Non-Maximum Suppression for Unknown Class Objects using Image Similarity

Yoshiaki Homma[1], Toshiki Kikuchi[1] and Yuko Ozasa[2]
[1]*Faculty of Science and Technology, Keio University, Kanagawa, Japan*
[2]*School of System Design and Technology, Tokyo Denki University, Tokyo, Japan*

Keywords: Object Detection, Unknown Class, Non-Maximum Suppression.

Abstract: As a post-processing step for object detection, non-maximum suppression (NMS) has been widely used for many years. Greedy-NMS, which is one of the most widely used NMS methods, is effective if the class of objects is known but not if the class of objects is unknown. To overcome this drawback, we propose an NMS method using an image similarity index that is independent of learning. Even if the overlap of bounding boxes that locate different objects is large, they are considered to have located different objects if the similarity of the images in the bounding boxes is low. In order to evaluate the proposed method, we built a new dataset containing unknown class objects. Our experimental results show that the proposed method can reduce the rate of undetected unknown class objects when using greedy-NMS.

1 INTRODUCTION

Object detection is a fundamental problem in computer vision in which an algorithm generates bounding boxes and assigns them classification scores. Ideally, one bounding box should be output for each object. However, in practice, it is often the case that more than one bounding box is output to a single object. Therefore, in order to remove the redundant bounding boxes, many object detection methods use the post-processing called Non-Maximum Suppression (NMS) (Redmon et al., 2015; Girshick, 2015; Ren et al., 2015) .

In greedy-NMS, which is one of the most widely used NMS methods, the intersection over union (IoU) of bounding boxes is calculated for each class, and the bounding boxes are considered to have located the same object when the IoU is greater than a threshold. The classes are the object classes in the dataset used to train the object detector (i.e., classes that are known to the object detector). If an image in which an unknown class object appears is input to the object detector, the unknown class object will be detected as the object of one of the known classes. Therefore, when an unknown class object is occluded by another object, it is assumed that they will be detected as the same known class objects and may be regarded as duplicate detections because greedy-NMS determines whether or not two different objects have been located only by using the objects' IoU.

While most object detection methods (Redmon et al., 2015; Girshick, 2015; Ren et al., 2015) do not take unknown class objects into account, some studies have, as they required the detection of unknown class objects, such as the acquisition of knowledge of unknown class objects. In such studies, it is detrimental for the NMS to mistakenly remove unknown class objects from the detection results.

To overcome the drawback of greedy-NMS, we propose a NMS method using an image similarity index. Even if an unknown class object is occluded by another object and they are classified as the same known class, they can be regarded as different objects if their image similarity index is low. To evaluate the ability of our method to detect unknown class objects, we constructed a new dataset in which unknown class objects appear. Our experiment shows that the proposed methods can reduce the number of unknown class objects mistakenly removed by NMS.

2 RELATED WORK

2.1 Object Detection

With the development of convolutional neural networks (CNNs), object detection methods have made

Homma, Y., Kikuchi, T. and Ozasa, Y.
Non-Maximum Suppression for Unknown Class Objects using Image Similarity.
DOI: 10.5220/0010240304440449
In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) - Volume 5: VISAPP*, pages 444-449
ISBN: 978-989-758-488-6

great progress. Most CNN-based object detectors can be categorized into two-stage (Girshick, 2015; Ren et al., 2015; Cai and Vasconcelos, 2018) and one-stage detectors (Redmon et al., 2015; Liu et al., 2016; Lin et al., 2017b). Two-stage detectors first generate a sparse set of candidate object boxes, each called a region of interest (RoI), using a region-proposal method like selective search (Uijlings et al., 2013), RPN (Ren et al., 2015), and then the RoIs are classified and regressed to refine their localizations. The main advantages of two-stage detectors is high localization and classification accuracy. In contrast to two-stage detectors, one-stage detectors predict bounding boxes from the input images directly without the region proposal-step. The main advantage of one-stage detectors is their inference speed, which is due to their high computational efficiency.

