbox size > bbox size threshold then
            | Decrease row (R) and column (C);
        else
            | Keep row (R), column (C);
        end
    end
end

```

The split image frame is converted to 640x640 for effective processing in the YOLOv5 network. Subsequently, the converted image frame is transformed into a PyTorch tensor. After inference, the results of the split images are appended to an empty list named 'result', merged, and bounding boxes are displayed on the original image.

2.2.2 Merging Results

To merge the detection results from the split image frames, a coordinate transformation aligned with the original image is essential. For this, the ratio between the split image and the resized image must be calculated, which can be done using the formulas split height/resized height and split width/resized width. Subsequently, the coordinates of the detected bounding boxes are multiplied by these calculated ratios to transform them to align with the original image. Therefore, each split image is resized for inference. Instead of remerging the split images, only the results of the inferred bounding boxes are merged into the original image, utilizing the calculated ratios between the original and resized images.

2.3 Model Lightweights

A lightweighting process in deep learning refers to reducing the size of the deep learning model and enhancing computational efficiency, which exponentially increases computational performance which is beneficial for a computer that has limited storage and memory, i.e., embedded boards. Therefore, the lightweighting process in deep learning is essential to be applied in the proposed tiling algorithm that will be processed on an AI board mounted on a UAV. In this paper, TensorRT, a deep learning inference library provided by Nvidia, was used. TensorRT operates in environments using NVIDIA GPUs and generates optimized engines that can quickly execute models trained in various deep learning frameworks. Training model optimization is achieved through Quantization and various optimization techniques. TensorRT offers various computational precisions, such as FP32 (Floating Point 32), FP16 (Floating Point 16, Half-Precision), and INT8 (Integer Quantization). Table 1 presents the

Table 1. TensorRT Performance Comparison

	Floating point 32	Floating point 16	Int8
mAP	0.329	0.329	0.098
Real-time FPS	16.7	38	66.6

measured mAP (mean Average Precision) for different operational precisions on the Visdrone Dataset, along with the real-time FPS (Frames Per Second) on the Nvidia Jetson Xavier NX. As observed in Table 1, there is not a significant difference in mAP between FP32 and FP16. However, a more than twofold difference in real-time FPS was noted. For INT8 precision, while the real-time FPS reached 66.6, there was a considerable decrease in mAP compared to FP precisions. Consequently, in this paper, the YOLOv5s.pt model has been optimized using TensorRT FP16 precision to maintain high performance in an embedded board environment.

III. Experiment

The experiment was conducted in two real environments where a UAV flew and performed object detection tasks. In this work, detection class is focused only on a "pedestrian" class. This is due to large objects, i.e., "car" which can relatively easy to be detected at high altitude, while pedestrian which classified as small object is hard to detect. Therefore, this study focuses on detecting small objects at various altitudes employing the proposed autonomous tiling algorithm, which is the objective of this work. In addition, the proposed tiling method can be beneficial for pedestrian monitoring with security and rescue scenarios, specifically in urban areas. For instance, the proposed tiling method which is applied on the UAV can be applicable in analyzing pedestrian flow at large-scale events, tourist spots, public facilities, and search and rescue missions during a disaster scenario.

3.1 Experimental Setup

In this paper, a Holybro X500 UAV frame was used as shown in Fig. 3. In order to conduct real-time object detection, Nvidia Jetson Xavier NX and Orin NX AI boards that were attached to the UAV were

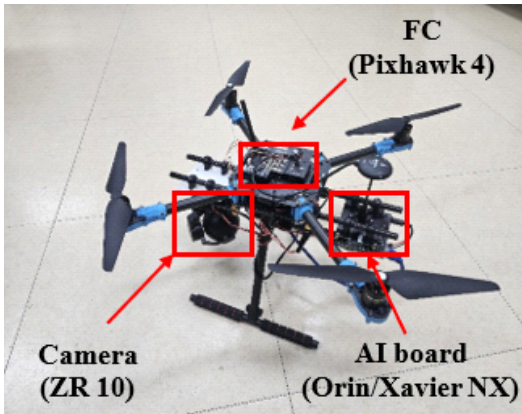


Fig. 3. Hardware

utilized. A real-time image frame was captured using the ZR10 monocular camera. The object detection was trained using a Visdrone dataset. Moreover, the 's' model provided by YOLOv5 was utilized to perform a real-time object detection. The TensorRT, one of the lightweight optimization processors, was employed to perform inference on the AI board.

