applied sciences

MDPI

Article

Improved YOLOv5: Efficient Object Detection Using Drone Images under Various Conditions

Hyun-Ki Jung * and Gi-Sang Choi

Department of Electrical and Computer Engineering, Graduate School, University of Seoul, 163 Seoulsiripdaero, Dongdaemun-gu, Seoul 02504, Korea; simpson@uos.ac.kr
* Correspondence: stillhk3@uos.ac.kr

Abstract: With the recent development of drone technology, object detection technology is emerging, and these technologies can also be applied to illegal immigrants, industrial and natural disasters, and missing people and objects. In this paper, we would like to explore ways to increase object detection performance in these situations. Photography was conducted in an environment where it was confusing to detect an object. The experimental data were based on photographs that created various environmental conditions, such as changes in the altitude of the drone, when there was no light, and taking pictures in various conditions. All the data used in the experiment were taken with F11 4K PRO drone and VisDrone dataset. In this study, we propose an improved performance of the original YOLOv5 model. We applied the obtained data to each model: the original YOLOv5 model and the improved YOLOv5_Ours model, to calculate the key indicators. The main indicators are precision, recall, F-1 score, and mAP (0.5), and the YOLOv5_Ours values of mAP (0.5) and function loss were improved by comparing it with the original YOLOv5 model. Finally, the conclusion was drawn based on the data comparing the original YOLOv5 model and the improved YOLOv5_Ours model. As a result of the analysis, we were able to arrive at a conclusion on the best model of object detection under various conditions.

Keywords: object detection; YOLOv5; drone images

Citation: Jung, H.-K.; Choi, G.-S. Improved YOLOv5: Efficient Object Detection Using Drone Images under Various Conditions. *Appl. Sci.* **2022**, *12*, 7255. https://doi.org/10.3390/app12147255

Academic Editors: Mauro Lo Brutto, Junchi Yan and Minghao Guo

Received: 19 June 2022
Accepted: 17 July 2022
Published: 19 July 2022

1. Introduction

Recently, drones have been a field that is developing a lot, and they are likely to be combined into various fields in the future to create high value. Especially, low-budget drone photography technology can boost the local economy or help scientists research cultural heritage areas on the coast [1,2]. In this paper, we study the performance improvement of object detection model using drone photography.

There are also many cases of searching for object using drones at accident or disaster sites. However, it is confusing to detect missing persons or objects in a situation where visibility is not secured due to heavy rain and snow.

On the 10th of 2021, at least 40 tornadoes occurred in six weeks, including Kentucky, Arkansas, Illinois, Missouri, Tennessee, and Mississippi, confirming that at least 84 people were killed [3]. In this case, the number of missing persons will be much higher than the death. In this situation of lifesaving, a detection technique using a drone [4]; could be a solution. Drones and UAVs (unmanned aerial vehicles) have done many missions recently.

For example, be studied in fields such as automatic license plate recognition [5]; detection of the diseased plant [6]; traffic light detector for self-driving vehicles [7,8]; for violent individual identification [9]; and detector for ship detection in SAR Images [10]. Searching for missing objects in a disaster situation or used in operational missions in war situations, and it is necessary in a situation where medical staff can quickly find injured people at the accident site [11–13].

However, detection using such drone is greatly affected by surrounding situations [14]. To solve this problem, object detection using drones has been researched and developed [15], but related research is lacking a lot.

Additionally, it can be used in numerous situations as well as the above-mentioned situations. In the future, object detection using drones will be further developed and necessary in various situations. This paper discusses how to detect well in environment that is confusing to recognize objects to solve these problems. We were able to efficiently improve the performance of the model through Conv layer modification, the main layer of the original YOLOv5. In this work, we demonstrate the association of activation function with mAP (0.5) and loss function.

In this paper, we can summarize our main contributions as follows:

- Firstly, we improved the performance of model that can detect object under various environmental and weather conditions, such as Clear, Cloudy, Rainy, Snowy day, Evening, Night, Low altitude, and High altitude.
- Secondly, the Precision and mAP (0.5) were increased by modifying the Conv layer, the main layer of the Original YOLOv5 model. We replaced the SiLU activation function of the Conv layer with the ELU activation function. We applied the replaced ConvELU layer to the original C3, SPPF, and Conv layer of the Backbone and head part, and we used CIoU in two models: Original YOLOv5 and YOLOv5_Ours to find association with ELU activation function. As a result, we were able to reduce the convergence speed of loss function at the training process.

2. Materials and Methods

2.1. YOLOv5_Ours Network

Currently, there are two types of detection methods based on deep learning: 1-stage detector and 2-stage detector. Firstly, 2-stage detector in which regional proposal and classification are performed sequentially. The faster R-CNN [16] and mask R-CNN [17] correspond to the kind of 2-stage detector. In contrast to 2-stage detector, in the 1-stage detector, a regional proposal and classification are performed simultaneously. In other words, it is a method of solving classification and localization problems at the same time. YOLO [18], TPH-YOLOv5 [19], SSD [20], SSD MobileNet [21], Focal Loss [22], and RefineDet [23]; are representative algorithm of 1-stage detector. While it was popular in the past, Fast R-CNN has an inefficient problem in learning and execution speed because the candidate area generation module is performed in a separate module independently of CNN [24].

The YOLO is a famous object detection algorithm with several versions. It is easy to implement and can train the entire image immediately. For this reason, YOLO has developed gradually [25]. In 2020, the fifth version of YOLO was released. Compared to fast R-CNN, speed and accuracy have increased. Since YOLO does not apply a separate network for extracting candidate regions, it shows better performance in terms of processing time than Fast R-CNN [26]. Because Fast R-CNN was the combining hand-crafted and deep convolutional features method is used, there are limitations in detecting objects or humans [27]. The basic structure of the previous YOLOv5 [28] is largely divided into the backbone network part, the neck part, and the head part, as shown in Figure 1 [29].

Backbone is a convolutional neural network formed by aggregating image features in various particle sizes. Neck is a series of layers that mix and combine image features to deliver prior to prediction, and Head consumes features from Neck (PAnet) and takes box and class prediction steps. The biggest feature of YOLOv5 is that it has Focus and CSP (cross-stage partial connections) [30] layer. The focus layer was created to reduce layers, parameters, FLOPS, and CUDA memory and improve forward and backward speed while minimizing the impact of mAP. Three layers were used in YOLOv3 [31], but in the previous YOLOv5, it was changed to one layer [32]. The CSP layer extends to shallow information in the focus layer to maximize functionality, while the feature extraction module is iterated to extract detailed information and functions more thoroughly [33].

