IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 February 2024, accepted 1 March 2024, date of publication 4 March 2024, date of current version 19 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3373463

RESEARCH ARTICLE

Canonical Plane Segmentation Without Annotating Pixel-Level Object Regions for Image Registration

SHUNSUKE YONEDA[1], GO IRIE[2], (Member, IEEE), AND MASASHI NISHIYAMA[3]

[1]Organization for Information Strategy and Management, Tottori University, Tottori 680-8550, Japan
[2]Faculty of Engineering, Tokyo University of Science, Katsushika City, Tokyo 125-8585, Japan
[3]Faculty of Engineering, Tottori University, Tottori 680-8550, Japan

Corresponding author: Shunsuke Yoneda (yoneda@tottori-u.ac.jp)

ABSTRACT Two-dimensional (2D) image registration is a natural choice for simultaneous object pose estimation and object recognition. However, it was not designed to perform object segmentation, which is critical for object-picking applications in warehouse automation scenarios. In this study, we propose a unified 2D image registration framework that simultaneously performs image registration and object segmentation by introducing a deep segmentation network module to the 2D image registration framework. Our method is designed to automatically generate annotations for training the segmentation network module through the process of 2D image registration, that is, no additional manual annotations are required. Specifically, given initial object regions from the 2D image registration results, our method trains the segmentation network module to refine a pseudo-pixel-level object region and remove background pixels based on the pixel-level similarity of an aligned image pair. The experimental results on several picking object datasets demonstrated that the segmentation accuracy of our method was superior to that of existing weakly supervised segmentation methods, and our method simultaneously achieved better performance for object recognition and pose estimation. Furthermore, our segmentation network module smoothly cooperated with many existing 2D image registration techniques.

INDEX TERMS Image registration, pseudo-pixel-level object region, weakly supervised segmentation.

I. INTRODUCTION

Recently, a strong demand has arisen for warehouse solutions that automate the picking process for planar objects, such as product boxes and books, using object-picking systems with cameras and robot arms [1], [2], [3]. These solutions are intended to address labor shortages in logistics. To accomplish the task by accurately controlling the robot arm to pick up the target object, such a system requires computer vision techniques to automatically estimate the pose parameters, object classes, and pixel-level regions of the object from images acquired from cameras mounted on the system. Hence, three tasks, that is, pose estimation, object recognition, and object segmentation, need to be performed automatically and accurately. In this study, we consider implementing these three tasks in a unified framework.

First, we begin with a common approach to the pose estimation task. Two-dimensional (2D) image registration with local features [4], [5] is frequently used to perform the pose estimation task. This technique estimates an object's pose parameter from an image pair using three major processes: local descriptor extraction, keypoint matching, and homography matrix estimation. Recently, deep neural-based 2D image registration techniques, such as SuperPoint [6] + SuperGlue [7] and LoFTR [8], have been proposed to perform these processes and achieved high pose estimation accuracy. One advantage of these techniques is that the results of local keypoint matching can provide information about both whether the objects in the two input images are identical and how much the poses of the two objects are misaligned:

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

That is, 2D image registration techniques can simultaneously perform the object recognition and pose estimation tasks. However, existing registration techniques were not designed to perform the object segmentation task and therefore cannot be applied to automatic object-picking.

A simple solution is to incorporate a pre-trained segmentation network into the registration framework. However, the network needs to be (pre-)trained in a fully supervised manner, which requires a large dataset with pixel-level object region labels, which is highly costly. To reduce the manual labor of annotation, an alternative approach is to use a weakly supervised segmentation network [9], [10], [11], [12], [13], [14], which is trained using only image-level class label annotation and therefore not using pixel-level object region annotation. Existing methods train the segmentation network using the class activation maps (CAMs) of the object recognition network and have the advantage of eliminating the cost of pixel-level object region annotation. However, existing methods cannot obtain high segmentation accuracy because they do not consider how to cooperate with 2D image registration techniques.

We propose a novel method that introduces a weakly supervised segmentation network module to the 2D image registration framework for performing the segmentation task with only image-level class label annotation. The main contributions of this study are as follows:

- We introduce a weakly supervised segmentation network that easily combines with existing 2D image registration techniques to simultaneously perform segmentation, pose estimation, and object recognition tasks.
- No pixel-level object region annotation is required for the segmentation training process. We train our segmentation network using only pseudo-object regions generated using keypoints detected by 2D image registration techniques.
- We design region loss and regularization terms to refine the pseudo-object regions by leaving object pixels and removing background pixels. Refining these loss terms accurately requires only image-level class label annotation.

The experimental results demonstrated that our segmentation network obtained higher accuracy than existing weakly supervised segmentation networks. Furthermore, our segmentation network smoothly cooperated with existing 2D image registration techniques that perform pose estimation and object recognition tasks.

This paper is an extended version of our previous paper [15]. The differences between the papers are described below. In our previous method [15], we required pixel-level object region annotation for a subset of the training samples. By contrast, in the current method, we use pseudo-object regions to train the segmentation network; thus, object region annotation is not necessary. In our previous method [15], we used only the region loss term for the segmentation training process. By contrast, in the current method, we use the regularization term in addition to the region loss term to prevent the over-refinement of pseudo-object regions caused by the region loss term.

II. RELATED WORK

A. IMAGE REGISTRATION

The 2D image registration technique simultaneously performs the pose estimation task and object recognition task using image pairs that belong to the same object acquired for different poses, as described in Section I. Generally, the 2D image registration technique consists of three processes: local descriptor extraction, keypoint matching, and homography matrix estimation. We base the concept of our method on this technique to compute the initial solution of the segmentation task. In the following, we introduce the details of the 2D image registration technique for the pose estimation and object recognition tasks.

Local descriptor extraction is the process of detecting keypoints in an image pair and computing local descriptors using the keypoints. The keypoints are located on characteristic points, such as corners, in images. Local descriptors are pose-invariant features computed from the appearances of small regions surrounding the keypoints. SIFT [16] and ORB [17] are well-known methods for the local descriptor extraction process. Recently, SuperPoint [6] and ALIKED [18], which are deep neural-based methods, have been proposed to detect keypoints and compute local descriptors with high repeatability. In our method, we simply use SuperPoint to perform the local descriptor extraction process to achieve repeatability.

Keypoint matching is the process of searching for correspondences of keypoints in an image pair using local descriptors. A brute-force matcher has been widely used with conventional local descriptors, such as SIFT and ORB, for matching the keypoints of an image pair. Recently, SuperGlue [7], LoFTR [8], and LightGlue [19], which are deep neural-based methods, have been proposed for matching keypoints with high accuracy. It is well known that SuperGlue is suitable for texture-full objects and LoFTR is suitable for texture-less objects. We consider using SuperGlue and LoFTR, depending on whether the images of the target objects are texture-full or texture-less. Our method can be easily combined with both existing methods.

