Deep Segmentation Network without Mask Image Supervision for 2D Image Registration

Shunsuke Yoneda[1], Go Irie[2], Takashi Shibata[2], Masashi Nishiyama[1], and Yoshio Iwai[1]

[1] Tottori University, 101 Minami 4-chome, Koyama-cho, Tottori, 680-8550, Japan
[2] NTT Corporation, 1 Morinosato-wakamiya 3-chome, Atsugi, 243-0198, Japan

Abstract. Two-dimensional (2D) image registration is a conventional technique for simultaneously performing object recognition and pose estimation tasks. Deep neural-based 2D image registration techniques recently emerged and achieved high performance in both tasks. However, these 2D image registration techniques are not designed to perform the segmentation task, which is one of the significant image processing techniques. Here, we consider introducing a deep segmentation network module into the framework of the 2D image registration. Especially, we consider training the segmentation network module with no supervision cost of mask images. To do this, we exploit the idea of the canonical plane, which is one surface observed mainly for each object in an image. We train the weakly supervised segmentation network module to perform the segmentation task of the canonical plane in the query and target images using the outputs of the 2D image registration. Experimental results show that our network can accurately perform the segmentation task without mask image supervision.

Keywords: Segmentation · Image registration · Supervision.

1 Introduction

There is a strong demand for solutions in warehouses to automate the picking process of planar objects such as product boxes and books to solve the labor shortage in logistics. Recently, some object-picking systems using cameras and robot arms [5, 10] have been developed as one of the automation solutions. These systems automatically infer foreground masks, class labels and pose parameters of target objects using image processing technologies to control robot arms to pick objects up. To do this, we need to perform the following tasks accurately; segmentation, object recognition and pose estimation.

We consider 2D image registration techniques, which are conventional techniques for simultaneously performing object recognition and pose estimation tasks. These techniques detect interest points from images containing the same object, extract local descriptors from the surroundings of the interest points, and search the correspondences between the interest points using the descriptors. The

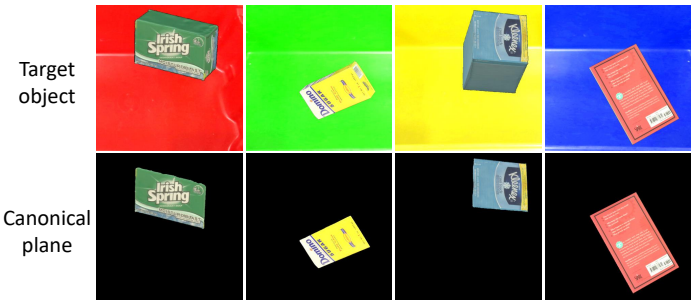

Fig. 1: Examples of the canonical plane in the target object. The canonical plane is one surface that is observed mainly for each object.

popular 2D image registration techniques are Scale-Invariant Feature Transform (SIFT) [11] + Random Sample Consensus (RANSAC) [7] and Oriented FAST and Rotated BRIEF (ORB) [13] + RANSAC. Recently, deep neural-based 2D image registration techniques, such as SuperPoint [6] + SuperGlue [14], have emerged and achieved high performance in object recognition and pose estimation tasks. However, these 2D image registration techniques [6, 7, 11, 13, 14] do not consider how to perform the segmentation task though they consider how to perform object recognition and pose estimation tasks.

Here, we propose a deep segmentation network module that is easily attached to the framework of the 2D image registration under the condition that mask image supervision is not required. Before explaining our segmentation network module, we consider a simple idea to use a fully supervised segmentation network in addition to using a 2D image registration technique. This idea requires collecting large datasets with mask image supervision, which is generally costly, to obtain high segmentation accuracy. Instead of using a fully supervised segmentation network, we introduce a weakly supervised segmentation network module into the framework of the 2D image registration. Experimental results show that our segmentation network module with no cost of mask supervision accurately performed the segmentation task with the help of pose parameters estimated by the 2D image registration technique.

2 Our deep segmentation network with 2D image registration

2.1 Overview

Our goal is to attach the deep segmentation network module to the framework of 2D image registration without explicit mask image supervision. To train the weakly supervised segmentation network module, we exploit the idea of the canonical plane, which has been introduced in [15]. The canonical plane is one surface that is observed mainly for each object. Figure 1 shows examples of the canonical plane when a camera acquires a rectangular or planar object image.

