Article

ECAP-YOLO: Efficient Channel Attention Pyramid YOLO for Small Object Detection in Aerial Image

Munhyeong Kim ¹ , Jongmin Jeong ² and Sungho Kim ^{1,*}

¹ Advanced Visual Intelligence Laboratory, Department of Electronic Engineering, Yeungnam University, 280 Daehak-ro, Gyeongsan 38541, Korea; hyoung5778@yu.ac.kr

² Agency for Defense Development, 111 Sunam-dong, Daejeon 34186, Korea; jmjeong@add.re.kr

* Correspondence: sunghokim@yu.ac.kr; Tel.: +82-810-3530

Abstract: Detection of small targets in aerial images is still a difficult problem due to the low resolution and background-like targets. With the recent development of object detection technology, efficient and high-performance detector techniques have been developed. Among them, the YOLO series is a representative method of object detection that is light and has good performance. In this paper, we propose a method to improve the performance of small target detection in aerial images by modifying YOLOv5. The backbone is was modified by applying the first efficient channel attention module, and the channel attention pyramid method was proposed. We propose an efficient channel attention pyramid YOLO (ECAP-YOLO). Second, in order to optimize the detection of small objects, we eliminated the module for detecting large objects and added a detect layer to find smaller objects, reducing the computing power used for detecting small targets and improving the detection rate. Finally, we use transposed convolution instead of upsampling. Comparing the method proposed in this paper to the original YOLOv5, the performance improvement for the mAP was 6.9% when using the VEDAI dataset, 5.4% when detecting small cars in the xView dataset, 2.7% when detecting small vehicle and small ship classes from the DOTA dataset, and approximately 2.4% when finding small cars in the Arirang dataset.

Keywords: object detection; small target detection; channel attention

Citation: Kim, M.; Jeong, J.; Kim, S.

ECAP-YOLO: Efficient Channel Attention Pyramid YOLO for Small Object Detection in Aerial Image.

Remote Sens. **2021**, *13*, 4851.

<https://doi.org/10.3390/rs13234851>

Academic Editor: Mi Wang

Received: 18 October 2021

Accepted: 26 November 2021

Published: 29 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Research on the field of deep learning-based object detection has been steadily conducted [1]. Object detection is generally composed of multi-labeled classification and bounding box regression, and it can be divided into a one-stage detector and a two-stage detector. In the case of a one-stage detector, localization to determine the location of an object and classification to identify an object are simultaneously performed, and two-stage object detection [2–4] is a method that performs these two sequentially. In this respect, one-stage object detection has the advantage of being faster than two-stage object detection.

For aerial images, one image has a high resolution, and in order to use one aerial image in a deep learning network, the image must be segmented according to the input size of the deep learning network. A one-stage object detection technique is adopted because a relatively high speed is required to detect the increased number of images. There are various deep learning networks, such as SSD [5], EfficientDet [6], and YOLO series [7–10], for one-stage object detection.

Aerial detection is a challenging problem because it has the difficulty of detecting a target with a small number of pixels at a low spatial resolution, and the target is often similar to background information. In addition, the existing object detection method is optimized for detecting common objects such as the MS COCO dataset [11] and the PASCAL VOC dataset [12]. In this paper, we aimed to optimize a network to find a small target during aerial detection and to increase the detection rate by using channel attention.

For the DOTA dataset [13], VEDAI dataset [14], xView dataset [15], and Arirang dataset [16], the results of the existing YOLOv5 and the results of the proposed method were compared.

The contributions of this paper are as follows:

We apply Efficient Channel Attention network (ECA-Net) [17] to YOLOv5 and propose an efficient channel attention pyramid network.

Transposed convolution [18] was used instead of upsampling via the existing nearest neighbor interpolation [19]. Using nearest neighbor interpolation for small targets can result in relatively greater information loss than for large objects, so we optimally upsample using Transposed Convolution.

Added a detect layer for smaller object detection in existing YOLO. This enables aerial small object detection to detect smaller objects than conventional YOLO. In addition, the existing YOLO prediction layer is designed to predict large objects, medium-sized objects, and small objects. For aerial small object detection, unnecessary computing power consumption was reduced by eliminating the detect layer that finds large and intermediate objects.

The ECAP-YOLO and ECAPs-YOLO proposed in this paper showed a 6.9% improvement in the mean average precision (mAP) in the VEDAI dataset. In the xView dataset, the small car class showed an improvement of 5.4%. The DOTA dataset showed a 2.7% improvement for the small vehicle and small ship classes. The Arirang dataset showed a performance improvement of approximately 2.4% for small cars.

The remainder of the paper is organized as follows: Section 2 introduces related work, Section 3 describes the proposed technique, Section 4 discusses the experimental results on the dataset, and finally, Section 5 provides the conclusion and future work.

2. Related Work

Many studies on object detection have been performed [1]. Among them, YOLO is a representative one-stage object detection method. The YOLO series is currently represented in published papers up to YOLOv4, and YOLOv5 has been uploaded to GitHub. Section 2.1 describes the YOLO series, and Section 2.2 describes the attention module.