Homography matrix estimation is the process of calculating a transformation matrix using a set of keypoint pairs matched using the above process. The homography transformation matrix represents the relative pose from one region to another region of the same object in an image pair. RANSAC [20] and CONSAC [21] are popular methods for the homography matrix estimation process.

The local descriptor detection process and keypoint matching process can be used to perform the object recognition task in addition to the pose estimation task, as described in [16]. To recognize object class labels, existing methods prepare a set of target images that contain objects. Existing methods perform the object recognition task by assigning the object

TABLE 1. Comparison of the advantages and disadvantages of our method and existing segmentation methods.

Method	Cooperates with 2D image registration	Does not require pixel-level object region annotation
Unet++ [22]	-	-
Deeplabv3+ [23]	-	-
PSA-Net [9]	-	✓
CONTA [10]	-	✓
SEAM [11]	-	✓
Puzzle-CAM [12]	-	✓
BECO [13]	-	✓
PPC [14]	-	✓
Co-segmentation [24], [25]	-	✓
Ours	✓	✓

class label of the target image with the largest number of keypoints corresponding to the input image to the predicted class label of the input image. Our method performs the object recognition task in accordance with this approach.

Generally, 2D image registration techniques, such as those in [6], [7], [8], [16], [17], [18], [19], and [20], are designed to simultaneously perform the pose estimation task and object recognition task. However, these techniques are not designed to perform the segmentation task. Our method is designed to cooperate with 2D image registration techniques using the outputs of the local descriptor extraction and keypoint matching processes to perform the segmentation task. Our method can be applied to the segmentation task, whereas existing 2D image registration techniques cannot.

B. SEGMENTATION

The segmentation task predicts pixel-level object regions from input images as described in [26]. There are two types of existing methods: fully supervised segmentation and weakly supervised segmentation. These methods depend on how the annotation is performed to train segmentation networks. In the following, we introduce the details of existing segmentation methods.

Fully supervised segmentation methods train deep neural networks with pixel-level object region annotation. Many methods have been proposed, such as Unet++ [22] and Deeplabv3+ [23]. To train segmentation networks, fully supervised segmentation methods require large datasets with pixel-level object region annotation. However, pixel-level object region annotation is time-consuming and requires expensive manual labor.

To reduce the manual labor of annotation, weakly supervised segmentation methods [9], [10], [11], [12], [13], [14] have been proposed in recent years. These segmentation networks are trained with only image-level class label annotation. In the training process, these methods use CAMs [27] generated by object recognition networks trained by image-level class labels. CAMs are multidimensional weight arrays that represent objects' characteristics for the object recognition task. Generally, CAMs have large weights only for partial object regions, therefore, not entire object regions. To train weakly supervised segmentation networks, existing methods, such as PSA-Net [9], CONTA [10], SEAM [11], Puzzle-CAM [12], BECO [13], and PPC [14], estimate pseudo-object regions using CAMs. Specifically, these existing methods train segmentation networks by propagating the weights of CAMs to expand partial object regions to entire object regions.

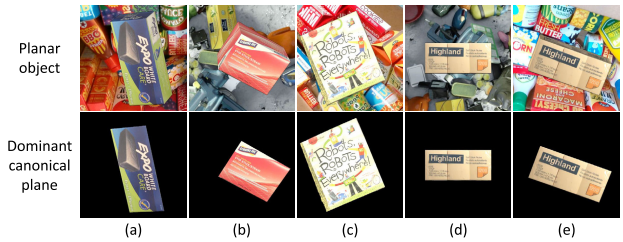


FIGURE 1. Examples of the dominant canonical planes in planar object images. The planar object handled in warehouses consists of a small number of "canonical planes" [28]. The dominant canonical plane is mainly observed in each planar object image. The appearance of the dominant canonical plane corresponds one-to-one with the three-dimensional (3D) pose of the planar object.

Co-segmentation methods [24], [25] that require no pixel-level object region annotation to perform the segmentation task have been proposed. Although such methods train segmentation networks using only image-level class labels, they require a set of images that contain the same object in both the training and inference processes. The acquisition of a set of images in the inference process is a constraint when designing object-picking systems. Two important approaches used to acquire a set of images are using multiple cameras that acquire the same object simultaneously or an object-tracking camera that acquires a video sequence of the same object. Regardless of which approach is used, large costs are required to set up accurate camera parameters to stably acquire a set of images. Therefore, co-segmentation methods are unsuitable for the design of low-cost object-picking systems.

In Table 1, we show the advantages and disadvantages of our method and existing segmentation methods. Our method estimates the pseudo-object region using a single image from the output of the local descriptor extraction and keypoint matching processes in 2D image registration techniques. This concept is completely different from that of existing weakly supervised segmentation methods and existing co-segmentation methods. Our pseudo-object region has large likelihoods for both entire object regions and surrounding background regions. Our method trains the segmentation network by propagating the likelihoods of the pseudo-object region to refine the target object regions in an image pair. In this study, we introduce a novel concept for the weakly supervised segmentation task.

III. INTRODUCTION OF THE WEAKLY SUPERVISED SEGMENTATION NETWORK MODULE TO 2D IMAGE REGISTRATION FRAMEWORK

A. OVERVIEW OF OUR METHOD

Before we describe the concept of our method, we introduce a canonical plane [28], which is a prerequisite for our method. In [28], planar objects handled in warehouses, such as product boxes and books, consist of a small number of planar surfaces

called "canonical planes." For example, daily necessary product boxes are often cuboid; hence, a box consists of six canonical planes. When a camera acquires the planar object image, a dominant canonical plane is observed in each image, as shown in Figs. 1 (a), (b), and (c). The appearance of the dominant canonical plane corresponds one-to-one with the 3D pose of the planar object.

We assume the 3D pose estimation of a planar object as the 2D transformation estimation of a dominant canonical plane contained in an image pair. The homography matrix is estimated between the keypoints of the same dominant canonical plane with different poses using existing 2D image registration techniques, such as those in [6], [7], and [8]. Suppose the homography matrix of an image pair in Figs. 1(d) and (e) is computed using 2D image registration techniques. This homography matrix represents the relative 2D pose change between the dominant canonical plane images and also represents the relative 3D pose change between the planar objects.