While there are some differences between the detection process of two-stage and one-stage detectors, the result is the same in that the detectors generate a large number of candidate object boxes and classify and refine their localization. Since the object detection methods generate a large number of candidate object boxes compared to the true number of objects appearing in the image, duplicate bounding boxes are often given for a single object. In order to remove the duplicate bounding boxes, most object detection methods use NMS post-processing. Although there are some object detection methods, such as CenterNet (Zhou et al., 2019), that do not necessarily require NMS, they are few, and even these are more accurate with NMS than without. Therefore, NMS is a fundamental component in object detection.

2.2 Non-Maximum Suppression

In greedy-NMS the intersection over union (IoU) of bounding boxes is calculated for each class, and the bounding boxes are considered to have located the same object when the IoU is greater than a threshold. Therefore, when bounding boxes classified as the same class overlap, they may be identified as duplicate detections, even though they actually locate the different objects, because greedy-NMS determines whether they locate the same object only by the IoU of the objects.

To solve this problem, several modified versions of NMS methods have been proposed. Instead of directly removing the highly overlapped bounding boxes, soft-NMS (Bodla et al., 2017) decreases the classification scores of the less confident bounding boxes according to their IoU with the most confident one. However, when the overlap of objects is large, the classification score is very small, so this is not effective in situations where occlusion is severe. In addition, unknown class objects are detected as one of the known class objects, and the score of the classification is likely to be small. In this situation, the unknown class object is more likely to be removed, even when the overlap is not large. R2NMS (Huang et al., 2020) estimates the visible bounding boxes, each of which encloses only the visible part of the object, in addition to full bounding boxes, each of which encloses the entire object, and the IoU of the visible bounding boxes is used to determine whether the bounding boxes locate the same object. While using the visible bounding boxes of objects, which have a smaller IoU even when the overlap of objects is large, can reduce the number of objects undetected by NMS, an additional annotation is required for training.

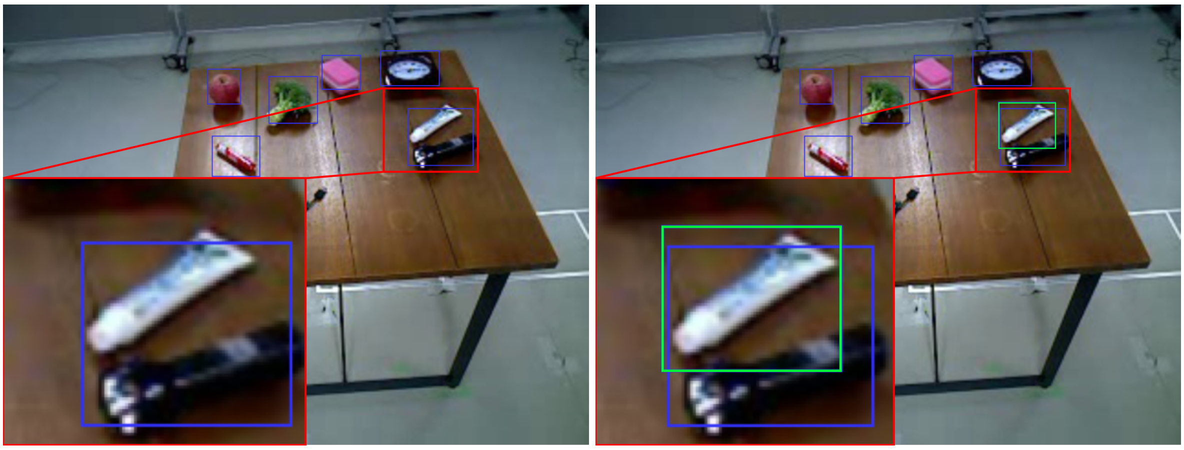
Some studies have proposed the use of neural networks with NMS. In addition to the IoU of the bounding boxes, pairwise-NMS (Liu et al., 2019) uses the L1 distance of the feature vectors of the RoIs corresponding to the bounding boxes. If the overlap of the bounding boxes and the L1 distance of the feature vectors of the bounding boxes are both large, they can be considered to have located different objects. However, the features extracted by CNNs are task dependent. Features extracted by CNNs trained to estimate whether two bounding boxes locate the same object using data consisting of only known classes objects are not expected to be effective for unknown class objects; this is the case with pairwise-NMS. Gnet (Hosang et al., 2017) attempts to learn a deep neural network to perform NMS using only bounding boxes and their scores as input. However, because the classification score of unknown class objects is likely to be low, this is sometimes not effective for scenes in which unknown class objects appear.

