On Improving the Accuracy of Object Detection for High Resolution Images Based on SSD

Kei Irie, Qiu Yicheng and Kiyoshi Nishikawa
Graduate School of Systems Design,
Tokyo Metropolitan University, Japan
E-mail: kiyoshi@tmu.ac.jp Tel: +81 42585 8423

***Abstract*— In this paper, we consider improving the accuracy of object detection using SSD (Single Shot multibox Detector). SSD is well known that it can execute the searching of region candidate and classifying the objects in a single process. One of the problems of the SSD is that if the object is not larger than a certain size, the accuracy will decrease. To solve this problem, we propose a method to improve the accuracy of the object detection by extending a conventional method. Although the proposed method applies the SSD after the segmentation of the image as in the conventional one, the difference is that SSD is performed using only partial information from the network. This allows us to detect small objects that could not be detected using the standard SSD. Then, region metrics are corrected using the result of detection without image segmentation and the result after image segmentation. The effectiveness of the proposed method is shown through the computer simulations.**

I. Introduction

Object detection is one of the fundamental and challenging tasks in computer vision, which is the task of detecting the location of an object in an image and classifying it as one of predefined categories [1], [2]. Object detection has been actively studied for several decades. With the advent of deep learning technology, the accuracy of object detection has improved. Currently, R-CNN (Regions with Convolutional Neural Network) [3], FastR-CNN [4], and FasterR-CNN [5] are known as the basic algorithms for object detection using deep learning. These networks use CNNs for feature extraction, which enables object detection with higher accuracy than conventional methods. They are based on a two-step prediction process: first, it searches for candidate regions where an object is likely to appear, and then, it classifies the image to determine whether it is an object or not. This type of algorithms is called two-stage, which is characterized by a relatively slower detection speed but a higher detection accuracy. On the other hand, there is also an algorithm called one-stage, which is another type of algorithms that executes two processes at once: searching of region candidate and classification, and thus has the feature of faster detection speed compared to two-stage. YOLO (You Only Look Once) and SSD (Single Shot multibox Detector) are examples of one-stage algorithms.

In this paper, we assume to use SSD for object detection. The problem we consider is the reduced detection accuracy of SSD for small objects. In this paper, the definition of a small object is one that occupies less than 1.2% of the area of the image. This value (1.2%) is derived from the results of experiments shown in II.B. When processing images using SSD, it is necessary to convert the resolution of the image to that of the images used for training the SSD. Due to this preprocessing, the sizes of objects in the image will be transformed, and the feature values of each object in the image will also be transformed. Besides, images are increasingly being recorded as high resolution such as so called 4K (4096 x 2160 pixels) or 8K (7680 x 4320 pixels). When those images are applied for SSD, their height and width are resized by one fifth or more. As a result, the area of each object in the image will be decreased, and the number of objects, which are categorized in our definition of small, increase. On the other hand, if SSD is trained directly using high-resolution images, the learning effect is degraded due to the increase in the number of parameters in the neural network, which leads to the problem of requiring long training times.

In this paper, we propose a method to improve the detection accuracy of small objects using the SSD by improving the previous method by Qiu Yicheng [6]. There are some differences between the method in [6] and the proposed method. Namely, the proposed method uses the observation of the relationship between the object area and the detection accuracy of SSD for the image segmentation method. Then, SSD is performed for segmented images using only partial information from the network. Also, we propose a method to mix the region metric for the whole image as well. Through the computer simulations, we show the effectiveness of the proposed method.

The paper is organized as follows: Section II summarizes the configuration of SSD and relation between object area and detection accuracy in SSD. Section III describes the previous method [6]. In section IV, the proposed method is described, and Section V shows the results of experiments using the proposed method.