As mentioned previously, the experiments in this paper were conducted in two distinct environments. The first environment, which was abundant in 'pedestrian' class small objects, served as the setting for evaluating and comparing the performance of each model at various altitudes. Experiments in this setting were conducted on a clear day over a grass field of 6084 square meters. As the altitude increased, an area up to 10000 square meters was covered. The number of pedestrians was observed, starting from one object at lower altitudes and increasing up to thirteen objects as the altitude rose. The second environment encompassed the vicinity of a building's parking lot, where the detection of small objects at varying altitudes was assessed. This setting, encompassing a mix of vehicular and pedestrian traffic as well as diverse background elements such as buildings, was deemed suitable for testing object detection performance in complex environments, and thus, was chosen for the experimental trials.

3.2 Datasets

As mentioned in Section 3.1, the dataset utilized for object detection is the Visdrone dataset. The

Visdrone dataset consists of data collected from drone views in various environments, primarily designed for object detection and object tracking from UAVs. This dataset possesses a broad range of characteristics suitable for object detection in multiple environments and conditions. However, the Visdrone dataset has a limitation where the size of the objects decreases as the altitude increases, which makes the object detection challenging. In this work, the Visdrone dataset, comprising 6471 training images, was utilized to train the models. The performance of the proposed tiling method at various altitudes was then measured, and the detection success rate and its limitations were analyzed.

3.3 Evaluation Metrics

In this study, three primary performance metrics are employed to evaluate the proposed method: precision, recall, and the F1 score. These indicators are essential for analyzing and assessing the performance of object detection models.

Precision is an indicator that reflects how accurately the models detects objects, denoted as the ratio of true positive (TP) detections to the sum of TP and false positive (FP) detections, where TP represents correctly identified objects, and FP represents incorrectly identified ones. Recall, on the other hand, indicates how well the model detects objects of a particular class, calculated as the ratio of TPs to the sum of TPs and false negatives (FN), with FNs being objects that were not detected but should have been. This metric represents the proportion of TPs against the ground truth.

However, there is a trade-off relationship between these two metrics. Increasing precision may result in a lower recall and vice versa. Due to this reason, the F1 score is utilized to conduct a balanced performance evaluation, representing the harmonic mean of precision and recall. The F1 score proves useful in assessing the performance of models when both metrics are considered simultaneously, providing a balance between them and offering an overall measure of model performance.

Thus, precision, recall, and the F1 score each evaluate the performance of the model from different

perspectives, ensuring an accurate reflection of object detection capabilities in complex and varied real-world scenarios. The fundamental rationale for employing these metrics in this research is to comprehensively assess whether the UAV-based object detection system fulfills the requirements in actual operational environments.

