Training Image Synthesis for Shelf Item Detection reflecting Alignments of Items in Real Image Dataset

Tomokazu Kaneko, Ryosuke Sakai, Soma Shiraishi
NEC Visual Intelligence Research Laboratories
211-8666, Kawasaki, Kanagawa, Japan
{tomokazu-kaneko, rsakai_zzkot, s-shiraishi}@nec.com

ABSTRACT

We propose a novel cut-and-paste approach to synthesize a training dataset for shelf item detection, reflecting the alignments of items in the real image dataset. The conventional cut-and-paste approach synthesizes large numbers of training images by pasting foregrounds on background images and is effective for training object detection. However, the previous method pastes foregrounds on random positions of the background, so the alignment of items on shelves is not reflected, and unrealistic images are generated. Generating realistic images that reflect actual positional relationships between items is necessary for efficient learning of item detection. The proposed method determines the pasting positions for the foreground images by referring to the alignment of the items in the real image dataset, so it can generate more realistic images that reflect the alignment of the real-world items. Since our method can synthesize more realistic images, the trained models can perform better.

Keywords

Object detection, Training data synthesis, Retail item recognition, Automatic annotation

1 INTRODUCTION

Image-based retail item recognition contributes to the efficient operation of stores. For example, monitoring item shelves with surveillance cameras can provide out-of-stock detection or planogram analysis services. The automatic method to create item image databases from shelf images has also been proposed in [6]. For these applications, item detection models are required to localize the position of items in the captured images.

Training data annotated with the bounding box of the item position is required to train item detection models. However, the annotation cost is high due to many items being densely aligned on the shelves. The SKU-110K [9] is a public dataset for item detection, but it only contains images taken in a specific country or region, which means it cannot support items sold locally.

The cut-and-paste method [4] is a method for synthesizing large amounts of training data for object detection. The cut-and-paste method can generate various patterns of images at a low cost by pasting foreground images onto background images. Therefore, by pasting images of local items onto the background shelf image, the training dataset for local item detection can be generated without shooting the items on shelves in real-world stores.

Conventional cut-and-paste methods paste the foreground image at a random position in the background image. Such random pasting methods are effective when objects appear in random positions in the image. However, in the case of shelf item detection, the items are regularly aligned, and there is less occlusion between items. As items can have complex textures, irregular occlusion between items due to pasting in random positions makes the boundaries and textures of the items too complex and difficult for training.

This paper proposes a new cut-and-paste method that reflects the alignments of the item positions. The proposed method realistically arranges shelf images by referring to the positional information of items from a real image dataset to determine the position to paste them (Figure 1). Using public datasets as reference datasets, no additional annotation costs are required, and realistic training data can be generated at a low cost.

Realistic images contribute to the training of high-performance detection models. In particular, the proposed method reproduces the regular alignment of items on real-world shelves, which allows the correct boundaries of items to be learned without generating too complex occlusions. We show that the proposed method can generate more realistic images, and the model trained on these images performs better in evaluation experiments on real store images.