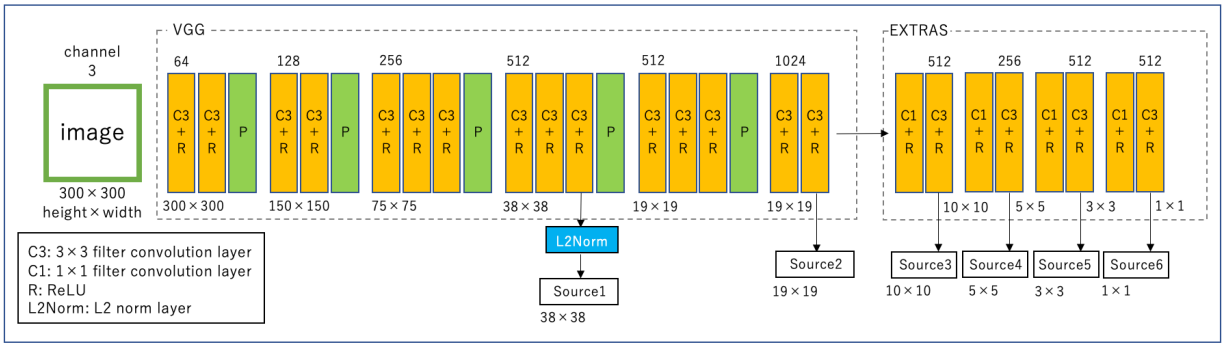

Fig.1: Components of the SSD network.

II. Configuration of SSD and relation between object area and detection accuracy in SSD

A. *Configuration of SSD*

The network configuration of SSD is shown in Fig.1. SSD performs classification and region estimation using six outputs from the network which undergone different number of convolutions. We refer to these six outputs as Source 1 to 6.

In the SSD network, the input image is first fed into the base network VGG-16 [7], which is referred as VGG in the following. The data that has undergone 10 convolutions in VGG is extracted separately, after normalized by the L2Norm layer, as the output Source 1. The output of the VGG module is then set to the output Source 2. At the same time, the output of VGG is inputted to the EXTRAS module of Fig.1, and after every two convolutions, the outputs are set as Source 3 to 6 respectively. The sizes of the feature maps for these outputs Source 1 to 6 are 38×38, 19×19, 10×10, 5×5, 3×3, and 1×1, respectively. Using Source 1, SSD searches an object in each sub region whose size is 1/38 of the whole image. On the other hand, Source 6 searches an object that spans over the whole image. Those differences in the sizes of the objects to be detected make it possible to detect objects of a variety of sizes in an image.

However, the differences in the number of convolutions which Source 1 to 6 undergo may cause a problem of SSD. Source 6 is the output of the deepest layer of the network as shown in Fig. 1, and it undergoes convolution 23 times in total, while Source 1 undergoes convolution only 10 times in total. Because of this difference, SSD may fail to extract features in small regions using Source 1 or Source 2, and this tends to reduce the detection accuracy of relatively small objects in the image compared to that of large objects.

B. *Relation between object area and detection accuracy in SSD*

Here, we show that the object size that can be detected differs for each Source, and the object size is determined by the ratio to the size of the image. So, we experimentally investigate how much area is required for SSD to be able to detect an object. Because the required area of an object to be detected is different for Source 1 to 6, we investigated the relation between the area of an object and the detection accuracy for each Source 1 to 6.

The SSD used in this study was trained to detect 20 classes of objects (e.g, aero plane, bird, boat, etc.) and an additional background class, and hence, a total of 21 classes. In order to experimentally see how much area Source 1 to 6 require for detecting an object in the center of the background image, we generated images which contain only one object, and the size of the object was varied as shown in Fig.2. In this figure, examples of target images generated by mixing the background and the target object, in this example, a horse, are shown. These target images were generated for all 20 classes.

By applying the SSD to all these images, we investigated how much area is required to detect an object by using one of Source 1 to 6. Table I shows the results of when Source 1 or 2 was applied.

In this table, the column ‘Source 2’ shows the ratio of the maximum area to the original image size when the detection accuracy becomes greater than 0.6 when Source 2 is used for each class. On the other hand, ‘Source 1’ in Table I shows the minimum area as a percentage of the original image size when the detection accuracy is greater than 0.5. Besides, the bottom of Table I shows the average and median values. From the table, it is possible to roughly predict how much areas are required to detect an object using Source 1 or Source 2. Note that, in Table I, “Nan” (Not a number) represents the case where there was no result exceeded the detection accuracy.