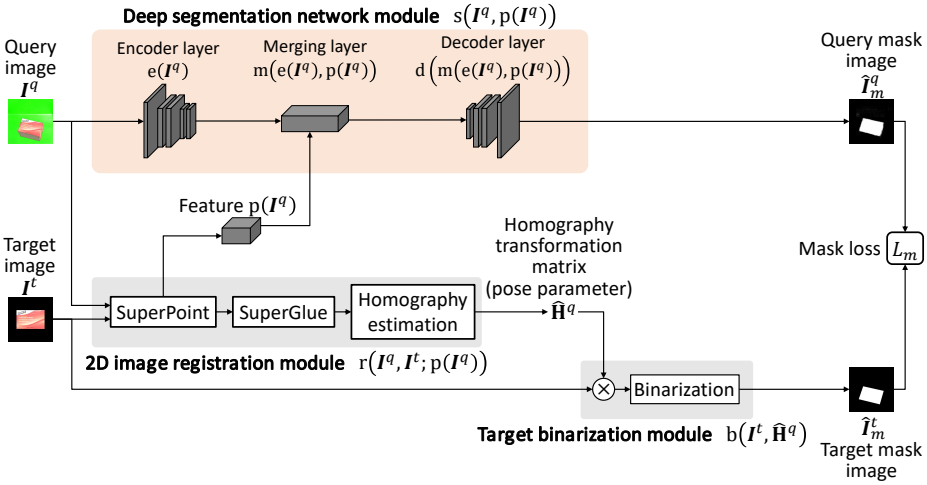

Fig. 2: Overview of our network for the training process. We introduce our deep segmentation network module s(), which does not require mask image supervision, into the framework of the 2D image registration. To predict the region of the canonical plane in the query image $\boldsymbol{I}^q$, we train the segmentation network module s() by reducing the mask loss L_m computed between the query mask image $\hat{\boldsymbol{I}}^q_m$ and the target mask image $\hat{\boldsymbol{I}}^t_m$.

We consider training the segmentation network module that predicts the region of the canonical plane in a query image. Our network performs the segmentation task using the output of 2D image registration between the query and target images belonging to the same canonical plane. Based on this idea, our training process can achieve no mask image supervision for the segmentation task. Figure 2 shows the overview of our network that trains the segmentation task of the canonical plane with the 2D image registration. We describe the detail of our overall network below.

In advance, we prepare the image pair as the training sample, where the pair consists of the query image $\boldsymbol{I}^q$ and target image $\boldsymbol{I}^t$. Note that the query $\boldsymbol{I}^q$ and the target $\boldsymbol{I}^t$ belong to the same canonical plane of the same object class (but with a different pose), which are used for the deep segmentation network module and the 2D image registration module.

Our overall network consists of the deep segmentation network module s(), the 2D image registration module r(), and the target binarization module b(). First, the segmentation network module s() predicts a query mask image $\hat{\boldsymbol{I}}^q_m$ from a query $\boldsymbol{I}^q$ by using the encoder layer, the merging layer, and the decoder layer. The encoder layer e() extracts the feature $e(\boldsymbol{I}^q)$ from the query $\boldsymbol{I}^q$. The merging layer m() combines the feature $e(\boldsymbol{I}^q)$ with the feature $p(\boldsymbol{I}^q)$, which is extracted in the 2D image registration module r(). We will explain this registration module r() later. In the merging layer m(), we consider that the feature $p(\boldsymbol{I}^q)$ highlights the informative region of the feature $e(\boldsymbol{I}^q)$ representing the canonical plane. The

decoder layer d() predicts the query mask $\hat{\boldsymbol{I}}_m^q$ using the feature $\mathrm{m}(\mathrm{e}(\boldsymbol{I}^q), \mathrm{p}(\boldsymbol{I}^q))$ combined in the merging layer.

Second, the 2D image registration module r() estimates the relative pose change from the target $\boldsymbol{I}^t$ to the query $\boldsymbol{I}^q$. According to [14, 15], this pose change can be represented by a homography transformation matrix $\hat{\mathbf{H}}^q$ between the regions of the target $\boldsymbol{I}^t$ and the query $\boldsymbol{I}^q$ belonging to the same canonical plane. For this module r(), we simply use the pre-trained SuperPoint [6] + SuperGlue [14] network. SuperPoint detects interest points and extracts deep local descriptors from the query $\boldsymbol{I}^q$ and the target $\boldsymbol{I}^t$ using a convolutional network. SuperGlue searches the correspondences between the interest points of the query $\boldsymbol{I}^q$ and those of the target $\boldsymbol{I}^t$ using a graph neural network and a matching layer. Additionally, the 2D image registration module r() uses the least squares method to estimate the homography matrix $\hat{\mathbf{H}}^q$ using the correspondences searched by SuperGlue.

Third, the target binarization module b() predicts a target mask image $\hat{\boldsymbol{I}}_m^t$ from a target $\boldsymbol{I}^t$ by using the following processes. This module transforms the target $\boldsymbol{I}^t$ using the homography matrix $\hat{\mathbf{H}}^q$ estimated in the 2D image registration module r() so that the pose parameter of the canonical plane in the target $\boldsymbol{I}^t$ is the same as that of the query $\boldsymbol{I}^q$. Next, this module binarizes pixel values of the canonical plane to 1 and those of other regions to 0. To check the similarity between the query mask $\hat{\boldsymbol{I}}_m^q$ and the target mask $\hat{\boldsymbol{I}}_m^t$, we compute the mask loss L_m in our network. By reducing the mask loss L_m, we can train the segmentation network module s() without the mask image supervision.