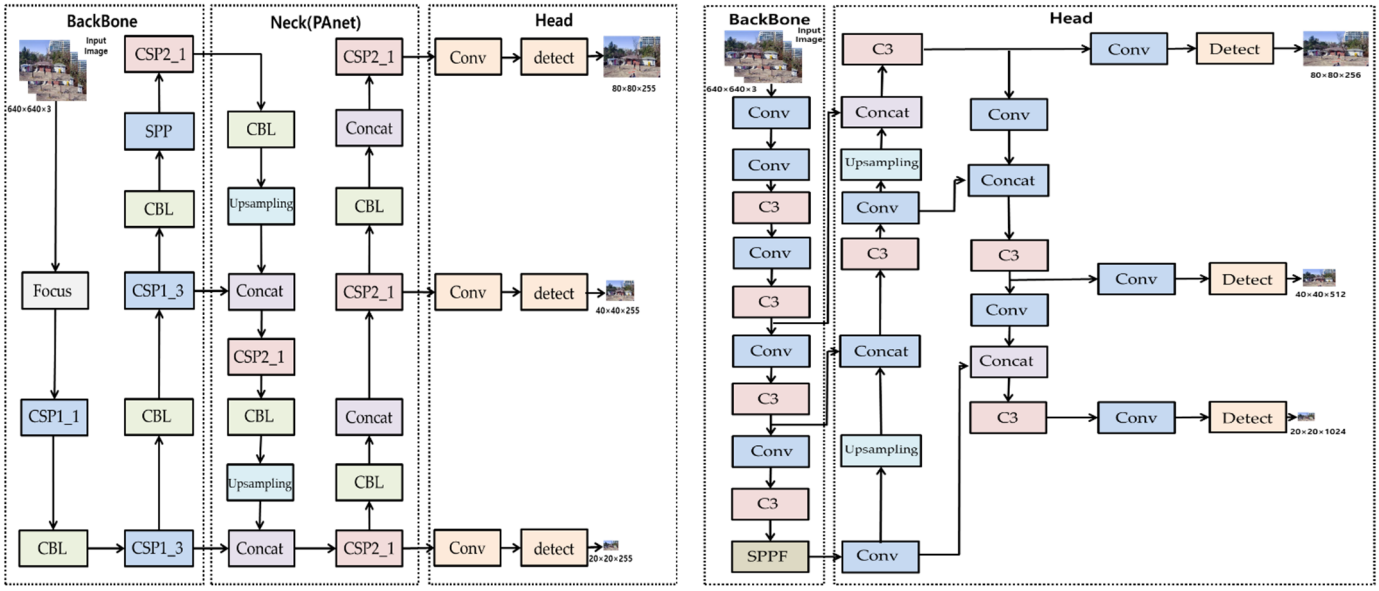

Figure 1. Structure comparison of previous and current YOLOv5 (**Left** figure is previous, **Right** figure is current YOLOv5).

The basic principle of YOLOv5 is similar to YOLOv4 [34]. YOLOv5 is an improvement base to YOLOv4, and YOLOv5 has the best performance in precision, recall, and average precision compared to Faster R-CNN, YOLOv3, and YOLOv4 [35,36]. In addition, YOLOv5 consists of four versions on its own, which are YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. This is classified according to the memory storage size, but the principle is the same. YOLOv5x has the largest storage size, and YOLOv5s has the smallest storage size. We improved the model based on the most basic YOLOv5s in this experiment.

There are two major differences between previous and current YOLOv5. Firstly, replaced the Focus layer with 6 × 6 Conv2d layer [37]. It is equivalent to a simple 2d-convolutional layer without the need for the space-to-depth operation. For example, a Focus layer with kernel size 3 can be expressed as a Conv layer with kernel size 6 and stride 2.

Secondly, the SPP layer was replaced by the SPPF layer. These operations increase the computational speed by more than double. This replacement is consequently efficient and faster in terms of speed. We noted the main layer of the current original YOLOv5 structure, the Conv layer, and we modified the Conv layer. In the original Conv layer, SiLU (Sigmoid-Weighted Linear Units) was used as an activation function.

Usually, the Conv layer uses ReLU (Rectified Linear Unit) as an activation function. This is because learning is fast and implementation is very simple due to the low amount of computation. However, the disadvantage of the ReLU activation function is that if it outputs a value less than zero, the gradient is likely to remain at zero, and the weight is likely to remain at zero forever until learning is completed. As a result, there is also a disadvantage in that learning is not conducted properly.

$$\mathrm{ReLU}(x) = \max(0,\ x) \tag{1}$$

$$\mathrm{ReLU}'(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The ELU activation function is a variant of the ReLU activation function. This reduces training time and improves the test set performance of neural networks. When $x <$ zero, the differential function is connected without breaking using the exponential function. If a broken function such as the step function is used, the loss function can be defined as uneven, resulting in local optima, as shown in Figure 2. The value of α is usually specified as 1. (If α is not 1, it is called SeLU.) In other words, the exclusive linear unit includes all

the advantages of ReLU and solves the Dying ReLU problem. The output value is almost zero-centered, and the exp function is calculated differently from the general ReLU.

$$\mathrm{ELU}(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha \exp(x) - 1, & \text{if } x \leq 0 \end{cases} \tag{3}$$

$$\mathrm{ELU}'(x) = \begin{cases} 1, & \text{if } x > 0 \\ f(x) + \alpha, & \text{if } x \leq 0 \end{cases} \tag{4}$$

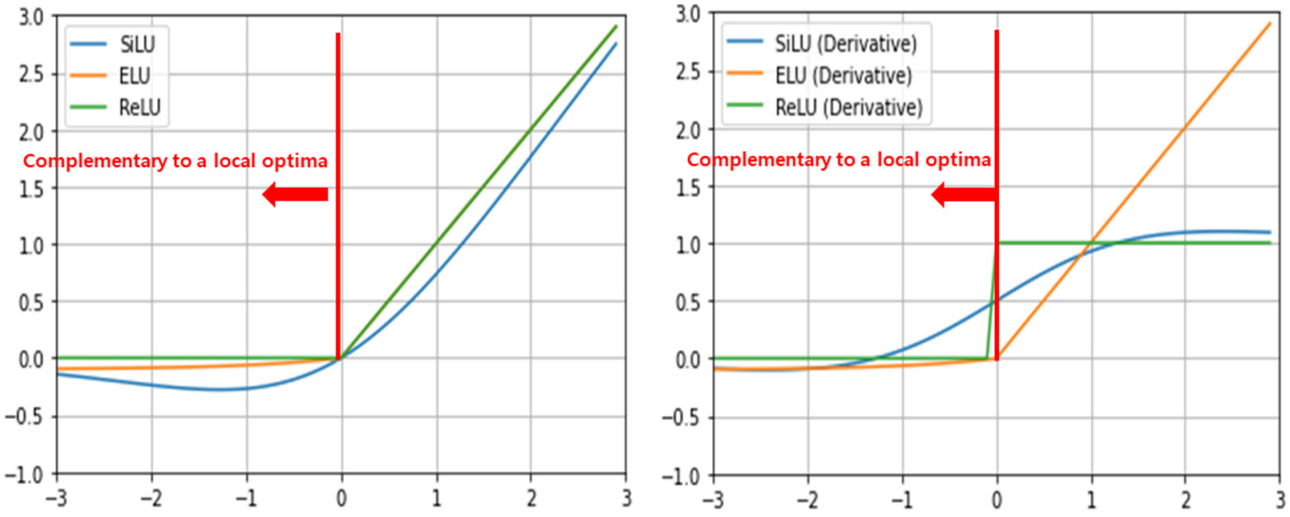

Figure 2. Graph comparison of SiLU, ELU, and ReLU activation function (**Left** figure is function, **Right** figure is a derivative function).

The SiLU (Swish) activation function can solve these problems, but it is only available in the hidden layers of deep neural networks and has the disadvantage that it can only be used in reinforcement learning-based systems. To solve this comprehensive problem, we used ELU (Exponential Linear Unit) as an activation function. SiLU activation function, which was previously used in the Conv layer, was replaced by the ELU activation function, as shown in Figure 3.

$$\mathrm{SiLU}(x) = x * \frac{1}{1 + \exp(-x)} \tag{5}$$

$$\mathrm{SiLU}'(x) = \frac{1}{1 + \exp(-x)} + \left\{ \left(x * \frac{1}{1 + \exp(-x)} \right) * \left(1 - \frac{1}{1 + \exp(-x)} \right) \right\} \tag{6}$$

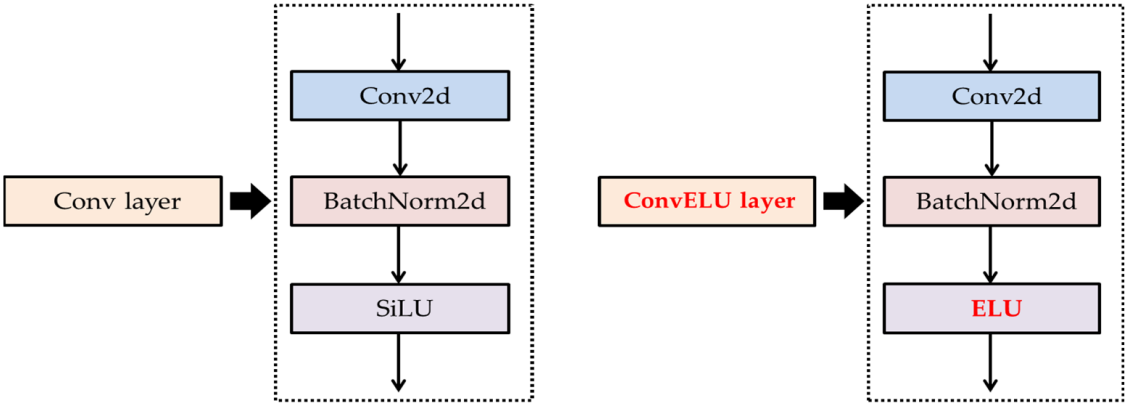

Figure 3. Comparison of Conv and Conv_Ours layer (**Left** figure is the existing layer, **Right** figure is the way we suggest it).