2.1. YOLO Series

The biggest difference between YOLO and the existing object detection at the time is that it has a structure in which the region proposal stage is removed and object detection is performed at once in the method divided into two stages, regional proposal and classification. After dividing the image into grid areas, it first predicts the bounding boxes corresponding to areas where objects are likely to be in each grid area. Next, the confidence indicating the reliability of the box is calculated. The network structure of YOLOv1 is based on GoogLeNet [20] and consists of 24 convolution layers and two fully connected layers.

YOLOv2 has been used to conduct various experiments to solve the low detection rate and considerable localization error of YOLOv1. First, the dropout [21] used in all of the convolution layers was removed, and batch normalization [22] was implemented. A fully connected layer was used as the last layer of YOLO, but in YOLOv2, this was changed to a convolution layer. This ensured that regardless of the input size, it does not matter. In YOLOv2, an anchor box is used, and training is stabilized by predicting the presence or absence of as many classes and objects as the number of anchor boxes. YOLOv2 proposes a new DarkNet-19.

From YOLOv3, three boxes with different scales are predicted. It predicts three boxes with different scales and extracts features in a similar way to feature pyramid networks. A few convolution layers are added, and the output is a 3D tensor. In this case, the bounding box uses multi-label classification and binary cross-entropy [23] loss instead of the existing SoftMax. The structure of the backbone uses Darknet-53. Training uses bounding box prediction L2 loss. Another aspect is that the feature map is taken from the beginning and the upsampled feature map and concat are performed. Using this method, semantic information (i.e., previous layer) and fine-grained information (i.e., initial layer)

can be obtained. This method is performed twice, and the same design is performed once more to predict the box of the final scale. Therefore, the prediction of the 3rd scale utilizes all previous layers and early segmented and meaningful information. The anchor box is clustered through k-means, and nine clusters and three scales are randomly selected to divide the clusters evenly.

The biggest features of YOLOv4 are Bag of Freebies [24] and Bag of Specials. The network backbone uses CSPDarkNet-53 [25] and uses the spatial attention module (SAM) [26], path aggregation network (PAN) [27], and cross-iteration batch normalization (CBN) [28]. Among them, SAM, PAN, and CBN were used with slight modifications, and augmentation was performed using mosaic augmentation.

In the case of YOLOv5, it is composed of a CBS block comprising a focus layer, convolution, batch normalization, SiLU [29], and a network of C3 blocks. The focus layer acts like a transformation from space to depth. The focus layer is shown in Figure 1. This serves to reduce the cost of the 2D convolution operation, reduce the space resolution, and increase the number of channels. The overall architecture is shown in Figure 2.

Figure 1. Focus layer.

Figure 2. YOLOv5 architecture.

2.2. Attention Module

Jie Hu et al. published the Squeeze-and-Excitation Network (SE-Net) [30]. The authors of this paper presented the idea of squeeze and excitation, and the SE-Net consists of a squeeze operation that summarizes the entire information of each feature map and an excitation operation that scales the importance of each feature. This is called the

squeeze and excitation block, and the model's performance improvement is very large compared to the parameter increase. This has the advantage that the model's complexity and computational burden do not increase significantly.

Qilong et al. proposed an Efficient Channel Attention Network [17], and the existing self-attention-based CNN model contributed to improving the performance, but the model complexity was high. In the case of SE-Net, avoiding dimensional reduction is important for training channel attention. We found that cross-channel interaction can significantly reduce the complexity of the model and maintain performance at the same time. In the case of the ECA-Net, the trade-off between performance and complexity was overcome, and a small number of parameters showed a clear performance improvement effect. They proposed a method for local cross-channel interaction without dimension reduction through efficient 1D convolution. The SE-Net is shown in Figure 3a, and the ECA-Net is shown in Figure 3b.

In the case of the ECA-Net, it is expressed as Equation (1), where k indicates the kernel size, $C1D$ indicates 1D convolution, and σ is a sigmoid function.

$$\omega = \sigma(C1D_k(y)) \quad (1)$$

Figure 3. (a) SE-Net and (b) ECA-Net architecture. Given the aggregated features obtained by global average pooling (GAP).

3. Proposed Methods

Three factors were considered for aerial small object detection. First, the detector should be optimized for small targets, and there should be no unnecessary detection. In addition, the small target should have little information loss at the deep learning network end. In this section, we propose the Efficient Channel Attention Pyramid YOLO (ECAP-YOLO) to satisfy this and describe the structure and contents of ECAP-YOLO after explaining the detect layer change, transposed convolution, and efficient channel attention pyramid.

3.1. Proposed Method Overview

The ECAP-YOLOv5 model was applied using EAC34, MEAP, and upsampling from the existing YOLOv5 to the transposed convolution. Then, we proposed two cases: ECAP-YOLO, where the detect layer detects intermediate objects, small objects, and smaller objects; ECAPs-YOLO, which consists of a detect layer that finds small objects and smaller objects. Section 3.2 describes the transposed convolution, Section 3.3 describes the efficient channel attention applied, Section 3.4 describes the efficient channel attention pyramid, and Section 3.5 describes the change detect layer. The overall architecture is shown in Figure 4.