The 2D image registration technique can simultaneously perform the pose estimation and object recognition tasks described in Section I, but cannot perform the segmentation task. The reasons for this are as follows: The 2D image registration technique uses the sets of keypoints on the dominant canonical plane, as shown in Fig. 2(a), to perform the pose estimation task. The sets of keypoints represent the characteristic points of the dominant canonical plane, but not the entire region of the dominant canonical plane. To perform the segmentation task, we exploit the sets of keypoints as an initial solution for training our segmentation network module. Specifically, we generate an initial solution that roughly covers the region of the dominant canonical plane using the sets of keypoints, as shown in Fig. 2(b). Based on the initial solution, our segmentation network module predicts the regions of the dominant canonical planes in an image pair. In the following, we refer to the initial solution as a pseudo-object region.

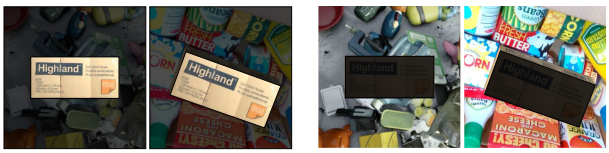
The pseudo-object region contains foreground pixels and background pixels. The foreground pixels represent the dominant canonical plane, for example, Fig. 3(a), and the background pixels do not represent the dominant canonical plane, for example, Fig. 3(b). To perform the segmentation task using the pseudo-object region, we exploit the similarity of the foreground pixels and the dissimilarity of the background pixels in an image pair. For this purpose, we introduce region loss and regularization terms to refine the initial pseudo-object region. Our loss terms have the effect of refining the region by leaving foreground pixels with similar appearances and removing background pixels with dissimilar appearances.

B. TRAINING PROCESS

Fig. 4 shows the overview of our weakly supervised segmentation network for the training process. In this process, our network uses a part of two images, referred to as a query image $\mathbf{I}^q$ and target image $\mathbf{I}^t$. We assume that

(a) Sets of keypoints on the dominant canonical plane (b) Pseudo-object regions

FIGURE 2. Examples of the sets of keypoints on the dominant canonical plane and pseudo-object regions. The sets in (a) represent the characteristic points of the dominant canonical plane. We generate the pseudo-object regions in (b) to roughly cover the region of the dominant canonical plane using the sets of keypoints.



(a) Foreground pixels (b) Background pixels

FIGURE 3. Examples of foreground pixels and background pixels. The foreground pixels in (a) represent the dominant canonical plane, whereas the background pixels in (b) do not represent it.

the query $\mathbf{I}^q$ and target $\mathbf{I}^t$ contain the same dominant canonical plane with different poses. We also assume that the backgrounds' appearances for the query $\mathbf{I}^q$ and target $\mathbf{I}^t$ are completely different. We believe that image pairs with different backgrounds are often acquired in real-world scenarios, such as warehouses.

In the training process, the 2D image registration module r() estimates a homography matrix $\hat{\mathbf{H}}^q$ from the target $\mathbf{I}^t$ to the query $\mathbf{I}^q$ as follows:

$$\hat{\mathbf{H}}^q = \mathrm{r}(\mathbf{I}^q, \mathbf{I}^t; \mathcal{K}^q, \mathcal{K}^t). \tag{1}$$

The module r() internally detects a query keypoint set $\mathcal{K}^q = \{\mathbf{k}_n^q\}_{n=1}^N$ and target keypoint set $\mathcal{K}^t = \{\mathbf{k}_n^t\}_{n=1}^N$ using existing methods for local descriptor extraction and keypoint matching, such as those in [6], [7], [8], [16], [17], [18], and [19].

After estimating the homography matrix, the pseudo-object region module p() generates the query pseudo-object region $\tilde{\mathbf{I}}_r^q$ using the query keypoint set $\mathcal{K}^q$ as follows:

$$\tilde{\mathbf{I}}_r^q = \mathrm{p}(\mathcal{K}^q) = \bigcup_{\mathbf{k}_n^q \in \mathcal{K}^q} \mathbf{I}_{\mathbf{k}_n^q}, \tag{2}$$

where the binary image $\mathbf{I}_{\mathbf{k}_n^q}$ is a one-keypoint region generated using the query keypoint $\mathbf{k}_n^q$ stored in the query keypoint set $\mathcal{K}^q$ and the operator $\bigcup$ represents pixel-wise logical disjunction. The binary image $\mathbf{I}_{\mathbf{k}_n^q}$ is determined by a circle of radius r centered on the keypoint $\mathbf{k}_n^q$. Our method assigns the binary pixel value 1 to the inside of the circle and 0 to the outside. Note that the radius r is a hyperparameter. The target pseudo-object region $\tilde{\mathbf{I}}_r^t$ is generated using the target keypoint set $\mathcal{K}^t$ using the same approach as the generation of the query region $\tilde{\mathbf{I}}_r^q$.

After generating the pseudo-object regions, the weakly supervised segmentation network module s() predicts the