3.4 Experiment Result

Fig. 4 depicts the experimental site encompassing multiple pedestrian classes. At this location, an UAV was flown up to an altitude of 130 meters, wherein the performance of small object detection across various models was compared using precision, recall, and F1 score as evaluation metrics. As mentioned in the Evaluation Metrics section, TP, FN, and FP were used to define cases of detection failures. Each pedestrian was counted individually, with successful detection (TP) being considered when the algorithm accurately identified a pedestrian. Conversely, if a pedestrian present in the image was not detected, it was counted as a FN. Furthermore, when an object other than a pedestrian was incorrectly identified as a pedestrian, it was counted as a FP. Table 2 presents the precision and recall values for each model at varying altitudes. It was observed that the conventional Yolov5 model exhibited a decline in object detection capabilities as the altitude increased. Tiling algorithms employing a fixed number of tiles, such as 2x2 or 3x3, demonstrated improved performance at specific altitudes only. However, the proposed method consistently showed superior performance in small object detection across a range of altitudes, mainly because the number of tiles was appropriately chosen for each altitude. For instance, in the altitude range of 30-50 meters, the performance



Fig. 4. Experiment environment

of the proposed algorithm was observed to be better than the 2x2 tiling, as at lower altitudes, it is advantageous not to use tiling. Consequently, tiling is not employed at lower altitudes in the proposed method, while it is applied at altitudes where tiling becomes necessary. Table 3 lists the F1 scores for each threshold across all altitudes, confirming that the proposed method outperforms other models in terms of performance, effectively adapting to varying altitudes by selectively employing tiling.

Fig. 5 shows the results of object detection performed at each altitude. The first row of Fig. 4 presents the ground truth. The second, third, and fourth row represent the object detection results of employing YOLOv5, 2x2 tiling, and the proposed method, respectively. As shown in the figure, both the object detection using YOLOv5 and the tiling algorithm with a fixed number of tiles failed to detect objects at all altitudes. However, our proposed method, which is based on altitude and bounding box size, successfully detected small objects at every altitude.

The proposed method has been experimentally validated to recognize small objects at any altitude, based on UAV's altitude and bounding box size. At

Table 2. Precision and Recall by Altitude

Alt(m)	Yolov5		2x2		3x3		Proposed	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
30-50	0.763	0.213	0.611	0.210	0.342	0.168	0.763	0.213
50-70	0.300	0.107	0.708	0.161	0.570	0.171	0.708	0.161
70-90	0.261	0.103	0.733	0.233	0.670	0.257	0.733	0.233
90-110	0.470	0.078	0.633	0.183	0.606	0.274	0.606	0.274
110-130	0.613	0.075	0.622	0.220	0.636	0.295	0.636	0.295

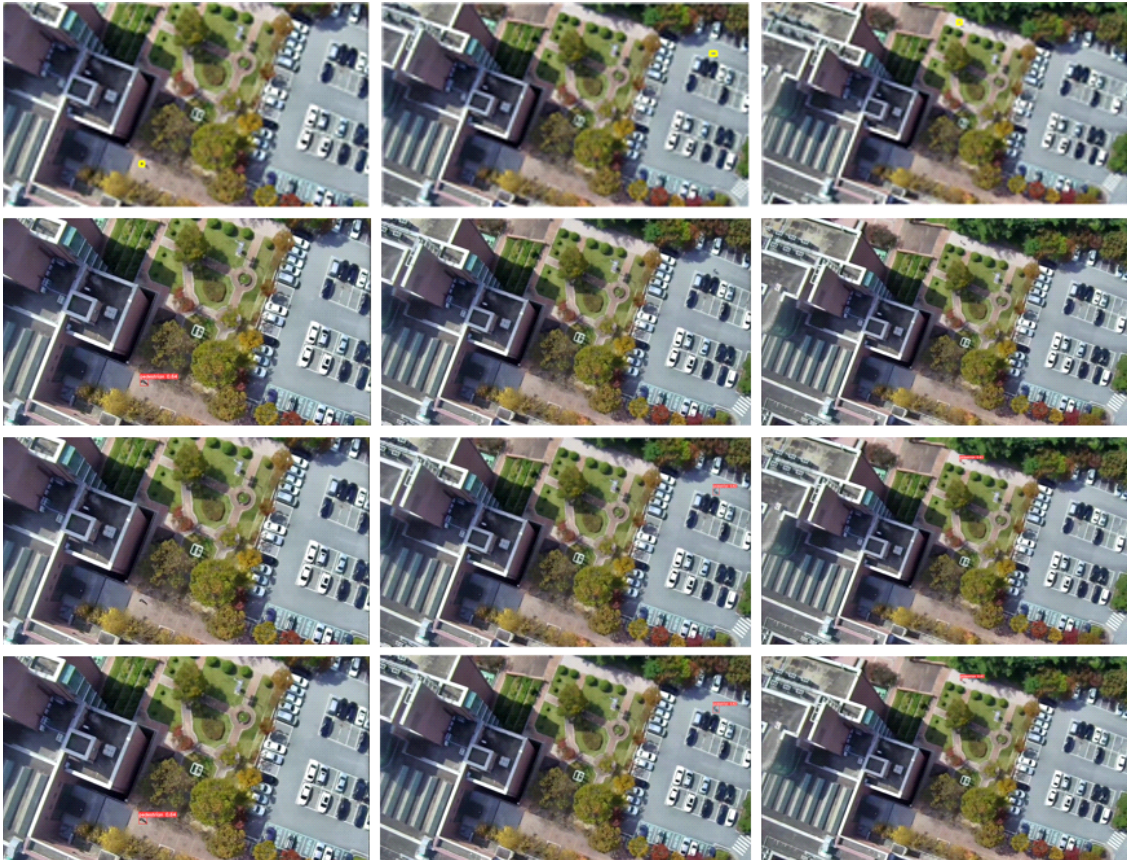