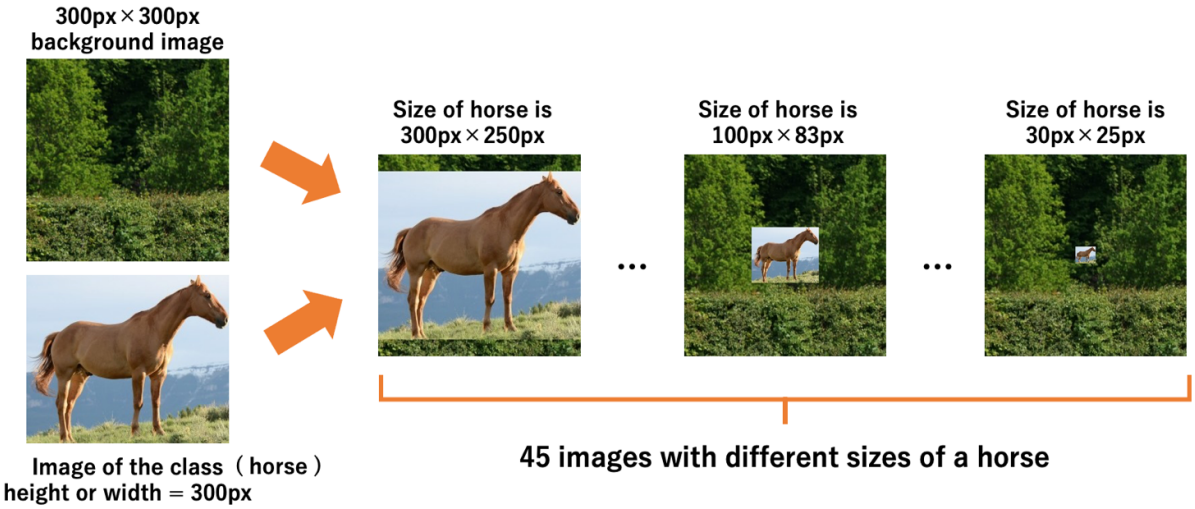

Fig.2: Generation of pseudo target images for determining the required object size for detection using each Source

Table I: Relation between detection accuracy and object area for Source 1 and Source 2.

	Source2 (%)	Source1 (%)
aeroplane	14.69	2.10
bicycle	16.67	1.50
bird	13.60	2.82
boat	12.27	1.47
bottle	3.67	0.88
bus	6.17	Nan
car	6.80	0.75
cat	12.50	1.18
chair	13.60	1.58
cow	14.36	0.70
diningtable	Nan	Nan
dog	14.69	1.52
horse	14.44	1.27
motorbike	15.00	1.10
person	7.08	Nan
pottedplant	11.81	1.00
sheep	14.36	0.70
sofa	6.94	Nan
train	11.64	Nan
tv monitor	19.19	2.20
Median	13.05	1.225
Average	11.93	1.33

III. Previous method [6]

In general, the input image to the SSD is decimated to be the same size as the images used to train the SSD, namely, 300×300 px or 500×500px. As a result, the sizes of the objects in the image are also reduced, which affects the accuracy of detection. The larger the size of the input image, the larger the reduction ratio. In [6], a method was proposed to improve the detection accuracy of smaller objects. Instead of reducing the image size, segmentation of the image is performed, and SSD is applied to each segmented region. However, if the image is simply segmented, the detection accuracy of objects spanning multiple regions deteriorates. For this reason, [6] proposes a method that combines two types of segmentation which we call Type I and Type II. (Note that in [6], they are referred as Normal and Misaligned segmentations.) The results are combined using a region metric correction method. The proposed method is a modification of [6].