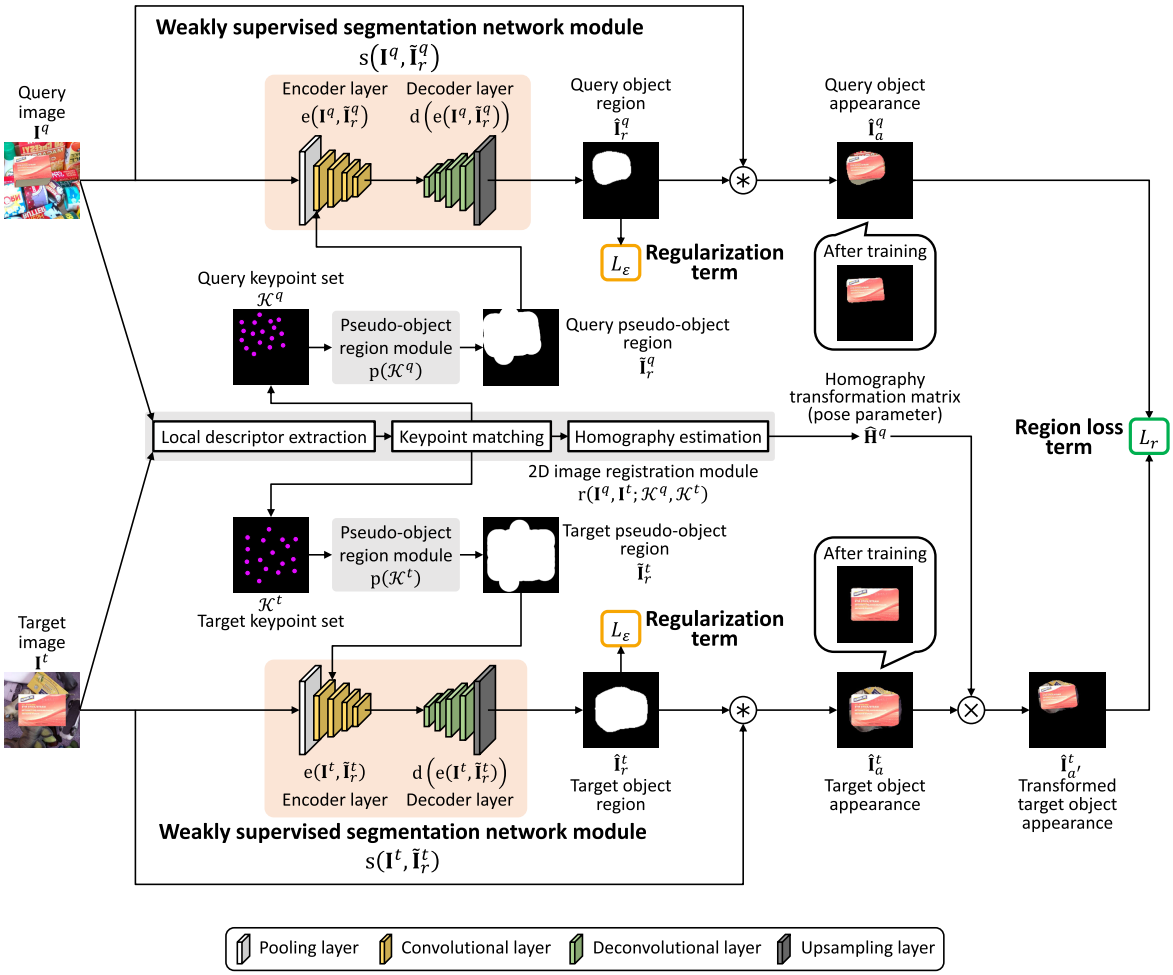


FIGURE 4. Overview of our weakly supervised segmentation network for the training process. Our network module predicts the query object appearance and target object appearance using a query pseudo-object region and target pseudo-object region. To predict the query object appearance and target object appearance adequately, our method trains the module by reducing the region loss term and regularization term.

query object region $\hat{\mathbf{I}}_r^q$ from the query $\mathbf{I}^q$ and query pseudo-object region $\tilde{\mathbf{I}}_r^q$ as follows:

$$\hat{\mathbf{I}}_r^q = \mathrm{s}(\mathbf{I}^q, \tilde{\mathbf{I}}_r^q) = \mathrm{d}(\mathrm{e}(\mathbf{I}^q, \tilde{\mathbf{I}}_r^q)). \tag{3}$$

In the module s(), the encoder layer e() extracts a feature map of the query $\mathbf{I}^q$ using internal layers. To represent the foreground region of the query $\mathbf{I}^q$ roughly, the layer e() multiplies the query pseudo-object region $\tilde{\mathbf{I}}_r^q$ by the feature map extracted by a certain internal middle layer. The decoder layer d() generates the query object region $\hat{\mathbf{I}}_r^q$ from the feature map extracted by the encoder layer e(). The query object appearance $\hat{\mathbf{I}}_a^q$ is predicted from the query $\mathbf{I}^q$ using the query object region $\hat{\mathbf{I}}_r^q$ as follows:

$$\hat{\mathbf{I}}_a^q = \mathbf{I}^q \circ \hat{\mathbf{I}}_r^q, \tag{4}$$

where $\circ$ is the operator for the pixel-wise product. The target object appearance $\hat{\mathbf{I}}_a^t$ is predicted using the target pseudo-object region $\tilde{\mathbf{I}}_r^t$ by generating a target object region $\hat{\mathbf{I}}_r^t$ in the same manner as the generation of the query object appearance $\hat{\mathbf{I}}_a^q$ by the module s().

Finally, the total loss term L is computed as follows:

$$L = L_r + \lambda L_\varepsilon, \tag{5}$$

where L_r is the region loss term, L_ε is the regularization term, and λ is a hyperparameter. In the following, we describe the region loss term L_r and regularization term L_ε. The region loss term L_r refines the query pseudo-object region $\tilde{\mathbf{I}}_r^q$ and target pseudo-object region $\tilde{\mathbf{I}}_r^t$. The loss term L_r is computed using the mean structural similarity (MSSIM) index [29] as follows:

$$L_r = 1 - \mathrm{m}(\hat{\mathbf{I}}_a^q, \hat{\mathbf{I}}_{a'}^t), \tag{6}$$

where m() returns the MSSIM index and $\hat{\mathbf{I}}_{a'}^t$ is a target object image transformed using the homography matrix $\hat{\mathbf{H}}^q$. The loss term L_r returns a small value when the query object appearance $\hat{\mathbf{I}}_a^q$ is close to the transformed target object appearance $\hat{\mathbf{I}}_{a'}^t$. The high similarity of the appearances $\hat{\mathbf{I}}_a^q$ and $\hat{\mathbf{I}}_{a'}^t$ means that the background pixels of the query $\mathbf{I}^q$ and target $\mathbf{I}^t$ were accurately removed from the query object

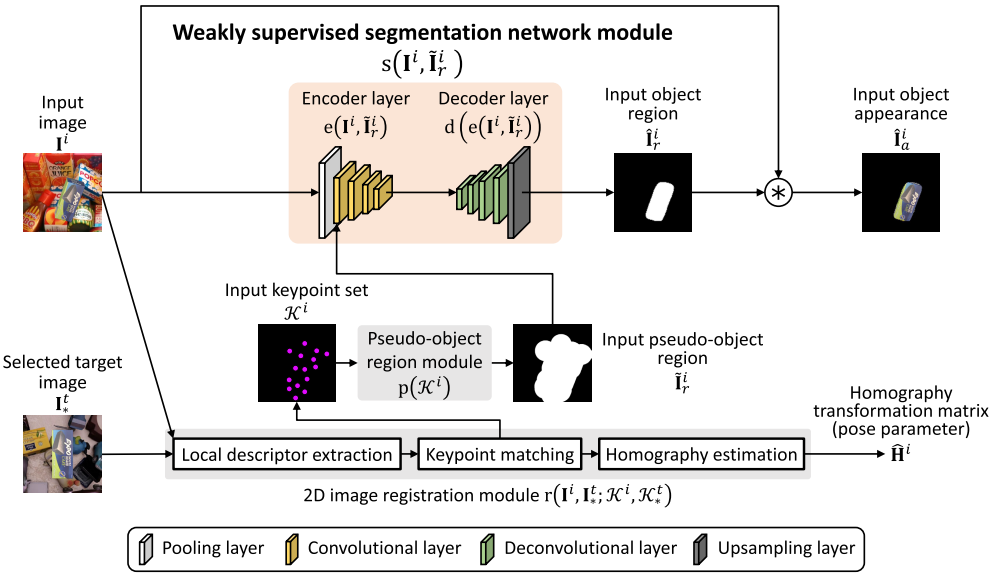


FIGURE 5. Overview of our weakly supervised segmentation network in the inference process. The object recognition task is performed in advance by assigning the object class label of the selected target that contains the same dominant canonical plane of the input to the predicted class label of the input and the pose estimation task is performed by estimating a homography matrix. Our network performs the segmentation task for the input using our network module trained in Section III-B.