2.2 Training image pairs

Our network uses the query $\boldsymbol{I}^q$ and the target $\boldsymbol{I}^t$ as the training image pair $\langle \boldsymbol{I}^q, \boldsymbol{I}^t \rangle$. The query $\boldsymbol{I}^q$ contains a rectangular or planar object with a random pose. We do not place any particular restrictions on the background of the query $\boldsymbol{I}^q$. In contrast, we assume that the pixel values of the background region outside the canonical plane in the target $\boldsymbol{I}^t$ are filled with 0, i. e., the background condition of the target $\boldsymbol{I}^t$ is black. We also assume that the target $\boldsymbol{I}^t$ contains only one canonical plane, which appears in the pair's query $\boldsymbol{I}^q$. We consider that it is reasonable to acquire the target $\boldsymbol{I}^t$ with the black background condition. Specifically, the target $\boldsymbol{I}^t$ can be acquired by placing the object parallel to the black floor and using a camera set up so that its optical axis passes through the object's center of gravity. Each of Fig. 3(a) and (b) show the examples of the target $\boldsymbol{I}^t$ representing the canonical planes for rectangular and planar objects. It is sufficient to acquire six targets per one rectangular object and two targets per one planar object for the training process.

2.3 Deep segmentation network module

We describe the details of the training process of the deep segmentation network module s(). As described in Section 2.1, this module s() performs the

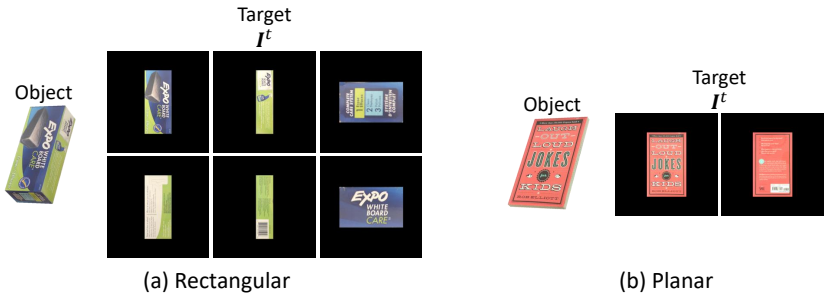

Fig. 3: Examples of the target $\boldsymbol{I}^t$ representing the canonical plane for rectangular and planar objects.

segmentation task with the help of the feature $\mathrm{p}(\boldsymbol{I}^q)$ extracted in the 2D image registration module r(). Hence, we first describe the pose estimation task using the module r(). This module estimates the homography matrix $\hat{\mathbf{H}}^q$ from the target $\boldsymbol{I}^t$ to the query $\boldsymbol{I}^q$ as

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

In the module r(), SuperPoint [6] internally extracts the feature $\mathrm{p}(\boldsymbol{I}^q)$ using a convolutional network. After performing the pose estimation task, the deep segmentation network module s() predicts the query mask $\hat{\boldsymbol{I}}^q_m$ from the query $\boldsymbol{I}^q$ using $\mathrm{p}(\boldsymbol{I}^q)$ as

$$\hat{\boldsymbol{I}}^q_m = \mathrm{s}(\boldsymbol{I}^q, \mathrm{p}(\boldsymbol{I}^q)) = \mathrm{d}(\mathrm{m}(\mathrm{e}(\boldsymbol{I}^q), \mathrm{p}(\boldsymbol{I}^q))). \tag{2}$$

As described in Section 2.1, the module s() uses the encoder layer e(), the merging layer m(), and the decoder layer d(). The encoder layer e() extracts the feature $\mathrm{e}(\boldsymbol{I}^q)$ that represents the region of the canonical plane in the query $\boldsymbol{I}^q$ using convolutional layers. The merging layer m() first combines the feature $\mathrm{e}(\boldsymbol{I}^q)$ with the feature $\mathrm{p}(\boldsymbol{I}^q)$ extracted by the SuperPoint network. Then, the merging layer reduces the dimensionality of the combined feature for fitting the dimensionality of the feature $\mathrm{e}(\boldsymbol{I}^q)$. The decoder layer d() predicts the query mask $\hat{\boldsymbol{I}}^q_m$ from the feature $\mathrm{m}(\mathrm{e}(\boldsymbol{I}^q), \mathrm{p}(\boldsymbol{I}^q))$ combined in the merging layer m() using deconvolutional layers. After performing the segmentation task, the target binarization module b() predicts the target mask $\hat{\boldsymbol{I}}^t_m$ from the target $\boldsymbol{I}^t$ as

$$\hat{\boldsymbol{I}}^t_m = \mathrm{t}(\boldsymbol{I}^t, \hat{\mathbf{H}}^q). \tag{3}$$