(a) ECAP-YOLOv5.

(b) ECAPs-YOLOv5

Figure 4. This figure illustrates (a) ECAP-YOLOv5 and (b)ECAPs-YOLOv5.

3.2. Using Transposed Convolution

In this paper, we propose a method using transposed convolution instead of upsampling via the nearest neighbor interpolation method. The upsampling method using the nearest neighbor interpolation method is a method of upsampling that fills in the blanks using neighboring pixels. However, in the case of a small target, since the number of pixels is relatively small, the surrounding background information will be mixed if only upsampling is performed. Transposed convolution is a method designed to learn how to optimally upsample a network. Transposed convolution does not use a predefined interpolation method and enables learning during upsampling. It can learn how to perform it when upsampling the feature map. In this regard, the decoder of semantic segmentation restores the original image size. In this paper, transposed convolution was used to reduce information loss during upsampling of small targets in the feature map and to learn the upsampling method.

3.3. Efficient Channel Attention Applied

Efficient channel attention achieves performance improvement at a low cost. This was applied in YOLOv5 to determine the optimal location to improve performance. The efficient channel attention modified C3 block of YOLOv5 was applied and tested in a total of four areas, and the best performance was selected. This chapter introduces the efficient channel attention C3 (EAC3) blocks modified from the C3 block. In this paper, EAC34 was adopted and used. In EAC3_n, n indicates the number of times to repeat a block. In EAC3_F, F

does not use the bottleneck structure. EAC31, EAC32, EAC33, and EAC34 are classified according to the location of the ECA-Net.

First, the EAC31 block adds efficient channel attention just before the sum occurs in the bottleneck structure. The EAC3_F EAC3 block has the structure as shown in Figure 5.

Figure 5. Composition of the EAC31 block.

The second EAC32 uses efficient channel attention after the C3 module is finished. The EAC32 has the structure as shown in Figure 6.

Figure 6. Composition of the EAC32 block.

The third EAC33 is when efficient channel attention enters a part of the C3 module without a bottleneck. The EAC33 block has the structure as shown in Figure 7.

Figure 7. Composition of the EAC33 block.

In the last case of EAC34, efficient channel attention is used between the convolution modules in the bottleneck structure. According to the test results in Section 4.1, ECA4 recorded the highest mAP, and EAC34 was adopted. EAC34 has the structure as shown in Figure 8.

Figure 8. Composition of the EAC34 block.

3.4. Efficient Channel Attention Pyramid

The existing YOLOv5 uses the spatial pyramid pooling (SPP) network [31]. In this paper, we aimed to improve SPP using channel attention. First, we propose the efficient channel attention pyramid (EAP). We create a pyramid model by applying the ECA-Net instead of the existing max pooling layer. As for the kernel size, it was the same as the SPP layer of the existing YOLOv5, and the kernel size was set to 5, 9, and 13 and then the CBS layer was performed by concatenating. EAP has the structure as shown in Figure 9.

Second, we propose the max pooling efficient channel attention pyramid network (MEAP). In the case of MEAP, global max pooling (GMP) is used instead of global average pooling in efficient channel attention. As for the kernel size, it was the same as the SPP layer of the existing YOLOv5, and the kernel size was set to 5, 9, and 13 and then the CBS layer was performed by concatenating. EAP has the structure as shown in Figure 10.

Figure 9. (a) Composition of the EAP architecture and the (b) EAC layer.

Figure 10. (a) Composition of the MEAP architecture and the (b) MEAC layer.

Third, we propose the atrous efficient channel attention pyramid network (AEAP). In the case of atrous spatial pyramid pooling [32], unlike general convolution, atrous convolution is used. In the case of atrous convolution, there is a space between the kernels. This provides a wider receptive field at the same computational cost as convolution, and it can be confirmed that features are extracted more densely and prominently. Atrous efficient channel attention was used. After global average pooling in the existing efficient channel attention, the kernel size of the 1D convolution was fixed, and the stride was different to create a pyramid structure. The EAP has the structure as shown in Figure 11.

Figure 11. (a) Composition of the AEAP architecture and the (b) AEAC layer.

MEAP was selected because it was confirmed that the highest performance was obtained when MEAP was applied as a result of the experiment described in Section 4.1 of this paper.

3.5. Change Detect Layer

In YOLO5, there are 3 detect layers in prediction. They detect small objects, middle objects, and large objects, respectively. Aerial object detection is not needed for mid-size and large objects, so eliminating them reduces unnecessary power usage. A detect layer was added in the detect layer to detect very small objects. If a detect layer is added, detection is performed in a larger grid cell, allowing for the detection of smaller objects. This increases the number of computations compared to the existing detector model but enables the detection of smaller objects.