Our proposed method only uses learning independent features. Therefore, it does not depend on whether the object is a known or unknown class and is not affected by an unknown class object with a low classification score.

3 METHODOLOGY

3.1 Greedy-NMS

In greedy-NMS, all detected bounding boxes are divided into the classes, and each set of bounding boxes is processed as follows. Let $\mathbf{B} = \{(b_i)_{i=1,2,...N}\}$ denote the set of bounding boxes and $\mathbf{Y} = \{(y_i)_{i=1,2,..N})\}$ the set of corresponding classification scores, where N is the number of bounding



(a) Greedy-NMS. (b) Proposed Method.

Figure 1: An example of detection results after NMS processing.

boxes. Also, let $\mathbf{D}$ denote the set of output bounding boxes.

Let y_m denote the maximum value of an element of $\mathbf{Y}$ and b_m denote the bounding box that corresponds to y_m. First, b_m is added to $\mathbf{D}$ and is removed from $\mathbf{B}$. Let $\mathbf{B_s}$ denote the set of elements of $\mathbf{B}$ whose IoU with b_m is higher than threshold θ_1. In NMS, $\mathbf{B_s}$ is defined as follows:

$$\mathbf{B_s} = \{b \mid I(b,b_m) > \theta_1,\ b \in \mathbf{B}\}. \tag{1}$$

In Equation (1), $I(b,b_m)$ is the IoU of b and b_m:

$$I(b,b_m) = \frac{Area(b \cap b_m)}{Area(b \cup b_m)}, \tag{2}$$

where $Area(b \cap b_m)$ is the area of the intersection of b and b_m, and $Area(b \cup b_m)$ is the area of the union of b and b_m. Let $\mathbf{Y_s}$ denote the set of elements of $\mathbf{Y}$ corresponding to each element of $\mathbf{B_s}$, with the elements of $\mathbf{B_s}$ and $\mathbf{Y_s}$ removed from $\mathbf{B}$ and $\mathbf{Y}$, respectively. The above process is repeated until $\mathbf{B}$ becomes an empty set.

3.2 Proposed Method

Let b and b' denote two different bounding-boxes. While IoU is only used in greedy-NMS, a new criterion using the image similarity index, such as the Sum of Squared Distance (SSD), is used in the proposed method in addition to IoU. The criterion used in the proposed method is defined as follows:

$$C(b,b',\lambda) = I(b,b') + \lambda f(b,b'), \tag{3}$$

where $f(b,b')$ is the value of the image similarity index between two images in b and b', and λ is a balancing parameter between IoU and the image similarity index. The sign of λ is determined such that $\lambda f(b,b')$ is high when the similarity between the bounding box b and b' is high.

In the proposed method, bounding-box b and b' are considered to have located the same object when $I(b,b') > \theta_1 \wedge C(b,b',\lambda) > \theta_2$. Here, θ_1 is the threshold of the IoU of the two bounding boxes, and θ_2 is the threshold of the criterion $C(b,b',\lambda)$. If the IoU of the bounding boxes is large, they can be considered to have located different objects if the criterion $C(b,b',\lambda)$ is large.

The algorithm of the proposed method is shown in Algorithm 1. The algorithm with $\lambda = 0$ and $\theta_2 = 0$ corresponds to the algorithm of greedy-NMS.