This module transforms the target $\boldsymbol{I}^t$ using the homography matrix $\hat{\mathbf{H}}^q$ estimated in the 2D image registration module r() and binarize the pixel value of the canonical plane to 1 and that of the other region to 0. Finally, the mask loss L_m is computed using the squared L2 norm between the query mask $\hat{\boldsymbol{I}}^q_m$ and the target mask $\hat{\boldsymbol{I}}^t_m$ as

$$L_m = ||\hat{\boldsymbol{I}}^q_m - \hat{\boldsymbol{I}}^t_m||^2_2. \tag{4}$$

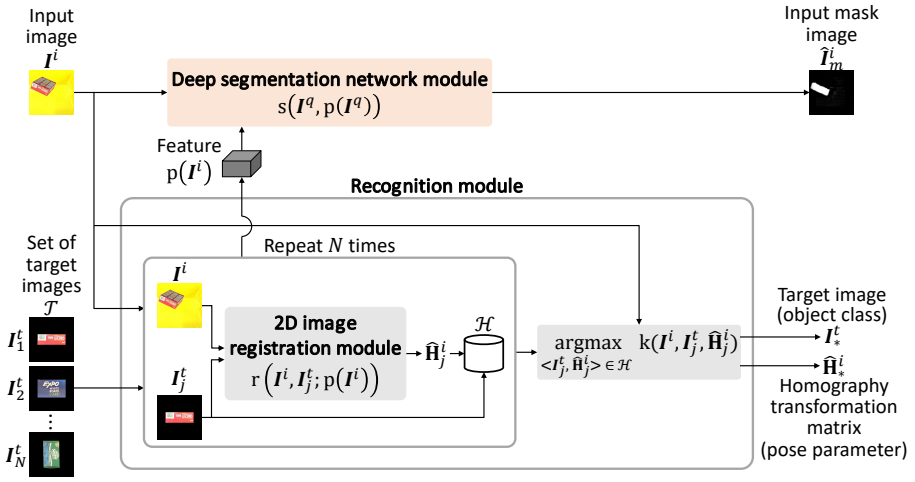

Fig. 4: Overview of our network for the inference process. The deep segmentation network module s() predicts the input mask image $\hat{\boldsymbol{I}}^i_m$ of the input image $\boldsymbol{I}^i$ using the feature $\mathrm{p}(\boldsymbol{I}^i)$ extracted in the 2D image registration module r(). The recognition module outputs the pair of the selected target image $\boldsymbol{I}^t_*$ and the homography transformation matrix $\hat{\mathbf{H}}^i_*$.

This loss L_m returns a small value when the query mask $\hat{\boldsymbol{I}}^q_m$ and the target mask $\hat{\boldsymbol{I}}^t_m$ are similar, i. e. when the prediction of the query mask $\hat{\boldsymbol{I}}^q_m$ is close to a correct solution.

2.4 Inference

Figure 4 shows the overview of our network for the inference process. Our inference process consists of the deep segmentation network module s() and the recognition module. We initially perform the object recognition and pose estimation tasks for an input image $\boldsymbol{I}^i$ using the recognition module and then perform the segmentation task using our module s(). We store a set of target images $\mathcal{T} = \{\boldsymbol{I}^t_j\}_{j=1}^N$ in advance. Note that each target $\boldsymbol{I}^t_j$ stored in the set $\mathcal{T}$ contains only one canonical plane belonging to a target object.

We describe the detail of the recognition module. The 2D image registration module r() in the recognition module estimates the homography transformation matrix $\hat{\mathbf{H}}^i_j$ for transforming the image from the target $\boldsymbol{I}^t_j$ to the input $\boldsymbol{I}^i$. We store a pair of the target $\boldsymbol{I}^t_j$ and the homography matrix $\hat{\mathbf{H}}^i_j$ in the set of the pairs $\mathcal{H}$ for all N target images. Here, the recognition module performs the object recognition task for the input $\boldsymbol{I}^i$ as

$$\langle \boldsymbol{I}^t_*, \hat{\mathbf{H}}^i_* \rangle = \underset{\langle \boldsymbol{I}^t_j, \hat{\mathbf{H}}^i_j \rangle \in \mathcal{H}}{\operatorname{argmax}} \ \mathrm{k}(\boldsymbol{I}^i, \boldsymbol{I}^t_j, \hat{\mathbf{H}}^i_j). \tag{5}$$

This equation means that a pair of a target image $\boldsymbol{I}^t_*$ and a homography transformation matrix $\hat{\mathbf{H}}^i_*$ is selected from the set $\mathcal{H}$, where the target $\boldsymbol{I}^t_*$ has the largest

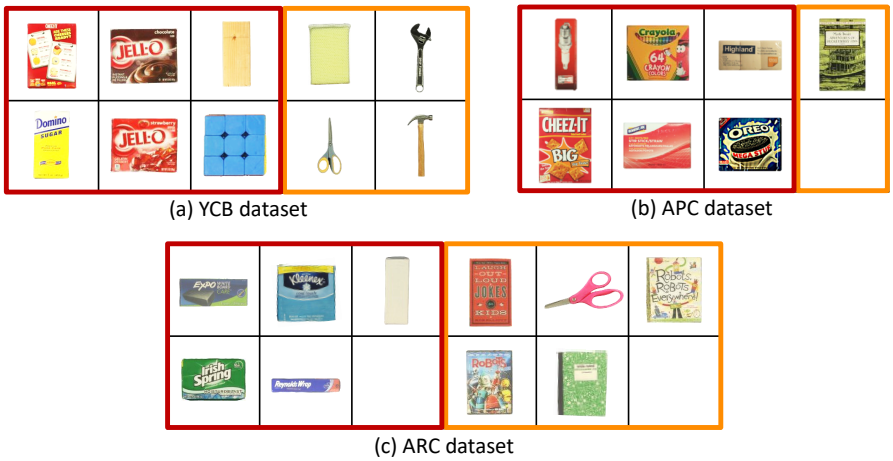

Fig. 5: Target objects from YCB dataset, APC dataset, and ARC dataset used in our experiments. The color frame indicates the type of the object shape (Red: rectangular, Orange: planar).