https://www.doi.org/10.24132/CSRN.3301.11

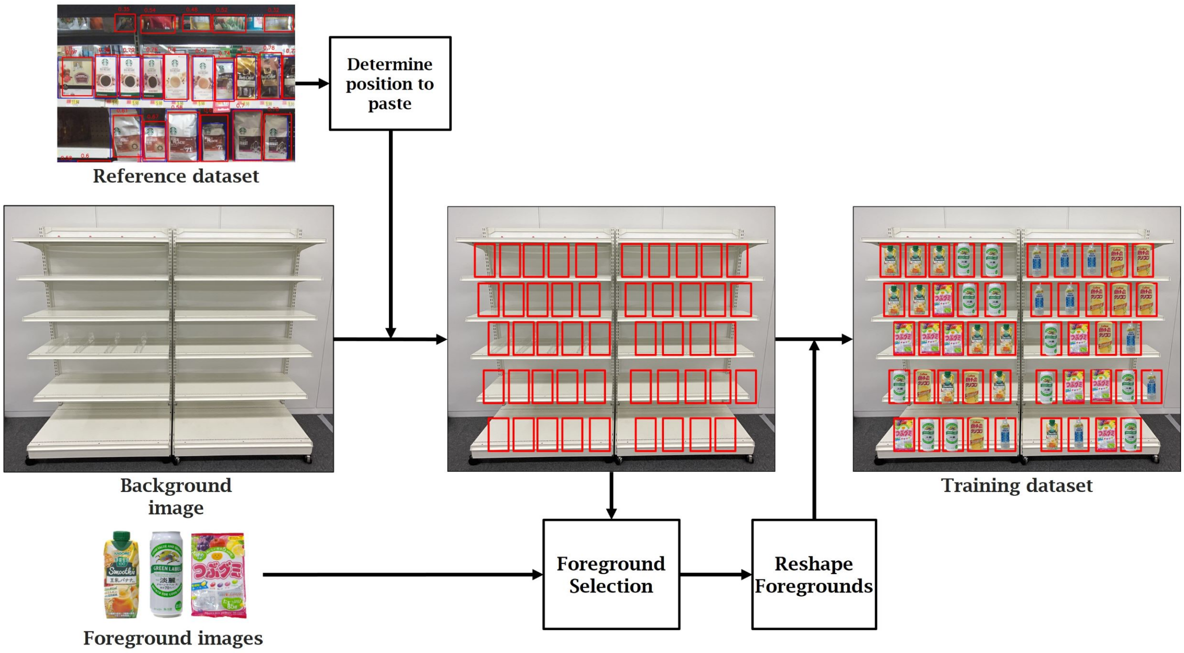


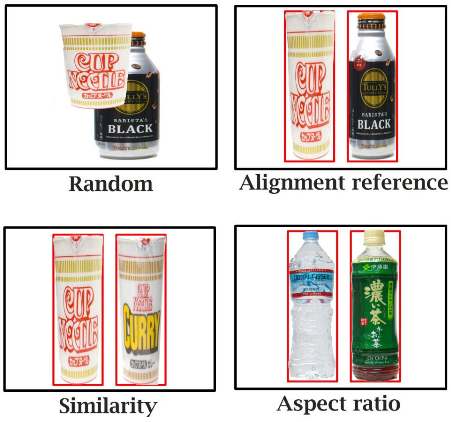
Figure 1: Outline of our approach. The details are explained in Section 3.

2 RELATED WORK

Cut-and-paste methods are proposed for many applications to synthesize a training dataset since it is low cost but effective [4, 5, 8, 10, 14, 19, 20, 21]. In these approaches, foreground images are pasted on a background image and positions of the pasted images are annotated automatically. Several papers reported the effect of the above approaches in industrial applications [14, 19]. It was shown that the cut-and-paste approach was effectively applied to an item recognition task for a self-checkout system. Since these methods determine positions on which foregrounds are pasted randomly, they are effective in a situation when target objects are placed on random positions such as a self-checkout system. However, the methods fail to synthesize realistic images in cases where the target objects are arranged following a pattern, as in the items on a shelf. The model, then, fails to learn the relationships between objects using the data.

There are advanced approaches based on a cut-and-paste method, which consider the positions on which the foreground images are pasted [1, 3, 7]. In [7], they use a depth sensor to estimate the support surface on which a real object is likely placed , floor and desk in background images. Foreground images are pasted on these surfaces, and realistic images are synthesized consequently. However, we need to prepare a depth sensor to use this approach, and moreover, generated images do not reflect the positional relationship between objects. The method in [1] also estimates realistic positions in a background image on which foregrounds are pasted. Its target is driving scenes, so the suitable positions to place car images are on the road. Since there are many driving scene datasets in public and the road is distinctive, the road estimator can be made robust through training on RGB images. It becomes strict when there are many variations of backgrounds and enough background images cannot be collected. For generic tasks, the approach in [3] is effective. In their approach, the context convolutional neural network (CNN) that estimates the context of backgrounds and foreground images is trained and selects the foreground image that is suitable to paste on each position of a background. In this way, extra sensors are not necessary, and the estimator can learn the context of the image from a generic image set. However, since this approach learns the relationship between foregrounds and backgrounds, it does not work on the scene like objects placed densely and the background is covered such as planogram analysis, and this method also does not reflect the positional relationship between foregrounds.

A 3D simulator is another way to synthesize realistic images. On a 3D simulator, we can place objects anywhere, and using a physics engine, we can simulate stability or interactions between objects. In fact, 3D simulators are used to synthesize training images for object detection [2, 11, 12, 13, 15, 18]. In these methods, the positions of objects in rendered images are annotated automatically, and therefore, a large amount of training data is generated. However, to reproduce target scenes on a 3D simulator, we need 3D models of target objects and backgrounds. Since preparing 3D models is



Random

Alignment reference

Similarity

Aspect ratio

Figure 2: Differences between approaches. The random approach pastes foregrounds on random positions, thus, there is no alignment and foregrounds may occlude each other. The alignment reference approach reproduces a realistic alignment of objects; however, foregrounds are reshaped unrealistically. In the similarity approach, objects in the same category are placed in the neighborhood. The aspect ratio approach selects foregrounds fit to bounding boxes, thus, its aspect ratio does not change so much, and therefore, a realistic appearance is achieved

expensive, the cost of covering many objects is higher than that in the cut-and-paste method. This is more critical in the item detection task for retail stores, in which hundreds of new products are introduced every week.

