IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 6 July 2022, accepted 24 July 2022, date of publication 1 August 2022, date of current version 4 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3195490

RESEARCH ARTICLE

A Method to Automatic Create Dataset for Training Object Detection Neural Networks

SHI ZHOU[1], (Member, IEEE), ZIJUN YANG[1], MIAOMIAO ZHU[2], HE LI[3], SEIICHI SERIKAWA[1], MITSUNORI MIZUMACHI[1], (Member, IEEE), AND LIFENG ZHANG[1], (Member, IEEE)

[1]Department of Electrical and Electronic Engineering, Kyushu Institute of Technology, Kitakyushu, Fukuoka 804-0015, Japan
[2]School of Computer Science and Engineering, Changshu Institute of Technology, Suzhou, Jiangsu 215500, China
[3]School of Mechanical Engineering and Automation, Northeastern University, Shenyang, Liaoning 110819, China

Corresponding author: Shi Zhou (zhou.shi403@mail.kyutech.jp)

ABSTRACT Numerous high-accuracy neural networks for object detection have been proposed in recent years. Creating a specific training dataset seems to be the main obstacle in putting them into practice. This paper aims to overcome this obstacle and proposes an automatic dataset creation method. This method first extracts objects from the source images and then combines them as synthetic images. These synthetic images are annotated automatically using the data flow in the process and can be used directly for training. To the best of our knowledge, this is the first automatic dataset creation method for object detection tasks. In addition, the adaptive object extraction method and created natural synthetic images make the proposed method maintain strong adaptation and generalization ability. To validate the feasibility, a dataset that includes 44 categories of objects is created for the object detection task in a vending supermarket. Under the strictest metric AP_{75}, both the trained EfficientDet and YOLOv4 achieve higher than 95% in accuracy on the common difficulty testing set and higher than 90% in accuracy on the high difficulty testing set.

INDEX TERMS Object detection, dataset creation, deep learning, computer vision, artificial intelligence.

I. INTRODUCTION

Recent advances in deep learning have contributed significantly to the performance improvement of many computer vision tasks [1]–[5]. Some of the notable achievements happened in the area of image classification [6], [7], object detection [8]–[10], and image segmentation [11], [12]. This paper focuses on the ultimate goal of putting deep learning-based object detection methods into practice and proposes an automatic dataset creation method to save time in manual labeling.

Object detection is a computer vision task for locating objects and identifying them in an image. It has a wide application in the industry, including face detection, object counting, and driving system. In the past decade, many neural networks have been proposed to achieve higher accuracy and more efficient object detection, such as YOLO [13]–[16], EfficientDet [17], and RetinaNet [18]. Those neural networks can perform well in both complex and simple environments. Especially in environments with a simple background, higher than 90% accuracy can be achieved. Fortunately, the background is simple and can be controlled well in most industry applications, such as vending supermarkets, and the assembly workshop. Applying these high accuracy methods to the industry is the future trend of the automation era.

The main difficulty in applying these object detection neural networks to practice is the training of deep learning models. Deep learning-based object detection is a supervised problem. The performance of the trained model depends heavily on the corresponding annotated training dataset. Usually, the pre-trained models provided by the proposer are trained on the public datasets or the proposed datasets, which can only detect these categories belonging to the training datasets and cannot be used directly for any other applications. Therefore, creating a corresponding training dataset is the pre-condition to applying these methods in practical applications. This leads to another challenge. How to annotate objects in these images efficiently? Can new datasets with annotations be created automatically?

Currently, the feasibility of most object detection neural networks is proved on popular public datasets, such as MS COCO [19] and PASCAL VOC [20]. Meanwhile, some

The associate editor coordinating the review of this manuscript and approving it for publication was Yiqi Liu.

datasets [21]–[23] for special situations have also been proposed to put object detection networks into practice. However, almost all these datasets are annotated manually with annotation tools, which require human annotators to spend 20-30 seconds per object in one image. In addition, the performance of the trained model in the testing set heavily depends on the amount and quality of the training dataset. Therefore, the annotation work is finished from coarse to fine. If practical applications use the same annotation methods as public datasets, it will be an extremely time-consuming and high-cost task.

Cityscape [24] is a popular synthetic dataset for autonomous driving, which artificially integrates different weather conditions into the images. In this way, creating datasets with various weather conditions by image processing saves a lot of time to prepare the training dataset.

Inspired by Cityscape, this paper explores the possibility of automatically creating object detection datasets through image processing. Considering the difficulty of creating undistorted synthetic images with complex backgrounds and the ease of controlling most industrial applications' backgrounds, this paper proposes a novel method to create datasets automatically for object detection applications with simple backgrounds. The dataset created with the proposed method consists of a set of synthetic images and a corresponding annotation file. The creation of the synthetic images follows the process of first extracting objects from the source images and then combining them.

The contributions of this work are summarized as follows:

1 Inputting only a set of source images, the output synthetic images with annotations can be used directly for training the network. All the processes are done automatically.
2 Adaptive extracting objects from source images makes the proposed method with strong adaptability to different conditions.
3 Creating natural synthetic images makes the trained model with strong robustness in practice.

To demonstrate the practicality of the proposed method, we created training datasets for a vending supermarket. Furthermore, the popular object detection networks, EfficientDet and YOLOv4, are trained on this dataset. The present results are encouraging and sufficient to prove the practicality.

II. RELATED WORKS

A. OBJECT ANNOTATION

The annotation of an object entails both stating its specified class and localizing it in the image [25]. Almost existing object detection datasets are annotated manually [19], [20]. The accurate annotation relies on a coarse to fine pipeline that labels objects step by step. In the end, the professionals check all the annotations to ensure the quality. These results in the annotation to being a complicated and extremely time-consuming task.

To simplify manual annotation, some label tools have been proposed, such as labelme [26] and coco annotator. With these tools, anyone without professional knowledge can complete the annotation work. However, although these tools make the annotation work easier, it is still a time-consuming task since the multi-step pipeline. What's more, due to a large amount of data, it cannot be finished by one or several persons in a short time. In this way, the workers from the famous Amazon's Mechanical Turk (AMT) are employed for all the crowd-sourcing tasks.

For complex object detection tasks, it is difficult to create artificial datasets. These label tools and crowd workers platforms can help annotate objects faster and easier. However, the cost of labors is high for tasks with simple backgrounds. Synthetic datasets are a good alternative. This paper proposed a method of automatically creating annotated datasets by synthesizing datasets for applications with simple background.

B. OBJECT EXTRACTION

Extracting objects from images is a practical and critical task in many computer applications. Various methods, such as interactive approaches [27], [28], background modeling [29], [30], and edge-based approaches [31], [32], have been proposed in this domain in the past decades. Nonetheless, accurate extracting objects is still a tricky problem since a multitude of challenges, such as illumination changes, automatic camera adjustments, and shadows. What's more, the requirements are unique for different applications.

Interactive approaches always extract objects based on the user-specified features. For example, Magic Wand [33] distributes foreground and background by adjusting the tolerance of color statistics in a specified region. Intelligent Scissors [34] trace the object's boundary by calculating the minimum cost based on specified seeds. GrabCut [35] extract object by comparing the pixels inside and outside the selection bounding box. These methods can accurately extract objects from complex scenes. But they are inefficient since the requirement for human intervention.

The general idea of background modeling is to make a pixel-wise comparison based on the built background model, so as to maintain the foreground pixels and eliminate the background pixels. Although many effective background modeling approaches have been proposed [36], [37], accurate foreground segmentation is still a problem due to the typical challenges, such as illumination changes and automatic camera adjustments. These challenges confuse the background modeling, and hence, cause incorrect segmentation.

Edge-based approaches generally produce a contour for object extraction by introducing image gradients into a predefined energy function [38], [39]. These approaches are sensitive to noises in the image. Conversely, they perform well on objects with high contrast. Moreover, the computation cost of edge-based approaches is low. In this paper, we extract the object with the mask that is conducted according to the distance of rich edges. Therefore, noises that have a few edges can be ignored well.