number of interest points corresponding to the input $\boldsymbol{I}^i$. The function k() counts the number of interest points in the correspondence between the input $\boldsymbol{I}^i$ and the target $\boldsymbol{I}^t_j$. The object recognition task is done by assigning the object class of the selected target $\boldsymbol{I}^t_*$. The pose estimation task is done by outputting the selected homography matrix $\hat{\mathbf{H}}^i_*$ as the relative pose change from the selected target $\boldsymbol{I}^t_*$ to the input $\boldsymbol{I}^i$.

In the inference process, we directly use our deep segmentation network module s() of Section 2.3 trained with no cost of mask image supervision and can perform the segmentation task efficiently. The segmentation network module s() predicts the input mask $\hat{\boldsymbol{I}}^i_m$ of the input $\boldsymbol{I}^i$ using the feature p($\boldsymbol{I}^i$) extracted in the 2D image registration module r() after the recognition module performs the object recognition and pose estimation tasks.

3 Experiments

3.1 Dataset

We evaluated the segmentation accuracy, recognition accuracy, and pose estimation error of our network on a dataset mixed with three popular datasets for the picking process scenario; YCB dataset [3], APC dataset [12], and ARC dataset [2]. We used 17 rectangular objects and 10 planar objects: six rectangular objects and four planar objects in YCB dataset as shown in Fig. 5(a), six rectangular objects and one planar object in APC dataset as shown in Fig. 5(b) and five rectangular objects and five planar objects in ARC dataset as shown

Fig. 6: Examples of the background condition used in our experiments.

in Fig. 5(c). Each object in these datasets consisted of a 3D mesh model and a texture map.

We generated the query $\boldsymbol{I}^q$ by applying 3D rendering with three degrees of freedom (3-DOF) rotation, translation, and scaling to the objects randomly. The 3-DOF rotation angles were sampled in the range of $[-30, 30]$ degrees. The translation parameters were sampled in the range of $[-150, 150]$ pixels. And, the scale parameters were sampled in the range of $[0.8, 1.2]$.

We used two different background conditions: tray background as shown in Fig. 6(a) and clutter background as shown in Fig. 6(b). In the tray background condition, we used red, blue, yellow, and green trays. In the clutter background condition, we randomly placed 28 objects in Household Objects for Pose Estimation (HOPE) datasets[3]. Note that the objects contained in HOPE dataset were completely different from the objects in the datasets of Fig. 5. We show examples of the query $\boldsymbol{I}^q$ in the tray background condition of Fig. 7(a) and the clutter background condition of Fig. 7(b).

We acquired the target $\boldsymbol{I}^t$ using the manner described in Section 2.2. We used $30,000$ pairs of the query $\boldsymbol{I}^q$ and the target $\boldsymbol{I}^t$ for the training process. We also used $3,000$ inputs $\boldsymbol{I}^i$ for the inference process. Note that we completely separated the pose parameters of the inputs $\boldsymbol{I}^i$ from those of the training image pairs $\langle \boldsymbol{I}^q, \boldsymbol{I}^t \rangle$. For the inference process, the set of target images $\mathcal{T}$ included all canonical planes of the 27 objects contained in the dataset of Fig. 5. The number of images contained in $\mathcal{T}$ was 122, i.e., $17 \times 6 + 10 \times 2$, because a rectangular object contains six canonical planes and a planar object contains two canonical planes. The size of each image was fixed to 600×600 pixels. In the 2D image registration module r(), we used the size of the input images at 600×600 pixels. In the deep segmentation network module s(), we resize the input images to 100×100 pixels.

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

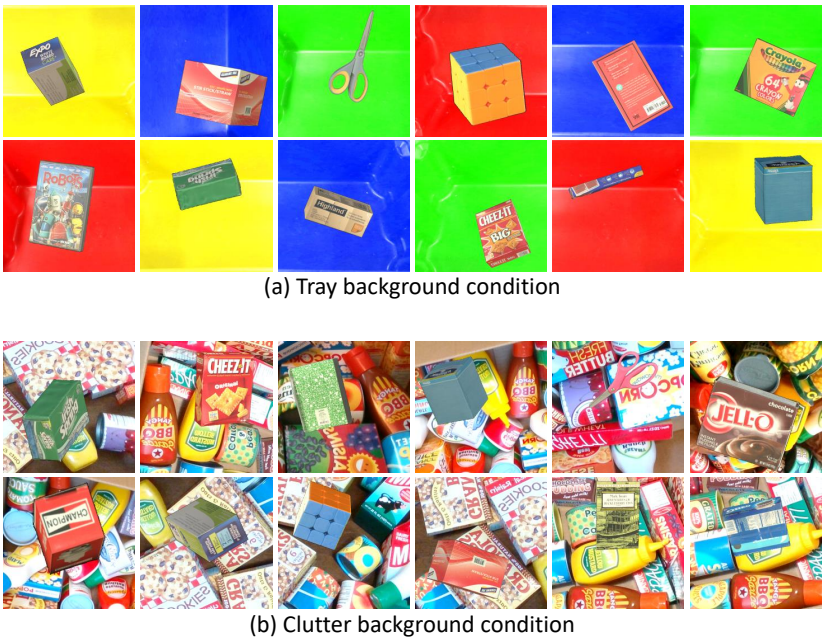

Fig. 7: Examples of the query $\boldsymbol{I}^q$ used in our experiments.