region $\hat{\mathbf{I}}_r^q$ and target object region $\hat{\mathbf{I}}_r^t$. The regularization term L_ε prevents the region loss term L_r from over-refining the query pseudo-object region $\tilde{\mathbf{I}}_r^q$ and target pseudo-object region $\tilde{\mathbf{I}}_r^t$. The loss term L_ε is computed using the L1 norms of the query object region $\hat{\mathbf{I}}_r^q$ and target object region $\hat{\mathbf{I}}_r^t$ as follows:

$$L_\varepsilon = \frac{1}{||\hat{\mathbf{I}}_r^q||_1 + c} + \frac{1}{||\hat{\mathbf{I}}_r^t||_1 + c}, \tag{7}$$

where c is a constant used to avoid division by zero. The loss term L_ε returns a large value as a penalty when the query object region $\hat{\mathbf{I}}_r^q$ and target object region $\hat{\mathbf{I}}_r^t$ are over-refined by the region loss term L_r. By reducing the region loss term L_r and regularization term L_ε, our method can train our segmentation network module s() without pixel-level object region annotation.

C. INFERENCE PROCESS

Our method initially performs the object recognition task for an input image $\mathbf{I}^i$ in the inference process. To perform the object recognition task, our method stores a target image set $\mathcal{T} = \{\mathbf{I}_m^t\}_{m=1}^M$. Each target $\mathbf{I}_m^t$ has one object class label corresponding to the dominant canonical plane.

In the object recognition task, our method detects an input keypoint set $\mathcal{K}_m^i$ and target keypoint set $\mathcal{K}_m^t$ using existing methods for local descriptor extraction and keypoint matching processes, such as those in [6], [7], [8], [16], and [17]. Then, our method selects a target $\mathbf{I}_*^t$ from the target set $\mathcal{T}$, where the target $\mathbf{I}_*^t$ has the largest number of keypoints corresponding to the input $\mathbf{I}^i$. The object recognition task is performed by assigning the object class label of the selected target $\mathbf{I}_*^t$ to the predicted class label of the input $\mathbf{I}^i$.

After the object recognition task, our method performs the pose estimation task and segmentation task using the input $\mathbf{I}^i$. Fig. 5 shows the overview of our inference process network for the pose estimation task and segmentation task. Our network simply uses the weakly supervised segmentation network module s() trained in Section III-B. The 2D image registration module r() estimates a homography matrix $\hat{\mathbf{H}}^i$ from the selected target $\mathbf{I}_*^t$ to the input $\mathbf{I}^i$ and detects an input keypoint set $\mathcal{K}^i$. The pseudo-object region module p() generates an input pseudo-object region $\tilde{\mathbf{I}}_r^i$ using the input keypoint set $\mathcal{K}^i$. The weakly supervised segmentation network module s() predicts an input object appearance $\hat{\mathbf{I}}_a^i$ using the region $\tilde{\mathbf{I}}_r^i$.

IV. EXPERIMENTS

A. DATASET

We evaluated the segmentation accuracy, pose estimation error, and recognition accuracy of the proposed network on a dataset that consisted of a mixture of three popular datasets for the picking process scenario: YCB dataset [30], APC dataset [31], and ARC dataset [32]. We used 13 cuboid objects and five plane objects to create a selected dataset: five cuboid objects in the YCB dataset, six cuboid objects and one plane object in the APC dataset, and two cuboid objects and four plane objects in the ARC dataset as shown in Fig. 6. Each object in these datasets consisted of a 3D mesh model and texture map. In our previous study [15], we required pixel-level object region annotation for a subset of the training samples. By contrast, in this study, we required no pixel-level object region annotation for the training samples.

FIGURE 6. Target objects from the YCB dataset, APC dataset, and ARC dataset used in our experiments. The color frame indicates the type of object shape (red: cuboid, orange: plane).

FIGURE 7. Examples of the background conditions used in our experiments.

We used two background conditions: HOPE background and HBDB background. For the HBDB background condition, we used publicly available images from the HOPE dataset,[1] as shown in Fig. 7(a). For the HBDB background condition, we used publicly available images from the HBDB dataset [33], as shown in Fig. 7(b). Note that the objects contained in the HOPE dataset and HBDB dataset were completely different from the target objects in Fig. 6.

For the training process, we used 10,000 pairs of the query image $\mathbf{I}^q$ and target image $\mathbf{I}^t$. Fig. 8 shows examples of the pairs of the query image $\mathbf{I}^q$ and target image $\mathbf{I}^t$. The query $\mathbf{I}^q$ and target $\mathbf{I}^t$ contained the same dominant canonical plane. We generated the query $\mathbf{I}^q$ by applying 3D rendering with random three degree-of-freedom (3-DOF) rotation, translation, and scaling to the object under the HOPE background condition. We sampled the 3-DOF rotation angles in the range $[-30, 30]$ degrees, translation parameters in the range $[-150, 150]$ pixels, and scale parameters in the range $[0.8, 1.2]$. We acquired the target $\mathbf{I}^t$ using a camera setup so that its optical axis passed through the target object's center of gravity under the HBDB background condition.

For the inference process, we used 1,000 input images $\mathbf{I}^i$ and the target image set $\mathcal{T} = \{\mathbf{I}^t_m\}_{m=1}^{M}$. We generated the input $\mathbf{I}^i$ in the same manner as the query $\mathbf{I}^q$. Note that we completely separated the pose parameters of the inputs $\mathbf{I}^i$ from those of the training image pairs $\langle \mathbf{I}^q, \mathbf{I}^t \rangle$. Each target $\mathbf{I}^t_m$ had one object class label corresponding to the dominant canonical plane. The target set $\mathcal{T}$ stored 88 targets, that is, $13 \times 6 + 5 \times 2$, because one cuboid object contains six canonical planes and one plane object contains two canonical planes.

[1] https://github.com/swtyree/hope-dataset

FIGURE 8. Examples of pairs of the query image and target image. The query was generated under the HOPE background condition and the target was generated under the HBDB background condition.