A. *Segmentation Type I*

Here, we describe Type I segmentation method. Set the target size $H_{target} \times W_{target}$ and segment the image evenly according to the target size. This results in several segmented sub-images. In [6], SSD is applied to all these sub-images independently where Source 1 to 6 are used. The result of Type I will have a boundary problem which is segmented as the failure to detect an object that spans the boundary of segmented areas as shown in Fig.3. Therefore, in [6], Type II segmentation method is also proposed.

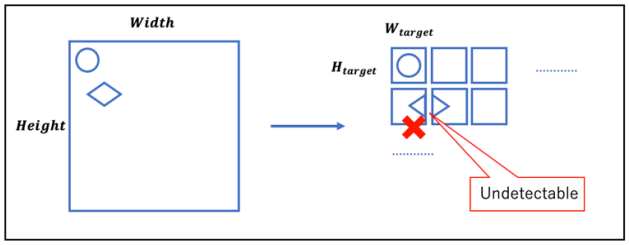

Fig.3: Type I segmentation method

B. *Segmentation Type II*

In Type I, objects near the boundary may be segmented, resulting in the loss of the original pixel features and semantic concept, and thus the object detection by the SSD model becomes impossible. To deal with this problem, [6] introduces Type II segmentation method shown in Fig.4. The target size of Type II is the same as that of Type I, but the cropping range of the original image is different from that of Type I. In Type II, the start and end positions of the segmentation are different points from those of Type I. The distance of the coordinates to the split point in Type II is $H_{target}/2$ in the vertical direction and $W_{target}/2$ in the horizontal. We expect the introduction of Type II enables us to detect the targets that could not be detected by Type I.

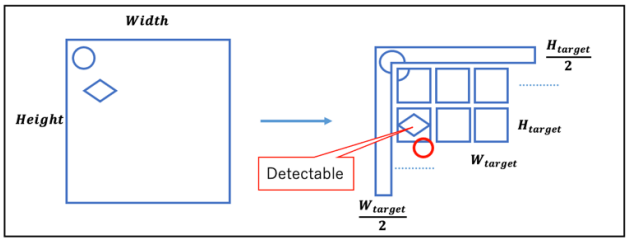

Fig.4: Type II segmentation method

C. *Mixing the region metrics for object detection*

As the final step, the method [6] executes mixing of the region metrics obtained by Type I and Type II to further improve the accuracy. Fig.5 shows the configuration of the mixing method for the region metrics. Its purpose is to correct the results of Type II to match those of Type I to improve the accuracy. There are two forms of modifying the results. First, for objects that could not be detected by Type I, we correct them with the results of Type II. On the other hand, if an object is detected by both Type I and Type II at the same time, it is corrected by filtering with a region metrics. Namely, we combine the resultant data obtained by Type I and by Type II, and then, select an appropriate part of the resultant data where the two overlap by comparing the region superposition and the region area. Details of the algorithm for mixing the region metrics are described in section IV.C.

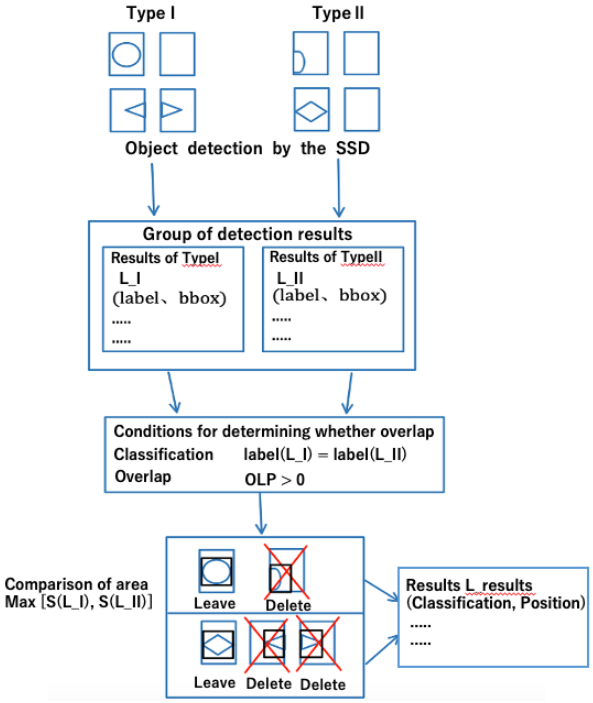

Fig.5: Configuration of the region metrics mixing method.