3 PROPOSED APPROACH

We propose a novel approach to synthesize a dataset for shelf item detection based on the cut-and-paste approach reflecting the alignments of real-world items. First, in this section, we explain the algorithm to determine positions on which foregrounds are pasted by referring to the alignment, which is the core idea of our approach. Then, we explain two specific methods for foreground selection to synthesize more realistic images: the first is based on the object similarity, the second is based on the aspect ratio of the bounding box and the foreground image. Finally, we explain how to combine the two methods. The differences between each approach are summarized in Figure 2.

3.1 Cut and paste referring to object alignment

Figure 1 shows the outline of our approach. The proposed method uses three datasets. The reference dataset is an image dataset of item shelves taken in real stores. The images of the reference dataset are taken at a specific location, so the items we want to detect do not appear. The reference dataset is annotated with a bounding boxes representing the item positions. The proposed method uses the annotation data and size information $(W^k_{\text{ref}}, H^k_{\text{ref}})$ of each image. If the size information is provided as metadata, preparing images of the item shelf is unnecessary for generating process. The background images are the image set of empty shelves. The foreground images are the image set of the items to be detected in which the foreground area has been cropped out.

We define the following notations for the background image set $\mathscr{B}$, foreground item image set $\mathscr{A}$, and bounding box annotations of the reference dataset $\mathscr{T}$ as,

$$\mathscr{B} = \left\{ b^k \in \mathbb{R}^{W^k_{\text{bg}} \times H^k_{\text{bg}} \times 3} \right\}_{k=1}^{N_{\text{bg}}}, \tag{1}$$

$$\mathscr{A} = \left\{ a^k \in \mathbb{R}^{W^k_{\text{fg}} \times H^k_{\text{fg}} \times 3}, m^k \in [0,1]^{W^k_{\text{fg}} \times H^k_{\text{fg}}} \right\}_{k=1}^{N_{\text{fg}}}, \tag{2}$$

$$\mathscr{T} = \left\{ T^k \in \mathbb{R}^{4 \times N_k} \right\}_{k=1}^{N_{\text{ref}}}. \tag{3}$$

Where $N_{\text{bg}}, N_{\text{fg}}, N_{\text{ref}}$, and N_k denote the number of background images, foreground images, reference dataset images, and objects in the k-th reference image, respectively. Furthermore, where $W^k_{\text{bg}}, H^k_{\text{bg}}, W^k_{\text{fg}}$, and H^k_{fg} are the width and height of the background image and the foreground image, respectively, for the k-th image, taking into account that they may differ from image to image. The foreground image is a transparent image to be pasted onto the background, where m^k represents the alpha mask of the foreground.

At first, in the pasting process, the proposed method selects one background image b^k and one reference annotation $T^k = \{(x^k_l, y^k_l, w^k_l, h^k_l)\}_{l=1}^{N_k}$ at random by the uniform distribution. Next, the method selects one bounding box $t^k_l = (x^k_l, y^k_l, w^k_l, h^k_l) \in T^k$ and determines the foreground pasting position by resizing the bounding box to fit the background image,

$$(x_l, y_l, w_l, h_l) = \left(\frac{W^k_{\text{bg}}}{W^k_{\text{ref}}} x^k_l, \frac{H^k_{\text{bg}}}{H^k_{\text{ref}}} y^k_l, \frac{W^k_{\text{bg}}}{W^k_{\text{ref}}} w^k_l, \frac{H^k_{\text{bg}}}{H^k_{\text{ref}}} h^k_l \right), \tag{4}$$

where W^k_{ref} and H^k_{ref} denote width and height of selected reference image, respectively. After that, one foreground image $(a^l, m^l) \in \mathscr{A}$ is selected and resized to fit into the bounding box,

$$(\tilde{a}^l, \tilde{m}^l) = \left(R_{w_l,h_l}(a^l), R_{w_l,h_l}(m^l) \right), \tag{5}$$

where $R_{w,h}(\cdot,\cdot)$ denotes the function that resizes an image to $w \times h$ size. Finally, the method pastes the resized image at the bounding box position with alpha blending. This is repeated until there are no more empty bounding boxes.

$$I_{ij} = \left(1 - \sum_{l=1}^{N_k} P_{x_l,y_l}(\tilde{m}^l_{ij}) \right) \cdot b^k_{ij} + \sum_{l=1}^{N_k} P_{x_l,y_l}(\tilde{m}^l_{ij} \cdot \tilde{a}^l_{ij}), \tag{6}$$

where i and j denote pixel coordinates of images and $P_{x,y}(\cdot)$ denotes the offset function, which shifts the image coordinates i and j to the pasting coordinates x and y.