TABLE 2. Parameters of our method used to achieve the best performance.

Parameter	Value
Radius r	80
Hyperparameter λ in Eq. (5)	0.005
Constant c in Eq. (7)	10^{-10}
Optimizer	SGD
Learning rate	1
Momentum	0.9
Epoch	200
Batch size	64
SuperPoint [6]	Default
SuperGlue [7]	Default

We fixed the sizes of the query $\mathbf{I}^q$, target $\mathbf{I}^t$, and input $\mathbf{I}^i$ to 600×600 pixels. In the 2D image registration module r() and pseudo-object region module p(), the size of the input images we used was 600×600 pixels. We resized the input images to 128×128 pixels in the weakly supervised segmentation network module s().

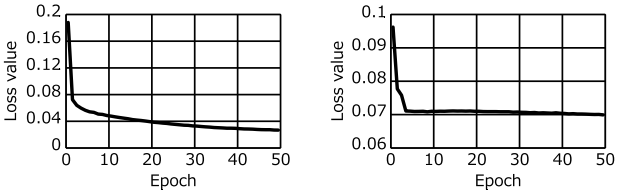
B. IMPLEMENTATION AND PERFORMANCE METRICS

In the weakly supervised segmentation network module s(), the encoder layer e() consisted of one pooling layer and four convolutional layers. The layer e() multiplied the pseudo-object region by the feature map extracted by the first convolutional layer. The decoder layer d() consisted of four deconvolutional layers and one upsampling layer. In the upsampling layer, the 1×1 convolutional layer converted the number of output channels of the final deconvolutional layer to one. The 2D image registration module r() used SuperPoint [6] for the local descriptor extraction process and SuperGlue [7] for the keypoint matching process. For SuperPoint and SuperGlue, we used the network weights and hyperparameters provided by default. We set the radius $r = 80$ of the pseudo-object region, hyperparameter $\lambda = 0.005$ in Eq. (5), and constant $c = 10^{-10}$ in Eq. (7). We trained our network using stochastic gradient descent with a learning rate of 1 and momentum parameter of 0.9 for 200 epochs. We present the parameters of our method used to achieve the best performance in Table 2.

We implemented our network using the PyTorch framework. We used RTX 4090 GPU with 24 GB and Intel Core

TABLE 3. Performance of our weakly supervised network and existing segmentation networks.

Method	Pixel-level annotation	Seg. Acc.	Pose Err.	Rec. Acc.
Unet++ [22]	w/	0.99±0.01	n/a	0.99±0.01
Deeplabv3+ [23]	w/	0.99±0.01	n/a	0.99±0.01
PSA-Net [9]	w/o	0.70±0.01	n/a	0.99±0.01
CONTA [10]	w/o	0.75±0.01	n/a	0.99±0.01
Ours	w/o	**0.88±0.01**	**0.05±0.01**	0.99±0.01



(a) Region loss term L_r (b) Regularization term L_ε

FIGURE 9. Loss value changes for our method.

i9-13900KF CPU with 128 GB. The training process of our weakly supervised segmentation network module s() required 2.6 GB of GPU memory and 19 hours per run. The inference process of the module s() required 1.0 GB of GPU memory and 42 milliseconds per image pair. The computational complexity increase incurred by adding our module was 0.9 GFLOPs and 3.9 M parameters. Note that we required 2.7 GB of GPU memory and 58 milliseconds per image pair on SuperPoint [6] and SuperGlue [7] as the 2D image registration network module r() for both the training and inference processes.

We used the following three metrics for performance evaluation. We used the intersection over union between the predicted input object region $\mathbf{I}_r^i$ and the ground truth object region to evaluate segmentation accuracy. We used the Frobenius norm of the difference between the estimated homography matrix $\hat{\mathbf{H}}^i$ and the ground truth homography matrix to evaluate the pose estimation error. Note that we computed the pose estimation error only when the object class of the input image $\mathbf{I}^i$ was the same as that of the target image $\mathbf{I}_*^t$ selected in the object recognition task. We used the correct match rate of the selected target $\mathbf{I}_*^t$ to evaluate recognition accuracy. We used the average performance over the three runs, each with different training-testing splits.

C. COMPARISON OF OUR NETWORK WITH EXISTING SEGMENTATION NETWORKS

We evaluated the effectiveness of our network by comparing it with the segmentation network methods [9], [10], [22], [23] described in Section II-B.

Unet++ [22]: Fully supervised network *with* pixel-level object region annotation using the Segmentation Models library[2].

Deeplabv3+ [23]: Fully supervised network *with* pixel-level object region annotation using the Segmentation Models library[2].

PSA-Net [9]: Weakly supervised network *without* pixel-level object region annotation using an official implementation[3].

CONTA [10]: Weakly supervised network *without* pixel-level object region annotation using an official implementation[4].

We applied fine-tuning to the provided default network weights for existing methods. We compared the segmentation accuracy of Unet++ [22] and DeepLabv3+ [23], which differ from our network because they use pixel-level object region annotation, to investigate the benefit of annotation. We also compared the segmentation accuracy of PSA-Net [9] and CONTA [10], which are the same as our network because they use only image-level class label annotation, to investigate the effectiveness of our segmentation network module.

The results are shown in Table 3. Regarding segmentation accuracy, our network outperformed existing weakly supervised segmentation networks [9], [10] by significant margins. The improved accuracy proves the importance of our weakly supervised segmentation cooperating with the 2D image registration techniques. The segmentation accuracy of our network was inferior to that of existing fully supervised segmentation networks [22], [23]. However, our network achieved segmentation accuracy closer to these networks [22], [23] than existing weakly supervised segmentation networks [9], [10]. Moreover, our network was the only one able to perform the pose estimation task. Regarding recognition accuracy, our network achieved performance equivalent to fully and weakly supervised segmentation networks [9], [10], [22], [23].

We show the loss value changes during the training process in Fig. 9. Until epoch 4, the values of the region loss term L_r in (a) and regularization term L_ε in (b) decreased drastically. We believe that this decrease was caused by the refinement of the pseudo-object regions from a randomly generated solution toward an approximate solution. After epoch 4, the value of L_r in (a) decreased gradually and the value of L_ε in (b) was almost constant. We believe that this decrease of L_r in (a) was caused by the refinement of the pseudo-object regions toward an approximate solution. We also believe that the almost constant value of L_ε in (b) was caused by the maintenance of the appropriate size of the pseudo-object regions.