3.2 Implementation and performance metrics

The encoder layer e() of the deep segmentation network module s() consisted of one pooling layer and four convolutional layers. The merging layer m() consisted of one upsampling layer for the features p($\boldsymbol{I}^q$) extracted in the 2D image registration module r(), one concatenate layer, and one convolutional layer for dimensionality reduction. The decoder layer d() consisted of two deconvolutional layers, one convolutional layer, and one upsampling layer. We simply used the pre-trained SuperPoint [6] + SuperGlue [14] network that are officially available. We set the hyperparameters of SuperPoint and SuperGlue to default provided values. We trained our network using stochastic gradient descent with a learning rate of 10 and a momentum parameter of 0.9 for 100 epochs.

We used the following three metrics for performance evaluation. The intersection over union (IoU) between the predicted mask image and the ground truth mask image was used to evaluate the segmentation accuracy. The correct match rate was used to evaluate the recognition accuracy. The Frobenius norm of the difference between the estimated homography matrix and the ground truth homography matrix was used to evaluate the pose estimation error. Note that we computed the pose estimation error only when the object class of the input $\boldsymbol{I}^i$ is the same as that of the target $\boldsymbol{I}^t_*$ selected by the recognition module. We used the average performance over the three runs, each with different training-testing splits.

Table 1: Performance of our network and SuperPoint [6] + SuperGlue [14].

	Seg. Acc.		Rec. Acc.		Pose Err.	
	Tray	Clutter	Tray	Clutter	Tray	Clutter
SuperPoint+SuperGlue	n/a	n/a	0.81±0.01	0.75±0.01	0.09±0.01	0.22±0.01
Ours	**0.82±0.01**	**0.75±0.01**	0.81±0.01	0.75±0.01	0.09±0.01	0.22±0.01

3.3 Comparison with SuperPoint [6] + SuperGlue [14]

We first evaluated the effectiveness of our network by comparing with original SuperPoint [6] + SuperGlue [14]. Our network performs the segmentation, object recognition, and pose estimation tasks, while the original SuperPoint + SuperGlue performs only the object recognition and pose estimation tasks. The results are shown in Table 1. The recognition accuracy and pose estimation error were consistent between the original and our network because we used the same SuperPoint + SuperGlue network. We confirmed that our network could perform the segmentation task in addition to the object recognition and pose estimation tasks by introducing the weakly supervised deep segmentation network module s() into the framework of the 2D image registration.

We show qualitative results of the deep segmentation network module s() in Fig. 8 of the tray background condition, and those in Fig. 9 of the clutter background condition. In each row of these figures (from left to right), we show the input $\boldsymbol{I}^i$ for the inference process, the segmentation region masked by the input mask $\hat{\boldsymbol{I}}^i_m$ of s(), the transformation region converted by the homography matrix $\hat{\mathbf{H}}^i_*$ of the recognition module, and the target $\boldsymbol{I}^t_*$ of the recognition module. Note that we replaced pixel values of the background region from 0 to 255 in these figures. When generating the transformation region, we used the inverse matrix of $\hat{\mathbf{H}}^i_*$. We see that the deep segmentation network module s() accurately works because the appearances of the object between the transformation region and the target $\boldsymbol{I}^t_*$ are similar.

3.4 Comparison with existing segmentation networks

To analyze the effectiveness of the deep segmentation network module s(), we compare our network with fully and weakly supervised segmentation networks. As fully supervised segmentation networks, we used Feature Pyramid Network (FPN) [9], DeepLabv3+ [4], and Unet++ [17]. As weakly supervised segmentation networks, we used Pixel-level Semantic Affinity (PSA) [1], Self-supervised Equivariant Attention Mechanism (SEAM) [16], and Puzzle-CAM [8]. Note that the existing segmentation networks do not consider performing the pose estimation task, and PSA does not even consider performing the object recognition task. We applied fine-tuning to the pre-trained networks of existing segmentation techniques. We used the default provided hyper-parameters. The results are shown in Table 2. Our network performs inferior to any fully supervised segmentation networks in both segmentation accuracy and recognition accuracy.

Fig. 8: Qualitative results of the deep segmentation network module s() in tray background condition. We see that the appearance of the transformation region is similar to that of the target $\boldsymbol{I}^t_*$.