4. Experimental Results and Discussion

In this paper, train methods such as loss function and parameters related to learning were performed in the same way as in the existing YOLOv5. For experimental data, VEDAI dataset, xView dataset, DOTA dataset, and Arirang dataset are used. Figure 12 depicts an example of the YOLOv5 learning method. In this section, after explaining the characteristics and composition of each dataset, we test the existing YOLOv5 and the proposed method, and organize and analyze the results. YOLOv5 has four models: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. The models are split into depth multiple and width multiple. The YOLOv5s depth multiple is 0.33 based on YOLOv5l, and the width multiple is 0.50 times. YOLOv5m has a depth multiple of 0.67 and a width multiple of 0.75 based on YOLOv5l. For YOLOv5x, depth multiple is 1.33 times, and the width multiple is 1.25 times, again based on YOLOv5l. The YOLOv5 that was used for this paper was the YOLOv5l model. For training the parameters, the same methods and parameters from the existing YOLOv5 were applied.

As for the evaluation method, the evaluation proceeded with precision, recall, F1-score, and mAP. Precision and recall were calculated using Equation (2). TP , FP , TN , and FN stand for true positive, false positive, true negative, and false negative. The F1-score was calculated using Equation (3).

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{All\ Detections} \quad Recall = \frac{TP}{TP + FN} = \frac{TP}{All\ Ground\ truths} \quad (2)$$

$$F1 - Score = 2 \times \frac{1}{\frac{1}{Precision} + \frac{1}{Recall}} = 2 \times \frac{Precision \times Recall}{All\ Ground\ truths} \quad (3)$$

Figure 12. Example of the training for YOLOv5.

4.1. VEDAI Dataset

The VEDAI dataset [14] consists of a total of 1200 images, 3757 instances, a spatial resolution of 25 cm at a 512×512 resolution, an object size of 8–20 pixels in width, and a total of 3757 instances. In the experiment, the dataset consisted of nine classes (i.e., car, truck, pickup, tractor, camper, boat, van, plane, and other instances) except these were set to other. Nine hundred and ninety-eight training data, 124 validation data, and 124 tests data were randomly divided. The VEDAI dataset was tested with RGB data with a 512×512 resolution.

The detector was modified to optimize YOLOv5 for a small target. In Table 1, the performance according to the change of the detect layer can be observed, which was different from the existing YOLOv5. Here, the detection layer of YOLOv5 was changed to a total of three. First, a detection layer was added to YOLOv5 to detect very small, small, medium,

and large objects. Second, after adding a detect layer, the large object detect layer was removed and configured as shown in Figure 4a, and the experiment was conducted. Third, the medium detect layer was removed from the second method, and two detect layers were configured as shown in Figure 4b and then tested. As a result of the experiment, it can be seen that the second method in YOLOv5 showed the best results with a mAP of 29.8%.

Table 1. Performance according to the addition and removal of detect layers in YOLOv5.

Model	Precision (%)	Recall (%)	F1-Score	mAP50	mAP50:95
YOLOv5l	50.3	64.4	56.5	52.5	27.6
YOLOv5l + 1 more Layer + XS, S, L, XL detect	60.1	51.4	55.4	51.4	29.5
YOLOv5l + 1 more Layer + XS, S, L detect	63.3	53.4	57.9	55.5	29.8
YOLOv5l + 1 more Layer + XS, S detect	62.0	52.8	57.0	52.8	26.8

Table 2 shows the performance results for each location of attention. When EAC3, EAC32, EAC33, and EAC34 were tested, mAP50 showed the best results for EAC34, and it was applied.

Table 2. Performance according to the addition and removal of detect layers in YOLOv5.

Model	Precision (%)	Recall (%)	F1-Score (%)	mAP50 (%)	mAP50:95(%)
YOLOv5l	50.3	64.4	56.5	52.5	27.6
YOLOv5l + EAC31	66.6	46.5	54.8	51.4	28.2
YOLOv5l + EAC32	61.7	55.8	58.6	56.3	33.1
YOLOv5l + EAC33	57.6	55.9	56.7	52.9	28.7
YOLOv5l + EAC34	73.0	49.9	59.27	56.4	29.5

Table 3 shows the performance of each attention pyramid that was tested by applying EAP, AEAP, and MEAP to the existing YOLOv5. In the case of EAP, as shown in the Figure 9, it is a method that simply stacks channel attention with different kernel sizes as pyramids, and the kernel sizes were 5, 9, and 13. In the case of AEAP, as shown in the Figure 11, convolutions dilated to the sizes of two, four, and six were used while maintaining the kernel size at three. In the case of AEAP2, the kernel size was maintained at three and the dilated convolutions of sizes 5, 9, and 13 were used. MEAP, as shown in Figure 10, was the same as the existing EAP, but instead of global average pooling, global max pooling and 1D convolutions with the kernel sizes of 5, 9, and 13 were used. Table 3 shows that AEAP had the best mAP50.