Both the SiLU activation function and ELU activation function can solve dying RELU, but the SiLU activation function has a problem of limited use, so we replaced it with the ELU activation function. We created the Conv layer with the activation function ELU applied, and we applied this to all of the ConvELU layers in the YOLOv5_Ours structure, as shown in Figures 3–5.

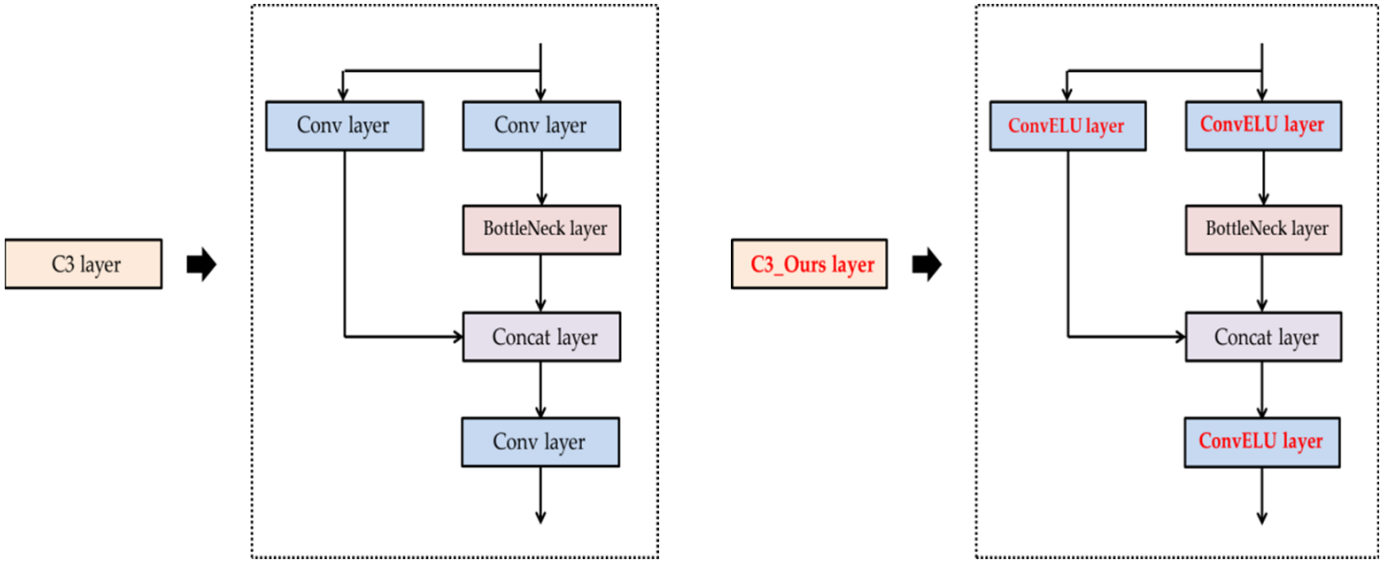

Figure 4. Comparison of original C3 and C3_Ours layer (**Left** figure is the existing layer, **Right** figure is the way we suggest it).

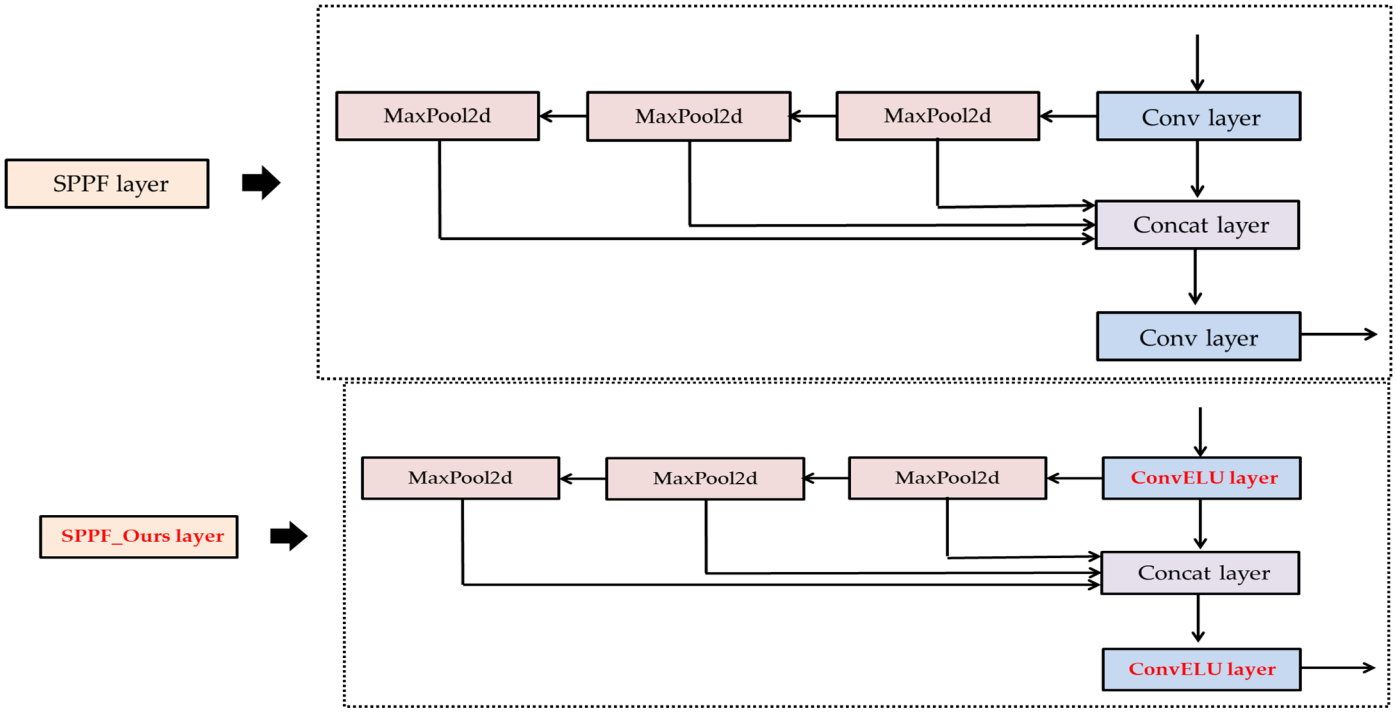

Figure 5. Comparison of original SPPF and SPPF_Ours layer (**Above** figure is the existing layer, **Under** figure is the way we suggest it).

The formula for calculating the output size at the Conv2d layer is Equation (7). In the equation, W is the size of the input data, F is the kernel size, P is the padding size, and S is the stride.

- Output size of Conv2d:

$$\text{Oupput size of Conv2d} = \frac{W - F + 2P}{S + 1} \tag{7}$$

The flowchart of the ConvELU layer is shown in Figure 6 as follows. BatchNorm2d layer means normalizing using average and variance, even if the data have various distributions for each batch unit in the training process. Figure 6 shows that the distribution of input values varies by batch unit or layer, but normalization makes the distribution Gaussian. This adjusts the distribution of the data to average zero and standard deviation to 1. Finally, the result of applying the Normalization and derivative activation function. And the final structure of applying all the measures is shown in Figure 7.