Images synthesized in this way reflect the alignment in the real scene, and therefore, they achieve more realistic appearances than randomly synthesized images.

3.2 Foreground selection based on object similarity

Real-world shelves have a feature that similar items are placed in the neighborhood of each other. For example, items on a beverage shelf may be collected from the same category, such as coffee, tea, or milk, in which similarly shaped bottles. To reproduce this appearance, we propose a method to select similar images when selecting foreground images.

To paste similar images close to each other, the proposed method first selects a bounding box in the neighborhood of the already pasted bounding box t_l^k,

$$(x_{l+1}, y_{l+1}, w_{l+1}, h_{l+1}) = \underset{t^k \in \bar{T}^k}{\arg\min}\, d\left(t_l^k, t^k\right), \tag{7}$$

where $\bar{T}^k \subset T^k$ represents the set of bounding boxes to which the foreground has not yet been pasted, and $d(\cdot,\cdot)$ is a function that calculates the distance between two bounding boxes. We use Euclidean distance between centers of bounding boxes. The foreground image is then selected from similar images to image (a^l, m^l) which was pasted to the neighboring bounding box,

$$(a^{l+1}, m^{l+1}) = \underset{(a,m) \in \bar{\mathscr{A}}}{\arg\max}\, S(a, a^l), \tag{8}$$

where $\bar{\mathscr{A}} \subset \mathscr{A}$ represents the set of foreground images that have not been pasted, and $S(\cdot,\cdot)$ is a function that outputs the similarity between the two images. The proposed method pastes the selected foreground image onto the selected bounding box position, as described in Section 3.1.

There are several ways to select a similar foreground. One way is to select from the same category or product code. Another way is to use a feature extractor and measure the feature similarity of extracted feature vectors of foreground images. In the following experiments in section 4, we select similar foregrounds by selecting the images of the same product code but viewed from different angles. This way, we can synthesize the appearance of the shelf on which the same objects are placed next to each other facing differently bit by bit.

3.3 Foreground selection based on aspect ratio

The bounding boxes in the reference dataset have various aspect ratios, and the aspect ratios change due to the transformation of Equation (4). The foreground image dataset may also contain images with varying aspect ratios. Due to these factors, the aspect ratio of the foreground image changes during the transformation in Equation (5).

Algorithm 1 Cut-and-paste procedure referring object alignment, similarity and aspect ratio.

```
Input: $\mathscr{A}$, $b^k$, $\bar{T}^k$, $p \in [0,1]$
SimilarityFlag $\leftarrow$ False
Last_bbox $\leftarrow$ [ ]
annotation $\leftarrow$ [ ]
for l $\leftarrow$ 1 ... len($\bar{T}^k$)
    if SimilarityFlag then
        Select $t_l^k$ from $\bar{T}^k$ by Eq. (7)
        Select $(a^l, m^l)$ from $\mathscr{A}$ by Eq. (8)
    else
        Randomly select $t_l^k$ from $\bar{T}^k$
        Select $(a^l, m^l)$ from $\mathscr{A}$ by Eq. (9)
    end if
    $t_l \leftarrow$ Reshape $t_l^k$ by Eq. (4)
    $(\tilde{a}^l, \tilde{m}^l) \leftarrow$ Reshape $(a^l, m^l)$ to fit $t_l$ by Eq. (5)
    Paste $\tilde{a}^l$ at position $t_l$ in $b^k$
    Append $t_l$ to annotation
    rand $\leftarrow$ a random number between 0 and 1
    if rand $< p$ then
        SimilarityFlag $\leftarrow$ True
    else
        SimilarityFlag $\leftarrow$ False
    end if
end for
return $b^k$, annotations
```

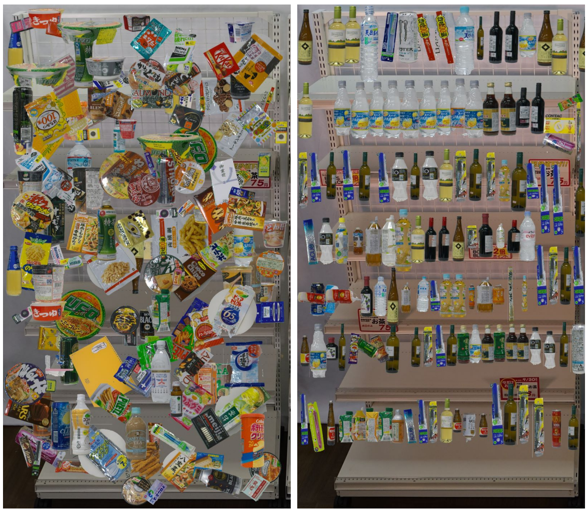
To synthesize a realistic image, the aspect ratio of the foreground image must not change too much from the original. The following algorithm selects a foreground image whose aspect ratio is close to the aspect ratio of the bounding box. The algorithm first calculates the aspect ratio of the bounding box and selects a foreground image with a similar aspect ratio.