We show qualitative results of the weakly supervised segmentation network module s() in Fig. 10. In the figure, we show the input image $\mathbf{I}^i$ for the inference process in (a), the input object appearance $\hat{\mathbf{I}}_a^i$ predicted by the module s() in (b), a transformed input object appearance converted by the homography matrix $\hat{\mathbf{H}}^i$ in (c), and the target image $\mathbf{I}_*^t$ selected in the object recognition task in (d). When generating the transformation input object appearance, we used the inverse matrix of $\hat{\mathbf{H}}^i$. We believe that the weakly supervised

[2]https://github.com/qubvel/segmentation_models

[3]https://github.com/jiwoon-ahn/psa

[4]https://github.com/dongzhang89/CONTA

FIGURE 10. Qualitative results of our weakly supervised segmentation network module. The dominant canonical plane in the transformed input object appearance of (c) is close to that in the selected target image of (d).

TABLE 4. Results of the combination of our network with other 2D image registration techniques.

Method	Seg. Acc.	Pose Err.	Rec. Acc.
SIFT [16] + RANSAC [20]	0.59±0.04	0.07±0.01	0.84±0.01
LoFTR [8]	0.76±0.01	0.08±0.01	0.91±0.01
SuperPoint [6] + SuperGlue [7]	0.88±0.01	0.05±0.01	0.99±0.01

segmentation network module s() worked accurately because the dominant canonical plane in (c) was close to the dominant canonical plane in (d).

D. COMBINATION OF OUR NETWORK WITH OTHER 2D IMAGE REGISTRATION TECHNIQUES

Our network can be easily combined with the existing 2D image registration framework described in Section III-B. We evaluated the performance of our network when it was combined with various 2D image registration techniques. We added the following combinations: SIFT [16] + RANSAC [20], LoFTR [8], and SuperPoint [6] + SuperGlue [7] for the local descriptor extraction and keypoint matching processes in the 2D image registration module r(). We evaluated the performance of all methods for these combinations by applying the same dataset used in Section IV-A. We simply used the network weights and hyperparameters provided by default for existing 2D image registration techniques [6], [7], [8], [16]. The results are shown in Table 4. SuperPoint + SuperGlue combined with our network outperformed the other techniques in terms of segmentation accuracy, pose estimation error, and recognition accuracy. We believe that SuperPoint and SuperGlue were better than the other techniques in terms of their combination with our weakly supervised segmentation network module s().

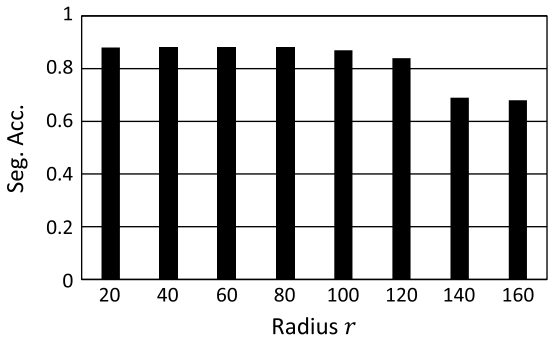


FIGURE 11. Segmentation accuracy for various values of the radius for the pseudo-object region.

E. INVESTIGATION OF OUR SEGMENTATION NETWORK'S CONFIGURATIONS

We compared the segmentation accuracy of our network for various values of radius r of the pseudo-object region. We generated pseudo-object regions with r from 20 to 160 pixels, in 20-pixel increments. The results are shown in Fig. 11. Segmentation accuracy was higher with $r = 20, 40, 60$, and 80 pixels than the other numbers of pixels. We believe that r set to 80 pixels or less can achieve higher segmentation accuracy on the dataset in Section IV-A. We also believe that r is unlikely to be parameter-sensitive.

We compared the segmentation accuracy of our network for various values of hyperparameter λ in Eq. (5). We set $\lambda = 0, 0.001, 0.005, 0.01, 0.05$. Setting $\lambda = 0$ meant that we ignored the regularization term L_ε in the training process. The results are shown in Fig. 12. We confirmed that our network with $\lambda = 0.005$ achieved the highest segmentation accuracy. By contrast, the segmentation accuracy of our network with $\lambda = 0$ was almost zero. We consider that the

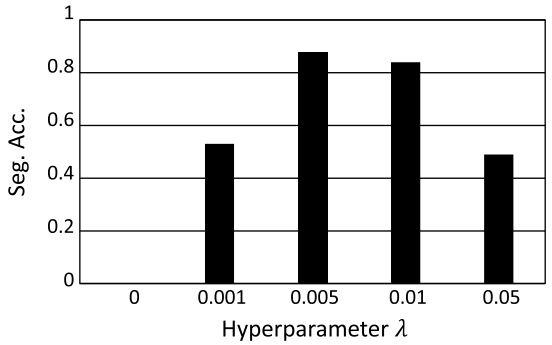


FIGURE 12. Segmentation accuracy while changing the hyperparameter for the regularization term.

correct λ adjustment between the region loss term L_r and regularization term L_ε is essential for the training process.

V. CONCLUSION

We proposed a method that introduces a weakly supervised segmentation network module to the 2D image registration framework for performing the segmentation task with only image-level class label annotation. To train our proposed network, we generated the pseudo-object region as the initial segmentation solution using the output of 2D image registration techniques. Moreover, we used the region loss term and regularization loss term to refine the pseudo-object region by leaving object pixels and removing background pixels. We demonstrated that our network achieved higher segmentation accuracy than existing weakly supervised segmentation networks in cooperation with 2D image registration techniques.

We discuss the limitation of our method as follows: As described in Section III-B, the region loss term correctly removes the background pixels of pseudo-object regions under the assumption that the background appearance of the query is different from that of the target. Our method has a limitation when the query and target do not satisfy this assumption, for example, when the query and target are acquired for the same color tray background. In this case, there is a risk that the region loss term cannot correctly remove the background pixels of the query and target.

In future work, we will develop a novel loss term to check the boundary between the foreground and background pixels. We will expand our approach to evaluate the performance of our network on objects of various shapes. We intend to design an algorithm for generating the pseudo-object region, which is more suitable for training the weakly supervised segmentation network module. We also intend to develop applications that use the weakly supervised segmentation module, such as multimodal image registration techniques [34] and camera pose reconstruction [35].

ACKNOWLEDGMENT

The authors would like to thank Dr. Takashi Shibata of NTT Corporation and Prof. Yoshio Iwai of Tottori University for their helpful advice on this study.

REFERENCES

[1] J. Leitner, A. W. Tow, N. Sünderhauf, J. E. Dean, J. W. Durham, M. Cooper, M. Eich, C. Lehnert, R. Mangels, C. McCool, and P. T. Kujala, "The ACRV picking benchmark: A robotic shelf picking benchmark to foster reproducible research," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 4705–4712.