Fig. 5. Comparison of object detection results: The first row presents the ground truth with yellow boxes, the second row presents results using YOLOv5, the third row presents results with 2x2 tiling, and the fourth row presents results using the proposed method.

lower altitudes, object detection is performed without tiling, and the number of tiles is adjusted based on the altitude value and the size of the bounding box

Table 3. F1 score by All Altitude

F1 score threshold	Yolov5	2x2	3x3	Proposed
0.1	0.442	0.573	0.524	0.601
0.2	0.359	0.542	0.581	0.598
0.3	0.249	0.447	0.541	0.523
0.4	0.198	0.368	0.425	0.442
0.5	0.143	0.292	0.265	0.336
0.6	0.063	0.202	0.129	0.203
0.7	0.026	0.076	0.024	0.056
0.8	0.005	0.000	0.000	0.005
0.9	0.000	0.000	0.000	0.000
Average	0.165	0.278	0.276	0.307

Table 4. Real-time FPS for Object Detection on AI Board

Real-time FPS	Xavier NX	Orin NX
No tiling	38	70
2x2 tiling	9.5	17.5
3x3 tiling	4.2	7.8

to determine the proper number of tiles for each altitude. Table 4 presents the real-time FPS in two different AI board environments. It is observed from Table 4 that the increase in 3x3 tiles results in a performance drop below 10FPS on the AI board. Therefore, its use becomes an issue for real-time object detection on the AI board.

IV. Conclusions

In this paper, an algorithm that receives UAV altitude values in real-time is proposed and the size of bounding boxes is also considered to detect small objects at any altitude. The proposed method solved the issue of existed tiling algorithms that detect small objects using a fixed number of tiles. To be specified, in the proposed method, the number of tiles was dynamically adjusted by considering the altitude value and the size of the bounding box simultaneously. Therefore, by employing the proposed method, the small objects were able to be detected more stable at any altitudes. Additionally, this study's performance has been verified through experiments in real-world environments, with a focus on measuring real-time frame rates, demonstrating its practicality in the changing environments of drones. This aspect differentiates it from previous research, showcasing the applicability of tiling algorithms in actual environments.

However, all experiments were conducted only in clear weather conditions and at two specific locations. Additionally, the experiments were limited to altitudes up to 130 meters. It was observed that the use of 3x3 tiling becomes impractical for real-time operations above this altitude due to the limitations in FPS on the AI board. To address this issue, it is anticipated that the use of more advanced onboard environments or the application of other lightweight techniques such as transformation or distillation could improve real-time FPS and accuracy. Moreover, if the real-time video can be transmitted to a server for processing, this issue can be resolved. Further, it is expected that conducting experiments in a more diverse range of environments will more conclusively validate the proposed method's effectiveness, with plans to explore other lightweight techniques in future research.