IV. Proposed method

Although the method in [6] improves the detection of small objects, there are some problems. Firstly, the target size in the segmentation is fixed, so the accuracy varies depending on the size of the original image. Secondly, it fails to detect the objects that appear in the entirety of a segmented image. Thirdly, it is not possible to detect objects larger than the target size. Here, we propose a method to overcome these problems

In the proposed method, we first perform detection on a whole image with SSD without segmentation. This allows us to detect objects that are large enough to be detected without segmentation. Then, by improving the segmentation method in section III, after image segmentation, SSD detection is performed using only Source 1 and 2, and the region metrics are modified. This allows us to detect small objects that could not be detected from the whole image. Then, the region metrics are corrected using the results of whole image and the segmented image.

A. *How to determine the size of segmented areas for image segmentation*

Based on the median values in Table I, we assume that Source 1 can detect object area larger than 1.2% of that of the image, and Source 2 can detect object area smaller than 13%. Let us show the area of the image when detection is performed without any segmentation as S_{origin}. Also, we define the variable $S_{division}$ to show the area of the image when detection is performed after the image has been segmented as

$$S_{division} = \frac{S_{origin} \times 0.012}{0.13}. \qquad (1)$$

In the proposed method, SSD is applied for the image segmented into the size of the $S_{division}$. In this case, only Source 1 and 2 are used for the detection as described later. The visual interpretation of equation (1) is shown in Fig.6. As a result, we can expect that it would detect small objects that could not be detected using the whole image.

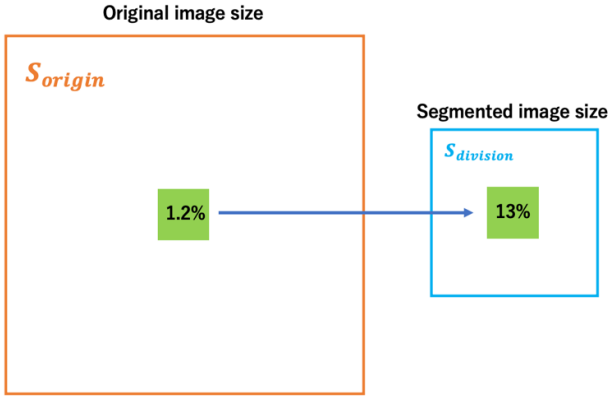

Fig.6 An illustrate of equation (1).

The target size $H_{target} \times W_{target}$ in Section III.A is obtained by the following equations (2) and (3), by letting H_{origin} and W_{origin} as the height and width before segmentation:

$$H_{target} = \frac{H_{origin} \times \sqrt{1.2} \times 0.01}{\sqrt{13} \times 0.01}, \quad (2)$$

$$W_{target} = \frac{W_{origin} \times \sqrt{1.2} \times 0.01}{\sqrt{13} \times 0.01}. \quad (3)$$

Hence, the size of the segmented image is equal to $H_{target} \times W_{target}$.

B. Detection using only Source 1 and 2

In the proposed method, only Source 1 and 2 are used for detection from segmented images. Let us explain the reason of this selection. In Fig.7, we show a result of detection for a segmented image with Source 3 to 6, the figure shows that there is a problem the SSD mistakenly predicts that it is a train, car, or so because the windows and other objects are reflected in the entire image.

On the other hand, when we use Source 1 and 2, we can avoid this problem because the configuration detects only small objects in the image. Therefore, we will use only Source 1 and 2 to detect for the segmented images.

Fig.7 False positive example of segmented image detection.