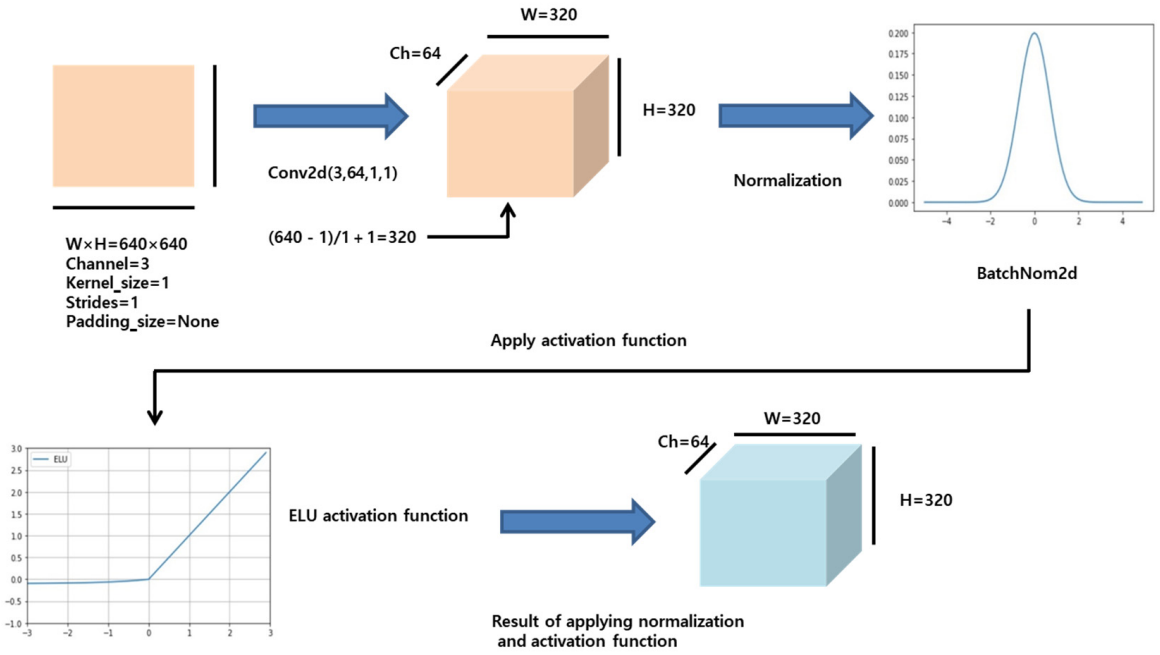

Figure 6. Flowchart of ConvELU layer (the way we suggest it).

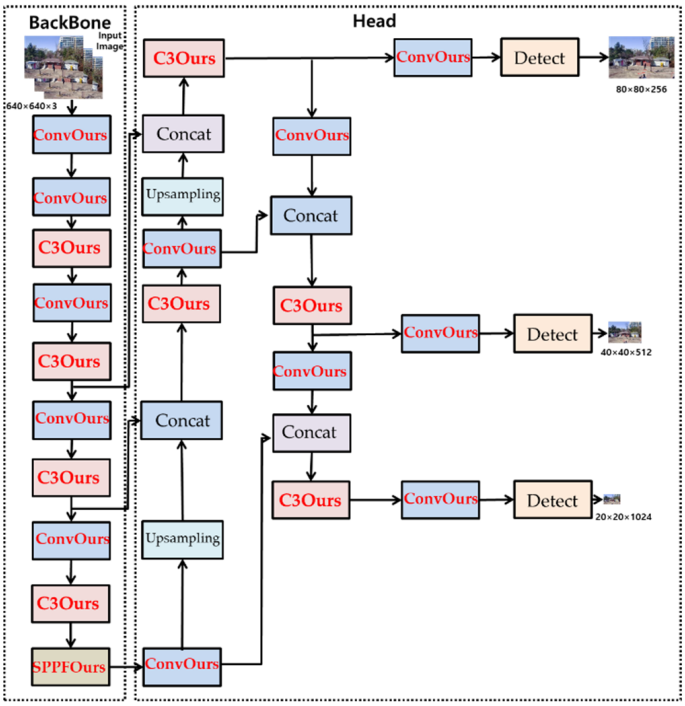

Figure 7. Structure of YOLOv5_Ours (the way we suggest it).

2.2. Data Preparation and Processing

Class selection and data collection are important to increase the accuracy of object search by training the model. The F11 4K PRO was used as the drone for filming. It has an adjustment distance of 10 m and a Wi-Fi image distance of 100 m. It is also suitable for object detection because it supports 4k camera image quality. According to the purpose of the study, the classes were designated as objects that are confusing to distinguish. Therefore, person, car, and notice were set as Classes, and the distance from the object was divided by less than 10 m: Low altitude and more than 10 m: High altitude. In addition, we took photos in various environments by changing the altitude of the drone, surrounding background, and weather. The shooting was conducted in the mountain and a downtown area, at low light: Evening and Night. In addition, it was filmed while changing the altitude of the drone. This is caused to create an environment where it is confused to identify objects.

Additionally, drone photographs were added from VisDrone (http://aiskyeye.com, accessed on 5 June 2021) [38] to collect more diverse data. VisDrone is a dataset used annually for object detection using drones and is very reliable [39]. This is to increase the accuracy of the experiment through reliable data combinations. In the VisDrone dataset, only data photographed above 10m: High altitude were added to meet the existing data and standard. Figure 8 shows the samples used in the experiment. In the final dataset used in the experiment were 2080 images: Clear, Cloudy, Rainy, Snowy day, Evening, Night, Low altitude, High altitude in training, 960 images: Clear, Cloudy, Rainy, Snowy day, Evening, Night, Low altitude, High altitude in validation, and 320 images: Clear, Cloudy, Rainy, Snowy day, Evening, Night, Low altitude, and High altitude in testing, prepared a total of 3360 images: Clear, Cloudy, Rainy, Snowy day, Evening, Night, Low altitude, and High altitude. Details are summarized in Table 1.

The collected data were then labeled from the online platform makesense (http://www.makesense.ai/, accessed on 14 July 2019) [40]. As shown in Figure 9, the label was created as three objects: person, car, and notice and annotated, and the annotated image was converted to a txt format according to the YOLO format.

Figure 8. Images of various experimental conditions captured by drone.

Table 1. Dataset classification by various experimental conditions.

Dataset	Experimental Conditions	Number of Images
Training	Clear day	260
	Cloudy day	260
	Rainy day	260
	Snowy day	260
	Evening	260
	Night	260
	Low altitude	260
	High altitude	260
Validation	Clear day	120
	Cloudy day	120
	Rainy day	120
	Snowy day	120
	Evening	120
	Night	120
	Low altitude	120
	High altitude	120
Testing	Clear day	40
	Cloudy day	40
	Rainy day	40
	Snowy day	40
	Evening	40
	Night	40
	Low altitude	40
	High altitude	40