4 DATASET

To evaluate the ability to detect unknown class objects, we need a dataset in which unknown class objects appear. Even using Microsoft COCO dataset (Lin et al., 2014), which is a common dataset in object detection, we can build a dataset in which unknown class objects appear by dividing classes into known and unknown classes and selecting only images in which only known class objects appear for training. However, there are a lot of unannotated objects in COCO, and the objects adversely affect the evaluation. Therefore, we built a new dataset in which the classes could be easily divided into known and unknown classes, and annotated all the objects in the dataset.

We selected 22 object classes from ones in the Microsoft COCO dataset and 33 object classes from ones in the RGB-D object dataset (Lai et al., 2011). Ten classes overlapped. Therefore, our dataset contained 45 object classes. We split the object classes into

Algorithm 1: Proposed Method.

```
Input: $\mathbf{B}, \mathbf{Y}, \lambda, \theta_1, \theta_2$
    $\mathbf{B}$ is the list of initial bounding boxes
    $\mathbf{Y}$ contains corresponding detection scores
    $\lambda$ is the balancing parameter
    $\theta_1$ is the IoU threshold
    $\theta_2$ is the threshold of $C(b, b', \lambda)$
Output: D
begin
    $\mathbf{D} \leftarrow \{\}$
    while $\mathbf{B} \neq empty$ do
        $m = argmax\ \mathbf{Y}$
        $\mathbf{D} \leftarrow \mathbf{D} \cup \mathbf{b_m}$; $\mathbf{B} \leftarrow \mathbf{B} - b_m$
        for $b_i$ in B do
            if
              $I(b_m, b_i) \geq \theta_1 \wedge C(b_m, b_i, \lambda) \geq \theta_2$
              then
                $\mathbf{B} \leftarrow \mathbf{B} - b_i$; $\mathbf{Y} \leftarrow \mathbf{Y} - y_i$
            end
        end
    end
    return D
end
```

known classes (present in both the training and test phases) and unknown classes (only present in the test phase). From the 45 classes, 26 classes were selected as known classes and 19 classes were identified as unknown. Table 1 shows how we selected each class from the two datasets and whether we assigned them to known or unknown classes.

As shown in Fig.1, we placed several objects on the table and recorded video sequences while circling the desk.

At first, several known class objects were placed on the table, and then the video sequences were recorded. After that, unknown class objects were added one by one as subsequent video was recorded. Each time an unknown class object was added, the placement of the objects was changed.

The collection of data in which the same known class objects are placed is called a scene. Table 2 shows the number of known and unknown class objects in each scene.

5 EXPERIMENTS

From the dataset, we selected 2763 images in which only known class objects appeared for training and 18763 images in which unknown class objects appeared for evaluation. Training data was only used for training the object detector.

Table 1: A table showing the classes that exist in both the COCO and RGB-D object dataset, the classes that exist only in COCO, and the classes that exist only in RGB-D. Classes without * indicate a known class, and classes with * indicate an unknown class.