$$(a^l, m^l) = \underset{(a,m) \in \bar{\mathscr{A}}}{\arg\min}\, |r(a) - w_l/h_l|, \tag{9}$$

where $r(\cdot)$ is a function to calculate the aspect ratio of the image. After that, we reshape the foreground image to fit the bounding box and paste the foreground. This approach allows the foreground image to be pasted to fit into a bounding box while preserving its aspect ratio. Thus, the synthesized image becomes more realistic with no extremely reshaped items.

3.4 Inclusion of all approaches

The method containing all of the above approaches is shown in Algorithm 1. Our method basically selects a foreground image in accordance with its aspect ratio.



Random [4] Ours

Figure 3: Examples of synthesized images

After pasting one foreground, the algorithm determines whether to select a foreground in accordance with similarity probabilistically. In this way, two foreground selection processes can be included in one algorithm.

4 EXPERIMENTS

We evaluate the proposed approach on the task of shelf item detection. The purpose of the proposed approach is to train a better object detector on synthesized images. To evaluate from this perspective, we compare detection scores of object detectors trained on the synthesized images by the baselines and the proposed approach. To evaluate in a realistic situation, we use shelf images shot in real stores as the evaluation dataset.

4.1 Training dataset

We prepare a training dataset in addition to the public dataset. We add the synthesized dataset to the public dataset to train from both real and synthesized images. This is because there is a domain gap between real and synthesized images and training only on synthesized images suffers from this gap. Training on both domains mitigates this adverse effect.

We adopt SKU-110K as the base dataset. Since SKU-110K does not contain images shot in locale-specific stores or shot from angles of surveillance cameras, the model trained only on SKU-110K does not work well enough in these situations. By adding synthesized data from foreground images of locale-specific items or background images of surveillance angles, the trained models become robust to the uncovered situation.

	Front	Upper	Upper-left	average
SKU-only [9]	0.946	0.893	0.712	0.850
Random [4]	0.945	0.880	0.735	0.853
Ours	**0.951**	**0.894**	**0.755**	**0.866**

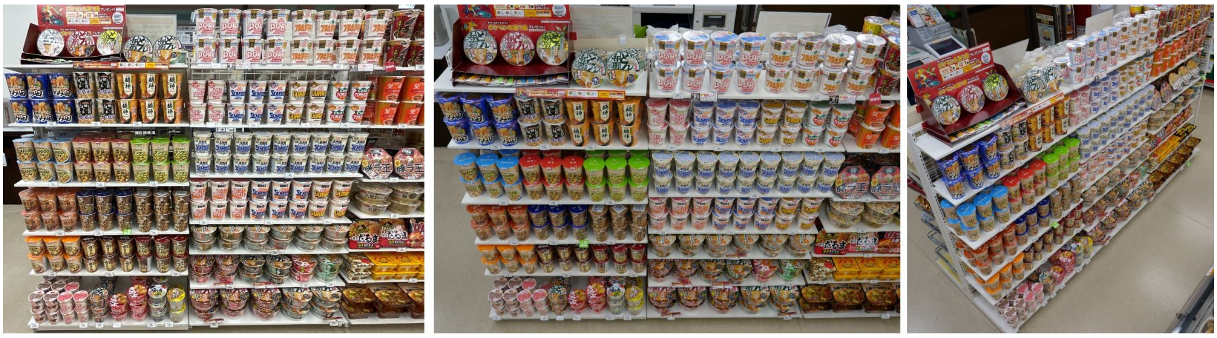
Table 1: Detection scores (AP_{50}) of trained models. SKU-only, Random, and Ours indicate the methods to synthesize training images. Front, Upper, and Upper-left indicate camera angles of the evaluation dataset. All of the scores are the means of three trials of training on different random seeds.

We use item images shot on a turntable as foregrounds. This image set contains 39,559 images of 1,000 items. Each item is captured from multiple orientations by rotating the turntable. We cut out foregrounds by GrabCut [16] from the captured images. These cut out images of items are used as the foreground images. We use images of empty shelves as backgrounds. This set contains 989 images of five types of shelves. These images are shot from various angles and under various lighting conditions. Using the above foreground and background images, we synthesize a training dataset by each approach. We synthesize 1,000 images for training by each approach and add to SKU-110K training dataset that contains 8,185 real images.