C. Area Metric Calculation

As the final step of the previous method [6], it obtains the results by mixing the area metrics from Type I and Type II. In Algorithm1, we show the algorithm proposed by [6]. The data Box_{I} of each object in L_{I}, which is a group of results obtained by Type I, and the data Box_{II} of each object in L_{II}, which is a group of results obtained by Type II, are compared one by one for the label of each object. If the labels are the same, the superposition of their regions $OLP_{\mathrm{II}}^{\mathrm{I}}$ is calculated as

$$OLP_{\mathrm{II}}^{\mathrm{I}} = \frac{S_{inter}}{S_{\mathrm{I}} + S_{\mathrm{II}} - S_{inter}}. \quad (4)$$

In (4), OLP_{II}^{I} is the superposition of the object Box_{I} detected by Type I and Box_{II} detected by Type II. S_{I} shows the area of Box_{I}, S_{II} the area of Box_{shift} and S_{inter} shows the area of the superimposed part of both.

If $OLP_{\mathrm{II}}^{\mathrm{I}} > 0$, the area of the two bounding boxes are assumed to be overlapped. The area of each region is calculated, and relatively large regions are left as high-precision results. On the other hand, relatively small portions are deleted as duplicates. L_{II}^{del} is the list of candidates to remove the result of Type II. $L_{\mathrm{II}}^{\mathrm{I}}$ is the list of candidates to remove the result of Type I. L_{result} is the group of results of this algorithm. Using the region metric correction method, the accuracy and consistency of the results can be ensured by the superposition and the area of the region.

Algorithm 1 Mixing the region metrics [6]

for *each* $Box_I \in L_I$ **do**
  for *each* $Box_{II} \in L_{II}$ **do**
    if $Label_I = Label_{II}$ **then**
      $OLP_{II}^{I} \leftarrow \frac{S_I \cap S_{II}}{S_I \cup S_{II}}$;
      if $OLP_{II}^{I} > 0$ **then**
        if $S_I \geq S_{II}$ **then**
          add Box_{II} *into* L_{II}^{del};
        else
          add Box_I *into* L_{II}^{del};
        end if
      end if
    end if
  end for
end for
$L_{result} \leftarrow (L_I - L_I^{del}) + (L_{II} - L_{II}^{del})$;
return L_{result};

In [6], the region metric correction method is performed only on results of Type I and Type II. However, in the proposed method, the detection is also performed on the image without the segmentation. After the region metric is calculated for results of Type I and Type II, the region metric is calculated again for the segmented and, non-segmented whole images.

However, a problem arises when we perform this region metric calculation. The standard SSD is designed to select the most accurate bounding box among the Source 1 to 6, so even if multiple detections are made for the same object, a certain % (45% in this paper) of the total area of the two bounding boxes will be selected. Even if multiple detections are made for the same object, if the area of the two bounding boxes overlap by 45%, the larger object is given priority. On the other hand, if the area is less than 45%, the system recognizes that the objects

are different and allows the overlap. However, in the image shown in Fig.8, the objects detected by Source 1 and 2 have an overlap of 45% or less, and the bounding box is not displayed correctly. As a solution to this problem, we propose a method using Small IoU (Intersection over Union). In the next sub-section, we will explain the Small IoU in detail.

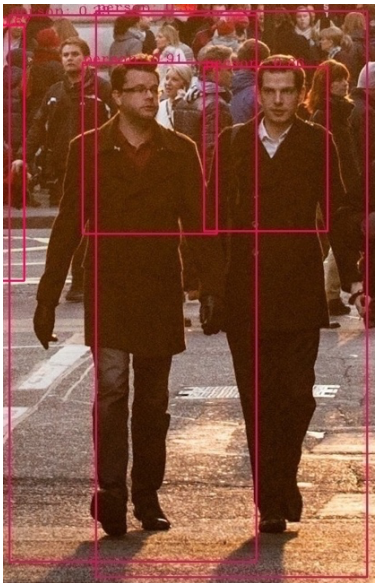

Fig.8 Example of False positives

(Source of image: https://pixabay.com/images/id-438393/)

D. *Small IoU*

Fig.9 shows the conditions for deleting or leaving the small bounding box when the bounding boxes are overlapping. As shown in Fig.9, on the left side, we want to leave the small bounding box, but on the right side, we want to delete the small bounding box. The image in Fig.8 is an example of the right side of Fig.9. In Fig.8, the same object is detected, and the bounding boxes overlap.