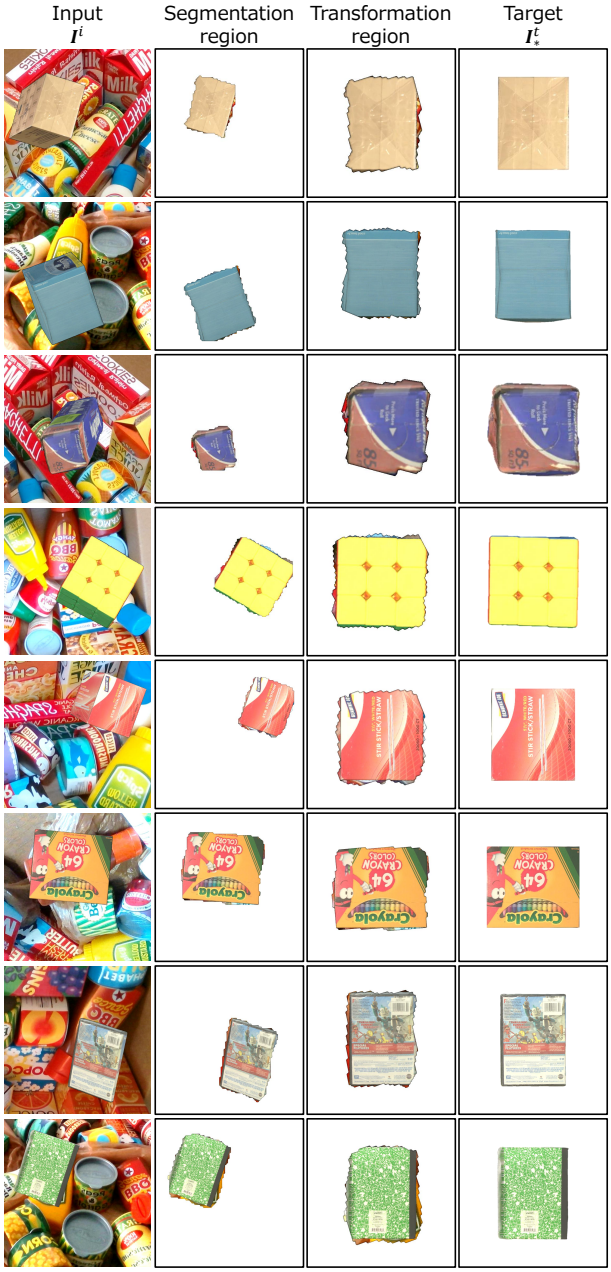

Fig. 9: Qualitative results of the deep segmentation network module s() in clutter background condition. We see that the appearance of the transformation region is similar to that of the target $\boldsymbol{I}^t_*$.

Table 2: Performance of our network and existing segmentation networks.

	Mask	Seg. Acc.		Rec. Acc.		Pose Err.	
	supervision	Tray	Clutter	Tray	Clutter	Tray	Clutter
FPN [9]	w/	0.98±0.01	0.98±0.01	0.99±0.01	0.99±0.01	n/a	n/a
DeepLabv3+ [4]	w/	0.98±0.01	0.98±0.02	0.99±0.01	0.99±0.01	n/a	n/a
Unet++ [17]	w/	0.99±0.02	0.99±0.01	0.99±0.01	0.99±0.01	n/a	n/a
PSA [1]	w/o	0.46±0.01	0.34±0.05	-	-	n/a	n/a
SEAM [16]	w/o	0.16±0.04	0.18±0.01	0.64±0.08	0.56±0.07	n/a	n/a
Puzzle-CAM [8]	w/o	0.33±0.07	0.06±0.02	0.45±0.04	0.15±0.09	n/a	n/a
Ours	w/o	**0.82±0.01**	**0.75±0.01**	**0.81±0.01**	**0.75±0.01**	**0.09±0.01**	**0.22±0.01**

Table 3: Performance of our network with and without features $p(\boldsymbol{I}^q)$ extracted in the 2D image registration module r().

	Seg. Acc.		Rec. Acc.		Pose Err.	
	Tray	Clutter	Tray	Clutter	Tray	Clutter
Ours (w/o $p(\boldsymbol{I}^q)$)	0.80±0.02	0.74±0.01	0.81±0.01	0.75±0.01	0.09±0.01	0.22±0.01
Ours (w/ $p(\boldsymbol{I}^q)$)	**0.82±0.01**	**0.75±0.01**	0.81±0.01	0.75±0.01	0.09±0.01	0.22±0.01

In contrast, our network outperforms all weakly supervised segmentation networks in both segmentation accuracy and recognition accuracy. Furthermore, our network was the only one that was able to obtain pose estimation outputs. We believe that our deep segmentation network module is advantageous for not requiring mask image supervision.

3.5 Performance without features extracted in 2D image registration module

As described in Section 2.1, our network uses the features $p(\boldsymbol{I}^q)$ extracted in the 2D image registration module r() to help the segmentation task by combining $p(\boldsymbol{I}^q)$ with the features $e(\boldsymbol{I}^q)$ in the merging layer m(). To analyze the impact of the features $p(\boldsymbol{I}^q)$, we evaluated the performance of our network without the use of $p(\boldsymbol{I}^q)$. The results are shown in Table 3. Comparing our network with and without $p(\boldsymbol{I}^q)$, the segmentation accuracy of our network with it was slightly better. We consider that the use of $p(\boldsymbol{I}^q)$ with $e(\boldsymbol{I}^q)$ has a little bit impact on the performance of our network.