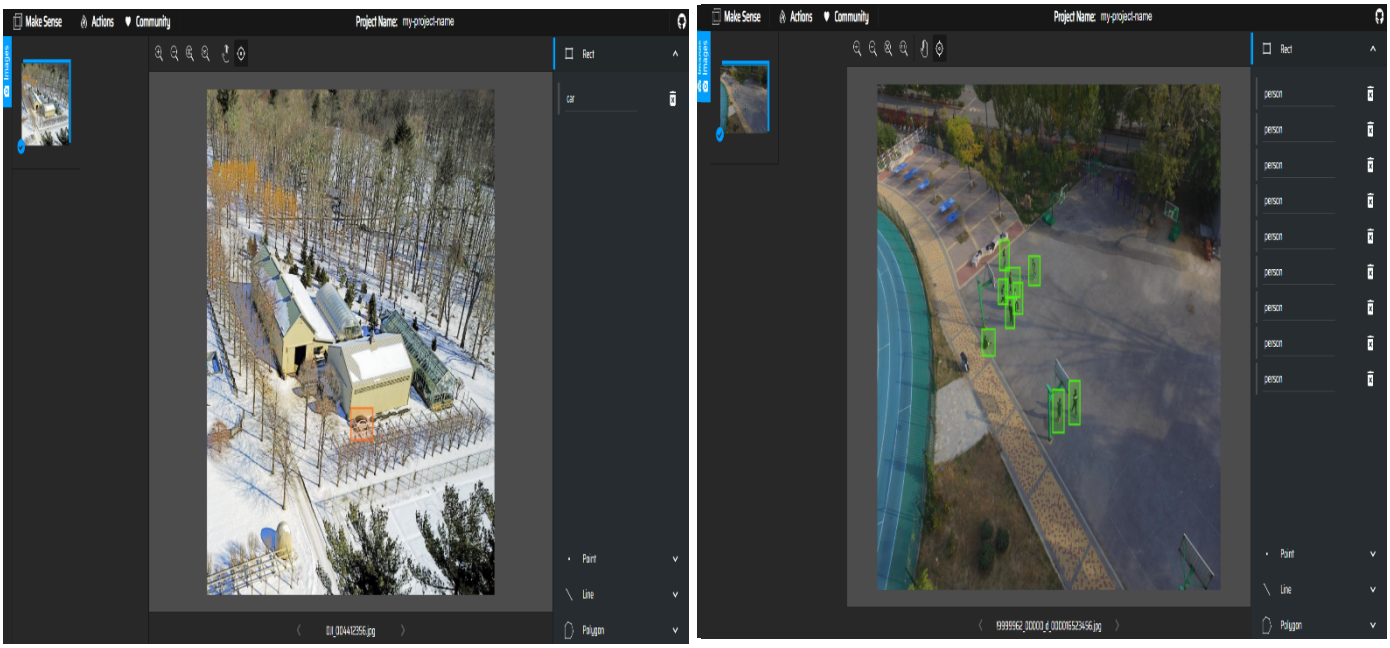

Figure 9. The process of changing data samples collected by drone to YOLO format (**Left** figure is the snowy day, **Right** figure is the evening).

3. Experiment and Results

3.1. Experimental Setup and Flowchart

For the experiment, the basic environment of the experiment was conducted in Google Colab. Colab is well organized with a GPU environment, so we used it. We also trained and compared with same data acquired by drone shooting. The difference between the original YOLOv5 model and the YOLOv5_Our model is as follows. The weight trained

by the original YOLOv5 model is put on the image data set as the pre-training weight of the configured data set [41]. That is, the original YOLOv5 model uses its own weight obtained by pre-learning on COCO (Common Object in Context) dataset. However, in this study, both the original YOLOv5 model and YOLOv5_Our model conducted experiments based on the same data. This is to compare the performance of the models under the same condition.

We were three classes: person, car, and notice labeled to be annotated according to the purpose of the study. This is because we thought it was the easiest thing to confuse with objects based on the photos taken. All data taken by drone were labeled with three objects: person, car, notice in this way. Through training, the loss function is calculated, and the best weight is updated in models: the original YOLOv5 model and YOLOv5_Our model. After that, we proceed with the validation and testing process with the best weight obtained through training. Then, predict the test data with the obtained weight.

To make an accurate comparison, the original YOLOv5 model and YOLOv5_Our model conduct the experiment completely separately. After the experiment, the following indicators were used to evaluate the performance of the model. In short, the research is conducted in the process shown in Figure 10.

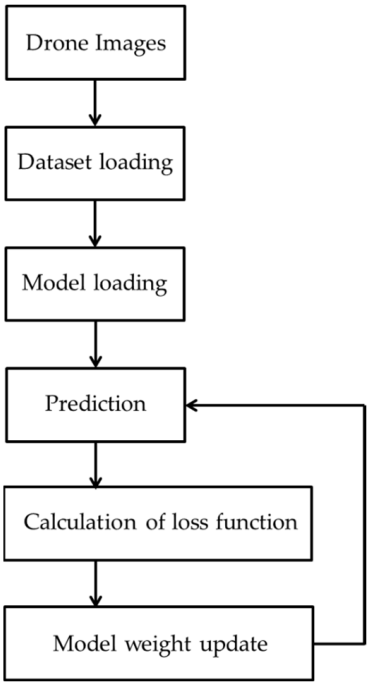

Figure 10. The structure of experimental method.

3.2. Experimental Key Indicators

In this paper, the performance of the original YOLOv5 model and YOLOv5_Ours model is evaluated based on Precision, Recall, F1-score, AP (average precision), and mAP (mean average precision).

- Precision:

$$\text{Precision} = \frac{\text{TP(True Positive)}}{\text{TP(True Positive)} + \text{FP(Fales Positive)}} = \frac{\text{TP(True Positive)}}{\text{All Detections}} \tag{8}$$

- Recall:

$$\text{Recall} = \frac{\text{TP(True Positive)}}{\text{TP(True Positive)} + \text{FN(Fales Nagative)}} = \frac{\text{TP(True Positive)}}{\text{All Ground Truths}} \tag{9}$$

Precision refers to the percentage of all detection results that are correctly detected. Recall is used to indicate how well a positive prediction is made when a positive input is given. Simply put, it means how well model detect it.

TP (True Positive) is a number detected to fit an object. FP (False Positive) means that it is detected as an object of another class. In other words, it is a false detection. FN (False Negative) means an object that should have been detected but not detected, and the TN (True Negative) means nothing that should not be detected.

- F1-score:

$$\text{F1} - \text{score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

It is calculated as the harmonic mean of precision and recall and not the arithmetic mean. F1-score has a value between zero and 1; the higher the value, the higher the accuracy of detecting an object. mAP (mean average precision) is the average value of the AP (average precision), indicating how accurate the predicted result is.

- AP:

$$\text{AP} = \int_0^1 P(R)dR \tag{11}$$

- mAP:

$$\text{mAP} = \frac{1}{|Q_R|} \sum_{q=Q_R} \text{AP}(q) \tag{12}$$

3.3. Experimental Loss Function

IoU (Intersection over Union) [42] is produced by the interaction between the predicted box and the ground truth box. That is, it is a value representing the size of the predicted Bounding Box and Ground Truth in the field of object detection as a value between zero and 1. The formula is as follows. A is the predicted box, and B is the ground truth box. C box is the smallest box, including A and B box, and $C \backslash A \cap B$ is the area in which the sum of A and B box is subtracted from the C box area. The GIoU (Generalized IoU) is the value obtained by subtracting the ratio of areas that do not overlap with both A and B in the C box. The larger the GIoU, the better the performance.

When $1 - \text{GIoU}$ is used as loss in object detection (the range of the loss value is zero ~2), the bounding box prediction process of GIoU loss according to Iteration is performed by expanding the B box area to overlap with GT and then reducing the B box area to increase IoU. This can improve the gradient vanishing problem for non-overlapping boxes, but there is a problem that the convergence rate is slow and the box is predicted incorrectly. To solve this problem, we use CIoU (Complete-IoU) in this paper to compare the loss function of the Original YOLOv5 model with the YOLOv5_Ours. In other words, the experiment is conducted under the condition that CIoU is applied equally to two models.

- IoU:

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|} \tag{13}$$

- GIoU:

$$\text{GIoU} = \text{IoU} - \frac{|C \backslash A \cap B|}{|C|} \tag{14}$$

- L_{GIoU}:

$$L_{GIoU} = 1 - \text{GIoU} \tag{15}$$

As can be seen from Equation (18), w is the width, and h is the height of the prediction box. Additionally, w^{gt} and h^{gt} are the width and height of the ground truth box. v measures the consistency of the aspect ratio of the two boxes, α is a positive trade-off parameter to adjust the balance between the non-overlapping case and the overlapping case. In particular, in the case of non-overlapping, the overlap area factor gives a higher priority to regression loss.