Finally, the results according to the fusion of the proposed techniques were tested. Table 4 shows the results. Experiments were carried out using the existing YOLOv5l and YOLOv5x, transposed convolution, using the attention pyramid, adding a detect layer to find small objects, removing the detect layer to find large objects, and finally, finding a medium-sized object in the third case. This was the case when the detect layer was removed. Table 4 shows that ECAP-YOLOv5l had the best mAP50. Compared to the existing YOLOv5l, the mAP50 of ECAP-YOLOv5l increased by 6.2% and that of ECAPs-YOLOv5l by 3.5%.

Table 3. Performance according to the addition and removal of detect layers in YOLOv5.

Model	Precision (%)	Recall (%)	F1-Score (%)	mAP50 (%)	mAP50:95 (%)
YOLOv5l	50.3	64.4	56.5	52.5	27.6
YOLOv5l + EAP	59.2	57.9	58.5	56.0	31.5
YOLOv5l + AEAP	53.5	63.9	58.2	56.2	31.2
YOLOv5l + AEAP2	61.0	54.1	57.34	54.0	30.8
YOLOv5l + MEAP	65.2	51.9	58.0	54.2	29.6

Table 4. Results by option in ECAP-YOLOv5.

Model	Precision (%)	Recall (%)	F1-Score (%)	mAP50 (%)	mAP50:95 (%)
YOLOv5l	50.3	64.4	56.5	52.5	27.6
YOLOv5x	64.0	52.7	57.8	55.9	30.8
YOLOv5l + Tconv	65.4	52.1	58.0	56.8	31.9
ECAPs-YOLOv5l (EAP)	63	47.7	54.3	52.6	29.4
ECAPs-YOLOv5l (AEAP)	62.8	60.1	61.4	55.8	32.4
ECAPs-YOLOv5l	64.6	51.4	57.2	56	31.5
ECAP-YOLOv5l	71.6	52.9	60.8	58.7	34.5

Class-wise accuracy and mean average precision for the VEDAI dataset are shown in Table 5. For cars, when AEAP was used, it showed the highest mAP50 at 86.7%. For trucks, YOLOv5x was 59.4%, indicating the highest mAP50. Pickups showed the highest mAP50 with ECAP-YOLOv5l at 77%. Tractors and campers showed the highest mAP50s at 72% and 61.3% for ECAPs-YOLOv5l using AEAP. For boats, ECAP-YOLOv5l was 35%, indicating the highest mAP50. For vans, the case of upsampling with transposed convolution in YOLOv5 was 54.7%, indicating the highest mAP50. For planes, ECAPs-YOLOv5l using EAP showed the highest mAP50 at 92%. In the other class, YOLOv5x showed the highest mAP50 with 40.7%.

Table 5. Class-specific accuracy and mean average precision (mAP50) in the VEDAI dataset.

Model	All (%)	Cars (%)	Trucks (%)	Pickups (%)	Tractors (%)	Campers (%)	Boats (%)	Vans (%)	Planes (%)	Other (%)
YOLOv5l	52.5	79.5	43.3	66.2	63.2	53.3	26.4	45.1	63.4	32.1
YOLOv5x	55.9	83.6	59.4	67.6	62.8	49.8	24.3	53.5	61.5	40.7
YOLOv5l + Tconv	56.8	83.6	50.1	67.3	56.2	53.4	31.7	54.7	78.5	36.0
ECAPs-YOLOv5l (EAP)	52.6	83.8	34.3	74.1	47.2	52.9	33.7	30.2	92.0	25.4
ECAPs-YOLOv5l (AEAP)	55.8	86.7	30	75.6	72	61.3	25.3	54.7	75.9	20.8
ECAPs-YOLOv5l	56	85.5	40.4	74.8	61.8	54.7	32.7	39.8	90.1	23.9
ECAP-YOLOv5l	58.7	85.6	52	77	62.7	50.3	35	47.8	91	27.2

The proposed method was compared with the other methods and is shown in Table 6. For comparison, YOLOR [33] and YOLOX [34] were used. In GFLOP, YOLOv5l was the smallest, but ECAPs-YOLOv5l had the smallest parameter with 33.6M. In the case of recall, YOLOv5l was the highest, but it was confirmed that the precision, F1-Score, mAP50, and mAP 50:95 showed high performance in ECAP-YOLOv5l.

Table 6. Comparison with other methods.

Model	Precision (%)	Recall (%)	F1-Score (%)	mAP50 (%)	mAP 50:95 (%)	GFLOPs	Parameters (M)
YOLOv5l	50.3	64.4	56.5	52.5	27.6	115.6	47.0
YOLOv5x	64.0	52.7	57.8	55.9	30.8	219.0	87.7
YOLOR_CSP	20.2	47.2	28.3	31.4	19.5	120.637	52.9
YOLOR_CSP_x	23.9	56.1	33.5	34.7	21.1	222.43	99.7
YOLOX-l	28.7	49.1	36.2	55.6	28.8	155.6	54.2
YOLOX-x	21.22	59.9	31.3	53.4	26.6	281.9	99.1
ECAPs-YOLOv5l	64.6	51.4	57.2	56	31.5	122.1	33.6
ECAP-YOLOv5l	71.6	52.9	60.8	58.7	34.5	132.4	36.8

Shown in Figure 13, YOLOv5l detected objects that were not found and performed more accurate classification. In the first image, an object misclassified as a pickup was found to be a boat, and the object was found. The second image performed additional plane detection, and the third image performed accurate classification of cars and additional detection of vans.