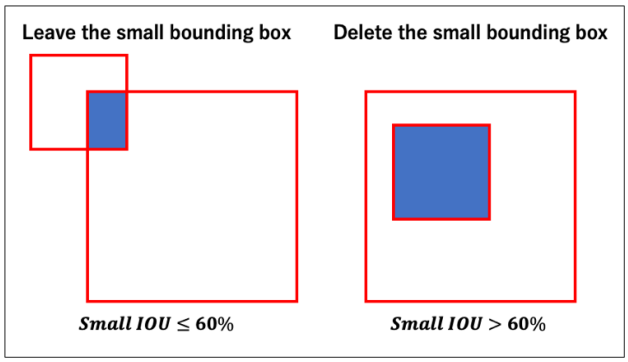

Fig.9 Conditions for deleting or leaving the small bounding box.

When the two bounding boxes are overlapping, the larger bounding box is called BB_{big} and the smaller bounding box BB_{small}. By modifying the IoU used in SSD, we propose a method using Small IoU to solve the problem in the case of Fig.9.Small IoU is expressed by the following equation,

$$Small\ IOU = \frac{BB_{small} \cap BB_{big}}{BB_{small}}. \qquad (5)$$

Small IoU is the ratio of the area covered by BB_{small} and BB_{big} to the area of BB_{small}. In this study, the threshold of this Small IoU is set to 60%, and when $(BB_{small} \cap BB_{big})$ to BB_{small} is more than 60%, BB_{small} is deleted. On the other hand, when it is 60% or less, BB_{small} is left.

E. *Flow of detection by the proposed method*

Here, we summarize the proposed method. Fig.10 shows the flow chart of the proposed method. Step 1 to Step 6 in the figure are described as the following.

Step 1. Apply the standard SSD to the whole image.
Step 2. Calculate the size for segmenting the image by the equations (2) and (3).
Step 3. Segment the image into the sizes defined in Step 2 using Type I and Type II methods.
Step 4. Apply SSD to each of the segmented images using the method described in IV.B.
Step 5. Using the method described in IV.C, calculate area metric of detection results in Step 4.
Step 6. Using the methods described in IV.C and IV.D, calculate area metric of detection results in Step 1 and results in Step 5.

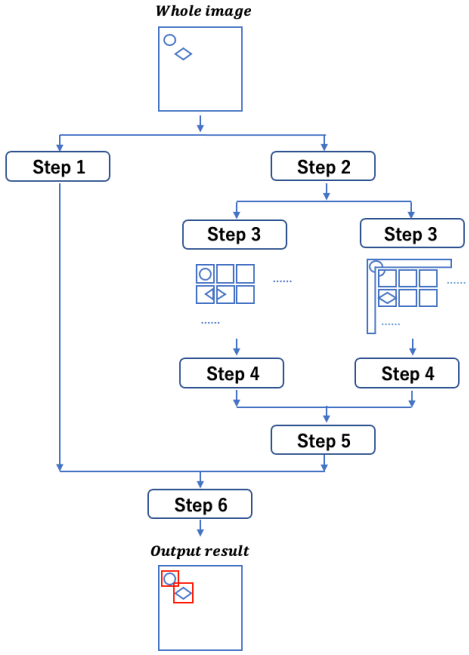

Fig.10 the summary in the form of a detailed flow chart

V. Experimental results

In this section, in order to confirm the effectiveness of the proposed method, we show the results of applying it to 4K images.

A. Conditions

The trained model of SSD300 used in this paper is 'ssd300_mAP_77_43_v2.pth' on GitHub [8]. Specific device resources are shown in Table II below.