3.6 Comparison with the other 2D image registration technique

We evaluated the performance of our network using ORB [13] instead of SuperPoint [6] in the 2D image registration module r(). Our network used RANSAC [7] instead of SuperGlue to search the correspondences of local descriptors between the interest points. Because ORB extracts local descriptors not using deep neural networks, we did not use the features $p(\boldsymbol{I}^q)$ for ORB. We used the default provided hyperparameters for ORB. The results are shown in Table 4. We confirmed that our network using SuperPoint is superior to that using ORB in terms of the segmentation accuracy, recognition accuracy, and pose estimation error. We

Table 4: Comparison with the other 2D image registration technique.

	Seg. Acc.		Rec. Acc.		Pose Err.	
	Tray	Clutter	Tray	Clutter	Tray	Clutter
Ours (ORB [13])	0.68±0.01	0.57±0.01	0.56±0.01	0.35±0.01	1.35±0.01	2.90±0.01
Ours (SuperPoint [6])	**0.82±0.01**	**0.75±0.01**	**0.81±0.01**	**0.75±0.01**	**0.09±0.01**	**0.22±0.01**

consider that SuperPoint is better than ORB for introducing our segmentation network module into the framework of the 2D image registration.

4 Conclusions

We proposed a deep segmentation network module without mask image supervision, which is easily attached to the framework of the 2D image registration. For this purpose, we trained the segmentation network module with the help of 2D image registration between the query and target images belonging to the same canonical plane. We demonstrated that our network accurately performs the segmentation, not requiring the cost of mask image supervision. In future work, we expand to evaluate the performance of our network on datasets of objects with various shapes. We intend to develop a network for handling more complex transformations than homography transformation. We would like to thank Mr. Tokachi SHIRAHATA for his cooperation in our experiments.

References

1. Ahn, J., Kwak, S.: Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4981–4990 (2018)
2. Araki, R., Yamashita, T., Fujiyoshi, H.: ARC2017 RGB-D dataset for object detection and segmentation. In: Proceedings of Late Breaking Results Poster on International Conference on Robotics and Automation (ICRA) (2018)
3. Calli, B., Singh, A., Walsman, A., Srinivasa, S., Abbeel, P., Dollar, A.M.: The YCB object and model set: Towards common benchmarks for manipulation research. In: Proceedings of International Conference on Advanced Robotics (ICAR). pp. 510–517 (2015)
4. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 801–818 (2018)
5. Correll, N., Bekris, K.E., Berenson, D., Brock, O., Causo, A., Hauser, K., Okada, K., Rodriguez, A., Romano, J.M., Wurman, P.R.: Analysis and observations from the first Amazon Picking Challenge. IEEE Transactions on Automation Science and Engineering (T-ASE) pp. 172–188 (2018)
6. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: Proceedings of CVPR Workshop on Deep Learning for Visual SLAM. pp. 337–33712 (2018)

7. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
8. Jo, S., Yu, I.J.: Puzzle-cam: Improved localization via matching partial and full features. In: Proceedings of International Conference on Image Processing (ICIP) (2021)
9. Kirillov, A., He, K., Girshick, R., Dollar, P.: A unified architecture for instance and semantic segmentation, `http://presentations.cocodataset.org/COCO17-Stuff-FAIR.pdf`
10. Leitner, J., Tow, A.W., Sunderhauf, N., Dean, J.E., Durham, J.W., Cooper, M., Eich, M., Lehnert, C., Mangels, R., McCool, C., Kujala, P.T., Nicholson, L., Pham, T., Sergeant, J., Wu, L., Zhang, F., Upcroft, B., Corke, P.: The ACRV picking benchmark: A robotic shelf picking benchmark to foster reproducible research. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA). pp. 4705–4712 (2017)
11. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60 pp. 91–110 (2004)
12. Rennie, C., Shome, R., Bekris, K.E., Souza, A.F.D.: A dataset for improved RGBD-based object detection and pose estimation for warehouse pick-and-place. IEEE Robotics and Automation Letters pp. 1179–1185 (2016)
13. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: Proceedings of 2011 International Conference on Computer Vision (ICCV). pp. 2564–2571 (2011)
14. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4938–4947 (June 2020)
15. Ueno, K., Irie, G., Nishiyama, M., Iwai, Y.: Weakly supervised triplet learning of canonical plane transformation for joint object recognition and pose estimation. In: Proceedings of International Conference on Image Processing (ICIP). pp. 2476–2480 (2019)
16. Wang, Y., Zhang, J., Kan, M., Shan, S., Chen, X.: Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12275–12284 (2020)
17. Zhou, Z., Rahman Siddiquee, M.M., Tajbakhsh, N., Liang, J.: Unet++: A nested u-net architecture for medical image segmentation. In: Proceedings of Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support (DLMIA). pp. 3–11 (2018)