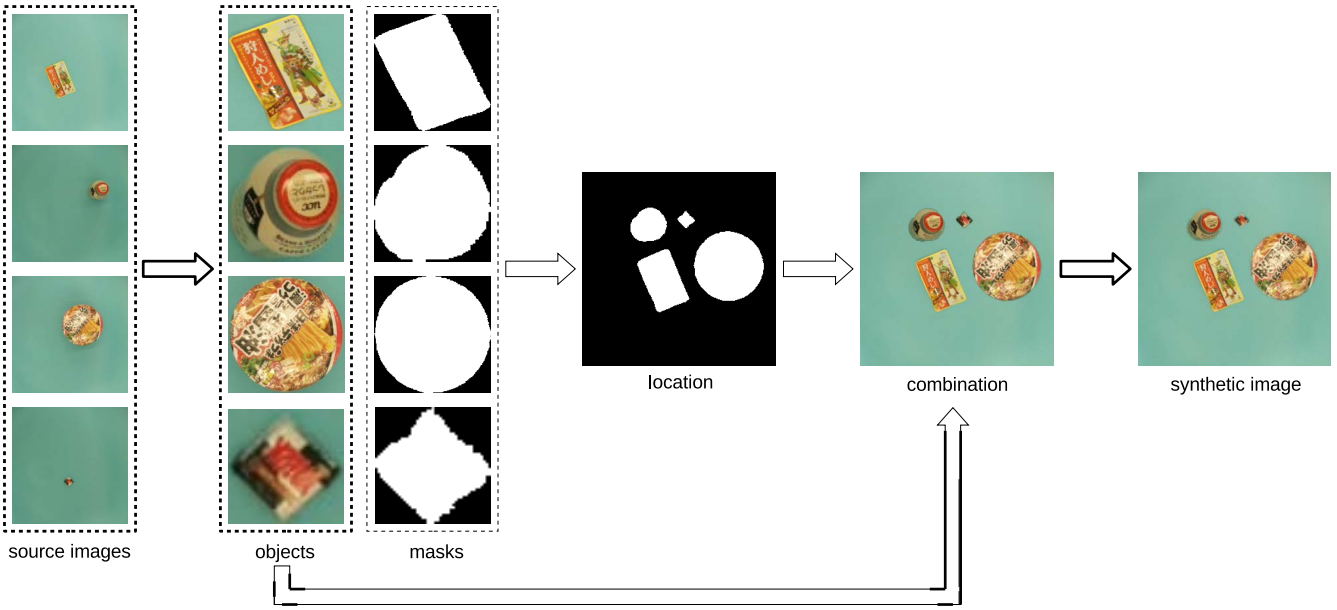


FIGURE 1. Pipeline of synthesizing image with multiple objects from several source images with a single object.

C. IMAGE FUSION

The goal of image fusion is to create a fused image, which gathers important information from multiple images [40]. In most cases, image fusion is a preparatory step to widespread computer vision applications, such as medical diagnostics [41] and image synthesis [42]–[44]. Since it is an important technique for various image processing, a large amount of image fusion methods have been proposed [45]–[47]. These methods usually are broadly classified into two groups: spatial domain methods and frequency domain methods.

Spatial domain methods directly fuse images with the intensity of pixels. These methods have a high efficiency by passing filters. But direct operation on intensity causes spatial distortion, which is a negative factor for further processing such as object detection.

Frequency domain methods first transfer the image from the spatial domain to the frequency domain with Fourier transform. after fusing, the result is obtained by inverse Fourier transform. Although the spatial distortion is well solved, it leads to high computational complexity. In this paper, image fusion is operated only in the boundary area of the object, and most of the image area is merged directly from the source image. Considering these, spatial distortion has almost no impact on object detection. So, we use the spatial domain method to fuse the combined image.

III. METHOD

The dataset for object detection consists of two parts, a set of images and a corresponding annotation file. This method proposes annotating while synthesizing images to enable the automatic creation of datasets. The pipeline of synthesizing image is shown in Fig. 1. The input is some source images with a single object, and the output synthetic image is a natural combination of these objects. The details are described in Sec. III-B, III-C, III-D. The automatic annotation depends on the data flow during synthesizing image, which is described in detail in Sec. III-E. Additionally, the collection of source images and the utilization of created dataset are introduced in Sec. III-A and III-F.

A. SOURCE IMAGES COLLECTION AND PREPARATION

The proposed method focuses on creating datasets with simple backgrounds. The 'simple' refers to solid-color. Since objects are extracted according to edge information (Sec. III-B), it is better to use a solid-color that has a gap with all objects. Additionally, to create a comprehensive dataset, multiple source images for each object taken from different viewpoints is necessary.

For situations with multiple conditions, such as multiple solid-color backgrounds, and multiple illuminations, each object should be taken the same amount of source images under each condition. In this way, a comprehensive and balanced dataset leads to good network learning.

After collecting the source images, preparation work is necessary to create a dataset. As the title described, the created dataset is annotated automatically. This is accomplished through the data flow from the source images to the synthetic image. The category information of each object in the source image comes from the corresponding file name. Therefore, naming each source image as a combination of its category information and a unique number is the key to starting the data flow.

For multiple condition cases, source images taken under different conditions should be saved in different folders. This presetting ensures that source images for a synthetic image

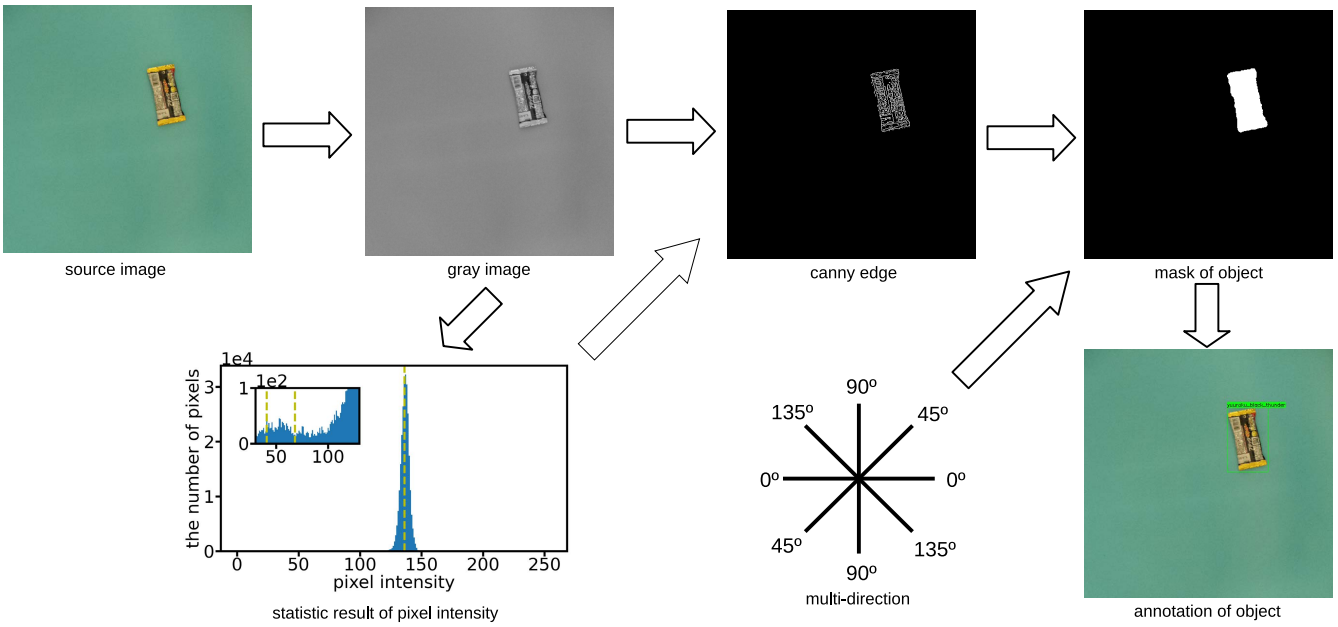


FIGURE 2. Adaptive masking and extracting object from source image according to the statistic result of pixel intensity.

come from the same condition. In more detail, the surface of objects located on a blue background is bluish due to light reflection. Likewise, the intensity of the image that takes for the same object under different illumination is different. If an object that extracts from one image with blue background is put into another image with a red background, the combined image will look unnatural. So, objects for one synthesized come from the same condition is necessary to avoid color distortion.