Table II Hardware indicators used for experiments

Hardware devices	Equipment parmeters
CPU	Intel Xeon W-2123 3.6GHz × 8
Memory	64GiB
GPU	Quadro P620
System environment	Ubuntu 18.04
Cuda	version: 11.2

B. Results

Fig.11 shows the results of object detection from a 4K (4096 px × 3657px) image using the standard SSD. On the other hand, Fig.12 shows the results of object detection from the same image using the proposed method. By comparing Fig.11 and Fig.12, we can see that the proposed method can improve the accuracy of object detection for small objects. As seen from Fig.11, there are many small objects, especially in the upper left corner of the image. Although the proposed method can detect them in Fig.12, they are not detected in Fig.11.

Fig.11 Result of object detection using standard SSD

(Source of image: https://unsplash.com/photos/9u7_4jwiiTw)

Fig.12 Result of object detection using the proposed method

In the image shown in Fig.11 and Fig.12, there are 40 cars, 2 motorbikes, and 2 persons, for a total of 44 labels. Using the labels in the image and each detection data of the proposed method and standard SSD, we calculate the AP (Average Precision). It is shown in Table III

Table III Comparison of the results provided by the standard SSD and the proposed method

	car (40)	motorbike (2)	person (2)
Standard SSD	15%	0%	0%
Proposed method	65%	50%	100%

We show another result of experiment using an image in which there are 70 people and 16 chairs with a total of 88 labels. Fig.13 shows the results of detection from a 4K (4096 px × 4096px) image using the standard SSD. On the other hand, Fig.14 shows the results using the proposed method. Similarly, using the labels in the image and each detection data of the proposed method and standard SSD, we calculate the AP. It is shown in Table IV

Fig.13 Result of object detection using standard SSD

(Source of image: https://www.pexels.com/photo/man-standing-in-front-of-people-1709003/)

Fig.14 Result of object detection using the proposed method

Table IV Comparison of the results provided by the standard SSD and the proposed method

	person (70)	chair (16)
Standard SSD	31.38%	25.00%
Proposed method	75.43%	43.75%

From Table III and Table IV, we can confirm that the proposed method has a higher detection rate of small targets and can obtain a higher AP.

VI. Conclusion

In this paper, we proposed a method to improve the detection accuracy of object detection using SSD for high resolution images with small objects. By investigating the relationship between object area and detection accuracy in SSD, we proposed a method to determine the size of image segmentation and performed two different image segmentations on the image. These segmented images were detected only from Source 1 and 2 in SSD, and calculate the region metric for each output result. Finally, we proposed a method to recalculate region metric based on the results of detection using segmented and non-segmented images. The results of experiments show that the proposed method can improve the accuracy of object detection for high resolution images containing small objects.

Although the proposed method achieves the above improvements, there are still some problems. First, even if Type I and Type II are used in image segmentation, there still exists a border of segmentation, and the detection of objects near the border will be less accurate. Secondly, we define the size of the object that can be detected by Source 1 and 2, but even if the object size is within the defined range, it may not be detected by Source 1 and 2.

VII. References

[1] Sergios Theodoridis and Konstantinos Koutroumbas, "Pattern Recognition", Canada, Elsevier Inc, 2009.

[2] Kevin P. Murphy, "Machine Learning A Probabilistic Perspective" Massachusetts Institute of Technology, 2021.

[3] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." In Proceedings of the IEEE conference on computer vision and pattern recognition, 2014.

[4] GIRSHICK, Ross. "Fast r-cnn." In Proceedings of the IEEE International Conference on Computer Vision, pp. 1440-1448, 2015

[5] Ren, S., He, K., Girshick, R., & Sun, J. "Faster r-cnn: Towards real-time object detection with region proposal networks." IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), pp.1137-1149, 2019.

[6] Qiu Yicheng, "Improving the Accuracy of Object Detection for High Resolution Images Based on SSD", 2020

[7] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Communications of the ACM 60.6 , pp. 84-90, 2017

[8] GitHub : zhang-can/ECO-pytorch https://github.com/zhang-can/ECO-pytorch