- L_{CIoU}:

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b,\ b^{gt})}{c^2} + \alpha v \tag{16}$$

- α:

$$\alpha = \frac{v}{1 - IoU + v} \tag{17}$$

- v:

$$v = \frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w^{p}}{h^{p}})^{2} \tag{18}$$

3.4. Results

The original YOLOv5 model and YOLOv5_Ours model were trained at 100 epochs and with the 3360 images: training images, validation images, and testing images. As a result of training all models, the average time spent training was about 2 h per model. The model that took the most time was the original YOLOv5 model, which took 2 h and 10 min. The object detection comparison results of the two models (the original YOLOv5 model and the YOLOv5_Ours model) are shown in Table 2 and Figure 11. Additionally, this table shows the Precision, Recall, F-1 score, and mAP of the original YOLOv5 model and YOLOv5_Ours. We compared based on the best of the 100 epochs result values. In order to objectively evaluate the performance of the models, the values of mAP (Mean average precision) were compared. The mAP value of the original YOLOv5 model is 94.6%, and YOLOv5_Ours is 95.5%. Overall, it may be seen that the YOLOv5_Ours model has higher than the original YOLOv5 model.

Table 2. Comparison table of best performance by models.

Model	Backbone	Precision	Recall	F-1 Score	mAP (0.5)
YOLOv5	CSPdarknet	90.1	89.8	89.9	94.6
YOLOv5_Ours	CSPdarknet	90.7	87.4	89.0	95.5

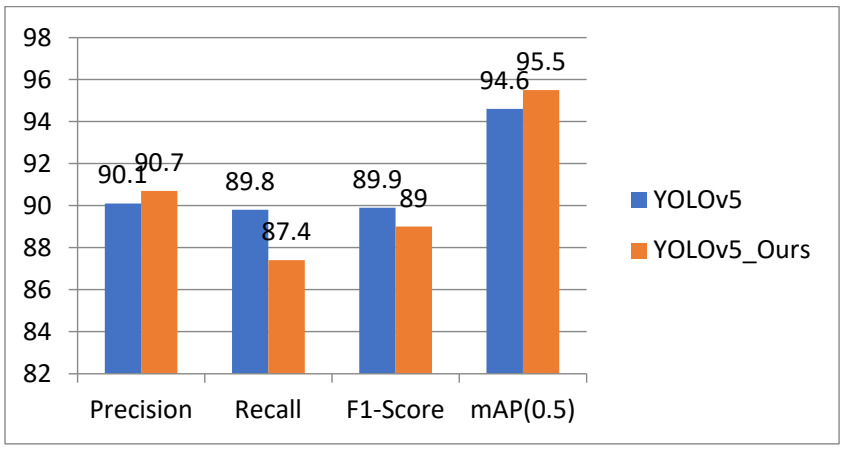

Figure 11. Comparison graph of result values for Original YOLOv5 and YOLOv5_Ours model.

As a result of the training and validation process, we found that the YOLOv5_Ours model was the best. Thus, the final prediction was made based on the weight obtained from the trained YOLOv5_Ours model, which was considered to have the best performance. The left part of Figure 12 shows the graphs of the metrics curves as training progresses. It is proved the detection accuracy of the YOLOv5_Ours model [43]. After evaluation, the YOLOv5_Ours model had a validation precision score of 90.7%, recall score of 87.4%, as well as F1-score of 88.8%, and mAP score is 95.5%. This result confirms the effectiveness of our approach in predicting experiment performed in several environments correctly.

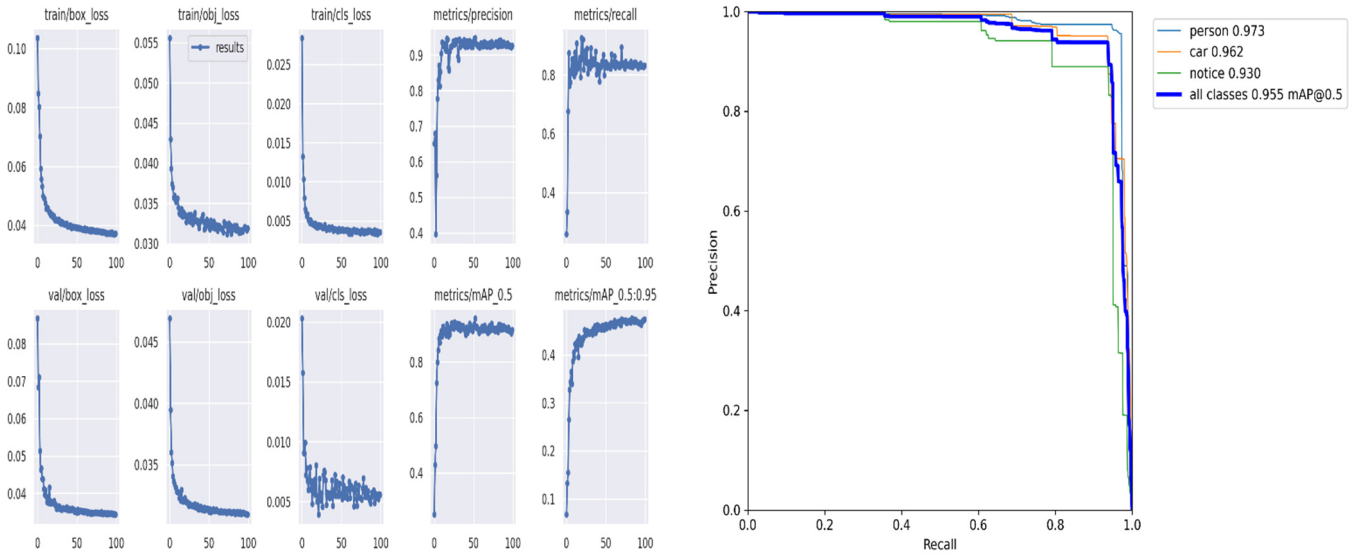

Figure 12. Graph of result values for YOLOv5_Ours model. (**Left** figure is changes in key indicators according to the epochs of training; **Right** figure is Precision–Recall curve.).

The first three columns are the YOLOv5_Ours model loss components, box loss, objectness loss, and classification loss, train the leftmost row and validation second row [44]. The box loss, objectness loss, and classification loss are indicators of how well an algorithm predicts an object [45]. These results mean that the three classes: person, car, and notice, which we use for detection, are accurately recognized during the training process.

Precision–Recall curve is a method of evaluating the performance of an object detector due to a change in the threshold value for the confidence level. The confidence level is a value that tells user how confident the algorithm is about the detection. In other words, the closer the number is to 1, the more confident the model is in detecting the target object. The right part of Figure 12 is the Precision–Recall curve graph of the YOLOv5_Ours model. It can be seen that the value of person is 97.3%, which is quite high.

The results are shown in Figure 13 by experimental conditions: Clear, Cloudy, Rainy, Snowy day, Evening, Night, Low altitude, and High altitude. For clear day and evening, object detection showed high accuracy above the value of about 87.0%. Rainy day is relatively low, about 57.0%, but overall, object detection is excellent.

It can be seen from Table 3 that the object detection results of the YOLOv5_Ours model. Among the detected objects, the value for a person was the highest. The person detection was calculated as 97.1% for Precision, 84.3% for Recall, 90.2% for F1-Score, and finally 97.3% for mAP. This means that the person detection rate is quite high.

The function loss difference between the two models results in a large gap at the beginning of the training. Therefore, the experiment was conducted by setting the epoch to 100.

It can be seen that YOLOv5 function loss occurs rapidly at the beginning of training. On the other hand, YOLOv5_Ours decreased function loss slowly. The gap appears to be narrowing until the epoch reaches 60. After that, the function loss of the two models: Original YOLOv5 and YOLOv5_Ours, is a little different. Figure 14 shows a graph comparing

the function loss value of the two models. That is, YOLOv5_Ours means an efficient model with low convergence speed.

Figure 13. Detection result for the YOLOv5_Ours model (**Red** is person, **Orange** is notice, and **Pink** is car).

Table 3. Key indicators of YOLOv5_Ours model.