B. OBJECT MASKING AND EXTRACTION

Synthesizing images by combining several objects based on the method described in Sec. III-C, the first step is to extract these objects from source images. The background of the source image is solid-color with no edge information, while the object is colorful and full of edge information. Therefore, objects are extracted from the source images based on edge information. This paper uses Canny edge detector. It is feasible to replace it with other edge detectors.

The pipeline of object extraction is shown in Fig. 2. The RGB image is converted to Gray first. This operation, on the one hand, satisfies the input requirement of Canny edge detector, on the other hand, reduces the computational cost by processing one channel gray-scale image instead of three channels RGB image.

Canny algorithm detects edge according to the image gradients, and whether there is an edge is determined by its double thresholds [48]. If the pixel's gradient is larger than the high threshold, it is a strong edge pixel. If that is smaller than the high threshold but larger than the low threshold, it is a weak edge pixel. Otherwise, it locates at no edge region. Therefore, adaptive finding the background and keeping a distance from the background is the key to achieving automatic edge information detection. Especially for source images from different conditions, adaptive determining the two thresholds is essential for automatic object extraction.

Considering the source images have a large solid-color background, their intensities have the same distribution even if the source images are from different conditions. The intensity distribution of a sample source image is shown in Fig. 2. It is a Gauss distribution with a small standard deviation. The background's intensity distribution is concentrated, while the object's intensity distribution is un-concentrated. Additionally, the relatively small area size of the object does not affect the distribution of the whole image. The median value of the background is assumed to be that of the full image. In this way, the two thresholds are determined adaptively by keeping a distance from the median value of the full image.

$$\begin{cases} t_1 = 0.5 * E \\ t_2 = 0.7 * E \end{cases} \tag{1}$$

where E is the median value of pixels intensity in the gray image. t_1 denotes the low threshold and t_2 denotes the high threshold. Pixels with an intensity gap greater than $0.5E$ from the background are considered as possible edges, and those greater than $0.7E$ are considered directly as edges.

The edge information output from the Canny edge detector is a series of pixels that denotes the boundary and location of the object. However, what object extraction require is a region corresponding to the object. Therefore, a multi-direction method is proposed to convert the scatted edge point to a mask region. The edges are checked from directions 0° and 90° to obtain an initialized mask. In addition, a distance

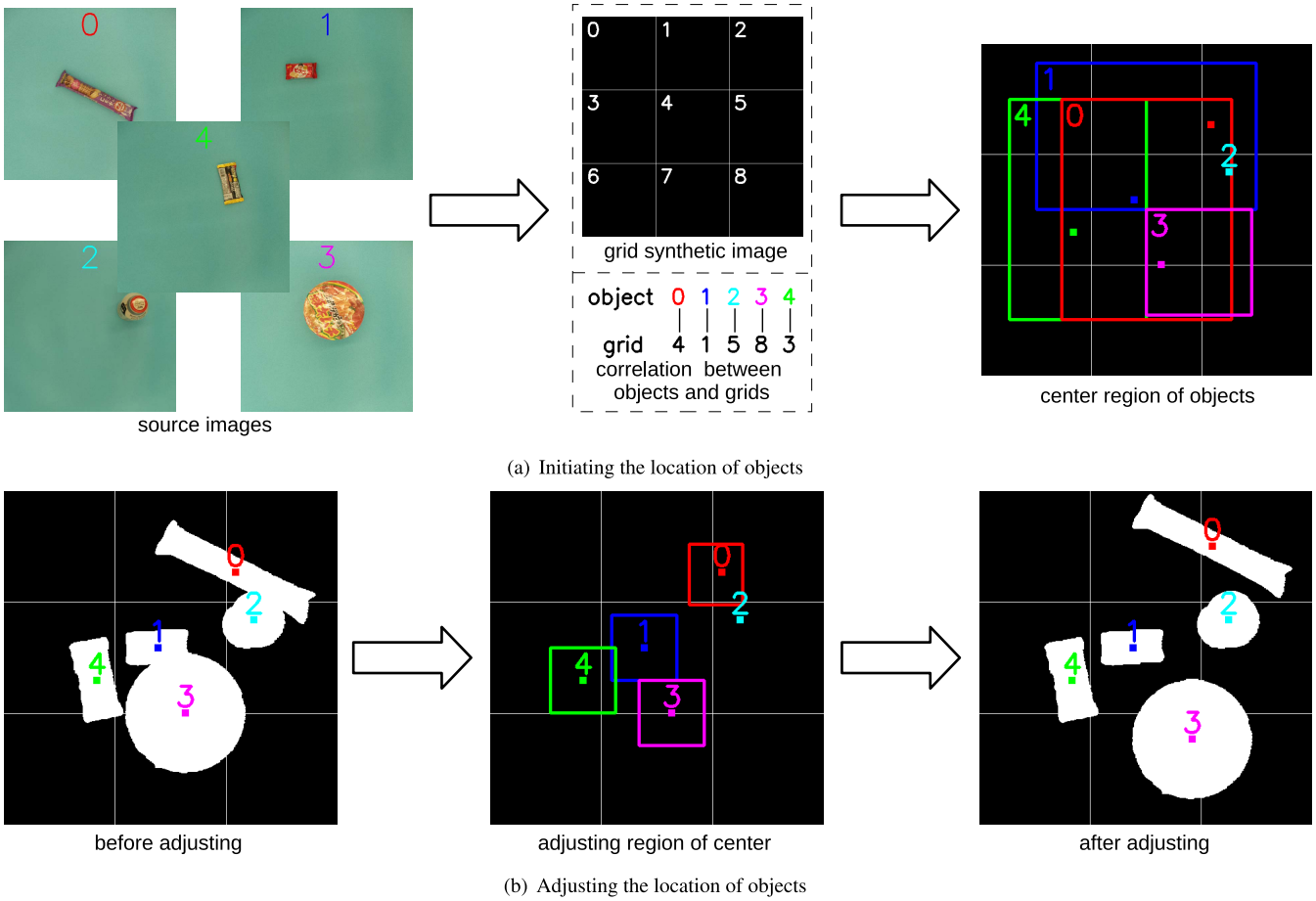


FIGURE 3. **Randomly locating objects into the synthetic image and making a reasonable combination for all objects according to their masks.**

constraint is set to avoid the effect of edge caused by noise.

$$\|a - b\| < \mu \tag{2}$$

where a and b denote two edge pixels. If the distance between the two edge pixels is smaller than μ, all pixels that locate between them are in the region of the object and should be masked as the object. This ensures that small texture-less regions on the object are not ignored and the object is completely masked.

However, due to the reflection of light, the shadow of the object, and overexposure, pixels' intensity gap in the object-background boundary area is small. As a result, parts of edge pixels cannot be detected, resulting in an uneven boundary. Furthermore, the combination of objects is performed based on the mask of objects, as described in Sec. III-C. Eventually, the uneven borders lead to unnatural synthetic images created. Therefore, it is important to preserve the boundary of the mask.

Considering the object's shape is regular, the mask's shape should be regular too. Thus, the mask is further modified in another two directions 45° and 135°. Of course, more or fewer directions can be considered to obtain better performance.

Finally, the object can be extracted accurately from the source image according to the mask.

C. OBJECT LOCATION AND COMBINATION

This method creates the synthetic image by combining several extracted objects. The natural distribution of objects in the synthetic image is the key. Inspired by some region-based image segmentation methods [49], [50], which first divide the whole image into many grids, and then adjust these grids to segment the whole image. In this work, to obtain a well initial state, the whole combined image is also divided into $s * s$ grids according to the number of objects in the corresponding synthetic image.

$$s = \left\{\sqrt{N}\right\} \tag{3}$$

where N denotes the number of objects in the synthetic image, $\{\cdot\}$ operation returns the minimum integer that is larger than itself. Therefore, the number of grids is more than objects. Each object can be assigned a unique grid randomly.