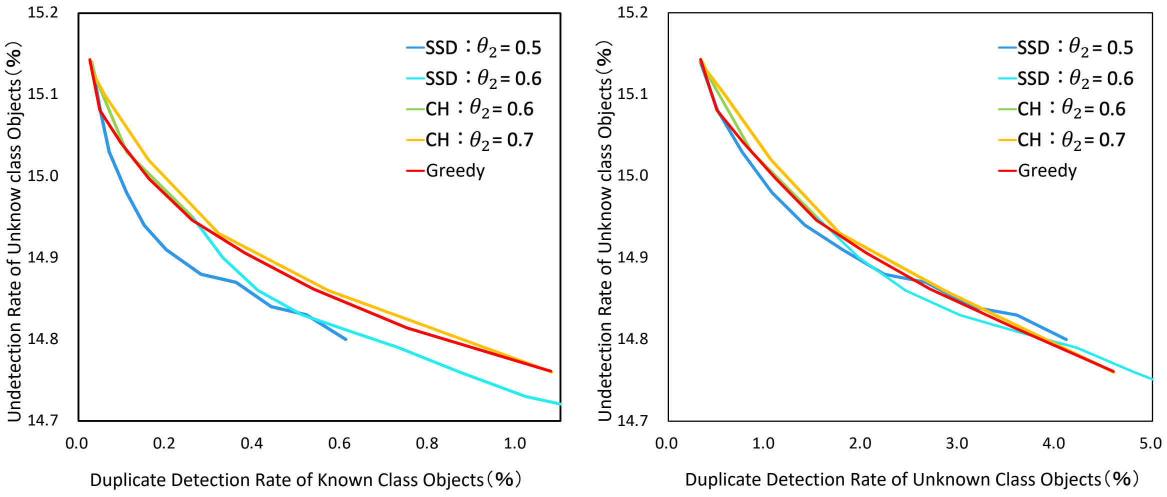
Both	only COCO	only RGB-D
bottle	fork	sponge
bowl	knife	soda can
scissors	clock	shampoo
toothbrush	vase	plate*
keyboard	sports ball	hand towel*
cell phone	sandwich	glue stick*
book	hot dog	flashlight*
banana	donut	cap*
apple	broccoli	calculator*
orange	carrot	toothpaste*
	pizza	instant noodles*
	mouse	food box*
		food bag*
		lemon*
		onion*
		food can*
		tomato*
		potato*
		lime*
		marker*
		camera*
		pitcher*

Table 2: The number of known and unknown class objects in each scene.

Scene	# of known class	# of unknown class
1	5	0
2	5	5
3	5	5
4	5	4
5	5	1
6	5	2
7	5	10
8	5	10
9	10	10
10	10	7

For object detection, we used the binary classification model to predict whether a bounding box contains an object or not. We used Fast R-CNN (Girshick, 2015) as the object detector, which outputs bounding boxes and object scores. We used ResNet-101-FPN (He et al., 2016; Lin et al., 2017a) as the backbone network in the object detector, which was pretrained on the ImageNet1k (Deng et al., 2009). We used stochastic gradient descent (SGD) with a minibatch size of 16. The model was trained for 4.0×10^4 iterations with an initial learning rate of 2.0×10^{-2}, which was subsequently divided by 10 at 3.0×10^4 iterations. We used a weight decay of 1.0×10^{-4} and a momentum of 0.9.

In the experiments, we used SSD and the color histogram as the image similarity index. SSD is de-



(a)

(b)

Figure 2: Undetection and duplicate detection rates for varying the threshold θ_1, θ_2 and λ.

fined as follows:

$$d = \frac{\Sigma_{x,y}(I(x,y) - I'(x,y))^2}{\sqrt{\Sigma_{x,y}I(x,y)^2 \cdot \Sigma_{x,y}I'(x,y)^2}}, \tag{4}$$

where, $I(x,y)$ and $I'(x,y)$ are pixel values at (x,y) of two different images.

We evaluated the proposed method using the percentage of undetected objects (undetection rate) and the percentage of duplicately detected objects (duplicate detection rate) when the bounding boxes were suppressed using the respective image similarity indices. Undetection rate r_{ud} and duplicate detection rate r_{dd} are defined as follows respectively:

$$r_{ud} = 100 \times \frac{N_{ud}}{N_{all}}, \tag{5}$$

$$r_{dd} = 100 \times \frac{N_{dd}}{N_{all}}, \tag{6}$$

where N_{all}, N_{ud}, and N_{dd} are the number of all objects, undetected objects, and duplicately detected objects throughout the test data respectively.

6 RESULT

The performance of the proposed method was compared with greedy-NMS by varying the threshold θ_1, θ_2 and the balancing parameter λ defined in Formula (3). In the proposed method using SSD as the image similarity index, θ_1 was fixed at 0.5, θ_2 was specified 0.5 or 0.6, and λ was varied from -0.5 to 0. In the proposed method using the coefficient of the color histogram as the image similarity index, θ_1 was fixed at 0.5, θ_2 was specified 0.6 or 0.7, and λ was varied from 0 to 0.5. In greedy-NMS, θ_1 was varied from 0.5 to 0.75, and θ_2 and λ was fixed at 0.