Parameter	Person	Car	Notice	Total
Precision/%	97.1	87.4	87.7	90.7
Recall/%	84.3	94.6	83.1	87.4
F1-Score/%	90.2	90.9	85.3	88.8
mAP (0.5)/%	97.3	96.2	93.0	95.5

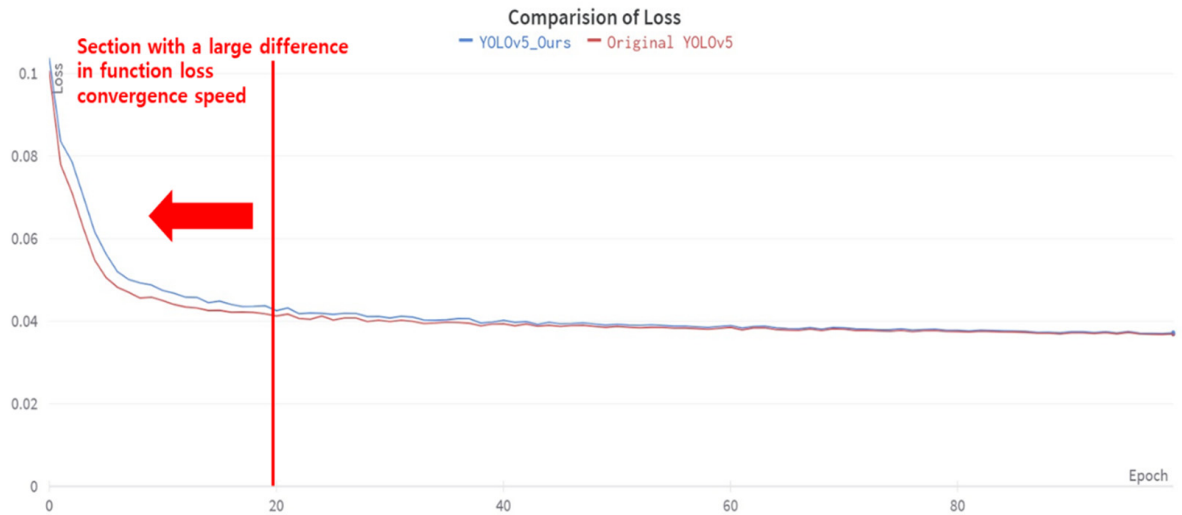

Figure 14. Comparison graph of function loss for Original YOLOv5 and YOLOv5_Ours model.

3.5. Comparison with Previous YOLO Models

For accurate verification of the study, it is necessary to compare performance with previous YOLO models. Therefore, we decided to experiment by applying the dataset to YOLOv3 and YOLOv4 model. The value of mAP was compared with the previous models: YOLOv3 and YOLOv4 model, and all the experiments were conducted independently. It is summarized as shown in Table 4 and Figure 15 for comparison of the data result value. As a result of comparing the final value, it was found that the performance of YOLOv5_Ours was the best.

Table 4. Comparison of performance by previous YOLO models and YOLOv5_Ours.

Model	Backbone	mAP (0.5)
YOLOv3	Darknet53	93.9
YOLOv4	CSPdarknet	91.0
YOLOv5	CSPdarknet	94.6
YOLOv5_Ours	CSPdarknet	95.5

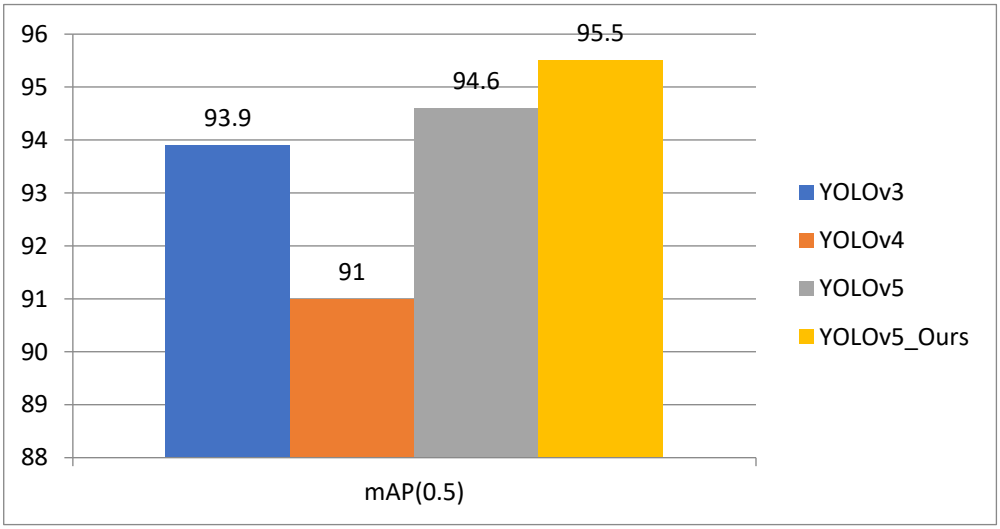

Figure 15. Comparison graph of result values for Previous YOLO models and YOLOv5_Ours. model.

4. Conclusions

In this paper, we studied a model for detecting objects in conditions that are confusing to detect objects. To create this environment, images were acquired using a drone in situations where it was confusing to detect objects such as various altitudes, weather, and background. In addition, it aimed to detect objects in these environments and increases detection performance.

The experimental method is based on the YOLOv5 structure. We compared the results with the original YOLOv5 model and improved the YOLOv5_Our model, and through training, it was selected for the YOLOv5_Ours model with the best performance. Then, the best weight obtained through validation is applied to the YOLOv5_Ours model and tested. As a result, we found that the mAP has increased to 0.9% compared with the original YOLOv5 model and improved the YOLOv5_Ours model. Finally, for a more accurate comparison, the key indicators were calculated with the previous version of YOLO: YOLOv3 and YOLOv4. The difference between the value of YOLOv3, YOLOv4, and mAP was 1.6% and 4.5%, respectively, which was greater than the original YOLOv5 model. In addition, it was confirmed that the convergence speed of loss function of YOLOv5_Ours model was reduce the compared to original YOLOv5 model at the beginning of training.

Object detection using drones is greatly influenced by the surrounding environment. We conducted research to improve the performance of the model under bad conditions, and we were able to obtain improved results. It may be applied to object recognition studies

using drones that have been previously conducted [46,47]. In the future, the results of this study will help use drones to detect objects in various conditions.

Author Contributions: Conceptualization, H.-K.J.; methodology, H.-K.J.; software, H.-K.J.; validation, H.-K.J.; formal analysis, H.-K.J.; investigation, H.-K.J. resources, H.-K.J.; data curation, H.-K.J.; writing—review and editing, H.-K.J.; visualization, H.-K.J.; supervision, H.-K.J.; project administration, H.-K.J. and G.-S.C.; funding acquisition, H.-K.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this paper were directly produced and processed.