Since the absorption of light is different for objects with different colors, even if all the other conditions are kept constant, the background of the captured image is still slightly different. Thus, no prepared backgrounds are used for

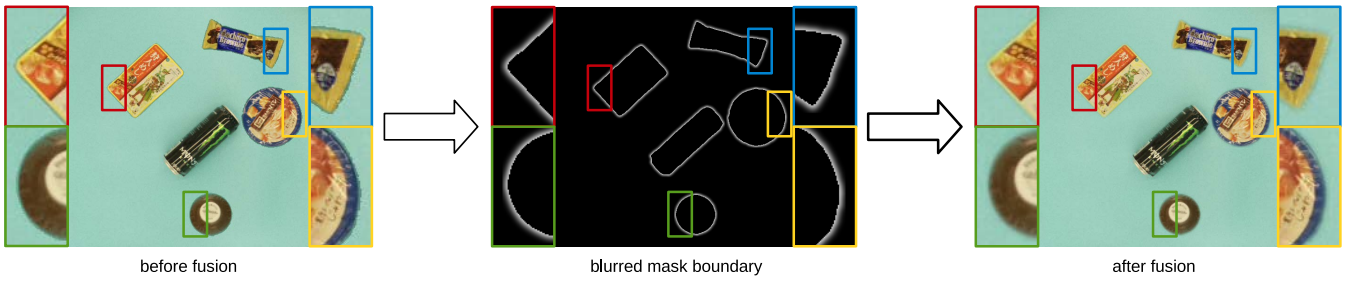


FIGURE 4. Fusing objects' boundaries into the background to obtain a natural synthetic image.

synthesizing images to ensure the diversity of backgrounds. Alternatively, the source image corresponding to any object in the synthetic image is selected as the initial synthetic image. In this way, the model trained on the natural dataset with diverse backgrounds can perform well in the testing set. A sample is shown in Fig. 3, where the source image labeled with 2 is selected as the initial synthetic image. The corresponding object is kept at the original place throughout the processing. In addition, the grid labeled with 5, whose center is closest to the object, is assigned to it. The other grids are assigned randomly to other objects. Therefore, there is a correlation between objects and grids.

To distribute all other objects in the synthetic image naturally, a region is delineated for each object's center.

$$\begin{cases} x \in [\max(\frac{O_c^x}{2}, G_c^x - G_w), \min(W - \frac{O_c^x}{2}, G_c^x + G_w)] \\ y \in [\max(\frac{O_c^y}{2}, G_c^y - G_h), \min(H - \frac{O_c^y}{2}, G_c^y + G_h)] \end{cases} \tag{4}$$

where (O_c^x, O_c^y) and (G_c^x, G_c^y) are the coordinate of the object's center and grid's center. G_w and G_h denote the width and height of the grid, while w and h denote the width and height of the synthetic image. In this way, each object has a center region, and the center of each object can be located randomly in its corresponding center region. Thus, all objects are located in the synthetic image randomly.

After initiating the location, these objects are merged into the synthetic image entirely. However, the previous work about localization only considers the scattered distribution of objects' centers and ignores the overlap between them. As shown in Fig. 3(b), before adjusting the center, some objects overlap others. Especially in the case of a small object completely covered by a large one, further adjustments are necessary.

To avoid large-scale movement and preserve the randomness of the initial position, another relatively small region is set to adjust the objects' center.

$$\begin{cases} x \in [O_c^x - s_{adj}, O_c^x + s_{adj}] \\ y \in [O_c^y - s_{adj}, O_c^y + s_{adj}] \end{cases} \tag{5}$$

where s_{adj} is the tolerance of the object center's movement. This results in a square region for each object's center as the adjusting region shown in Fig. 3(b). The size of the adjusting region is $2s_{adj}$. In the adjusting region, the center of the object is adjusted randomly until it satisfies the requirement.

Inspired by object detection tasks, which use the Intersection-over-Union (IoU) between the ground truth bounding box and the predicted one to evaluate the accuracy. This paper uses the IOU between two objects to control the overlap. Differently, to ensure the synthetic image with natural distribution and reduce the impact of object's background, the corresponding IoU is calculated based on objects' mask rather than their bounding box.

$$\mathrm{IoU}_{ab} = \frac{S_a + S_b - S_{ab}}{S_a + S_b} \tag{6}$$

where S_a denotes the mask's area size of object a, S_b denotes the mask's area size of object b, S_{ab} denotes the overlap region's area size between object a and b. For any two objects, if the IoU between them is lower than the maximum tolerance overlap, it will satisfy the requirement. Otherwise, one of these objects' locations is reset again until satisfying the requirement. After adjusting the center, these objects are combined with a suitable location in the synthetic image.

D. OBJECT-BACKGROUND FUSION

According to the location, all objects are merged into the initial synthetic image to create a new synthetic image for output. A sample is shown in Fig. 4, five objects are placed in the synthetic image. However, due to the slight difference between the background of the synthetic image and source images, sudden color changes appear in the boundary of the objects. If these images are used for training, the network will focus on strong boundary features. As a result, the trained model has a weak perception of the object and shows weak generalization ability during testing. Thus, eliminating these sudden color changes is necessary.

To keep the features of the object intact, a fusion operation is performed on the boundary region between objects and the background, instead of fusing the entire object into the synthetic image. For details, this method first erodes object's mask, then Gaussian blur the mask. The blurred mask boundary is a gradual change from 0 to 255, instead of a sudden change. After this operation, the mask is 0 in the non-object region, 255 in the object region, and between 0 and 255 in the object's boundary region. Finally, the synthetic image's intensity is reset according to the mask.

$$P = P_b * m + P_o * (1 - \frac{m}{255}) \tag{7}$$

Algorithm 1 Automatic Annotation Method

Input: $I_{in} = \{I^i | i \in [0, N]\}$, $[lim_s, lim_l]$
Output: $I_{out} = \{I_s^j | j \in [0, M]\}$, F_{json}

$ant = [\ ]$
shuffle I_{in}
$n = random(lim_s, lim_l)$
$j = 0, s = 0, e = n$
while $e \leq N$ **do**
  $objs = [\ ]$
  for $I \in \{I_i | i \in [s, e]\}$ **do**
    $obj = \{\ \}$
    $M_o, I_o \Leftarrow extraction(I)$ (Sec. III-B)
    $C_o \Leftarrow$ *source image file name*
    $obj[\text{“mask”}] = M_o$
    $obj[\text{“category”}] = C_o$
    $obj[\text{“object”}] = I_o$
    $obj[\text{“image id”}] = j$
    $objs.append(obj)$
  end for
  for $obj \in objs$ **do**
    $L_o, S_o \Leftarrow location(obj)$ (Sec. III-C)
    $obj[\text{“bbox”}] = L_o$
    $obj[\text{“area”}] = S_o$
  end for
  $objs.uppdate()$
  $I_s^j \Leftarrow fusion(objs)$ (Sec. III-D)
  $ant.append(objs)$
  $s = e$
  $n = random(lim_s, lim_l)$
  $e = min(s + n, N)$
  $j = j + 1$
end while
$F_{json} \Leftarrow ant$

where P_b and P_o denote the intensity of the background and the object. m is the corresponding mask. After the fusion operation, strong features in the boundary region are eliminated. Meanwhile, objects' features are preserved perfectly.

E. AUTOMATIC ANNOTATION AND ADDITIONAL CONSIDERATIONS

The automatic annotation of datasets is done through a data flow from object's extraction to synthetic images' creation. The pseudocode for automatic annotation is presented in Algorithm. 1. The input is a set of source image I_{in} and the number range $[lim_s, lim_l]$ of objects in the synthetic image. It the beginning, the order of these source images is shuffled, so that they are combined randomly. Then, each synthetic image is created following a process of extraction, location, combination, and fusion. At the same time, the required annotation information of each object in this synthetic image is transferred along with the whole process. Thus, synthetic images are annotated automatically.

In more detail, the data transformation is performed by using a list *objs* and a dictionary *obj*. *objs* stores the information of all the objects, and *obj* stores that of each object. For example, to create a synthetic image I_s^j with n objects, the information of each object that includes the mask, category, object, and corresponding synthetic image ID is saved in *obj*. Before extracting the next object, the information is transferred to *objs*, and *obj* is reset. According to the mask of all the objects, they are located in the synthetic image according to the method described in Sec. III-C. Meanwhile, their location and area size information is added to *obj*, and *objs* is updated to include all the information. After collecting the information about each object, they are fused into the background to create a synthetic image. Finally, *ant* collects all the information of each synthetic image and is saved as an annotation file.