[2] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman, "Analysis and observations from the first Amazon picking challenge," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 1, pp. 172–188, Jan. 2018.

[3] S. D'Avella, P. Tripicchio, and C. A. Avizzano, "A study on picking objects in cluttered environments: Exploiting depth features for a custom low-cost universal jamming gripper," *Robot. Comput.-Integr. Manuf.*, vol. 63, Jun. 2020, Art. no. 101888.

[4] B. Zitová and J. Flusser, "Image registration methods: A survey," *Image Vis. Comput.*, vol. 21, no. 11, pp. 977–1000, Oct. 2003.

[5] X. Jiang, J. Ma, G. Xiao, Z. Shao, and X. Guo, "A review of multimodal image matching: Methods and applications," *Inf. Fusion*, vol. 73, pp. 22–71, Sep. 2021.

[6] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 224–236.

[7] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4937–4946.

[8] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "LoFTR: Detector-free local feature matching with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 8922–8931.

[9] J. Ahn and S. Kwak, "Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4981–4990.

[10] Z. Dong, Z. Hanwang, T. Jinhui, H. Xiansheng, and S. Qianru, "Causal intervention for weakly supervised semantic segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 655–666.

[11] Y. Wang, J. Zhang, M. Kan, S. Shan, and X. Chen, "Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation," in *Proc. Comput. Vis. Pattern Recognit.*, 2020, pp. 12275–12284.

[12] S. Jo and I.-J. Yu, "Puzzle-CAM: Improved localization via matching partial and full features," in *Proc. Int. Conf. Image Process. (ICIP)*, 2021, pp. 639–643.

[13] S. Rong, B. Tu, Z. Wang, and J. Li, "Boundary-enhanced co-training for weakly supervised semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 19574–19584.

[14] Y. Du, Z. Fu, Q. Liu, and Y. Wang, "Weakly supervised semantic segmentation by pixel-to-prototype contrast," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4310–4319.

[15] S. Yoneda, G. Irie, T. Shibata, M. Nishiyama, and Y. Iwai, "Deep segmentation network without mask image supervision for 2D image registration," in *Proc. Int. Workshop Frontiers Comput. Vis. (IW-FCV)*, 2022, pp. 227–241.

[16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[17] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.

[18] X. Zhao, X. Wu, W. Chen, P. C. Y. Chen, Q. Xu, and Z. Li, "ALIKED: A lighter keypoint and descriptor extraction network via deformable transformation," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–16, 2023.

[19] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, "LightGlue: Local feature matching at light speed," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 17627–17638.

[20] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[21] F. Kluger, E. Brachmann, H. Ackermann, C. Rother, M. Y. Yang, and B. Rosenhahn, "CONSAC: Robust multi-model fitting by conditional sample consensus," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4633–4642.

[22] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A nested U-Net architecture for medical image segmentation," in *Proc. Deep Learn. Med. Image Anal. Multimodal Learn. Clin. Decis. Support (DLMIA)*, 2018, pp. 3–11.
[23] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.
[24] K.-J. Hsu, Y.-Y. Lin, and Y.-Y. Chuang, "Co-attention CNNs for unsupervised object co-segmentation," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2018, pp. 748–756.
[25] Y.-C. Chen, Y.-Y. Lin, M.-H. Yang, and J.-B. Huang, "Show, match and segment: Joint weakly supervised learning of semantic matching and object co-segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3632–3647, Oct. 2021.
[26] J. S. Sevak, A. D. Kapadia, J. B. Chavda, A. Shah, and M. Rahevar, "Survey on semantic image segmentation techniques," in *Proc. Int. Conf. Intell. Sustain. Syst. (ICISS)*, 2017, pp. 306–313.
[27] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.
[28] K. Ueno, G. Irie, M. Nishiyama, and Y. Iwai, "Weakly supervised triplet learning of canonical plane transformation for joint object recognition and pose estimation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 2476–2480.
[29] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[30] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The YCB object and model set: Towards common benchmarks for manipulation research," in *Proc. Int. Conf. Adv. Robot. (ICAR)*, Jul. 2015, pp. 510–517.
[31] C. Rennie, R. Shome, K. E. Bekris, and A. F. De Souza, "A dataset for improved RGBD-based object detection and pose estimation for warehouse pick-and-place," *IEEE Robot. Autom. Lett.*, vol. 1, no. 2, pp. 1179–1185, Jul. 2016.
[32] R. Araki, T. Yamashita, and H. Fujiyoshi, "ARC2017 RGB-D dataset for object detection and segmentation," in *Proc. Late Breaking Results Poster Int. Conf. Robot. Autom. (ICRA)*, 2018.
[33] R. Kaskman, S. Zakharov, I. Shugurov, and S. Ilic, "HomebrewedDB: RGB-D dataset for 6D pose estimation of 3D objects," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 2767–2776.
[34] B. Zhu, L. Zhou, S. Pu, J. Fan, and Y. Ye, "Advances and challenges in multimodal remote sensing image registration," *IEEE J. Miniaturization Air Space Syst.*, vol. 4, no. 2, pp. 165–174, Jun. 2023.
[35] S. Kim, R. Manduchi, and S. Qin, "Multi-planar monocular reconstruction of Manhattan indoor scenes," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2018, pp. 616–624.

SHUNSUKE YONEDA received the M.S. degree in engineering from the Graduate School of Sustainability Science, Tottori University, in 2022. He is currently attending a doctoral course with the Graduate School of Engineering, Tottori University. He is engaged in studies relating to object image recognition techniques using weakly supervised learning.

GO IRIE (Member, IEEE) received the M.E. degree in system engineering from Keio University, Japan, in 2006, and the Ph.D. degree in information science and technology from The University of Tokyo, Japan, in 2011. He was a Research Scientist with NTT Corporation, Japan, from 2006 to 2022, and a Visiting Research Scholar with Columbia University, from 2012 to 2013. He is currently an Associate Professor with Tokyo University of Science, Japan. His research interests include pattern recognition, machine learning, and media understanding.

MASASHI NISHIYAMA received the M.S. degree in engineering from the Graduate School of Natural Science and Technology, Okayama University, Japan, in 2002, and Ph.D. degree in interdisciplinary information studies from the Graduate School of Interdisciplinary Information Studies, The University of Tokyo, Japan, in 2011. He worked with the Corporate Research and Development Center, Toshiba Corporation, from 2002 to 2015. He is currently a Professor with the Graduate School of Engineering, Tottori University, Japan. In his recent research, he has focused on developing novel principles for representing identities, attributes, and human behavior in video sequences.