Figure 13. Cont.

(3)

Figure 13. Comparison of the detection results of YOLOv5l (a) and ECAP-YOLOv5l (b) in the VEDAI dataset.

4.2. xView Dataset

The xView dataset [12] was taken with the WorldView-3 satellite and has a ground sample distance of 0.3 m. The xView data set consisted of a total of 60 classes and over 1 million objects. In this paper, only one small car class out of 60 classes was ascertained to detect small targets. The image was cropped to a 640×640 size with a 10% overlap. As a result of the experiment, the proposed method, as shown in Table 7, showed a 9.4% and 8% improvement in mAP50 compared to the existing YOLOv5l.

Table 7. Performance comparison with xView dataset.

Model	Precision (%)	Recall (%)	F1-Score (%)	mAP50 (%)	mAP 50:95 (%)
YOLOv5l	76.8	70.4	73.5	73.9	29.2
YOLOv5x	78.5	74.5	76.4	79.3	31.9
ECAP-YOLOv5l	82.0	78.5	80.2	81.9	34.6
ECAPs-YOLOv5l	83.0	78.8	80.8	83.3	35.9

From the images in Figure 14(a1,b1), it can be seen that the proposed method detected the difference on the left side of the image better. In addition, it can be confirmed that it well-detected vehicles that were in the middle. In Figure 14(a2,b2), it was confirmed that the detection was better for the aligned vehicles. In Figure 14(a3,b3), the existing YOLOv5l missed a part with dense cars, but it was confirmed that the proposed method detected dense objects better. In Figure 14(a4,b4), it was also confirmed that the proposed technique detected dense objects better.

Figure 14. Comparison of the detection results for YOLOv5l (a) and ECAP-YOLOv5l (b) with the xView dataset.

4.3. DOTA Dataset

The DOTA images are collected from the Google Earth, GF-2 and JL-1 satellite provided by the China Centre for Resources Satellite Data and Application, and aerial images provided by CycloMedia B.V. DOTA consists of RGB images and grayscale images. The RGB images are from Google Earth and CycloMedia, while the grayscale images are from the panchromatic band of GF-2 and JL-1 satellite images.

For the DOTA dataset [9], grayscale images were from the panchromatic bands of the GF-2 and JL-1 satellite images. The DOTA dataset was from 800×800 to $20,000 \times 20,000$ pixels, and DOTAv1.5 was used in this paper. DOTAv1.5 uses the same number of images as DOTAv1.0. DOTAv1.0 consists of 1441 training datasets and 458 validation datasets. The data included a total of 403,318 instances and had 16 classes, i.e., plane, ship, storage tank, baseball diamond, tennis court, basketball court, ground track field, harbor, bridge, large vehicle, small vehicle, helicopter, roundabout, soccer ball field, swimming pool, and container crane. The existing train set was used as it was, and the validation set was randomly divided in half to make 229 for the validation set and the test set each. Only two classes (i.e., small vehicle and ship) were used to detect small targets. For training, the data were cropped by 640 pixels horizontally and 10% overlap. The training results are shown in Table 8. For ECAPs-YOLOv5l, the mAP increased by 2.1% compared to YOLOv5l. ECAP-YOLOv5 increased by 2.7% compared to YOLOv5l.

Table 8. Performance comparison with DOTA dataset.

Model	Precision (%)	Recall (%)	F1-Score (%)	mAP50 (%)	mAP 50:95 (%)
YOLOv5l	85.2	70.3	77.0	77.2	49.3
YOLOv5x	82.7	74.8	78.6	79.8	50.1
ECAP-YOLOv5l	80.3	79.3	79.8	81.5	52.0
ECAPs-YOLOv5l	78.8	79	78.9	81.1	51.4

As shown in Figure 15(a1,b1), YOLOv5ls was not detected, but ECAP-YOLOv5s was detected. In the remaining images in Figure 15, it can be seen that the proposed method detected dense objects well.

In Table 9, the mAP increased by 3.9% for ECAPs-YOLOv5l and 4.3% for ECAP-YOLOv5l compared to YOLOv5l for automobiles. For ship, ECAPs-YOLOv5l decreased by 0.3% and ECAP-YOLOv5l increased by 1.2%.

Table 9. Class-specific accuracy and mean precision (mAP50) with the DOTA dataset.