Acknowledgments: We are thankful to Jae-Sub Jung and Yong-Gi Jeong who helped us shoot drone.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Marco, W.; Dennis, E.; Andreas, R. Potentials of Low-Budget Microdrones: Processing 3D Point Clouds and Images for Representing Post-Industrial Landmarks in Immersive Virtual Environments. *Front. Robot. AI* **2022**, *9*, 886240.
2. Apostolos, P.; Dimitris, K.; Yannis, K.; Michail, C.; Vasilis, K.; Konstantinos, T.; Michail, V. Mapping Cultural Heritage in Coastal Areas with UAS: The Case Study of Lesvos Island. *Heritage* **2019**, *2*, 1404–1422.
3. Chosun News. Available online: https://chosun.com/international/us/2021/12/12/2A4EGF613NC7RM3S5XI4OA6LW4/ (accessed on 12 December 2021).
4. Hung, G.-L.; Sahimi, M.-S.-B.; Samma, H.; Almohamad, T.-A.; Lahasan, B. Faster R-CNN Deep Learning Model for Pedestrian Detection from Drone Images. *SN Comput. Sci.* **2020**, *10*, 1007. [CrossRef]
5. Laroca, R.; Severo, E.; Zanlorensi, L.-A.; Oliveira, L.-S. A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector. *arXiv* **2018**, arXiv:1802.09567v6.
6. Chaschatzis, C.; Karaiskou, C.; Mouratidis, E.-G.; Karagiannis, E.; Sarigiannidis, P.-G. Detection and Characterization of Stressed Sweet Cherry Tissues Using Machine Learning. *Drones* **2022**, *6*, 3. [CrossRef]
7. Kim, J.-k.; Cho, H.-k.; Hwangbo, M.; Choi, J.-H. Deep Traffic Light Detection for Self-driving Cars from a Large-scale Dataset. In Proceedings of the 21st International Conference on Intelligent Transportation Systems, Maui, HI, USA, 4–7 November 2018; pp. 280–285.
8. Jensen, M.-B.; Philipsen, M.-P.; Møgelmose, A.; Moeslund, T.-B.; Trivedi, M.-M. Vision for Looking at Traffic Lights: Issues, Survey, and Perspectives. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 1800–1815. [CrossRef]
9. Singh, A.; Patil, D.; Omkar, S. Eye in the Sky: Real-time Drone Surveillance System (DSS) for Violent Individuals Identification using ScatterNet Hybrid Deep Learning Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1629–1637.
10. Yang, R.; Pan, Z.; Jia, X.; Zhang, L.; Deng, Y. A Novel CNN-Based Detector for Ship Detection Based on Rotatable Bounding Box in SAR Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 1938–1958. [CrossRef]
11. Jayalath, K.; Munasinghe, S.-R. Drone-based Autonomous Human Identification for Search and Rescue Missions in Real-time. In Proceedings of the 2021 10th International Conference on Information and Automation for sustainability, Negambo, Sri Lanka, 11–13 August 2021; pp. 518–523.
12. Rizk, M.; Slim, F.; Charara, J. Toward AI-Assisted UAV for Human Detection in Search and Rescue Missions. In Proceedings of the 2021 International Conference on Decision Aid Sciences and Application, Sakheer, Bahrain, 7–8 December 2021; pp. 781–786.
13. Tariq, R.; Rahim, M.; Aslam, N.; Bawany, N.; Faseeha, U. Dronaid: A smart human detection drone for rescue. In Proceedings of the 15th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT, Islamabad, Pakistan, 8–10 October 2018; pp. 33–37.
14. Sharma, T.; Debaque, B.; Duclos, N.; Chehri, A.; Kinder, B.; Fortier, P. Deep Learning-Based Object Detection and Scene Perception under Bad Weather Conditions. *Electronics* **2022**, *11*, 563. [CrossRef]
15. Zhu, P.; Wen, L.; Bian, X.; Ling, H.; Hu, Q. Vision Meets Drones: A Challenge. *arXiv* **2018**, arXiv:1804.07437v2.
16. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Neural* **2015**, 91–99. [CrossRef]
17. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
18. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *CVF* **2016**, 779–788.
19. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios. *CVF* **2021**, *2108*, 11539.

20. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.-C. SSD: Single Shot MultiBox Detector. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 21–37.
21. Rahmaniar, W.; Hernawan, A. Real-Time Human Detection Using Deep Learning on Embedded Platforms: A Review. *J. Robot. Control* **2021**, *2*, 462–468.
22. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
23. Zhang, S.; Wen, L.; Bian, X.; Lei, Z.; Li, S. Single-Shot Refinement Neural Network for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4203–4212.
24. Cheng, D.; Gong, Y.; Zhou, S.; Wang, J.; Zheng, N. Person Re-Identification by Multi-Channel Parts-Based CNN with Improved Triplet Loss Function. *CVF* **2016**, 1335–1344.
25. Jiang, P.; Ergu, D.; Liu, F.; Cai, Y.; Ma, B. A Review of Yolo Algorithm Developments. *Procedia Comput. Sci.* **2022**, *199*, 1066–1073. [CrossRef]
26. Lee, Y.-H.; Kim, Y.-S. Comparison of CNN and YOLO for Object Detection. *J. Semicond. Disp. Technol.* **2020**, *19*, 1.
27. Zhang, L.; Lin, L.; Liang, X.; He, K. Is Faster R-CNN Doing Well for Pedestrian Detection? *ECCV* **2016**, *10*, 1007.
28. Ultralytics/yolov5. Available online: https://github.com/ultralytics/yolov5 (accessed on 25 June 2020).
29. Jia, W.; Xu, S.; Liang, Z.; Zhao, Y.; Min, H.; Li, S.; Yu, Y. Real-time automatic helmet detection of motorcyclists in urban traffic using improved YOLOv5 detector. *IET Image Process.* **2021**, *10*, 1049. [CrossRef]
30. Wang, C.-Y.; Mark Liao, H.-Y.; Wu, Y.-H.; Chen, P.-Y.; Hsieh, J.-W.; Yeh, I.-H. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. *CVF* **2020**, 14–19.
31. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
32. Glenn-Jocher. Available online: https://github.com/ultralytics/yolov5/discussions/3181m1 (accessed on 16 May 2021).
33. Li, S.; Li, Y.; Li, Y.; Li, M.; Xu, X. YOLO-FIRI: Improved YOLOv5 for Infrared Image Object Detection. *IEEE Access* **2021**, *2021*, 3120870. [CrossRef]
34. Bochkovskiy, A.; Wang, C.-Y.; Mark Liao, H.-Y. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
35. Wang, H.; Zhang, S.; Zhao, S.; Wang, Q.; Li, D.; Zhao, R. Real-time detection and Tracking of fish abnormal behavior based on improved YOLOV5 and SiamRPN++. *Comput. Electron. Agric.* **2022**, *192*, 106512. [CrossRef]
36. Li, Z.; Lu, K.; Zhang, Y.; Li, Z.; Liu, J.-B. Research on Energy Efficiency Management of Forklift Based on Improved YOLOv5 Algorithm. *J. Math.* **2021**, *2021*, 5808221. [CrossRef]
37. Glenn-Jocher. Available online: https://github.com/ultralytics/yolov5/issues/4825 (accessed on 16 September 2021).
38. VisDrone. Available online: http://aiskyeye.com (accessed on 5 June 2021).
39. Yaru, C.; Zhijian, H.; Lujia, W. VisDrone-DET2021: The Vision Meets Drone Object detection Challenge Results. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 2847–2854.
40. Piotr Skalski. Available online: https://makesense.ai (accessed on 14 July 2019).
41. Zhang, H.; Tian, M.; Shao, G.; Cheng, J.; Liu, J. Target Detection of Forward-Looking Sonar Image Based on Improved YOLOv5. *IEEE Access* **2022**, *2022*, 3150339. [CrossRef]
42. Borui, J.; Ruixuan, L.; Jiayuan, M.; Tete, X.; Yuning, J. Acquisition of Localization Confidence for Accurate Object Detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 784–799.
43. Yang, H.; Chen, L.; Chen, M.; Ma, Z.; Deng, F.; Li, M.; Li, X. Detection of coal and gangue based on improved YOLOv5.1 which embedded scSE module. *Measurement* **2021**, *10*, 1016.
44. Glenn-Jocher. Available online: https://github.com/ultralytics/yolov5/issues/2180 (accessed on 12 February 2021).
45. Kasper-Eulaers, M.; Hahn, N.; Berger, S.; Sebulonsen, T.; Myrland, Q.; Kummervold, P.-E. Short Communication: Detecting Heavy Goods Vehicles in Rest Areas in Winter Conditions Using YOLOv5. *Algorithms* **2021**, *14*, 114. [CrossRef]
46. Changrui, C.; Yu, Z.; Qingxuan, L.; Shuo, W.; Xiaorui, W.; Xin, S.; Junyu, D. RRNet: A Hybrid Detector for Object Detection in Drone-captured Images. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Korea, 27–28 October 2019.
47. Ali, R.; Mohammed, R.; Sung-Ho, K. Convolutional Neural Network-Based Real-Time Object Detection and Tracking for Parrot AR Drone 2. *IEEE Access* **2019**, *7*, 69575–69584.