Fig. 2 shows the undetection rates and the duplicate detection rates. In both Fig. 2a and Fig. 2 b, the vertical axis represents undetection rate of unknown class objects. The horizontal axis represents the duplicate detection rate of known class objects and unknown class objects in Fig. 2a and Fig. 2b respectively.

In Fig. 2, when SSD was used as the image similarity index, the undetection rate was lower with the proposed method than with the greedy-NMS. On the other hand, in Fig. 2, when the coefficient of the color histogram was used as the image similarity index, the undetection rate was higher with the proposed method than with the greedy-NMS. The results show that the proposed method is effective in reducing the undetection rate of unknown class objects without significantly increasing the duplicate detection rate of known and unknown class objects when SSD is used as the image similarity index. Furthermore, the results shows that the effectiveness of the proposed method depends on which image similarity index is used.

In Fig. 2, the increase in duplicate detection rate was different between known and unknown classes. The reason for this is probably due to the low accuracy of the localization of unknown class objects. Because localization depends on learning, unknown class objects are more difficult to localize than known class objects. Also, because SSD compares pixel values at the same location in two images, even if the two different bounding boxes locate the same object, the similarity is lower if the gap between the bounding boxes is large. Therefore, the proposed method is assumed to be more effective in improving the localization accuracy of unknown class objects or using a robust image similarity index for the misalignment of the bounding box.

We also show some the visual results of the greedy-NMS and the proposed method for comparison. As shown in Fig. 1, "toothpaste", which is an unknown class object to be detected, was removed when greedy-NMS was used (Fig. 1a), whereas it was detected in the position indicated by the green box in the proposed method (Fig. 1b).

7 CONCLUSION

In this paper, in addition to the IoU of the bounding boxes, we present an NMS method using the image similarity index of the images in the two bounding boxes. To evaluate our method's ability to detect unknown class objects, we constructed a new dataset including unknown class objects. Our experiment shows that the proposed method can reduce the number of unknown class objects mistakenly removed by NMS. In the future, we plan to develop an effective feature extraction method for unknown class objects and to use it with NMS.

REFERENCES

Bodla, N., Singh, B., Chellappa, R., and Davis, L. S. (2017). Soft-nms–improving object detection with one line of code. In *ICCV*, pages 5561–5569.

Cai, Z. and Vasconcelos, N. (2018). Cascade r-cnn: Delving into high quality object detection. In *CVPR*, pages 6154–6162.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE.

Girshick, R. (2015). Fast r-cnn. In *ICCV*, pages 1440–1448. IEEE.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE.

Hosang, J., Benenson, R., and Schiele, B. (2017). Learning non-maximum suppression. In *CVPR*, pages 4507–4515.

Huang, X., Ge, Z., Jie, Z., and Yoshie, O. (2020). Nms by representative region: Towards crowded pedestrian detection by proposal pairing. In *CVPR*, pages 10750–10759.

Lai, K., Bo, L., Ren, X., and Fox, D. (2011). A large-scale hierarchical multi-view rgb-d object dataset. In *ICRA*, pages 1817–1824. IEEE.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017a). Feature pyramid networks for object detection. In *CVPR*, volume 1, page 4. IEEE.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017b). Focal loss for dense object detection. In *ICCV*, pages 2980–2988.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *ECCV*, pages 21–37. Springer.

Liu, Y., Liu, L., Rezatofighi, H., Do, T.-T., Shi, Q., and Reid, I. (2019). Learning pairwise relationship for multi-object detection in crowded scenes. *arXiv preprint arXiv:1901.03796*.

Redmon, J., Divvala, S. K., Girshick, R. B., and Farhadi, A. (2015). You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99.

Uijlings, J. R., Van De Sande, K. E., Gevers, T., and Smeulders, A. W. (2013). Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171.

Zhou, X., Wang, D., and Krähenbühl, P. (2019). Objects as points. *arXiv preprint arXiv:1904.07850*.