Note that if source images are taken from multiple conditions, this algorithm should be performed in each condition. Also, keeping the balance between each category and between each condition is important. As we all know, data augmentation facilitates the expansion of the sample size. If the same object appears in several different synthetic images, it is a better data augmentation method compared with traditional rotation and resize operations. Therefore, each source image can be used for multi-times to create different synthetic images. But each source image should be used the same number of times to keep the balance of the dataset.

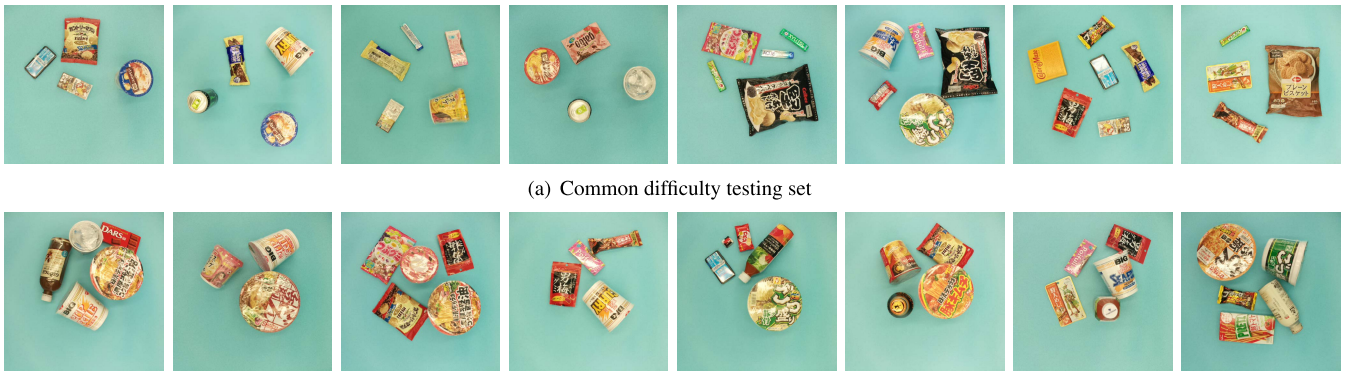
F. DATASET UTILIZING

The created dataset includes a set of synthetic images and a corresponding annotation file. We follow the same form as MS COCO to annotate objects. The area size, category, and location information of each object in the synthetic image are stored using JSON form [51]. In addition, all the data are saved with the same data structure as MS COCO. Therefore, all MS COCO APIs can be used directly for the datasets.

IV. EXPERIMENT

In this section, according to the proposed method, datasets are created to train the popular EfficientDet and YOLOv4 for the object detection task in the vending supermarket. These datasets involve 44 categories of objects, including noodles, snacks, and drinks. Source images are taken from nine illuminations to train the model's adaptation to illumination changes from morning to night. Six source images are taken for each object at each illumination. In this way, 54 source images for each object in total. When creating the training dataset, each source image is used 12 times. Each synthetic image has 3 to 6 objects.

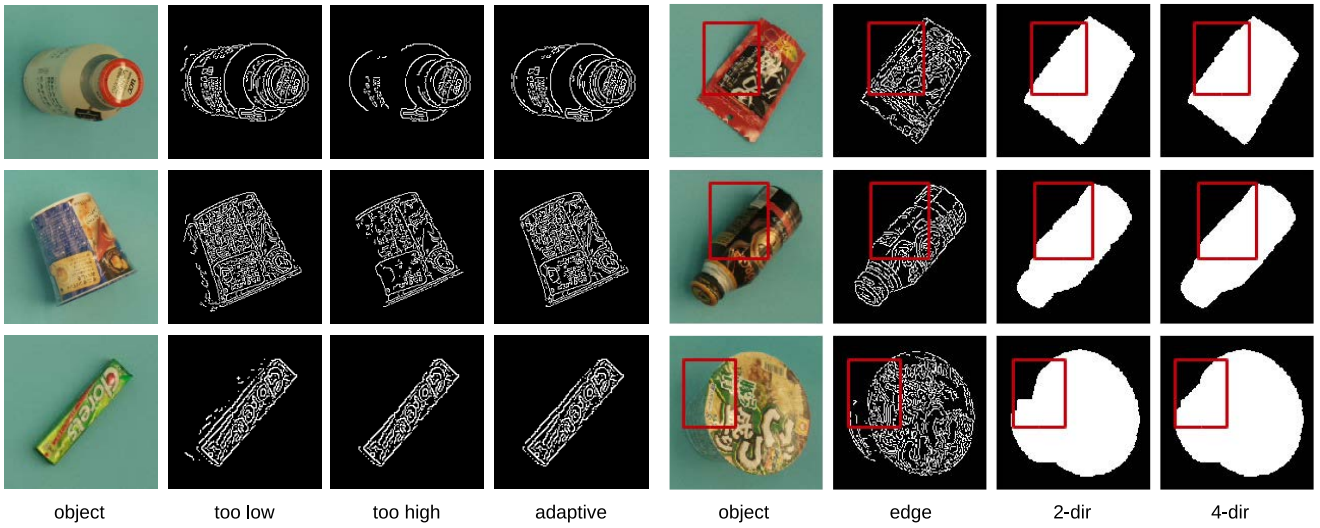
In addition, two testing sets with different difficulties are annotated manually to evaluate the trained model, as shown in Fig. 5. The common difficulty testing set is similar to the training set. The distribution of objects is scattered with little overlap. Differently, objects in the high difficulty testing set are concentrated and have relatively more overlap area size.



(a) Common difficulty testing set

(b) High difficulty testing set

FIGURE 5. Two manual annotated testing sets with different difficulty for evaluating the reliability of the proposed method.



(a) Edge detection with different thresholds

(b) Multi-direction modify the edge to mask the object

FIGURE 6. Component analysis for extracting and masking object from source image.

In the following, the proposed method is analyzed according to the performance of the trained EfficientDet model on the two testing sets. Meanwhile, the popular YOLOv4 is trained to prove the feasibility of the proposed method.

A. OBJECT MASKING ANALYSIS

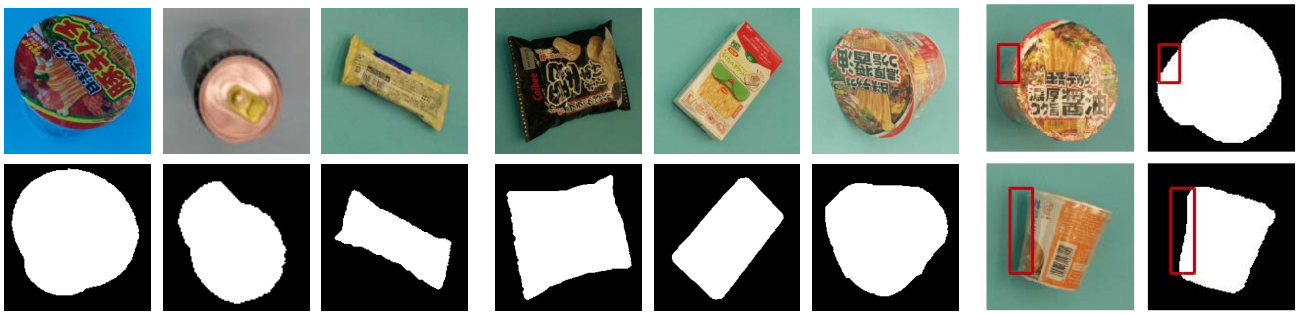
Masking object accurately is important for creating synthetic images. Incorrect masking means that too large or too small regions are considered as the object. E.g., a part of the background near the object is considered as the object, and the object cannot be completely masked. These incorrect masks eventually lead to an unnatural synthetic image being created. To better demonstrate the effectiveness of the proposed method, we illustrate the contributions of the adaptive edge detection method and the multi-direction modification method. Furthermore, we analyze the performance of the proposed method in terms of its adaptability to different backgrounds and illuminations and its coping ability to some classic problems in image capture.