References

[1] D. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice Hall, 2011.
 [2] A. Voulodimos, N. Doulamis, A. Doulamis,

and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational Intell. and Neurosci.*, vol. 2018, no. Article ID 7068349, pp. 1-13, Feb. 2018.

(<https://doi.org/10.1155/2018/7068349>)

- [3] L. Liu, W. Ouyang, X. Wang, et al., "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261-318, Oct. 2019.
 (<https://doi.org/10.1007/s11263-019-01247-4>)
- [4] B. Zhao, J. Feng, X. Wu, et al., "A survey on deep learning-based fine-grained object classification and semantic segmentation," *Int. J. Autom. Comput.*, vol. 14, no. 2, pp. 119-135, Jan. 2017.
 (<https://doi.org/10.1007/s11633-017-1053-3>)
- [5] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," in *Proc. IEEE*, vol. 111, no. 3, pp. 257-276, Mar. 2023.
 (<https://doi.org/10.1109/JPROC.2023.3238524>)
- [6] X. Wu, D. Sahoo, and S. C. H. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, vol. 396, pp. 39-64, Jul. 2020.
 (<https://doi.org/10.1016/j.neucom.2020.01.085>)
- [7] Q. Liu, Z. Li, S. Yuan, Y. Zhu, and X. Li, "Review on vehicle detection technology for unmanned ground vehicles," *Sensors*, vol. 21, no. 4, p. 1354, Feb. 2021.
 (<https://doi.org/10.3390/s21041354>)
- [8] K. Tong, Y. Wu, and F. Zhou, "Recent advances in small object detection based on deep learning: A review," *Image and Vis. Computing*, vol. 97, p. 103910, May 2020.
 (<https://doi.org/10.1016/j.imavis.2020.103910>)
- [9] W. Liu, et al., "SSD: Single shot multiBox detector," in *Computer Vision - ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9905, LNCS, Springer, Cham, Sep. 2016.
 (https://doi.org/10.1007/978-3-319-46448-0_2)
- [10] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M.

- Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, Apr. 2017.
(<https://doi.org/10.48550/arXiv.1704.04861>)
- [11] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. CVPR*, Jun. 2020.
(<https://doi.org/10.1109/CVPR42600.2020.01079>)
- [12] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. CVPR*, Jul. 2017.
(<https://doi.org/10.1109/CVPR.2017.690>)
- [13] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, Apr. 2018.
(<https://doi.org/10.48550/arXiv.1804.02767>)
- [14] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, Apr. 2020.
(<https://doi.org/10.48550/arXiv.2004.10934>)
- [15] 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," in *Proc. IEEE/CVF Conf. CVPR*, Jun. 2023.
(<https://doi.org/10.1109/CVPR52729.2023.00721>)
- [16] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. CVPR*, Jul. 2017.
(<https://doi.org/10.1109/CVPR.2017.106>)
- [17] Y. Zhong, et al., "Anchor box optimization for object detection," *arXiv preprint arXiv:1812.00469*, Jan. 2020.
(<https://doi.org/10.48550/arXiv.1812.00469>)
- [18] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1-48, Jul. 2019.
(<https://doi.org/10.1186/s40537-019-0197-0>)
- [19] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: A technical overview," *IEEE Signal Process. Mag.*, vol. 20, no. 3, pp. 21-36, May 2003.
(<https://doi.org/10.1109/MSP.2003.1203207>)
- [20] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *arXiv preprint arXiv:1708.02002*, Feb. 2018.
(<https://doi.org/10.48550/arXiv.1708.02002>)
- [21] F. C. Akyon, S. Onur Altinuc, and A. Temizel, "Slicing aided hyper inference and fine-tuning for small object detection," in *2022 IEEE ICIP*, pp. 966-970, Bordeaux, France, 2022.