We compare three methods: SKU-only [9], random [4], and our approach. SKU-only means training on SKU-110K dataset only. The random approach is the cut-and-paste method whose pasting position is determined randomly. In the random approach, we paste 147 foregrounds on average on one background image. This number is the same as the average number of objects in one image of SKU-110K. With this condition, the number of foreground objects contained in one training image is the same among comparison methods. In the proposed method, we also use SKU-110K as a reference dataset. We set the parameter p to 0.5, which is the probability of selecting a foreground by similarity and pasting it on the neighborhood. To increase the variation of the appearance of the foreground, for all comparison methods, we randomly rotate the foreground image with a probability of 0.1 when pasting it.

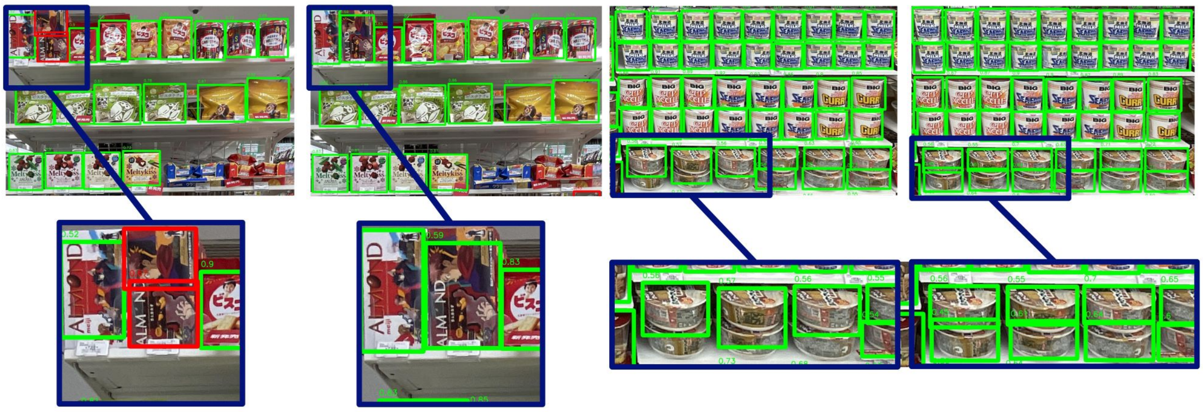
Figure 3 shows examples of synthesized images. Figure 3-(a) is synthesized by the random approach, whose foregrounds are pasted on random positions and frequently occlude each other. On the other hand, in the proposed approach shown in Figure 3-(b), items are lined up in accordance with the object alignment in SKU-110K. The reshaping of foregrounds is realistic, and the same items shot from various angles are pasted close together, this reproduces a more realistic appearance of shelves.

We adopt EfficientDet-d0 [17] as a detector model. Hyper-parameters, training epochs, and the learning rate, are tuned by the validation dataset that consists of the SKU-110K test-set and 100 synthesized images by each method.



(a) Front (b) Upper (c) Upper-left

Figure 4: Examples of evaluation images. (a), (b), and (c) show the images taken from the Front, Upper, and Upper-left angles, respectively. The items in the Upper and Upper-left are deformed compared to the Front image due to parallax.



(a) Random (b) Ours (c) Random (d) Ours

Figure 5: Detection results on the front angle data. Green and red bounding boxes represent outputs of the model with confidence scores more than 0.4. Green bounding boxes have IoU with ground truth more than 0.5, and red bounding boxes are less than 0.5.

4.2 Evaluation dataset

We use images shot in real stores as the evaluation set (Figure 4). This image set was shot in two convenience stores for four days. The target shelves are drinks, snacks, and instant noodles. There are three variations of camera angles: front, upper and upper-left, where each set consists of 32 images.

The detection targets are items on the target shelves, whose whole body is within the image, and therefore, items on non-target shelves are not subject to aggregation. The metric of evaluation is average precision (AP) of all items in one class.

4.3 Results

The experimental results are shown in Table 1. For all evaluation sets, the proposed method performs the best. For the front and the upper angle data, the scores of random approach decrease relative to SKU-only. This shows that unrealistic images synthesized by the random approach adversely affect the training. On the other hand, the proposed approach positively affects all of the targets.

Detection results are shown in Figure 5. One notable example is the chocolate box on the upper left corner of Figure 5-(a) and (b). In the random approach, the chocolate box is recognized as two objects. The random approach synthesizes crowded and complex images as shown in Figure 3. Due to this, the detection model trained on such images tends to split objects of complex texture into two different objects. On the other hand, in the proposed approach, the model recognizes the chocolate box correctly.