1) EFFECTIVENESS OF ADAPTIVE EDGE DETECTION METHOD

The object is masked according to the edge information. So, accurate detecting edges is the precondition for accurate masking objects. As shown in Fig. 6, we compare the edge result with different thresholds. Too low threshold causes some of the weak edges in the background to be detected. These weak edges are so close to the object that they are considered as objects. As a result, the large-size background is considered as the object incorrectly. On the contrary, a too high threshold results in some weak edges of the object being ignored. This causes some regions of the object to be considered as background, and the object cannot be fully masked. The proposed adaptive method masks weak edges reasonably and achieves accurate edge detection.

2) EFFECTIVENESS OF MULTI-DIRECTION MODIFICATION

High-quality object masking refers that the mask is not only complete but also has a natural boundary. To illustrate



(a) Different backgrounds (b) Different illuminations (c) Classic problems

FIGURE 7. Adaptability in extracting objects from source images with different conditions and problems.

the effectiveness of the multi-direction modification method, we present the mask result in Fig. 6(a). The detected edge is scattered points, while the mask corresponds to a region. After modifying the edge from two directions (0° and 90°), the object is masked as a region instead of scattered points. However, the mask still has an uneven boundary. Further modifying from another two directions (45° and 135°), the shape of the object's mask looks natural.

3) ADAPTABILITY TO DIFFERENT BACKGROUNDS

The same background color is not suitable for all scenes, so the adaptability to different backgrounds determines whether this method can be used widely. To test the adaptability of the proposed method, we do experiments on three types of backgrounds under soft illuminations, and the result masks of objects are shown in Fig. 7(a). The three objects are masked well, and the quality is not affected by the background's color. This result proves that the proposed method has strong adaptability to different backgrounds.

4) ADAPTABILITY TO DIFFERENT ILLUMINATIONS

Illumination cannot be fully controlled in most real scenes since it is sensitive to the surroundings. Therefore, the adaptability to different illuminations suggests the proposed method's stability. To test the adaptability of the proposed method to different illuminations, we do experiments on three illuminations with the same background. The three types of illumination are weak, soft, and strong. The corresponding result masks are shown in Fig. 7(b). Even if the weak illumination causes the background to be so dark and the strong illumination causes the object cannot be seen clearly, it can mask the object as well as that in the soft illumination. Therefore, the proposed method has strong adaptability to different illuminations and performs stably in real scenes.

5) COPING ABILITY TO CLASSIC PROBLEMS IN IMAGE CAPTURE

Reflection and shadows are two phenomena that are difficult to eliminate in image capture, which lead them to be classic problems in image processing. To illuminate the coping ability to the two problems, we present the corresponding result mask in Fig. 7(c). For the top image, the background's color is reflected on the object, which results in an unclear boundary between them. The mask result still separates the object from the background. For the bottom image, there is a shadow on the left side of the object. The corresponding result masks the object and ignores the shadows well. Therefore, the proposed method has a strong coping ability for the two classic problems.

B. SYNTHETIC IMAGE CREATION ANALYSIS

The naturalness of the created synthetic image is important to train the model's generalization ability. The more natural and realistic the synthetic image looks, the better the trained model can perform in real scenes. To illustrate the effectiveness of the proposed method for creating natural synthetic images, we analyze each step from objects to synthetic images and their impacts on practical application.

1) LOCATION ANALYSIS

The random and reasonable distribution of objects in the synthetic image greatly contributes to the training of the network's adaptability to objects' location. To illustrate the rationality of the proposed method, we do experiments that locate objects according to the bounding boxes and the masks, respectively. As shown in Fig. 8(a), objects in the left column of synthetic images are located according to their bounding boxes, while that in the right column are located according to their masks. Since the object is not allowed to locate in the bounding boxes of others, their distribution in the left column of synthetic images is scattered. Conversely, objects in the right column of synthetic images have a low distribution limitation. They are allowed to locate anywhere except the region where other objects are located. As a result, the right column of synthetic images is more natural than the left column.

2) COMBINATION ANALYSIS

Combining objects in the right way is important to create natural synthetic images. To illuminate the advantage of combining objects according to mask, we compare the synthetic image with that combined based on the bounding box,

(a) Object location (b) Object combination (c) Object-background fusion

FIGURE 8. Analysis of each component in synthesizing image from objects.

as shown in Fig. 8(b). The left column of synthetic images that combines objects according to the bounding box introduces parts of object's background. This result in a sharp boundary between the objects and the background. Conversely, the right column combines objects according to their masks. Merging only objects into the synthetic image results in these synthetic images looking natural and are not affected by objects' backgrounds.

3) OBJECT-BACKGROUND FUSION ANALYSIS

The fusion operation further modifies the synthetic image from detail. To illustrate the effectiveness of the fusion operation, we make a comparison as shown in Fig. 8(c). This synthetic image includes four objects. The source image of the center object with pink color is initiated as the initial synthetic image, while the other three objects are merged into this initial synthetic image. Before fusing, clear edges appear in the boundary between object and background, as shown in the top row. After fusing, objects have smooth boundaries, as shown in the middle row. By comparing with the corresponding source images of objects in the bottom row, the synthetic image with fusion operation is closer to the actual image. Therefore, the fusion operation makes the synthetic image look more natural in detail.

4) INFLUENCE ON PRACTICAL APPLICATION

To show the importance of each component, four datasets are created to train the popular object detection neural network, EfficientDet. Regarding the performance in the testing dataset, we do quantitative and qualitative analyses to illuminate the influence on actual application. Regarding the quantitative analyses, we used the same evaluation method as MS COCO, as shown in Table. 1. Average precision (*AP*) is the primary challenge metric, which uses ten IoU thresholds from 0.50 to 0.95. AP_{50} is computed with a single IoU of 0.50, and AP_{75} is computed with a single IoU of 0.75. AP_{75} is the strictest metric.

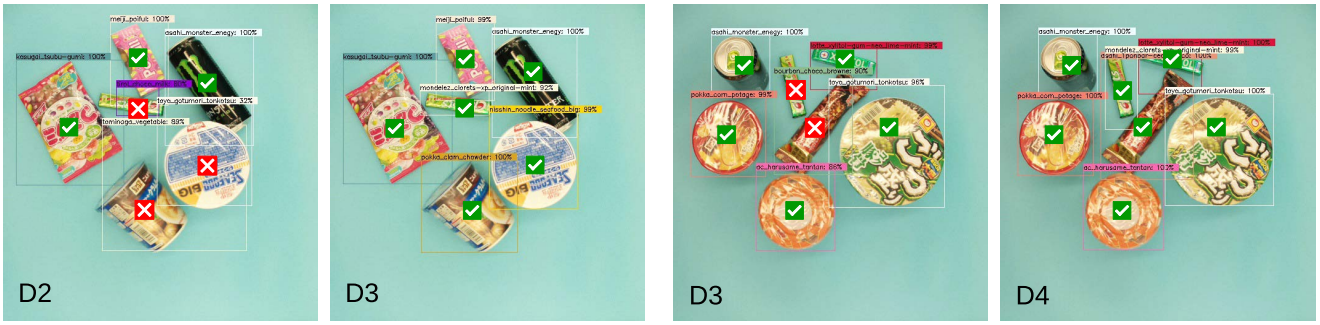
Objects' locations in both D1 and D2 are according to the bounding boxes. However, the combination in D1 and D2 is according to the bounding box and mask, respectively. The *AP* of D2 in both testing sets is a litter higher than that of D1. This is because D2 is combined with masks, which leads the network to learn more about the object without the influence of the object's background. As shown in Fig. 9, this image is from the common difficulty testing set with four objects. The top two objects have very similar objects in this dataset. The bottom two objects are quite different from the other objects. The result shows the model trained with D1 recognized the top two objects as others that are similar to them and correctly detects the bottom objects, while that trained with D2 correctly detects all four objects. Therefore, the model trained by dataset combining objects according to the mask learns the details of objects better and has a stronger ability in distinguishing similar objects.