(<https://doi.org/10.1109/ICIP46576.2022.9897990>)
- [22] F. Ö. Ünel, B. O. Özkalayci, and C. Çiğla, "The power of tiling for small object detection," *2019 IEEE/CVF Conf. CVPRW*, pp. 582-591, Long Beach, CA, USA, 2019.
(<https://doi.org/10.1109/CVPRW.2019.00084>)
- [23] Y. Kwon, Y. Jeong, and C. Lee, "Improved image tiling method for real-time vespa detection," *J. Apiculture*, vol. 37, no. 3, pp. 243-253, 2022.
(<https://doi.org/10.17519/apiculture.2022.09.37.3.243>)
- [24] S. T. Nguyen, T. Tulabandhula, and D. Nguyen, "Dynamic tiling: A model-agnostic, adaptive, scalable, and inference-data-centric approach for efficient and accurate small object detection," *arXiv preprint arXiv:2309.11069*, 2023.
(<https://doi.org/10.48550/arXiv.2309.11069>)
- [25] J. Lin, H. Lin, and F. Wang, "STPM_SAH: A small-target forest fire detection model based on swin transformer and slicing aided hyper inference," *Forests*, vol. 13, no. 10, pp. 1603, Sep. 2022.
(<https://doi.org/10.3390/f13101603>)
- [26] V. De Ridder, B. Dey, V. Blanco, S. Halder, and B. Van Waeyenberge, "Improved defect detection and classification method for advanced ic nodes by using slicing aided hyper inference with refinement strategy,"

arXiv preprint arXiv:2311.11439, 2023.

(<https://doi.org/10.48550/arXiv.2311.11439>)

- [27] A. Pereira, C. Santos, M. Aguiar, D. Welfer, M. Dias, and M. Ribeiro, "Improved detection of fundus lesions using yolov-csp architecture and slicing aided hyper inference," in *IEEE Latin Am. Trans.*, vol. 21, no. 7, pp. 806-813, Jul. 2023.
(<https://doi.org/10.1109/TLA.2023.10244179>)
- [28] H. Zhang, C. Hao, W. Song, B. Jiang, and B. Li, "Adaptive slicing-aided hyper inference for small object detection in high-resolution remote sensing images," *Remote Sens.*, vol. 15, no. 5, p. 1249, Feb. 2023.
(<https://doi.org/10.3390/rs15051249>)
- [29] Z. Yang, et al., "EdgeDuet: Tiling small object detection for edge assisted autonomous mobile vision," in *IEEE/ACM Trans. Netw.*, vol. 31, no. 4, pp. 1765-1778, Aug. 2023.
(<https://doi.org/10.1109/INFOCOM42981.2021.9488843>)
- [30] M. C. Keles, B. Salmanoglu, M. S. Guzel, B. Gursoy, and G. E. Bostanci, "Evaluation of YOLO models with sliced inference for small object detection," *arXiv preprint arXiv:2203.04799*, 2022.
(<https://doi.org/10.48550/arXiv.2203.04799>)

Mun Kyu Choi



Feb. 2022 : B.S. degree,
Kumoh National Institute
of Technology

Mar. 2022~Current : M.S
student, Kumoh National
Institute of Technology

<Research Interests> Deep

Learning, Object Detection, Drone application

[ORCID:0009-0008-0815-5732]

Soo Young Shin



Feb. 1999 : B.S. degree, Seoul
National University

Feb. 2001 : M.S. degree, Seoul
National University

Feb. 2006 : Ph.D, Seoul
National University

Feb. 2010~Current : Professor,

Kumoh National Institute of Technology

<Research Interests> 5G/6G wireless communi-
cations, Internet of things, Drone applications

[ORCID:0000-0002-2526-239]