Model	Class	Precision (%)	Recall (%)	F1-Score (%)	mAP50 (%)	mAP 50:95 (%)
YOLOv5l	Car	71.8	65.5	68.5	68.6	36.8
	Ship	93.7	83.2	88.1	90.9	65.2
YOLOv5x	Car	77.4	57.5	65.9	63.7	34.4
	Ship	93.0	83.2	87.8	90.7	64.2
ECAP-YOLOv5l	Car	65.7	75.1	70.1	71.6	38.7
	Ship	94.9	83.5	88.8	91.4	65.4
ECAPs-YOLOv5l	Car	63.6	76.0	69.2	71.3	38.3
	Ship	93.9	81.9	87.5	90.9	64.5

Figure 15. Comparison of the detection results of YOLOv5l (a) and ECAP-YOLOv5l (b) with the DOTA dataset.

4.4. Arirang Dataset

In the case of Arirang dataset [13], it was provided as a processed patch (1024×1024 size) by pre-processing the original satellite image as a dataset created using the Arirang 3/3A satellite image dataset. There were a total of 16 classes, including small ship, large ship, civilian aircraft, military aircraft, small car, bus, truck, train, crane, bridge, oil tank, dam, athletic field, helipad, and roundabout. It consisted of 132,436 objects, and the number of objects by class is shown in Table 10.

Table 10. Number of objects per class in the Arirang dataset.

Small Cars	Trucks	Small Ships	Trains	Buses	Military Aircrafts	Oil Tankers	Others
86,305	19,011	13,107	5123	3840	937	868	953
Civilians	Aircrafts	Large Ships	Cranes	Helipads	Roundabouts	Bridges	Dams
424	306	260	135	1131	93	40	3

In this paper, only small car was used among the data for small target detection, and 1024×1024 of the input image was learned. The experimental results for the Arirang dataset are shown in Table 11. The mAP50 improved by 4.4% in ECAPs-YOLOv5l and 4.8% in ECAP-YOLOv5l compared to YOLOv5l. The mAPs were 2.4% in ECAPs-YOLOv5 and 2% in ECAP-YOLOv5l than in YOLOv5l.

Table 11. Performance comparison with the Arirang dataset.

Model	Precision (%)	Recall (%)	F1-Score (%)	mAP50 (%)	mAP50:95 (%)
YOLOv5l	65.6	45.7	53.9	42.8	15.2
YOLOv5x	66.0	47.2	55.0	43.8	15.7
ECAP-YOLOv5l	72.9	47.7	57.7	47.6	17.6
ECAPs-YOLOv5l	68.9	49.3	57.5	47.2	17.2

It can be seen that the proposed method in the image shown in Figure 16 detected small cars better. In images in Figure 16(a1,b1), it can be seen that the alignment difference in the upper left corner was better detected. In the Figure 16(2,3) images, the existing YOLOv5l missed the small car part, but it was confirmed that the proposed method detected dense objects better.

4.5. Discussion

We would like to discuss the experimental results. First, it can be seen in Table 1 that the mAP increased for aerial small target detection when the Vedai dataset was changed to the method proposed by the existing YOLOv5 detect layer. By applying the ECA-Net to YOLOv5, it was confirmed that an mAP50 of 55.5% was obtained. It can be seen in Table 2 that the highest mAP of 56.4% was obtained when it was arranged as shown in Figure 8. As shown in Table 3, it was confirmed that the highest mAP was obtained when AEAP was applied. However, when comparing ECAP-YOLOv5l with other attention pyramids, it was confirmed, as shown in Table 4, that MEAP showed the highest mAP. The proposed method was compared with other methods. This can be seen in Table 6. For recall, it was higher for YOLOv5l than for ECAP-YOLOv5l. Precision was lower with YOLOv5l. This indicates that YOLOv5l found more truth but had lower precision, resulting in more false positives. However, it can be seen that ECAP-YOLOv5l showed better performance when mAP is considered. Comparing Figure 13(a3,b3), van was not found in YOLOv5l, but was found in the proposed method. Also, in the proposed method that YOLOv5l recognized as Pickup Truck, it was recognized as a car, so it can be seen that it shows better performance in recognition.

Figure 16. Comparison of the detection results of YOLOv5l (a) and ECAP-YOLOv5l (b) with the Arirang dataset.

5. Conclusions and Future Works

In this paper, we introduced the YOLO series and modified the YOLOv5 network for aerial small target detection. As for the modification method, the backbone was changed by applying the first efficient channel module, and various channel attention pyramids were proposed. The channel attention pyramid was proposed, and the efficient channel attention pyramid network, the atrous efficient channel attention pyramid network, and the max pooling efficient channel attention pyramid network were tested. The channel attention pyramid was proposed, and the efficient channel attention pyramid network, the atrous efficient channel attention pyramid network, and the max pooling efficient channel attention pyramid network were tested. Second, in order to optimize the detection of small objects, the module for detecting large objects was eliminated, and a detect layer was added to find smaller objects, thereby reducing the computing power used for detecting small targets and improving the detection rate. Finally, upsampling was performed using transposed convolution instead of upsampling using existing nearest neighbor interpolation.