Objects' combinations in both D2 and D3 are according to the mask. However, the location in D2 and D3 is according to the bounding box and mask, respectively. In the common difficulty, both achieve comparable *AP*. In the high difficulty testing set, *AP* of the model trained by D3 is 15% higher than that of the model trained by D2. There is a huge *AP* gap between the models trained on both datasets. The qualitative analysis result is shown in Fig. 9(b). This image is from the high difficulty testing set, and the distribution of the six objects is concentrated. The result shows that the model trained with D2 only correctly detects three objects, while that trained with D3 correctly detects all these objects. Therefore, the model trained by dataset locating objects according to the mask performs well in both common difficulty and high difficulty testing sets and has a strong generalization ability in complex scenes.

TABLE 1. Quantitative analysis the importance of each components in synthesizing image.

Dataset	Location		Combination		Fusion	Common difficulty			High difficulty		
	Mask	BBox	Mask	BBox		AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
D1		✓		✓		0.890	0.968	0.968	0.641	0.810	0.729
D2		✓	✓			0.891	0.971	0.968	0.662	0.819	0.734
D3	✓		✓			0.883	0.964	0.961	0.816	0.932	0.885
D4	✓		✓		✓	0.897	0.987	0.987	0.816	0.931	0.909

(a) D1 VS D2



(b) D2 VS D3 (c) D3 VS D4

FIGURE 9. Qualitative analysis the importance of each components in synthesizing image.

Compared with D3, D4 has a fusion operation after merging these objects into the initial synthetic image. The quantitative result in Table. 1 shows that D3 and D4 achieve comparable AP, but that of D4 is a little higher than D3. This is because D4 eliminates some strong features that shouldn't exist. The qualitative analysis result is shown in Fig. 9(c). This image is from high difficult testing set with seven objects. The result shows that the model trained with D3 misrecognizes two objects that have a relatively large overlap boundary, while that trained with D4 correctly detects all the seven objects. Therefore, fusion operation improves AP slightly, especially for objects with overlap boundaries with others.

C. THE FEASIBILITY OF PROPOSED METHOD

To illustrate the proposed method is effective for different object detection frameworks, the popular YOLOv4 is trained on the created dataset in addition to the basic B0 of EfficientDet. The result on each dataset is shown in Table. 2. Under the strictest metric AP_{75}, both the trained models achieve higher than 95% accuracy in the common difficulty testing set and hold the strong generalization ability in the high difficulty testing set. Therefore, the proposed method maintains the feasibility of creating datasets for training various object detection networks.

TABLE 2. Result of different networks trained with the created dataset.

Network	Dataset	AP	AP_{50}	AP_{75}
EfficientDet (B0)	Common difficulty	0.897	0.987	0.987
YOLOv4		0.917	0.985	0.985
EfficientDet (B0)	High difficulty	0.816	0.931	0.909
YOLOv4		0.901	0.989	0.981

V. CONCLUSION

This paper presents a novel method that automatically creates datasets for training object detection networks. This method proved that a dataset can be created automatically by using the data flow from object extraction to image synthesis. The

trained model's strong adaption and generalization ability suggest that the proposed method can create reliable datasets for object detection tasks. In the future, we plan to explore the possibility of automatically creating datasets with complex backgrounds.

REFERENCES

[1] J. Gao, J. Yi, and Y. L. Murphey, "Attention-based global context network for driving maneuvers prediction," *Mach. Vis. Appl.*, vol. 33, no. 4, pp. 1–11, May 2022, doi: 10.1007/s00138-022-01305-x.

[2] M. Zhu, S. Gong, Z. Qian, S. Serikawa, and L. Zhang, "Person re-identification in the real scene based on the deep learning," *Artif. Life Robot.*, vol. 26, no. 4, pp. 396–403, Jul. 2021, doi: 10.1007/s10015-021-00689-9.

[3] S. Zhou, M. Zhu, Z. Li, H. Li, M. Mizumachi, and L. Zhang, "Self-supervised monocular depth estimation with occlusion mask and edge awareness," *Artif. Life Robot.*, vol. 26, no. 3, pp. 354–359, May 2021, doi: 10.1007/s10015-021-00685-z.

[4] L. Tabelini, R. Berriel, T. M. Paix ao, A. F. De Souza, C. Badue, N. Sebe, and T. Oliveira-Santos, "Deep traffic sign detection and recognition without target domain real images," *Mach. Vis. Appl.*, vol. 33, no. 50, pp. 1–12, Apr. 2022.

[5] S. Zhou, Z. Yang, M. Zhu, H. Li, S. Serikawa, M. Mizumachi, and L. Zhang, "Higher accuracy self-supervised visual odometry with reliable projection," *Artif. Life Robot.*, vol. 27, no. 3, pp. 568–575, Jun. 2022, doi: 10.1007/s10015-022-00766-7.

[6] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "CNN-RNN: A unified framework for multi-label image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2285–2294.

[7] H. Phan, F. Andreotti, N. Cooray, O. Y. Chen, and M. D. Vos, "Joint classification and prediction CNN framework for automatic sleep stage classification," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 5, pp. 1285–1296, May 2019, doi: 10.1109/TBME.2018.2872652.

[8] M. Csontho, A. Rovid, and Z. Szalay, "Significance of image features in camera-LiDAR based object detection," *IEEE Access*, vol. 10, pp. 61034–61045, 2022, doi: 10.1109/ACCESS.2022.3181137.

[9] A. Ozcan and O. Cetin, "A novel fusion method with thermal and RGB-D sensor data for human detection," *IEEE Access*, vol. 10, pp. 66831–66843, 2022, doi: 10.1109/ACCESS.2022.3185402.

[10] A. Dhillon and G. K. Verma, "Convolutional neural network: A review of models, methodologies and applications to object detection," *Prog. Artif. Intell.*, vol. 9, no. 2, pp. 85–112, Dec. 2019, doi: 10.1007/s13748-019-00203-0.

[11] R. Yang and Y. Yu, "Artificial convolutional neural network in object detection and semantic segmentation for medical imaging analysis," *Frontiers Oncol.*, vol. 11, Mar. 2021, Art. no. 638182, doi: 10.3389/fonc.2021.638182.

[12] N. Paluru, A. Dayal, H. B. Jenssen, T. Sakinis, L. R. Cenkeramaddi, J. Prakash, and P. K. Yalavarthy, "Anam-Net: Anamorphic depth embedding-based lightweight CNN for segmentation of anomalies in COVID-19 chest CT images," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 3, pp. 932–946, Mar. 2021, doi: 10.1109/TNNLS.2021.3054746.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[14] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2017, pp. 7263–7271.

[15] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[16] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[17] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10781–10790.

[18] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[19] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2014, pp. 740–755.

[20] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010, doi: 10.1007/s11263-009-0275-4.

[21] Y. Lv, Y. Fang, W. Chi, G. Chen, and L. Sun, "Object detection for sweeping robots in home scenes (ODSR-IHS): A novel benchmark dataset," *IEEE Access*, vol. 9, pp. 17820–17828, 2021, doi: 10.1109/ACCESS.2021.3053546.

[22] S. Wei, X. Zeng, Q. Qu, M. Wang, H. Su, and J. Shi, "HRSID: A high-resolution SAR images dataset for ship detection and instance segmentation," *IEEE Access*, vol. 8, pp. 120234–120254, 2020, doi: 10.1109/ACCESS.2020.3005861.

[23] S. Shao, Z. Li, T. Zhang, C. Peng, G. Yu, X. Zhang, J. Li, and J. Sun, "Objects365: A large-scale, high-quality dataset for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8430–8439.

[24] C. Sakaridis, D. Dai, S. Hecker, and L. Van Gool, "Model adaptation with synthetic and real data for semantic dense foggy scene understanding," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 687–704.

[25] D. Acuna, H. Ling, A. Kar, and S. Fidler, "Efficient interactive annotation of segmentation datasets with Polygon-RNN++," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 859–868.

[26] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: A database and Web-based tool for image annotation," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 157–173, 2008, doi: 10.1007/s11263-007-0090-8.

[27] C. P. Austin, J. F. Battey, A. Bradley, M. Bucan, M. Capecchi, F. S. Collins, W. F. Dove, G. Duyk, S. Dymecki, and J. T. Eppig, "The knockout mouse project," *Nature Genet.*, vol. 36, no. 9, pp. 921–924, Sep. 2004, doi: 10.1038/ng0904-921.

[28] S. D. Jain and K. Grauman, "Predicting sufficient annotation strength for interactive foreground segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1313–1320.