In Figure 5-(c), some noodles stacked in two layers on the bottom row are detected as one object in the random method. On the other hand, in our approach they are detected correctly as shown in Figure 5-(d). This is the effect of the alignment approach with object similarity, that is, our approach can detect objects in the scene with similar objects that are stacked and aligned densely.

	Alignment reference	Object similarity	Aspect ratio	Front	Upper	Upper-left	average
Random [4]				0.945	0.880	0.735	0.853
Align only	✓			0.943	0.868	0.718	0.843
Align + Sim	✓	✓		0.946	0.872	0.732	0.850
Align + Aspect	✓		✓	0.949	0.843	0.701	0.831
Align + Sim + Aspect	✓	✓	✓	**0.951**	**0.894**	**0.755**	**0.866**

Table 2: Ablation study of our method. Align, Sim, and Aspect denote the approaches described in Section 3.1, 3.2, and 3.2, respectively.

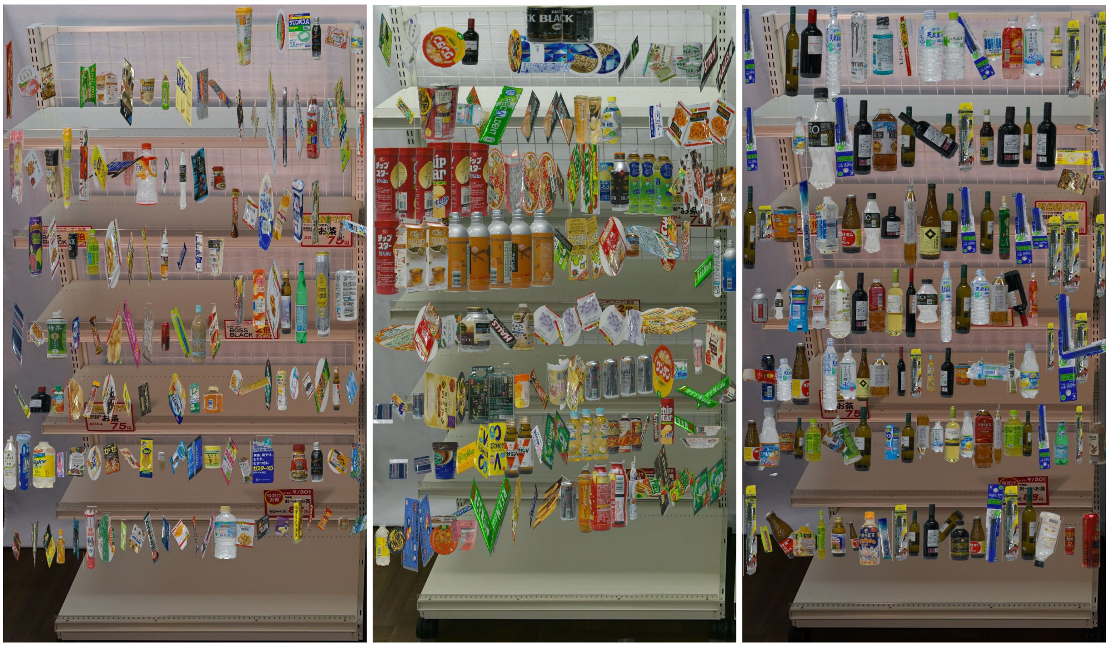

Alignment only   Alignment + Similarity   Alignment + Aspect

Figure 6: Examples of synthesized images in ablation study.

4.4 Ablation study

We conduct an ablation study of our approach in Table 2. The alignment reference approach without considering the object similarity and aspect ratio (Align only) is worse than the random approach. One reason is unrealistic reshaping of foreground images. This can be confirmed in results on the front data, that is, the score of the proposed approach while considering the aspect ratio (Align + Aspect) is higher than that of the random approach. However, on the other data, this tendency is not clear on the upper and upper-left images. This is because the objects shot from the upper or upper-left angles have reshaped appearance, so there are cases where it is better not to select foregrounds on the basis of the aspect ratio. The approach considering object similarity (Align + Sim + Aspect) is better on all of the evaluation data.

Figure 6 shows synthesized images by the approaches in the ablation study. In the alignment only approach, some foreground images are reshaped extremely and their appearance is unrealistic. On the other hand, in the alignment + aspect approach that takes into account the aspect ratio of the box, the appearance of foregrounds is not so different from reality and objects are aligned. In the alignment + similarity approach that takes into account the object similarity, similar objects are placed in the neighborhood, which achieves a similar appearance to shelves in stores. As above, each approach has advantages for realistic synthetization, and combining all of the approaches, the most realistic image in Figure 3-(b) is synthesized.