As a result of the proposed method, the mAP was 6.9% with the VEDAI dataset, 6.7% in the case of small cars in the xView dataset, 2.7% in the small vehicle and small ship class with the DOTA dataset, and approximately 2.4% in the small car class in the Arirang dataset, which shows an improvement in performance.

The proposed technique is an EO-based object detection technique for aerial images. EO has difficulty detecting objects at night, in fog, and with clouds. However, if various heterogeneous sensors, such as infrared radiation sensors, synthetic aperture radar sensors, LiDAR sensors, and hyperspectral sensors, are used together, it is expected to show not only strong performance in the aforementioned environment and also better performance overall.

Author Contributions: Conceptualization, M.K.; methodology, M.K.; software, M.K.; formal analysis, M.K.; investigation, M.K.; writing—original draft preparation, M.K.; writing—review and editing, S.K. and J.J.; visualization, M.K.; supervision, S.K.; project administration, S.K. and J.J.; funding acquisition, S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Research Foundation of Korea (NRF) (grant number: NRF-2018R1D1A3B07049069), the Institute of Industrial Technology (KIAT) (grant number: P0008473), and the Agency for Defense Development (grant number: UD200005F).

Informed Consent Statement: Not applicable.

Data Availability Statement: Vehicle Detection in Aerial imagery (VEDAI) dataset (<https://downloads.greyc.fr/vedai/>, accessed on 25 June 2021), DOTA dataset (<https://captain-whu.github.io/DOTA/dataset.html>, accessed on 25 June 2021), Arirang dataset (<https://dacon.io/en/competitions/open/235644/overview/description>, accessed on 25 June 2021).

Acknowledgments: This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2018R1D1A3B07049069). This (performance) research was conducted with the support of the Korea Institute of Industrial Technology (KIAT) with the funding of the Ministry of Trade, Industry and Energy's "Industrial Professional Capacity Enhancement Project" (Project number: P0008473). This research was carried out with research funding support from the Agency for Defense Development (contract number UD200005F).

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

References

1. Zou, Z.; Shi, Z.; Guo, Y.; Ye, J. Object detection in 20 years: A survey. *arXiv* **2019**, arXiv:1905.05055.
2. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp 580–587.
3. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.

4. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [\[CrossRef\]](#)
5. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
6. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2020), Washington, DC, USA, 14–19 June 2020; pp. 10781–10790.
7. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
8. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
9. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
10. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
11. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Doll, Z.; Lawrence, C. Microsoft coco: Common objects in context. In Proceedings of the 13th European Conference on Computer Vision, Zurich, Switzerland, 1–6 September 2014.
12. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [\[CrossRef\]](#)
13. Xia, G.S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A large-scale dataset for object detection in aerial images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3974–3983.
14. Razakarivony, S.; Jurie, F. Vehicle detection in aerial imagery: A small target detection benchmark. *J. Vis. Commun. Image Represent.* **2016**, *34*, 187–203. [\[CrossRef\]](#)
15. Lam, D.; Kuzma, R.; McGee, K.; Dooley, S.; Laielli, M.; Klaric, M.; Bulatov, Y.; McCord, B. Xview: Objects in context in overhead imagery. *arXiv* **2018**, arXiv:1802.07856.
16. Arirang Satellite Image AI Object Detection Contest. Available online: <https://dacon.io/competitions/open/235644/overview/description> (accessed on 25 June 2021).
17. Wang, Q.L.; Wu, B.G.; Zhu, P.F.; Li, P.H.; Zuo, W.M.; Hu, Q.H. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11531–11539.
18. Dumoulin, V.; Visin, F. A guide to convolution arithmetic for deep learning. *arXiv* **2016**, arXiv:1603.07285.
19. Rukundo, O.; Cao, H. Nearest neighbor value interpolation. *arXiv* **2012**, arXiv:1211.1768.
20. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
21. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
22. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on Machine Learning (ICML), Lille, France, 6–11 July 2015; Volume 37, pp. 448–456.
23. Zhang, Z.; Sabuncu, M.R. Generalized cross entropy loss for training deep neural networks with noisy labels. In Proceedings of the 32nd Conference on Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018.
24. Zhang, Z.; He, T.; Zhang, H.; Zhang, Z.; Xie, J.; Li, M. Bag of freebies for training object detection neural networks. *arXiv* **2019**, arXiv:1902.04103.
25. Wang, C.Y.; 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. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 390–391.
26. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
27. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.
28. Yao, Z.; Cao, Y.; Zheng, S.; Huang, G.; Lin, S. Cross-iteration batch normalization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 12331–12340.
29. Elfving, S.; Uchibe, E.; Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Netw.* **2018**, *107*, 3–11. [\[CrossRef\]](#) [\[PubMed\]](#)
30. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [\[CrossRef\]](#) [\[PubMed\]](#)
32. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [\[CrossRef\]](#) [\[PubMed\]](#)

-
33. Wang, C.Y.; Yeh, I.H.; Liao, H.Y. You only learn one representation: Unified network for multiple tasks. *arXiv* **2021**, arXiv:2105.04206.
 34. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.