[29] Y. Sun, M. Liu, and M. Q.-H. Meng, "Active perception for foreground segmentation: An RGB-D data-based background modeling method," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 4, pp. 1596–1609, Oct. 2019, doi: 10.1109/TASE.2019.2893414.

[30] Z. Zhong, B. Zhang, G. Lu, Y. Zhao, and Y. Xu, "An adaptive background modeling method for foreground segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1109–1121, May 2017, doi: 10.1109/TITS.2016.2597441.

[31] L. Wang, G. Chen, D. Shi, Y. Chang, S. Chan, J. Pu, and X. Yang, "Active contours driven by edge entropy fitting energy for image segmentation," *Signal Process.*, vol. 149, pp. 27–35, Aug. 2018, doi: 10.1016/j.sigpro.2018.02.025.

[32] M. Á. Castillo-Martínez, F. J. Gallegos-Funes, B. E. Carvajal-Gámez, G. Urriolagoitia-Sosa, and A. J. Rosales-Silva, "Color index based thresholding method for background and foreground segmentation of plant images," *Comput. Electron. Agricult.*, vol. 178, Nov. 2020, Art. no. 105783, doi: 10.1016/j.compag.2020.105783.

[33] M. Xiaobo and Y. Jing, "Research on object-background segmentation of color image based on LabVIEW," in *Proc. IEEE Int. Conf. Cyber Technol. Autom., Control, Intell. Syst.*, Mar. 2011, pp. 190–194.

[34] E. N. Mortensen and W. A. Barrett, "Intelligent scissors for image composition," in *Proc. 22nd Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*, 1995, pp. 191–198.

[35] C. Rother, V. Kolmogorov, and A. Blake, ""GrabCut" interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, Aug. 2004, doi: 10.1145/1015706.1015720.

[36] S. Y. Elhabian, K. M. El-Sayed, and S. H. Ahmed, "Moving object detection in spatial domain using background removal techniques-state-of-art," *Recent Patents Comput. Sci.*, vol. 1, pp. 32–54, Jan. 2008.

[37] M.-H. Yang, C.-R. Huang, W.-C. Liu, S.-Z. Lin, and K.-T. Chuang, "Binary descriptor based nonparametric background modeling for foreground extraction by using detection theory," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 4, pp. 595–608, Apr. 2015, doi: 10.1109/TCSVT.2014.2361418.

[38] C. Li, C.-Y. Kao, J. C. Gore, and Z. Ding, "Implicit active contours driven by local binary fitting energy," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–7.

[39] R. Kaur, M. Juneja, and A. K. Mandal, "A hybrid edge-based technique for segmentation of renal lesions in CT images," *Multimedia Tools Appl.*, vol. 78, no. 10, pp. 12917–12937, May 2019, doi: 10.1007/s11042-018-6421-7.

[40] Y. Zhang, "Understanding image fusion," *Photogram. Eng. Remote Sens*, vol. 70, no. 6, pp. 657–661, 2004.
[41] A. P. James and B. V. Dasarathy, "Medical image fusion: A survey of the state of the art," *Inf. Fusion*, vol. 19, pp. 4–19, Sep. 2014, doi: 10.1016/j.inffus.2013.12.002.
[42] F. Zhan, H. Zhu, and S. Lu, "Spatial fusion GAN for image synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3653–3662.
[43] B. Xin, Y. Tian, Y. Wang, and W. Gao, "Background subtraction via generalized fused lasso foreground modeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4676–4684.
[44] K. Cui, G. Zhang, F. Zhan, J. Huang, and S. Lu, "FBC-GAN: Diverse and flexible image synthesis via foreground-background composition," 2021, *arXiv:2107.03166*.
[45] S. Li, J. T. Y. Kwok, I. W. H. Tsang, and Y. Wang, "Fusing images with different focuses using support vector machines," *IEEE Trans. Neural Netw.*, vol. 15, no. 6, pp. 1555–1561, Nov. 2004, doi: 10.1109/TNN.2004.837780.
[46] M. Kumar and S. Dass, "A total variation-based algorithm for pixel-level image fusion," *IEEE Trans. Image Process.*, vol. 18, no. 9, pp. 2137–2143, Sep. 2009, doi: 10.1109/TIP.2009.2025006.
[47] R. Shen, I. Cheng, J. Shi, and A. Basu, "Generalized random walks for fusion of multi-exposure images," *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3634–3646, Dec. 2011, doi: 10.1109/TIP.2011.2150235.
[48] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986, doi: 10.1109/TPAMI.1986.4767851.
[49] S. Wazarkar, B. N. Keshavamurthy, and A. Hussain, "Region-based segmentation of social images using soft KNN algorithm," *Proc. Comput. Sci.*, vol. 125, pp. 93–98, Dec. 2018, doi: 10.1016/j.procs.2017.12.014.
[50] M. V. D. Bergh, X. Boix, G. Roig, B. D. Capitani, and L. V. Gool, "Seeds: Superpixels extracted via energy-driven sampling," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2012, pp. 13–26.
[51] *The Javascript Object Notation (JSON) Data Interchange Format*, Standard RFC 7159, 2014.

SHI ZHOU (Member, IEEE) was born in Shaodong, Hunan, China, in 1994. She received the B.S. and M.S. degrees in mechanical engineering and automation from Northeastern University, China, in 2017 and 2020, respectively. She is currently pursuing the Ph.D. degree with the Department of Electrical and Electronic Engineering, Kyushu Institute of Technology, Japan. Her research interests include computer vision, image processing, and artificial intelligence. She is a member of IIAE.

ZIJUN YANG was born in Ningxia, China, in 1992. She received the B.S. degree from Suzhou University, China, in 2016, and the M.S. degree in electrical and electronic engineering from the Kyushu Institute of Technology, Japan, in 2021, where she is currently pursuing the Ph.D. degree with the Department of Electrical and Electronic Engineering. Her research interests include image processing, signal processing, and artificial intelligence.

MIAOMIAO ZHU was born in Shanxi, China, in 1981. She received the M.S. degree in engineering from the Xi'an University of Architecture and Technology, China, in 2008, and the Ph.D. degree in electrical and electronic engineering from the Kyushu Institute of Technology, Japan, in 2021. Since 2008, she has been a Research Assistant with the School of Computer Science and Engineering, Changshu Institute of Technology, China. Her research interests include image processing, image enhancement, and wireless sensor networks.

HE LI received the B.S., M.S., and Ph.D. degrees in mechanical engineering and automation from Northeastern University, China, in 1999, 2002, and 2005, respectively. He is currently a Professor with Northeastern University. He is the author of four books, more than 60 articles, and more than ten inventions. His research interests include nonlinear vibration, mechanical system dynamics, and machine tool dynamics. He is a member of the Liaoning Association for Science and Technology and the Chinese Mechanical Engineering Society.

SEIICHI SERIKAWA was born in Kumamoto, Japan, in June 1961. He received the B.S. and M.S. degrees in electronic engineering from Kumamoto University, in 1984 and 1986, respectively, and the Ph.D. degree in electronic engineering from the Kyushu Institute of Technology, Japan, in 1994. He is currently a Professor with the Department of Electrical Engineering and Electronics, Kyushu Institute of Technology. His research interests include computer vision, sensors, and robotics. He is a member of IEICE, IEEJ, IPSJ, and IIAE.

MITSUNORI MIZUMACHI (Member, IEEE) received the B.S. degree in design from the Kyushu Institute of Design, in 1995, and the Ph.D. degree in information science from the Japan Advanced Institute of Science and Technology (JAIST), in 2000. He is currently an Associate Professor with the Kyushu Institute of Technology. His research interests include acoustic information processing and statistical signal processing. He is a member of AES, ASA, ASJ, IEICE, and RISP.

LIFENG ZHANG (Member, IEEE) received the B.S. degree in electronic engineering from Southeast University, in 1994, and the M.S. and Ph.D. degrees in electrical engineering from the Kyushu Institute of Technology, Japan, in 1999 and 2001, respectively. He is currently a Professor with the Kyushu Institute of Technology. His current research interests include computer vision, image processing, biometrics authentication, and remote sensing systems. He is a member of IIAE and IEICE.