5 CONCLUSION

We proposed a novel cut-and-paste method for training shelf item detection models. The proposed method determines the pasting positions for the foreground images, referring to the annotations of a dataset of real-world shelves images. Furthermore, to generate more realistic images, the proposed method selects the foreground image with reference to the similarity of the items and the aspect ratio of the bounding box of the pasting position. Experiments show that the proposed method can generate more realistic images of the item shelves than the conventional random pasting method and that the dataset of the images can be used to train more accurate item detection models.

As the proposed method determines the pasting positions without specifying the positions of the shelves in the image, if the positions of the shelves in the reference dataset image and the background image change drastically, the items will be placed at locations other than the shelves. In order to generate a more realistic image where the items are accurately placed on the shelves, annotations of the shelf positions in the background image should be prepared, and the pasting positions should be adjusted based on the annotations. Verification of such a generation method is future work.

6 REFERENCES

[1] H. Abu Alhaija, S. K. Mustikovela, L. Mescheder, A. Geiger, and C. Rother. Augmented Reality Meets Computer Vision: Efficient Data Generation for Urban Driving Scenes. *International Journal of Computer Vision*, 126(9):961–972, Sept. 2018.

[2] E. Bochinski, V. Eiselein, and T. Sikora. Training a convolutional neural network for multi-class object detection using solely virtual world data. In *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 278–285, 2016.

[3] N. Dvornik, J. Mairal, and C. Schmid. Modeling visual context is key to augmenting object detection datasets. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, *Computer Vision – ECCV 2018*, pages 375–391, Cham, 2018. Springer International Publishing.

[4] D. Dwibedi, I. Misra, and M. Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017.

[5] H.-S. Fang, J. Sun, R. Wang, M. Gou, Y.-L. Li, and C. Lu. Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 682–691, 2019.

[6] M. Filax, T. Gonschorek, and F. Ortmeier. Semi-automatic Acquisition of Datasets for Retail Recognition. *Computer Science Research Notes*, 3201:86–94, 2022.

[7] G. Georgakis, A. Mousavian, A. C. Berg, and J. Kosecka. Synthesizing training data for object detection in indoor scenes. 2017.

[8] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2917–2927, 2021.

[9] E. Goldman, R. Herzig, A. Eisenschtat, J. Goldberger, and T. Hassner. Precise detection in densely packed scenes. In *Proc. Conf. Comput. Vision Pattern Recognition (CVPR)*, 2019.

[10] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[11] S. Hinterstoisser, V. Lepetit, P. Wohlhart, and K. Konolige. On pre-trained image features and synthetic images for deep learning. In L. Leal-Taixé and S. Roth, editors, *Computer Vision – ECCV 2018 Workshops*, pages 682–697, Cham, 2019. Springer International Publishing.

[12] S. Hinterstoisser, O. Pauly, H. Heibel, M. Marek, and M. Bokeloh. An Annotation Saved is an Annotation Earned: Using Fully Synthetic Training for Object Instance Detection. Feb. 2019.

[13] T. Hodaň, V. Vineet, R. Gal, E. Shalev, J. Hanzelka, T. Connell, P. Urbina, S. N. Sinha, and B. Guenter. Photorealistic image synthesis for object instance detection. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 66–70, 2019.

[14] S. Koturwar, S. Shiraishi, and K. Iwamoto. Robust multi-object detection based on data augmentation with realistic image synthesis for point-of-sale automation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):9492–9497, July 2019.

[15] S. R. Richter, V. Vineet, S. Roth, and V. Koltun. Playing for data: Ground truth from computer games. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *European Conference on Computer Vision (ECCV)*, volume 9906 of *LNCS*, pages 102–118. Springer International Publishing, 2016.

[16] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": Interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 309–314, New York, NY, USA, 2004. Association for Computing Machinery.

[17] M. Tan, R. Pang, and Q. V. Le. EfficientDet: Scalable and efficient object detection. *CoRR*, abs/1911.09070, 2019.

[18] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield. Training deep networks

with synthetic data: Bridging the reality gap by domain randomization. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1082–10828, 2018.

[19] X.-S. Wei, Q. Cui, L. Yang, P. Wang, and L. Liu. RPC: A large-scale retail product checkout dataset. *CoRR*, abs/1901.07249, 2019.

[20] S.-F. Wu, M.-C. Chang, S. Lyu, C.-S. Wong, A. K. Pandey, and P.-C. Su. FlagDetSeg: Multi-nation flag detection and segmentation in the wild. In *2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–8, 2021.

[21] W.-H. Yun, T. Kim, J. Lee, J. Kim, and J. Kim. Cut-and-Paste Dataset Generation for Balancing Domain Gaps in Object Instance Detection. *IEEE Access*, 9:14319–14329, 2021.